Lernen mit Computerspielen

Programmieren in Minecraft

Ein Unterrichtsmaterial der Initiative Code your Life

Lernen mit Computerspielen

Programmieren mit Minecraft

Die Initiative Code your Life

Code your Life ist eine Initiative des 21st Century Competence Center im Förderverein für Jugend und Sozialarbeit e.V.
Marchlewskistraße 27, 10243 Berlin
Internet: www.code-your-life.org www.21ccc.de

Die Initiative Code your Life ist Teil des weltweiten Programms Microsoft YouthSpark und wird vom 21st Century Competence Center im Förderverein für Jugend und Sozialarbeit e.V. umgesetzt
Microsoft Berlin, Unter den Linden 17, 10117 Berlin
Internet: www.microsoft.de/politik

Lernen mit Computerspielen – Programmieren mit Minecraft

Herausgeber: Förderverein für Jugend und Sozialarbeit e. V.
Verantwortlich: Thomas Schmidt
Konzeption und Umsetzung: Helliwood media & education im fjs e.V.
Marchlewskistraße 27, 10243 Berlin, www.helliwood.com
In Zusammenarbeit mit
Playtypus GbR
William-Zipperer-Straße 118, 04179 Leipzig
Autoren: Julia Huke, Max Strohmeyer
Jörg Peleikis, Thomas Schmidt
Gestaltung und Satz: Helliwood media & education, Christiane Herold

Inhalt

Lernen mit Computerspielen

Digitale Spiele sind heute aus der Lebenswelt von Kindern und Jugendlichen nicht mehr wegzudenken. Egal ob PC, Konsole, Tablet oder Smartphone: Fast 70% aller Jugendlichen in Deutschland spielen täglich bis mehrmals die Woche (JIM-Studie 2015). Bei den Kindern sind es bereits über 50 %, die mindestens einmal pro Woche spielen (KIM-Studie 2014). Tendenz steigend.

> **Warum also nicht den Versuch wagen, ein Stück dieser Lebenswelt auch in die Schule zur holen? Wie lässt sich die Faszination, die vom Medium Computerspiel ausgeht, konstruktiv mit bestehenden Bildungsinhalten verknüpfen?**

Ein gutes Beispiel dafür, dass dies wirklich möglich ist, geben die immer zahlreicher werdenden Bildungsprojekte mit dem überaus beliebten Videospiel Minecraft. Warum ausgerechnet dieses Spiel? Mit dieser Handreichung beleuchten wir das Bildungspotential des Phänomens Minecraft, indem wir uns zunächst das grundlegende Spielprinzip anschauen und didaktische Überlegungen hinsichtlich der Verwendbarkeit im formellen Bildungsbereich anstellen. Wo liegen die Chancen eines „digital Game-based Learning“, dem Lernen mit digitalen Spielen und was haben die verschiedenen Spielertypen mit Lerntypen, wie man sie aus der Schule kennt, gemein?

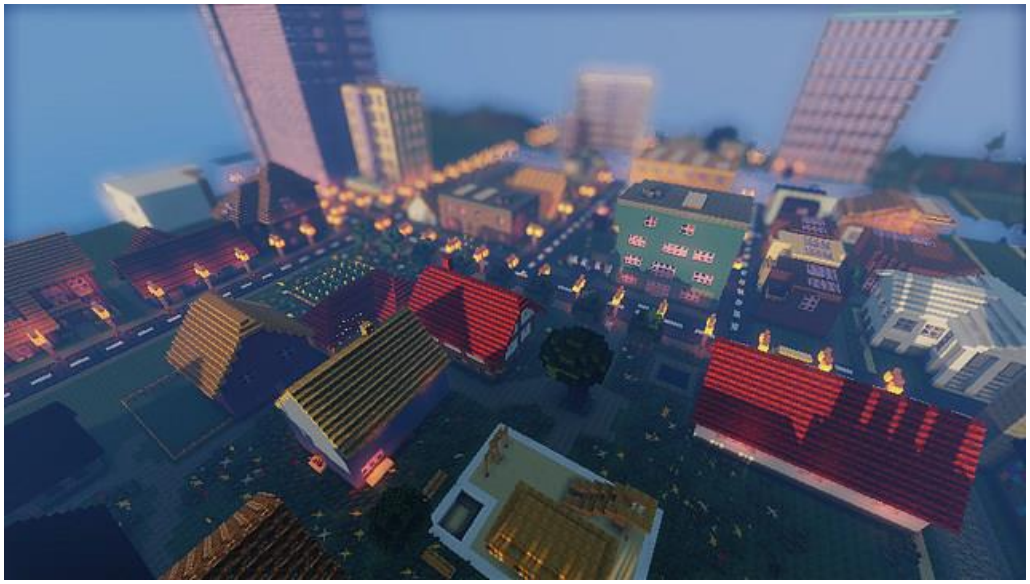
Natürlich darf auch die Praxis nicht fehlen! Unser mitgeliefertes Minecraft-Curriculum basiert auf dem Grundcurriculum von Code your Life zum spannenden Thema „Programmierung“ und gibt Ihnen das nötige Rüstzeug an die Hand, mit Minecraft noch einen Schritt weiterzugehen, wenn Sie mögen. In den von uns vorstrukturierten Unterrichtseinheiten, tauchen Sie gemeinsam mit Ihren Schülerinnen und Schülern in das Minecraft-Universum ein und schlüpfen in der ebenfalls von uns zur Verfügung gestellten Minecraft-Welt, in die Rolle von jungen Angestellten eines Bauunternehmens. Mit Hilfe der Skriptsprache Lua und kleiner Roboter, den Turtles, gilt es dort in mehreren kleinen Challenges, Lektion für Lektion gemeinsam eine neue Filiale aufzubauen.

Unsere Installationsanleitung hilft Ihnen Schritt für Schritt bei der Klärung technischer Voraussetzungen und beim Setup der Spielumgebung. Die einzelnen Lektionen enthalten, wie Sie es gewohnt sind, Anweisungen zur Vorbereitung der Einheiten, Vorbemerkungen, Schlüsselkompetenzen, empfohlene Arbeitsphasen, Anregungen zur Reflexion und mehr. Angemessen ist das Curriculum für Fortgeschrittene im Alter ab 12 Jahren, die das Grundcurriculum von Code your Life bereits durchlaufen haben.

Das Phänomen Minecraft

Abbauen – Kombinieren – Neuerschaffen: Mit diesem simplen Grundprinzip gelingt es dem Computerspiel „Minecraft“, heute weltweit über 100 Millionen Nutzer zu begeistern, darunter viele Kinder und Jugendliche. Verglichen wird es dabei gern mit einer Art digitalem Lego, nicht zuletzt aufgrund seiner blockigen und pixeligen visuellen Präsentation.

Entwickelt wurde Minecraft ursprünglich vom schwedischen Programmierer Markus „Notch“ Persson und seinem Unternehmen Mojang, bevor letzteres 2014 in den Besitz von Microsoft überging. Seit seiner erstmaligen Veröffentlichung 2011 ist das Spiel inzwischen nicht nur für den PC, sondern in anderen Version auch für viele Spielkonsolen und mobilen Systeme verfügbar.



Worum geht es?

Ohne weitere Erklärung wird der Spieler mit seiner Spielfigur, dem Avatar, zu Beginn des Spiels in eine offene, meist zufällig generierte Welt aus Wäldern, Wüsten, Ebenen, Gebirgsketten, Meeren und mehr gesetzt. Angereichert ist die Welt mit Rohstoffen wie Stein, Holz, wertvollen Erzen und Mineralien, bei Bedarf zusätzlich mit Tieren, Dorfbewohnern und kleinen Monstern.

Ein vorgegebenes Spielziel oder Spielende gibt es im Grunde nicht. Zu bewältigende Aufgaben stellen sich Spielerinnen und Spieler selbst. So werden Häuser, Burgen und Schlösser gebaut, aber auch Weizenfarmen automatisiert, Monster gejagt und Stromleitungen verlegt. Der Fantasie sind kaum Grenzen gesetzt.

Benötigte Ressourcen müssen die Spieler im sogenannten „Überlebensmodus“ mit der Hand oder speziellen Werkzeugen abbauen bzw. zu neuen Blöcken kombinieren, die sie wiederum in der Spielwelt platzieren können.

Ein einfaches Beispiel: Lilly fällt einen Minecraft-Baum. Sie erhält ein paar Rohholzblöcke, aus denen sie in ihrem Inventar Holzbretter herstellen kann. Ein Holzblock ergibt in Minecraft vier Holzbretter. Perfekt! Genau die richtige Menge, um eine Werkbank herzustellen. Die braucht Lilly nämlich, um noch bessere Werkzeuge und hübschere Blöcke herzustellen, die sie beim Bau ihres Hauses verwenden möchte.

Der „Kreativmodus“ ist eine andere Variante, Minecraft zu spielen. Hier stehen bereits alle Ressourcen von Beginn an in unendlicher Menge zur Verfügung. Möchte Lilly beispielsweise Fenster in die Wände ihres Häuschens einsetzen, könnte sie sich diese direkt über eine separate Auswahl nehmen, anstatt zuerst Sandblöcke zu sammeln und in einem Ofen zu Glas zu schmelzen, so wie sie es im oben geschilderten „Überlebensmodus“ tun müsste. Sie kann einfach frei drauflos bauen und schnell das ein oder andere ausprobieren, was anderenfalls viel Zeit und Ressourcen benötigen würde.

Gemeinsam spielen

Einen ganz besonderen Reiz bietet der Mehrspieler-Modus in Minecraft. Denn auch die größten Bauwerke, Entdeckungen und andere Errungenschaften werden erst zu richtigen Schätzen, wenn man sie mit anderen Spielern teilt oder sich darüber austauscht. Sehr beliebt sind z.B. gemeinsame Großbauprojekte, aber auch das freundschaftlich kompetitive Spiel, bei welchem Spielerinnen und Spieler sich im Internet auf speziellen Minispiel-Servern miteinander messen. Geschicklichkeit, Logik, Kreativität und vor allem Kommunikationstalent sind gefragt, denn ein konstruktives gemeinsames Spiel bedeutet in Minecraft vor allem eine gemeinsame Absprache. Diese findet über das direkte Gespräch, den Spielchat, Teamspeak, Skype und andere Wege statt. Besonders der soziale Aspekt des Spiels machte Minecraft zu einem der beliebtesten Spiele weltweit.

Mehr Inhalt durch Erweiterungen

Wie zuvor geschildert, bietet bereits das Grundspiel allein Raum für eine Fülle an unterschiedlichen Spielerlebnissen. Sein ganzes Potential schöpft Minecraft jedoch erst durch das Einfügen und Spielen zusätzlicher Inhalte bzw. Modifikationen, kurz „Mods“, aus.

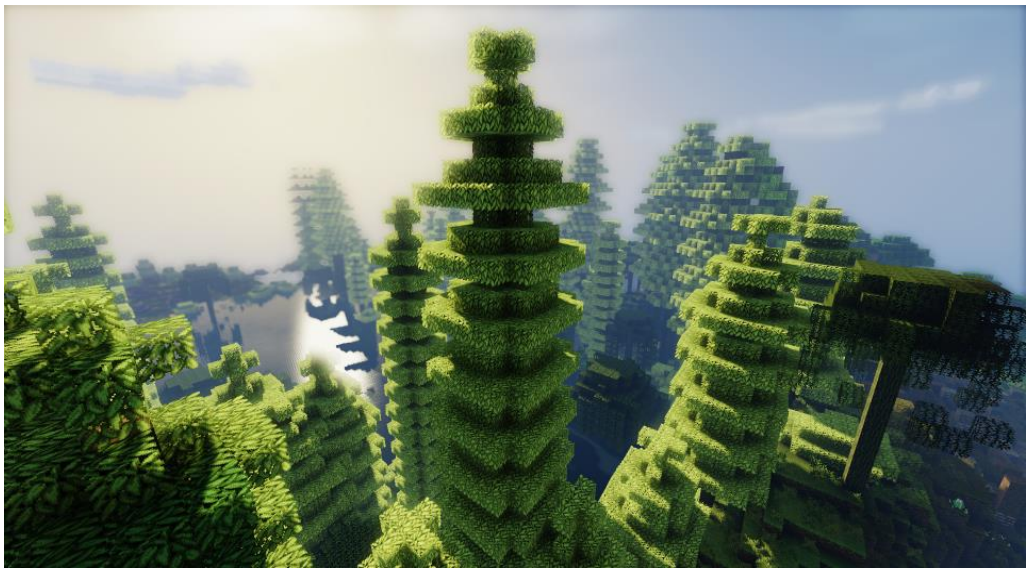
Dampfmaschinen, Möbel, exotische Pflanzen, Dinosaurier, Zauberei, Windenergie, kulinarische Gerichte? Mit Mods in Minecraft kein Problem! Meist entwickelt von Hobby-Programmierern und Grafikern sind diese Erweiterungen frei im Netz erhältlich und sorgen für viele weitere Stunden Spielspaß. Darüber hinaus können, je nach Thematik, komplexe Zusammenhänge vermittelt werden. Mods zu Bereichen wie Quantenphysik und Programmierung sind keine Seltenheit. Mit Hilfe dieser Zusatzmodule lässt sich das Spiel nach Belieben anpassen.

Genau in dieser Freiheit, die Minecraft seinen Spielerinnen und Spielern lässt, liegt auch die große Chance für einen konstruktiven Einsatz des Spiels im Bildungsbereich, die inzwischen weltweit wahrgenommen wird.

Minecraft als Bildungsplattform?

Vielerlei Orts erfreut sich Minecraft in pädagogischen Einrichtungen bereits großer Beliebtheit. Egal, ob im Bereich der Gemeinschaftsbildung, Naturwissenschaften, Kunst oder Informatik: Weltweit entwickeln mehr und mehr Lehrer gemeinsam mit ihren Schülern Minecraft-Projekte, die auf innovative Weise und nah an der Lebenswelt der Kinder Wissen vermitteln. So werden Werke bekannter Künstler nachgebaut und „begehbar“ gemacht oder zu einem Spaziergang durch eine Pflanzenzelle eingeladen. Außerdem erweist sich Minecraft für viele Kinder und Jugendliche als hilfreiches und vertrautes Artikulationswerkzeug. In Partizipationsprojekten zu Stadtgestaltung & Co. visualisieren sie ihre Zukunftsvisionen und Wünsche, sodass diese auch für die „Erwachsenenwelt“ greifbar werden.

Im 21. Jahrhundert gehört der natürliche Umgang mit digitalen Werkzeugen, wie es auch digitale Spiele sind, zum Alltag von Kindern und Jugendlichen. Warum davon also nicht mehr in die Schule holen und sich vertrauten Umgangsweisen anpassen? Das noch recht junge Prinzip des „Digital Game-based Learning“, das Lernen mit und durch digitale Spiele, findet in dieser Zeit immer mehr Anklang. Minecraft hat sich über die Zeit zu einer Art Paradebeispiel auf diesem Gebiet entwickelt.



Das Prinzip des Digital Game-based Learning

Ansatz für das Digital Game-based Learning in der Schule ist es meist, curricular vorgegebene Inhalte für Schüler mit Hilfe digitaler Spiele so aufzubereiten, dass sich behandelte Thematiken für sie interessanter und relevanter gestalten als im sonstigen formellen Unterricht und zu einer aktiven Auseinandersetzung motivieren.

Inhalte werden häufig als langweilig empfunden, wenn die Lernenden in ihnen keine konkrete Relevanz erkennen kann, da beispielweise der wichtige Praxisbezug fehlt.

Bei digitalen Spielen, wie Minecraft, handelt es sich um in sich geschlossene Umgebungen, mit festen Regelsätzen und Konsequenzen für bestimmte Aktionen, die der Spielerin oder dem Spieler ständiges Feedback geben und lerntechnisch mit einem Try-and-Error-Prinzip arbeiten, was einem natürlichen Lernverhalten entgegenkommt:

> **„Wir lernen nur durch Versuch und Irrtum. Unsere Versuche sind aber immer unsere Hypothesen. Sie stammen von uns, nicht von der Außenwelt. Von der Außenwelt lernen wir nur, dass gewisse unserer Versuche Irrtümer sind“ Karl Raimund Popper“**

Das Lernen in digitalen Spielen basiert auf dem aktiven Sammeln von Erfahrung, dem Experimentieren und Explorieren in einer in sich schlüssigen Welt, die durch zugrundeliegende Geschichten, Charaktere und Systeme Kontext schafft.

Genau wie in der Literatur- und Filmgeschichte, hat sich auch im Bereich der Computerspiele über die Zeit eine Fülle an unterschiedlichen Genres herausgebildet. So liegt es nahe, dass sich verschiedene Spiele auch in verschiedenem Maße für den Einsatz in der Schule eignen. Hier kann Minecraft auf besondere Art und Weise trumpfen: Durch die große Vielfältigkeit und Offenheit des Spiels lassen sich verschiedene Spielertypen sinnvoll zusammenbringen. Aber warum ist genau das so wichtig?

Verschiedene Spielertypen und ihr Lernverhalten

Jeder, der selbst in einer Schule tätig ist oder sich auch nur an seine Schulzeit zurückerinnert, wird es kennen: Der eine lernt lieber allein, der andere findet bestärkende Sicherheit in Lerngruppen. Die einen wollen immer alles selbst ausprobieren und experimentieren gerne, andere halten sich lieber an genaue Anweisungen des Lehrenden oder des Lehrbuchs. Je nach Klassenverband herrscht zusätzlich häufig ein sehr unterschiedliches Lerntempo, je nach Thema und Interesse der Schülerinnen und Schüler. Die Lehrenden haben es nicht leicht auf einem Niveau zu unterrichten, dass alle „mitnimmt“. Jede und jeder lernt offenbar anders.

Dass die unterschiedlichen Lerntypen vieles mit den verschiedenen Spielertypen gemeinsam haben, zeigt das Modell des britischen Computerspielforschers Richard Bartle. Er unterscheidet zwischen Killer, Socializer, Explorer und Achiever, alles Typen, die sich unter bestimmten Voraussetzungen auch in Minecraft antreffen lassen und dort wohlfühlen.

Killer

Bartles „Killer“ liebt das actionreiche Spiel und misst sich gern im freundschaftlichen Kampf und in direkter Auseinandersetzung mit anderen Spielern. Eine (Spiel-) Umgebung, die sich nach Belieben auseinandernehmen und nach ihrem Geschmack neuaufbauen lässt, ist wie für ihn geschaffen. Verständlich, dass er sich in einer Spielwelt wie Minecraft sie bietet, oft zu Hause fühlt. Das Zerstören und Neubauen von Strukturen ist schließlich einer der Hauptbestandteile des Spiels.

Der Killer ist der geborene Eroberer und Jäger. Er genießt es in gewisser Weise, Macht über andere Spieler zu haben, was sich aber meist auch darin widerspiegelt, dass er für schwächere Mitglieder seiner Gruppe einsteht und sie gegen „gegnerische Angriffe“ loyal verteidigt. In der

Realität ist ein Mensch, der in der virtuellen Welt eher der Fraktion der Killer angehört, oft körperlich sehr aktiv, treibt Sport und liebt den Wettkampf. Die Auseinandersetzung mit (Lern-) Themen ohne Konflikt und Herausforderung durch andere empfindet er eher als langweilig, sodass er nicht selten dadurch auffällt, dass er sich selbst diese Reibungsherde schafft, wenn kein entsprechendes Angebot vorliegt. Manche Lehrkraft mag den ein oder anderen Killer bereits als liebenswerten, vereinnahmenden „Klassenclown“ kennengelernt haben. In Minecraft trifft man den Killer typischerweise auf Wettkampf-Servern oder bei großen Bauprojekten an.

Socializer

Auch für den Socializer (engl. von „socialize“: „sozialisieren“) spielt der direkte Kontakt mit anderen Spielern eine große Rolle, jedoch auf einer harmonischeren Basis als der Killer sie sucht. Ihm ist es wichtig, sich ein gemeinschaftliches Geflecht einzufügen, dieses zu pflegen oder sogar selbst ein neues zu schaffen. Weiß er nicht weiter, wendet er sich meist an andere Spieler und bittet um Rat, was zu tun ist. Im Gegenzug findet der Socializer Erfüllung darin, seine Erfahrung wiederum mit neuen Spielern zu teilen, bereitwillig Auskunft zu geben und zwischen einzelnen Spielerinnen und Spielern weiterzuvermitteln.

Er organisiert die soziale Gemeinschaft, verwaltet, weist Aufgaben zu und interessiert sich generell für die Belange und Wünsche der Gruppe. Gemeinsam mit anderen etwas im Spiel zu (er)schaffen steht für Socializer eindeutig im Vordergrund. Äquivalent dazu, lernen Socializer in der Schule auch lieber gemeinsam mit anderen und fragen bei Unsicherheit darüber, wie eine Aufgabe zu lösen sei, lieber zweimal nach, anstatt drauflos zu stürmen.

Die Bestätigung durch die Gemeinschaft gibt den Socializern festigende Sicherheit. So ist es auch nachvollziehbar, dass sie in Minecraft eher selten Einzelprojekte angehen, sind dafür aber häufig auf Bau- und Rollenspielservern zu finden, engagieren sich dort unterstützend als Helfende oder sind als talentierte Händlerinnen und Händler unterwegs, die untereinander Spielgegenstände tauschen. Sie sorgen immer für Gesprächsstoff. Die Interaktion mit anderen Spielerinnen und Spielern ist für Socializer das A und O in Minecraft.

Explorer

Im Gegensatz zum Socializer, zählt für den Explorer (engl. von „explore“: entdecken) nicht so sehr die Interaktion mit seinen Mitspielenden, dafür aber umso mehr mit der ihn umgebenden Welt. Er liebt es, auch allein (!) auf Abenteuerreise zu gehen und jeden Winkel der Spielwelt zu durchstöbern, Geheimnisse zu entdecken und sämtliche Informationen über Spielfiguren, Spielobjekte und Spielgeschichte, sofern vorhanden, zu sammeln. Fühlt sich der Explorer zu sehr vom Spiel eingeschränkt, wird er häufig versuchen, Schlupflöcher in der Entwicklung zu finden, die es ihm gestatten, das Spiel anders als herkömmlich gedacht zu spielen.

Für kämpferische Auseinandersetzungen interessieren sich Explorer kaum, da diese bei der Entdeckungsreise eher als störend empfunden werden können. So wundert es nicht, dass oft

gegenseitige Antipathien zwischen reinen Explorern und Killern entstehen. Kontakt mit Socializern interessiert meist nur, sofern sich dadurch Wissen über die Spielwelt vermehren lässt. Repetitive Aufgaben schrecken den Explorer eher ab.

Auf die reale Welt übertragen sind Explorer meist experimentierfreudige Lernende, die sich gern für sich zurückgezogen mit einem Thema auseinandersetzen und gerne selbst entdecken. Sie fühlen sich am wohlsten, wenn sie Art und Weise, sowie auch das Lerntempo selbst bestimmen können. Lerngruppen sind weniger nach ihrem Geschmack, da sie sich hier gegebenenfalls stark anpassen müssen.

Interessiert den Explorer ein Aspekt besonders, zieht er alle möglichen Informationsquellen, die er finden kann, zur Weiterbildung heran und kann nahezu darin versinken. Sind diese Voraussetzungen gegeben können Explorer zu wahren Experten für ein Fachgebiet aufsteigen. Minecraft, welches in der Lage ist, nahezu unendliche Welten zu generieren, bietet diesem Spielertypus eine ausgiebige Spielwiese. Die zahlreichen Modifikationen, die stetig Zusatzinhalte für das Spiel bereithalten, sorgen immer wieder für ganz neue Spielerlebnisse – perfekt für den Explorer!

Achiever

Hohe Punktzahlen, Ranglisten, ein hohes Spielerlevel und die besten Spielgegenstände sind alles für den Achiever (engl. von „achieve“: „erreichen, erzielen“) nach Bartle. Sein größtes Ziel ist es, auf der Bestenliste ganz oben zu stehen. Bietet ein Spiel verschiedene Wege, an ein Ziel zu gelangen, muss der Achiever sie, nach eigenem Anspruch, möglichst alle meistern, es auf einfache Weise zu erreichen, genügt oft nicht. Dabei schreckt er auch vor sehr repetitiven, monotonen Aufgaben nicht zurück, denn diese helfen ihm sein Spiel zu perfektionieren.

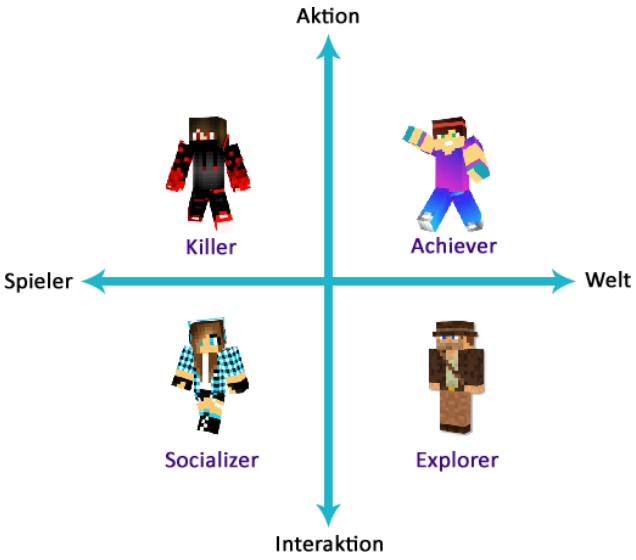
Während es für den Explorer ein langweiliger Graus wäre, eine Aufgabe à la „Springe hundert Mal hintereinander über die gleiche Schlucht, ohne herunterzufallen“ zu bewältigen (, denn das bringt ihn auf seiner persönlichen Entdeckungsreise nicht weiter), läuft der Achiever bei solch einer Herausforderung zur Höchstform auf.

Dabei lässt er sich nicht nur vom Spiel selbst, sondern vor allem auch von anderen Spielerinnen und Spielern zur Lösung ungewöhnlicher Aufgaben motivieren. Achiever vergleichen und messen sich untereinander gern anhand ihrer Errungenschaften (engl. „Achievements“) und reizen alles aus, was das Spiel ihnen zu bieten hat. Die Abgrenzung zum klassischen Killer besteht darin, dass Herausforderungen zwar auch in einer kämpferischen Auseinandersetzung mit anderen Spielerinnen und Spielern stattfinden können, aber nicht müssen. Dem Achiever geht es nicht darum, andere Spieler kontrollieren zu können. Das pure Erreichen und „Vorzeigen-können“ eines vorgegebenen Ziels ist für ihn Anerkennung genug.

Der Achiever sammelt im übertragenen Sinne Fähigkeits-Abzeichen, die von seiner Kompetenz auf diversen Gebieten zeugen. In der Schule sind Achiever häufig gute Sportler oder „sammeln“ generell gern gute Noten. Je nach Interesse gibt es unter ihnen auch schnelle

Kopfrechner und gute „Auswendiglerner“. Lerngruppen sehen sie eher als weitere Möglichkeit, sich direkt miteinander zu vergleichen. Im Gegenzug kann es aber ebenso eine reizvolle Herausforderung sein, allein zu lernen und zu beweisen, was aus eigener Kraft zu erreichen möglich ist.

Es kommt ganz auf die Aufgabenstellung an und ob diese vom Achiever als anspruchsvoll genug befunden wird. Im Vordergrund steht stets der Vergleich mit anderen. So auch in Minecraft: Wer hat das höchste Gebäude gebaut? Wer hat die meisten Gegner besiegt? Wer hat die meisten Erfahrungspunkte gesammelt?



Beziehungen der Spielertypen untereinander

Wie man sich vorstellen kann, sind die Spielertypen nach Richard Bartle in Reinform eher selten. Vielmehr hat jeder Spieler in gewissem Maße mehr oder weniger Anteile der verschiedenen Typen. Im 1999 entwickelten Bartle Test (of Gamer Psychology) werden, mit Hilfe einer Fragenserie von 30 Fragen, genau diese Anteile ermittelt. Im Netz ist dieser Test in vielen Variationen kostenfrei zugänglich.

Die vorangegangenen Schilderungen haben es bereits deutlich gemacht: Genauso wie es in einer Schulklasse unterschiedliche Charaktere und Lerntypen gibt, verhält es sich in auch in der Computerspielwelt. Das Verhältnis zueinander ist, aufgrund der unterschiedlichen Vorlieben, mal harmonischer, mal schwieriger gefärbt. Nicht jede und jeder kann jeden leiden. Das ist ganz normal.

Recht bekannt ist in der Computerspielforschung, aber auch in der Industrie, das „Character theory chart“ nach Bartle, ein Diagramm, das die einzelnen Spielertypen miteinander und in

Bezug auf Aktion und Interaktion mit der Spielwelt in Beziehung setzt (siehe Grafik) und hilft die Zusammenhänge besser zu verstehen und zusammenzufassen.

So haben beispielsweise Socializer und Explorer ihre Vorliebe für die Interaktion gemein, jedoch bevorzugen die Socializer eher die Interaktion mit anderen Spielern und die Explorer mit Elementen der Spielwelt. Killer sind weniger auf die Spielwelt als solche fokussiert, sondern auf das Agieren und/oder Instrumentalisieren gegen andere Spieler und stehen damit maximal weit von den Explorern entfernt, die sich durch Killer in ihrem Spielerlebnis, der Entdeckungsreise, oft gestört fühlen können. Ebenso wenig können reine Achiever puren Socializern abgewinnen, da sie im gemeinsamen Gespräch, so wie die Socializer es lieben, nur wenig Motivation finden. Achiever wollen viel lieber Herr über alle Herausforderungen werden, die ihnen die Spielwelt zu bieten hat.

Ähnlich wie bei der Wissensvermittlung in der Schule, grenzt es dementsprechend auch in der Welt der Computerspiele oft an ein wahres Kunststück, die verschiedenen Spielertypen konstruktiv zusammenzubringen. Minecraft ist ein Kandidat, dem dieses Kunststück gelingen kann.

Minecraft bietet Spielerinnen und Spielern eine offene Welt, die sie sich je nach Belieben gestalten können. Hier liegen gleichzeitig die Herausforderung, aber auch die Chance, das Spielerlebnis für alle Spielertypen attraktiv zu gestalten.

Kinder und Jugendliche als Expertinnen und Experten

Das dies auch im Bildungsbereich möglich ist, zeigen die bereits zahlreich durchgeführten, pädagogisch betreuten Projekte mit Minecraft. Eine wesentliche Rolle spielt hier das spielerische Expertentum der Kinder. Nicht häufig haben sie im sonstigen Unterricht die Gelegenheit, den Spieß einmal umzudrehen und den Lehrenden in die Rolle des Lernenden zu versetzen, denn im Gegensatz zu diesem haben sie in den meisten aller Fällen viel mehr Spiel- und Bedienkompetenz, so auch in Minecraft.

Auch wenn viele Pädagoginnen und Pädagogen zunächst genau diesen Umstand fürchten: dieses Expertentum zuzulassen, lohnt sich! Lassen Sie sich über die Grundlagen in Minecraft von ihren Schülerinnen und Schülern aufklären und Sie werden in so manch begeistert strahlendes Gesicht blicken. Die Grundlage dafür, schulische Inhalte und Spiel zusammenzubringen ist nun eine ganz andere, viel motivierendere. Sie als Lehrende oder Lehrender sind die Expertin oder der Experte für die Wissensinhalte, ihre Kinder für das Spiel. Ihre Schülerinnen und Schüler freuen sich, wenn auch Sie bereit sind, Neues zu lernen und dies nicht nur von Ihren Schützlingen fordern.

Minecraft kann ein hilfreiches Werkzeug sein, um die verschiedenen Lerntypen abzufangen und Ihnen bei Ihrem Unterricht unter die Arme zu greifen, ganz allein organisiert es diesen natürlich nicht. Für einen sinnvollen Einsatz sollten Sie sich zu einem Mindestmaß mit dem Spiel auseinandergesetzt haben und die grundlegende Handhabung beherrschen. Wissen Sie mal nicht weiter, greifen Sie gern auf Ihre Kinder als Lernquelle zurück!

In den weiteren Kapiteln begleiten wir Sie Schritt für Schritt von der Vorbereitung für den Einsatz im Unterricht bis hin zur Durchführung der einzelnen Unterrichtseinheiten.

Vorbereitung und technischer Rahmen

Eine freundliche Warnung vorweg: Sie werden während des Spielens mit Ihren Schülerinnen und Schülern sehr wahrscheinlich das ein oder andere spielinterne Problem zu lösen haben. Das ist im Minecraft-Universum aber ganz normal.

Unter Kindern und Jugendlichen hat es sich zu einer Art Sport entwickelt, Schlupflöcher in einer Minecraft-Welt ausfindig zu machen und diese auszunutzen. Mal geht dabei vielleicht ein Spielgegenstand kaputt, der nicht kaputtgehen sollte, mal benötigt eine Spielerin oder ein Spieler Ihre Hilfe, um aus einer Sackgasse zu gelangen, in die er nicht hätte gelangen sollen. All das kann passieren und Sie sollten definitiv darauf gefasst sein.

Sofern sinnvoll, haben wir an diversen Punkten der Code your Life Minecraft-Welt Vorkehrungen getroffen, um derartigen Umständen vorzubeugen. Zu viele Restriktionen nehmen dem beliebten Spiel jedoch den Spielspaß und schränken seine Möglichkeiten stark ein. Sollte einmal etwas schiefgehen, haben wir in jeder Lektion des Curriculums die gängigsten Fragen, auftauchenden Probleme und Lösungsvorschläge für Sie zusammengestellt. Behalten Sie im Hinterkopf, dass es sich bei Minecraft immer noch um ein kommerzielles digitales Spiel aus dem Unterhaltungssektor handelt, das nicht ursprünglich und explizit für den Bildungsbereich entwickelt wurde. Genau hier liegt aber nach wie vor die große Chance für die Entwicklung einer zielgruppengerechten, innovativen Lernkultur.

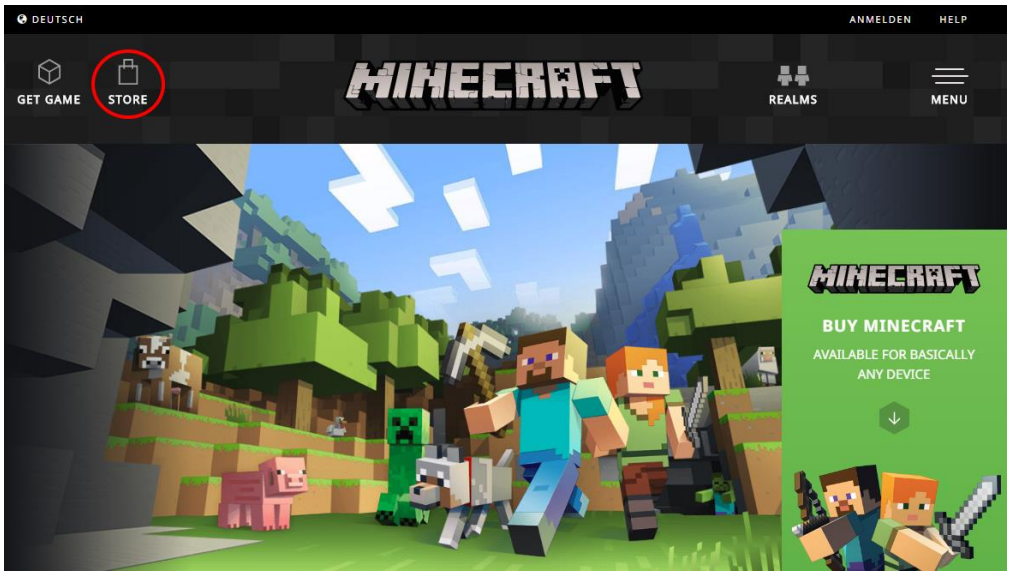
Unser Code your Life Minecraft-Curriculum ist so aufgebaut, dass es an das Vorwissen aus dem Grundprogramm anschließt und geht von der Annahme aus, dass Sie dieses zuvor mit Ihren Schülerinnen und Schülern bereits durchlaufen haben. Kenntnisse über grundlegende Strukturen und Prinzipien von Schleifen, Bedingungen und Verschachtelungen werden dementsprechend vorausgesetzt.

Bevor Sie sich mit Ihren Kindern und unserem Curriculum in die spannende Welt der Klötzchen stürzen, empfehlen wir, dass Sie sich selbst einen Moment Zeit nehmen, um sich nicht nur mit den einzelnen Lektionen, sondern auch mit den Basics von Minecraft selbst vertraut zu machen. Lassen Sie sich auch dabei gerne von Ihren Schülern unterstützen!

Systemvoraussetzungen

Um unserer Curriculum mit einer Schülergruppe durchführen zu können, sollten Sie in Ihrer Einrichtung Zugriff auf ein stabiles Netzwerk haben. Ein Internetzugang ist einmalig für die Installation des Spiels und unseres Pakets notwendig, für das spätere Spielen genügt die lokale Netzwerkverbindung. Sofern es sich einrichten lässt, sollte jeder Schülerin und jedem Schüler ein Rechner zur Verfügung stehen. Auch eine Besetzung mit zwei Kindern pro Gerät ist möglich, mehr als zwei Teilnehmende sollte aber auf keinen Fall an einem Gerät sitzen.

Je Gerät benötigen Sie wiederum eine Minecraft-Lizenz. Sollten Sie diese noch nicht besitzen, können Sie diese im Store auf https://minecraft.net käuflich erwerben.



Minecraft lässt sich prinzipiell auf verschiedenen Systemen spielen. Wir werden uns in dieser Handreichung jedoch auf den Umgang in Windows-Betriebssystemen konzentrieren.

Eine Empfehlung für die technischen Mindestkonfigurationen der Geräte:

- Windows 7, 8, 10
- 4 GB Arbeitsspeicher
- 500 MB Festplattenspeicher
- Aktuelles Java (je nach Gerät 32 oder 64 Bit)
- Prozessor mit mindestens 2,5 GHz
- Grafikkarte mit mindestens 512 MB Grafikspeicher

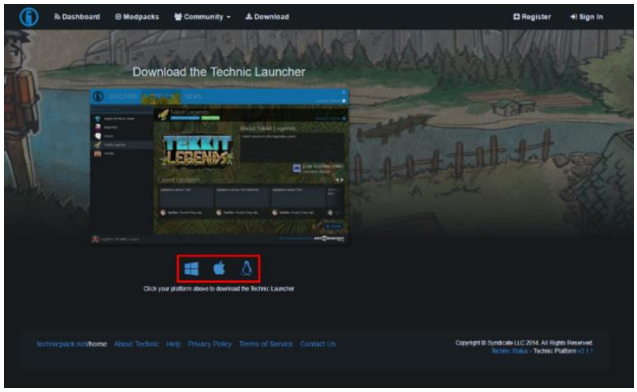
Sollten Sie sich nicht sicher sein, ob ihre Systeme die Mindestansprüche erfüllen, wenden Sie sich am besten an den oder die IT-Zuständige/n.

Installation

Sind alle technischen Voraussetzungen gegeben, kann es mit der Installation von Minecraft weitergehen, die Sie auf jedem Gerät vornehmen müssen. Für unser Minecraft-Curriculum nutzen wir dabei nicht den normalen Launcher, über den das Spiel üblicherweise gestartet wird. Wir greifen auf den Minecraft Technic Launcher zurück, da er die Installation des Code your Life Modpacks (ein für das Curriculum extra zusammengestelltes Paket an Modifikationen) erleichtert.

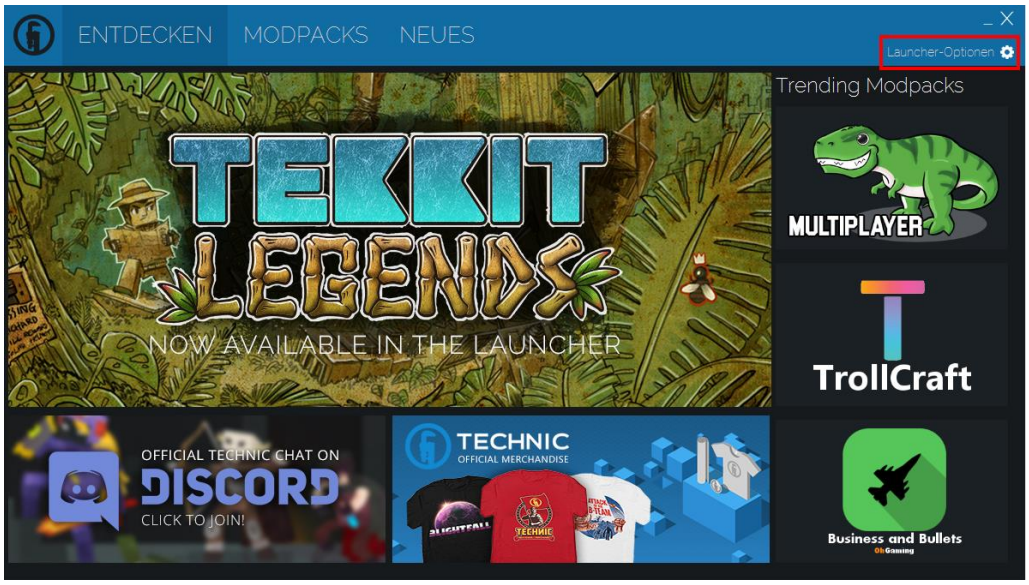
Technic Launcher

Zunächst müssen Sie sich den Technic Launcher downloaden. Gehen Sie dazu am besten auf http://www.technicpack.net/download und wählen dort den Download für ihr entsprechendes Betriebssystem.

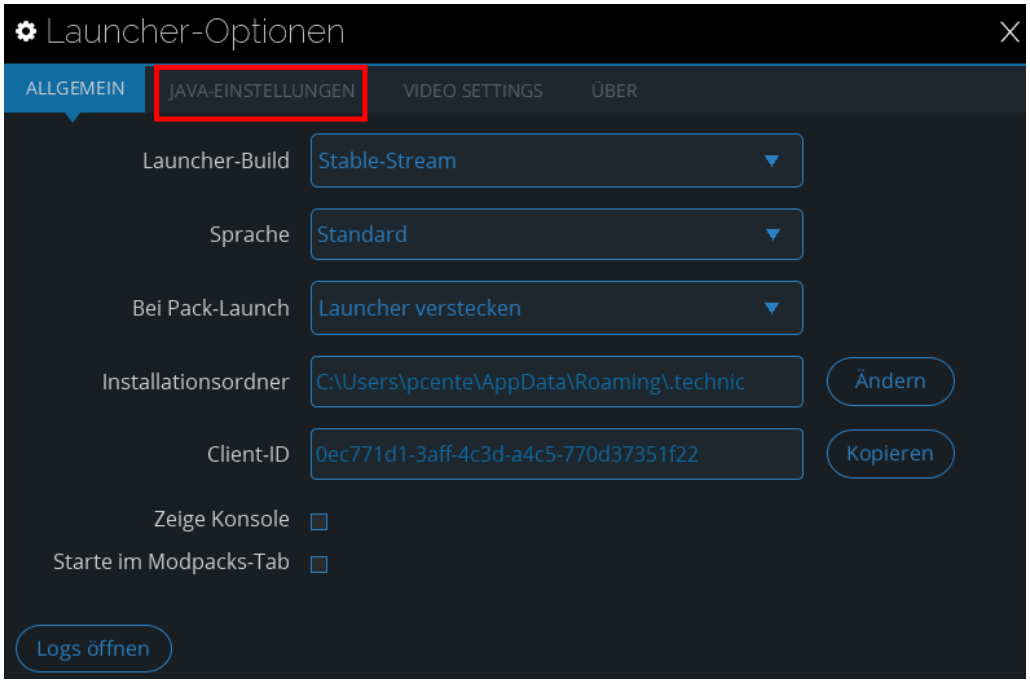


Anschließend führen Sie die heruntergeladene TechnicLauncher.exe aus. Sie werden nun aufgefordert, Ihre Minecraft-Accountdaten einzugeben, d.h. die je Account die E-Mail-Adresse, über welche Sie den Account registriert und das dazugehörige Passwort, dass Sie bei der Anmeldung vergeben haben.

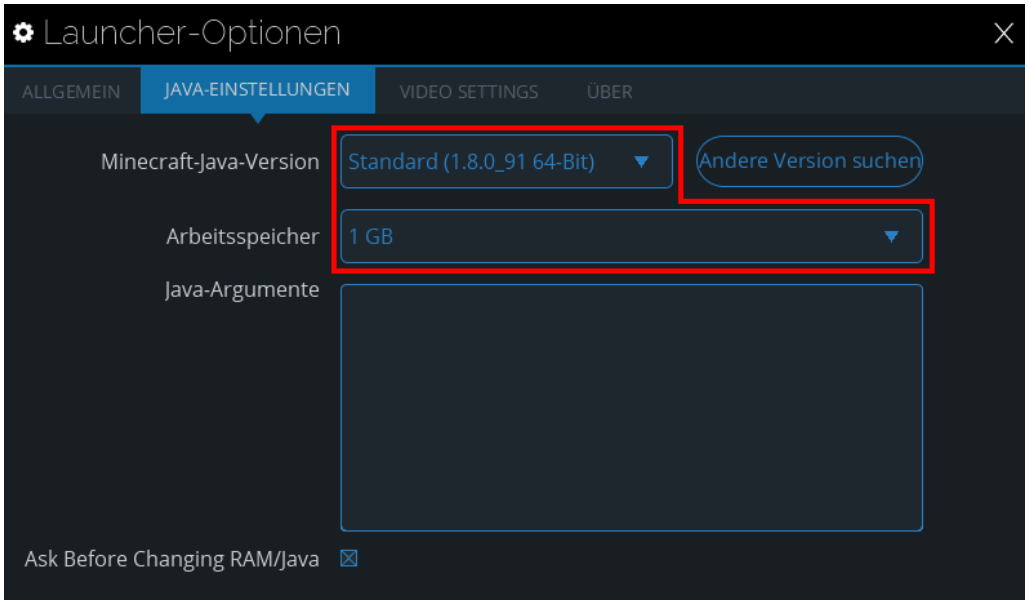
Nach dem Login öffnet sich der eigentliche Launcher. Bei der erstmaligen Nutzung sollten Sie sich zunächst vergewissern, dass die Grundeinstellungen des Launchers an Ihr System angepasst sind. Klicken Sie dafür oben rechts neben dem kleinen Zahnrad auf „Launcher-Optionen“:



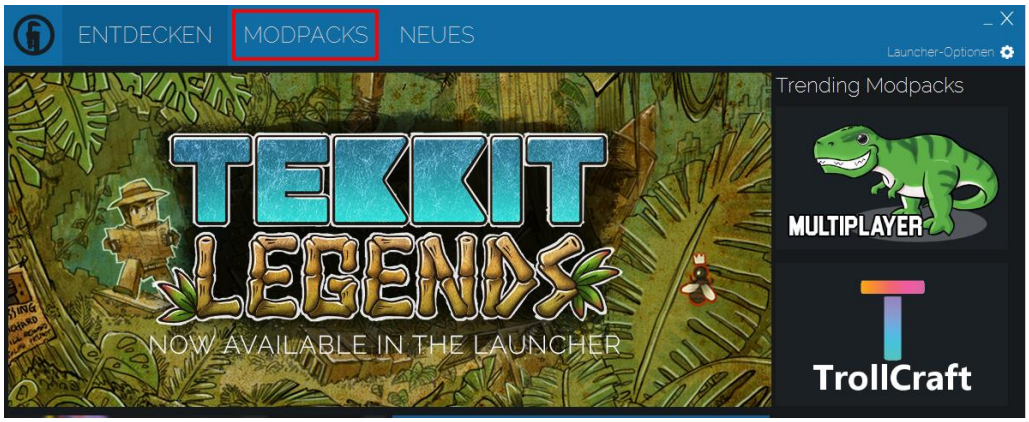
Wählen Sie ihm neu geöffneten Fenster nun den Reiter „Java-Einstellungen“.



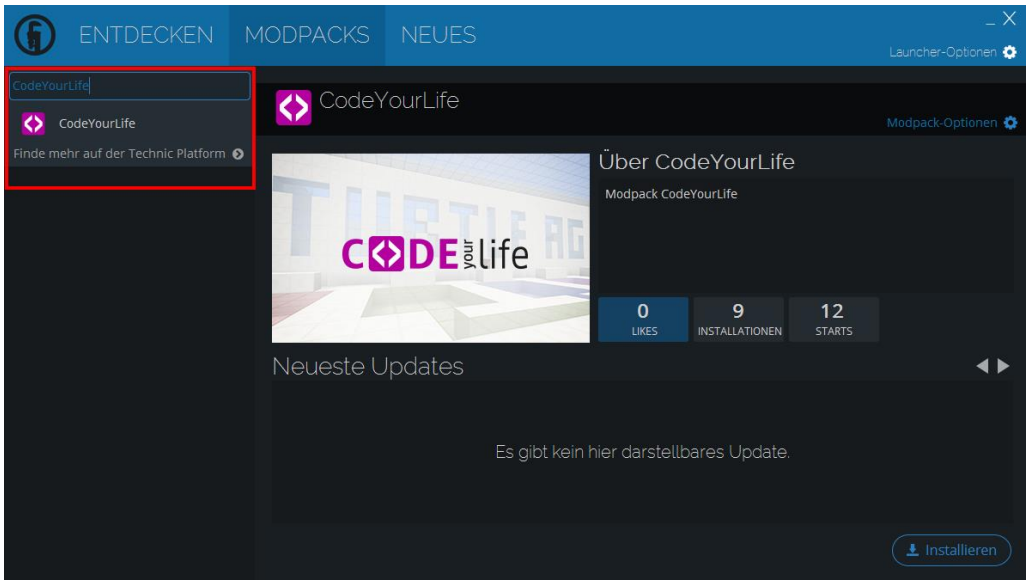
Neben „Minecraft-Java-Version“ sollte hier in der Regel automatisch das aktuelle Java ausgewählt sein. Sollte dies nicht der Fall sein, können Sie über das Drop-Down-Menü die passende Version wählen. Unter „Arbeitsspeicher“ ist standardmäßig 1 GB auswählt. Normalerweise sollte diese Menge genügen, um flüssig spielen zu können. Verfügt Ihr Gerät über mindestens 6 GB Arbeitsspeicher, empfehlen wir im Drop-Down-Menü 1,5 GB zu wählen, ab 8 GB Arbeitsspeicher können Sie aber auch 2 GB wählen. Die Performanz des Spiels kann so merklich verbessert werden. Sind Sie sich unsicher, ziehen Sie am besten auch hier einen IT-Kundigen zu Rate.



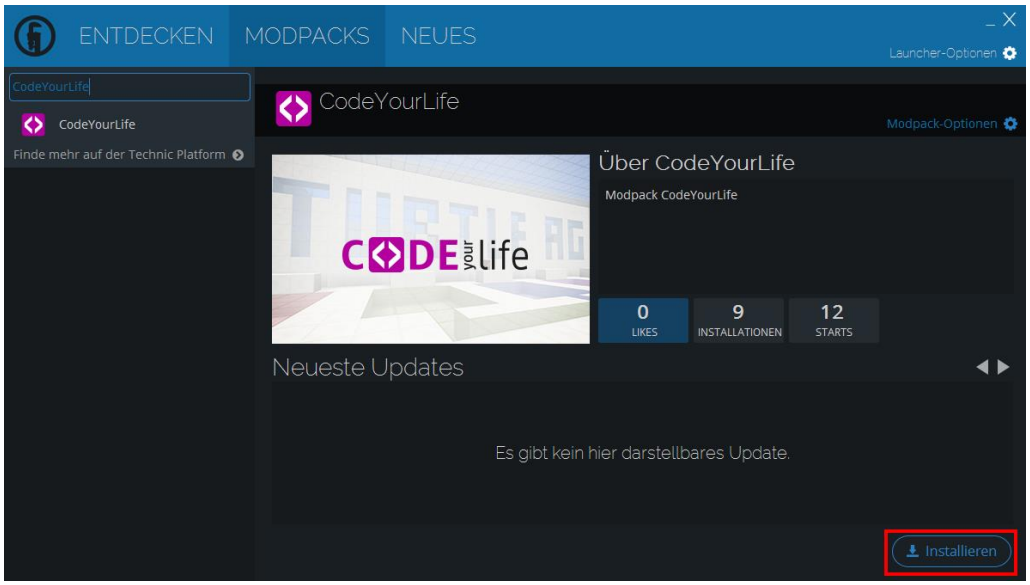
Schließen Sie das Optionsfenster mit einem Klick auf das X in der oberen rechten Ecke und klicken Sie nun in der Hauptansicht auf den Reiter „Modpacks“:



Geben Sie „CodeYourLife“ (achten Sie auf die korrekte Schreibweise) in das Suchfeld in der oberen linken Ecke ein:



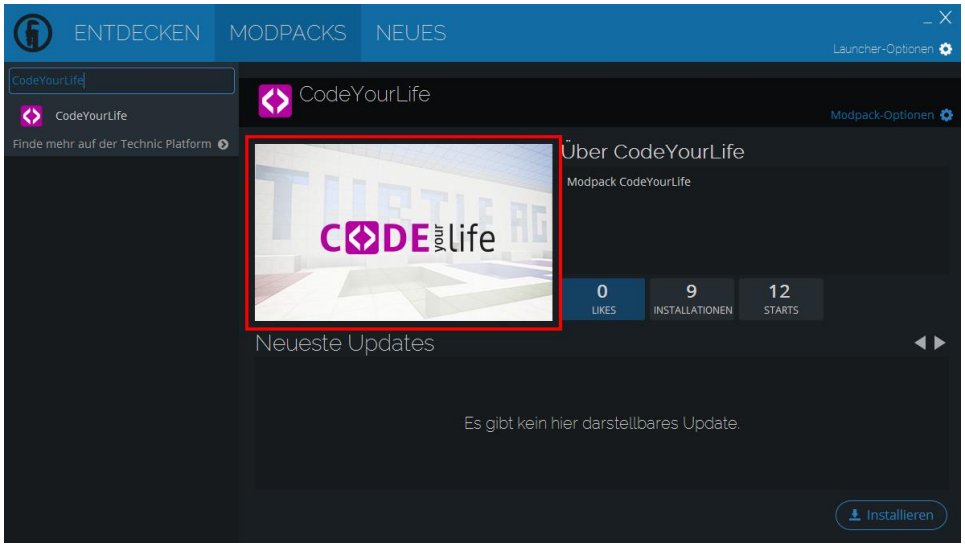
Wählen Sie das gefundene Paket in der linken Liste aus und klicken Sie unten rechts im Launcher auf „Installieren“. Das Code your Life Modpack wird nun auf Ihren Rechner heruntergeladen und entpackt. Je nach Internetverbindung kann dieser Vorgang mehrere Minuten in Anspruch nehmen.



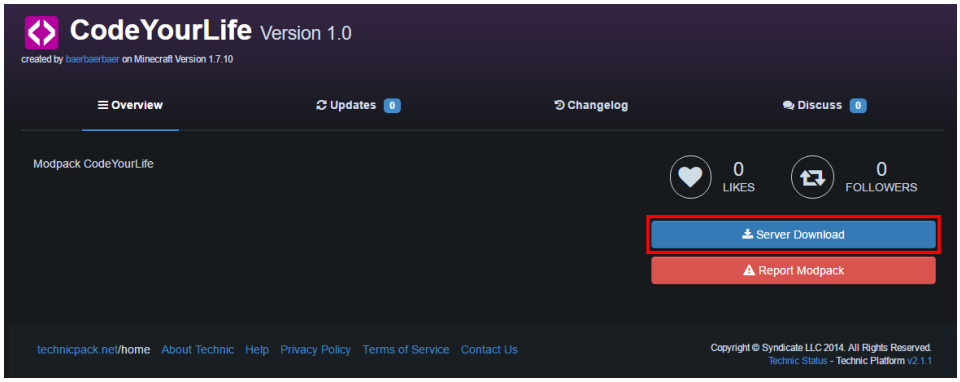
Sobald das Paket fertig installiert wurde, erscheint nun der Button mit der Aufschrift „Spielen“. Gleich kann es losgehen! Was noch fehlt, ist ein laufender Server mit der Code your Life Minecraft-Welt, auf den sich später alle verbinden sollen.

Den Server starten

Hinweis: Den Server mit der Minecraft-Welt müssen Sie nur auf einem einzelnen Gerät starten und dort laufen lassen. Es empfiehlt sich meist, dafür den leistungsstärksten Rechner zu wählen, der Ihnen zur Verfügung steht, da dieser in der Regel eine größere Rechenlast zu bewältigen hat. Dies sollte außerdem der Rechner sein, an dem Sie selbst spielen. So können Sie gegebenenfalls schneller einige Einstellungen vornehmen, wenn nötig. Sollten Sie den Technic Launcher nicht mehr geöffnet haben, starten Sie diesen erneut und melden sich an. Klicken Sie nun wieder in der oberen Leiste auf „Modpacks“ und wählen aus der Liste links „CodeYourLife“. Klicken Sie nun direkt auf das Vorschaubild des Modpacks:

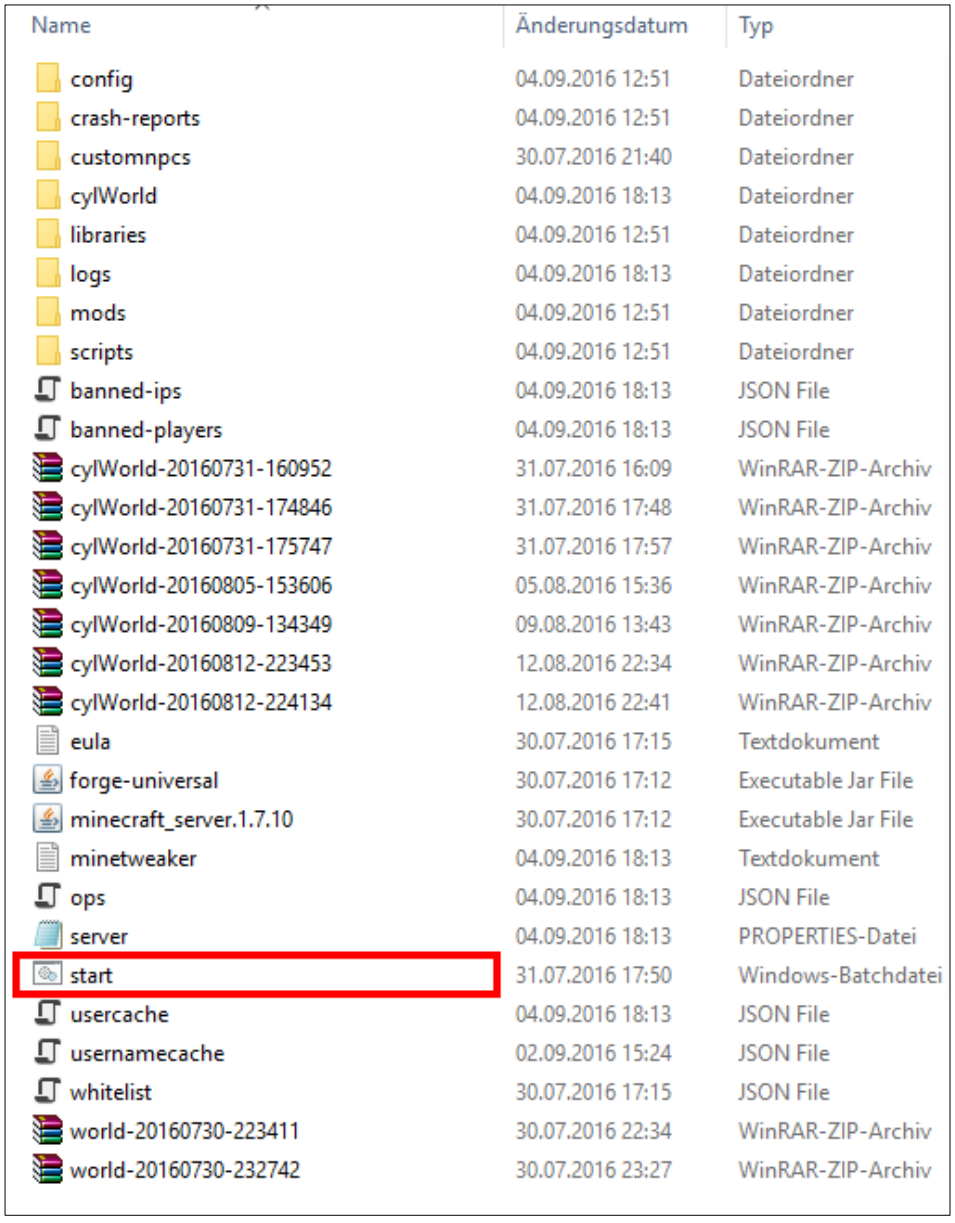


In Ihrem Standard-Browser öffnet sich nun die Projektseite des Modpacks auf der Webseite des Technic Launchers. Klicken Sie dort auf den Button mit der Aufschrift „Server Download“.



Die Datei codeYourLife_server.zip wird nun auf Ihren Computer heruntergeladen. Bevor Sie den Server nun mit der passenden Welt starten können, müssen Sie die .zip-Datei per Klick noch an einen Ort ihrer Wahl entpacken.

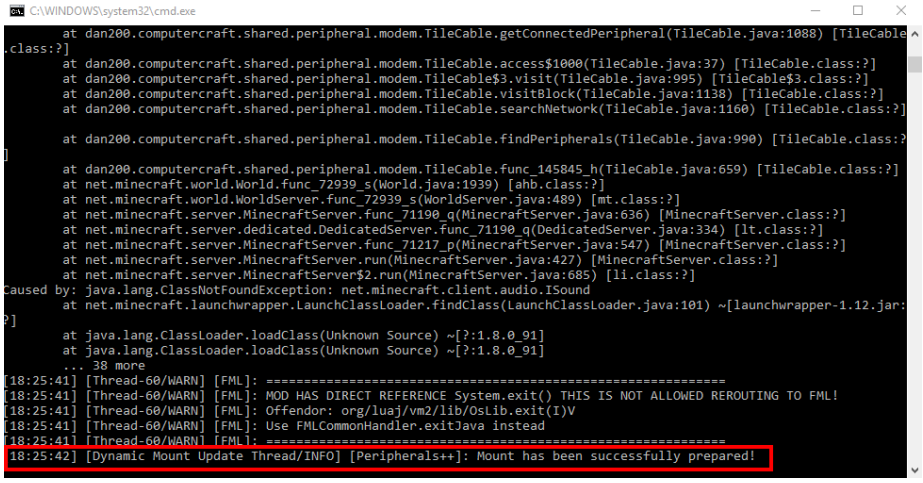
Sobald Sie mit dem entpacken fertig sind, navigieren Sie zum entsprechenden Speicherort und öffnen den Ordner „codeYourLife_server“. Starten Sie dort die start.bat.



Name	Änderungsdatum	Typ
config	04.09.2016 12:51	Dateiordner
crash-reports	04.09.2016 12:51	Dateiordner
customnpcs	30.07.2016 21:40	Dateiordner
cylWorld	04.09.2016 18:13	Dateiordner
libraries	04.09.2016 12:51	Dateiordner
logs	04.09.2016 18:13	Dateiordner
mods	04.09.2016 12:51	Dateiordner
scripts	04.09.2016 12:51	Dateiordner
banned-ips	04.09.2016 18:13	JSON File
banned-players	04.09.2016 18:13	JSON File
cylWorld-20160731-160952	31.07.2016 16:09	WinRAR-ZIP-Archiv
cylWorld-20160731-174846	31.07.2016 17:48	WinRAR-ZIP-Archiv
cylWorld-20160731-175747	31.07.2016 17:57	WinRAR-ZIP-Archiv
cylWorld-20160805-153606	05.08.2016 15:36	WinRAR-ZIP-Archiv
cylWorld-20160809-134349	09.08.2016 13:43	WinRAR-ZIP-Archiv
cylWorld-20160812-223453	12.08.2016 22:34	WinRAR-ZIP-Archiv
cylWorld-20160812-224134	12.08.2016 22:41	WinRAR-ZIP-Archiv
eula	30.07.2016 17:15	Textdokument
forge-universal	30.07.2016 17:12	Executable Jar File
minecraft_server.1.7.10	30.07.2016 17:12	Executable Jar File
minetweaker	04.09.2016 18:13	Textdokument
ops	04.09.2016 18:13	JSON File
server	04.09.2016 18:13	PROPERTIES-Datei
start	31.07.2016 17:50	Windows-Batchdatei
usercache	04.09.2016 18:13	JSON File
usernamecache	02.09.2016 15:24	JSON File
whitelist	30.07.2016 17:15	JSON File
world-20160730-223411	30.07.2016 22:34	WinRAR-ZIP-Archiv
world-20160730-232742	30.07.2016 23:27	WinRAR-ZIP-Archiv

Der Server wird nun gestartet, das Server-Fenster öffnet sich. Anhand der durchlaufenden Meldungen sehen Sie, dass die Serverdateien geladen werden.

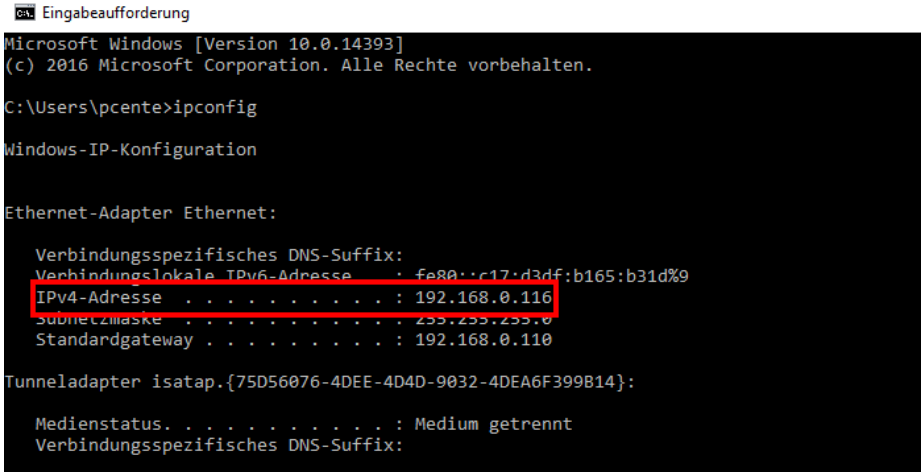
Sobald Sie die Nachricht „[Dynamic Mount Update Thread/INFO] [Peripherals++]: Mount has been successfully prepared!“ lesen können bzw. keine neuen Meldungen mehr geladen werden, ist der Server mit der Code Your Life Welt fertig hochgefahren:



Nun können sich alle Rechner im gleichen Netzwerk mit der Serverwelt verbinden. Benötigt wird dafür nur noch die entsprechende Netzwerkadresse bzw. IP.

Die Server-IP identifizieren

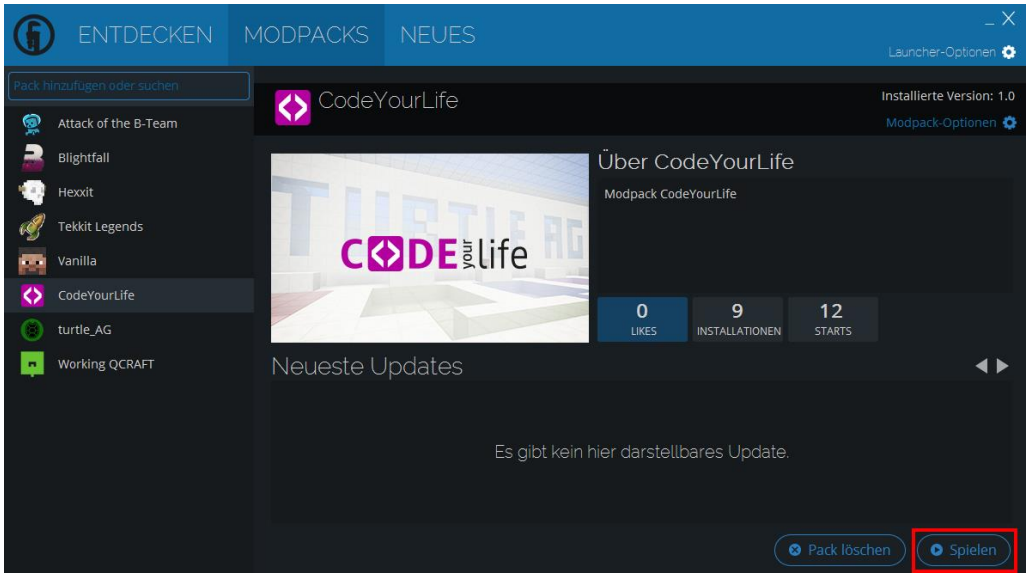
Wissen Sie Ihre IP nicht auswendig, begeben Sie sich in das Windows-Start-Menü, geben dort „cmd“ ein und bestätigen mit Enter. Wieder öffnet sich ein neues Fenster. Geben Sie hier einfach „ipconfig“ ein und bestätigen wieder mit Enter. Suchen Sie nun nach einem Eintrag wie etwa „IPv4-Adresse“.



Die Nummer, die Sie dort finden, ist Ihre IP-Adresse. Notieren Sie sich diese Nummer! Jetzt kann losgespielt werden.

Minecraft starten und die Code your Life Minecraft Welt betreten

Wenn der Server läuft, navigieren Sie im Technic Launcher wieder zum Code your Life Modpack, sollten Sie nicht mehr dort sein. Klicken Sie hier auf „Spielen“ in der unteren rechten Ecke.



Minecraft öffnet sich und lädt.

Sobald fertig geladen, wählen Sie nun „Mehrspieler“ im Hauptmenü. Klicken im folgenden Bildschirm auf „Direkt verbinden“ und geben anschließend die IP, die Sie sich zuvor notiert haben, in das Feld ein. Danach klicken Sie auf Server beitreten:

Nach einer kurzen Ladezeit stehen Sie nun mit Ihrer Minecraft-Figur am Startpunkt unserer Code your Life Minecraft Welt. Herzlichen Glückwunsch, jetzt kann es an die Curricula gehen.

Jedes Mal, wenn Sie oder Ihre Schülerinnen und Schüler sich mit dem Server verbinden wollen, gilt es die hier beschriebene Routine auszuführen.

Die Curricula

Nun kann es in die Praxis gehen. Die Minecraft Curricula von Code your Life nehmen Ihre Schülerinnen und Schüler nicht nur in die von ihnen so sehr geliebte Klötzchenwelt mit, sondern führen an die spannende Skriptsprache Lua heran. Genutzt wird dafür intern die Mod „Computercraft“.

Warum Computercraft? Warum Lua?

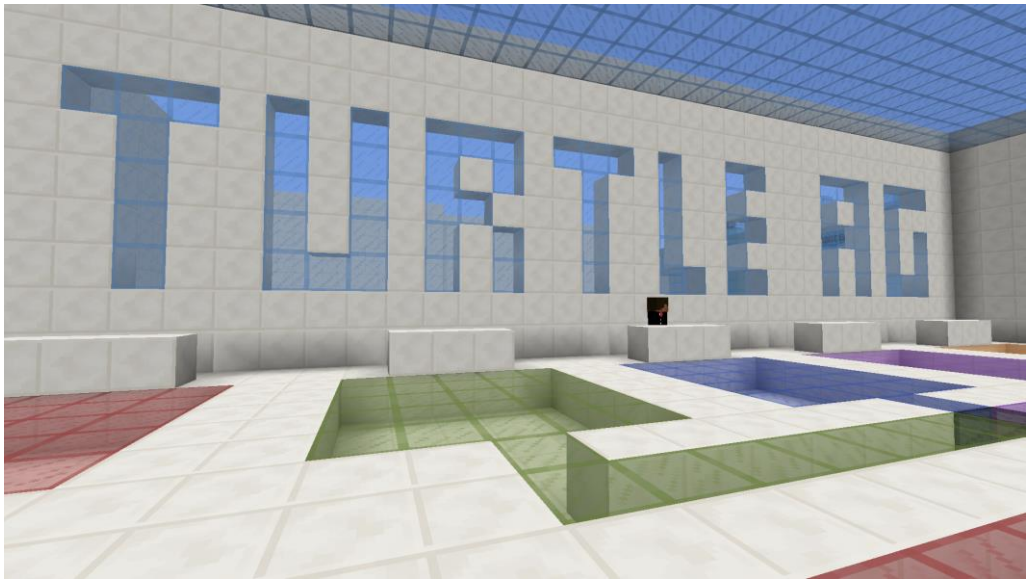
Das Schöne an dieser Modifikation und der Sprache Lua: Bis auf wenige Ausnahmen werden die genutzten Programmierbefehle und -strukturen schon vertraut sein. Diese kennen sie nämlich aus dem Grundcurriculum „Programmieren mit Logo“ und den dortigen Turtles. Auch in der Mod Computercraft sind, neben den Spielern selbst, Turtles die Hauptakteure, die hier als kleine Roboter Aufgaben erledigen können.

Die Syntax ähnelt dabei stark der von Logo bzw. Touch Develop, sodass kaum eine Umgewöhnung notwendig ist.

Das Setting – die „Turtle AG“

Über den Verlauf der Curricula schlüpfen Ihre Schülerinnen und Schüler in die Rolle von jungen Angestellten des fiktiven Bau-Unternehmens „Turtle AG“.

In der von uns zur Verfügung gestellten Code your Life Minecraftwelt lernen sie zunächst in einer „Mitarbeiterschulung“ das Wichtigste zur Steuerung ihrer Avatare und natürlich zur Handhabung der Turtles. Dabei werden die wesentlichen Befehle vorgestellt und Wissen zu bekannten Programmierprinzipien, wie der for-Schleife, wiederaufgefrischt. Außerdem wird erklärt, wie die Transportrohre und das Energiesystem in dieser Welt funktionieren.



Gespielt wird von Beginn an im Challenge-Format in Teams bis zu fünf Personen. Nach erfolgreichem Mitarbeitertraining muss nun jede Gruppe, mit Hilfe der Turtles, eine eigene kleine Filiale aufbauen, die möglichst schnell und energieeffizient Stein abbaut und weiterverarbeitet. Da die Turtles zur Verrichtung ihrer Arbeit Energie benötigen, müssen sich die Schüler um eine entsprechende Versorgung kümmern. Selbst der emsigste Angestellte wird einmal Hunger bekommen. Also muss auch für genügend Nachschub an Nahrungsmitteln in der Minecraftwelt gesorgt werden.

Aber was wäre ein Bau-Unternehmen ohne den eigentlichen Bau? In einer besonderen Challenge können die Fortgeschrittenen unter den kleinen Codern ihren Turtles einen kompletten Bauplan für ein einfaches Haus einprogrammieren, sodass diese das gespeicherte Muster immer wieder und wieder bauen kann, egal an welchem Ort.

> **Wer programmiert die besten Turtles? Wer baut die coolste Filiale? Wie so oft in der Programmierung gilt auch hier: Viele Wege führen nach Rom.**

Zum Ende einer jeden Unterrichtseinheit erhalten die Teams die Gelegenheit, sich gegenseitig ihre Ergebnisse, aber vor allem auch Lösungswege zu präsentieren und sich Anregungen und Motivation für die nächste Challenge zu holen.

CURRICULUM 1 – Turtle Trainingscenter 1

Auf geht es in die spannende Welt von Minecraft und der Programmiersprache Lua! Die Kinder schlüpfen in die Rolle neuer Angestellter des Bau-Unternehmens „Turtle AG“. Bevor sie gemeinsam eine neue Filiale aufbauen, müssen sie aber zunächst das Turtle Trainingscenter durchlaufen. Gespielt wird in 5er-Teams.

Die Unterrichtseinheit „Turtle Trainingscenter“ führt die Kinder an die grundlegende Steuerung von Minecraft und die wichtigsten Befehle der Computercraft Turtles heran. Altes Wissen aus den vorherigen Curricula von Code your Life, z.B. zur Bewegung einer Turtle (siehe Programmieren mit Logo) oder auch der for-Schleife, wird aufgefrischt und in seiner Funktionalität leicht erweitert. Die Kinder lernen, wie sie im Computercraft internen Editor navigieren, Programme erstellen, speichern und editieren.

Die wichtigsten Funktionen und Befehle tragen die kleinen Nachwuchscoder über den Verlauf der Lektion selbständig in ein Arbeitsblatt ein. Diese Übersicht wird sie später dabei unterstützen, bestimmte Kommandos schneller parat zu haben und in ihre Programme zu integrieren.

Überblick

Kompetenzen

Zeitaufwand	**90 Minuten**
Technik	Laptops / Standrechner, je Gerät eine Mouse, LAN/WLAN, Beamer
Methoden	Gruppenarbeit, Simulation, Frage und Antwort, Arbeitsblatt
Vorkenntnisse	Programmieren mit Logo

Die Schülerinnen und Schüler …

- lernen die grundlegende Handhabung der Minecraft (Computercraft) Turtles kennen.
- machen ihre ersten Schritte in der Programmiersprache Lua auf Basis ihrer Erfahrung mit der Programmierung in Logo.
- durchlaufen gemeinsam den ersten Part des Turtle Trainingscenters.
- lassen die Minecraft Turtles erste knifflige Aufgaben für sie lösen.

Ablauf

Phase	Aufgabe	Methode	Zeit
Sensibilisierung	Was ist Minecraft und was hat das mit Programmierung zu tun?	F & A	5´
Vorbereitung	Erarbeitung grundlegender Regeln für das Miteinander in Minecraft		15´
	Einteilung in Gruppen und Einstieg ins Spiel		
Arbeitsphase	Der Fitnesstest	Gruppenarbeit, Simulation und Arbeitsblatt	50´
	Minecraft Turtles bedienen und Programme ausführen		
	Mit Turtles craften		
	Die Turtles bewegen und Signale ausgeben lassen		
Nachbereitung	Besprechung des Arbeitsblattes	F & A, Arbeitsblatt	15‘
	Präsentation der Lösungswege	Simulation	
Ausblick	Turtle Trainingscenter 2		5‘

Unterrichtsverlauf

Vorbemerkung

Während der Arbeitsphasen kann es an einigen Stellen notwendig werden, dass die Lehrkraft die Kinder im Spiel aktiv unterstützen muss. Mal muss jemand teleportiert werden, mal muss ein neuer Gegenstand her, auf den die Kinder selbst keinen Zugriff haben. Das heißt in erster Linie, dass Sie mitspielen müssen und schon ein wenig Vorerfahrung, vor allem auch im Umgang mit dem Kreativmodus, haben sollten. Nehmen Sie sich zuvor ruhig 1-2 Stunden Zeit zum Üben und durchlaufen Sie bestenfalls selbst einmal das Curriculum.

Die wichtigsten Befehle für die Lehrkraft:

Diese Minecraft-Befehle werden Sie im Verlauf der Unterrichtseinheiten immer wieder brauchen. Öffnen Sie dazu die Spielkonsole in Minecraft, indem Sie „T“ auf Ihrer Tastatur drücken und geben Sie bei Bedarf folgende Befehle ein:

/gamemode 1 (für den Kreativmodus)

/gamemode 0 (für den Überlebensmodus)

/tp spielername1 (teleportiert Sie zum Spieler mit dem Spielernamen „spielername1“)

/tp spielername1 spielername2 (teleportiert spielername1 zu spielername2: „/tp Hans Peter“ teleportiert also den Spieler mit dem Minecraftnamen „Hans“ zum Spieler mit dem Minecraftnamen „Peter“)

Phase 1 | Sensibilisierung

Was ist Minecraft und was hat das mit Programmierung zu tun?

Sicherlich haben nahezu jede Schülerin und jeder Schüler zumindest von Minecraft gehört. Viele werden es sogar selbst schon gespielt haben. Stellen Sie die Frage nach den Minecrafterfahrungen offen in den Raum und lassen Sie die Kinder kurz von ihren Erlebnissen berichten. Bitten Sie Ihre Minecraftexpertinnen und -experten, den Minecraftneulingen das Spiel grob zu erklären, wenn nötig. Um nun die Überleitung zum Curriculum zu finden, schauen Sie gemeinsam mit den Kindern, was Minecraft mit Programmierung gemeinsam haben könnte.

- Was könnte Minecraft mit Programmierung zu tun haben? (Antwort: Minecraft ist ein Computerspiel. Computerspiele müssen programmiert werden. Man kann auch in Minecraft programmieren, …)

Erklären Sie Ihren Schülerinnen und Schülern nun, dass Sie sich gemeinsam in die Welt von Minecraft begeben und dort mit Hilfe kleiner Turtles aus der Mod Computercraft einige spannende Aufgaben lösen werden, ähnlich wie schon mit den Logo Turtles. Erläutern Sie den Kindern das Setting (wie unter „Das Setting – die Turtle AG“ beschrieben).

Phase 2| Vorbereitung

2.1 Schritt eins | Klare Regeln

Neben dem Einstieg in das Spiel, ist das wichtigste Ziel der Vorbereitungsphase die Erarbeitung eines grundlegenden Regelsatzes für das Miteinander in Minecraft. Vergessen Sie nie: es handelt sich immer noch um ein Spiel, das normalerweise auch Spielweisen zulässt, die für den Durchlauf des Curriculums eher unpassend sind (z.B. kampforientiert).

Zwar ist der Spieler-gegen-Spieler-Modus (englisch: „PvP“ oder „Player vs. Player) in unserem Curriculum deaktiviert, dennoch ist es theoretisch möglich, Spieler auf andere Art und Weise vom Fortkommen abzuhalten, indem z.B. Gegenstände absichtlich falsch platziert oder abgebaut werden. Diese Form des „Störens“ wird einigen Kindern vielleicht auch unter dem Begriff „Griefing“ („Leid zufügen“) bekannt sein.

Machen Sie deutlich, dass das Arbeiten mit Minecraft ein kleiner Vertrauensvorschuss Ihrerseits (und auch eine kleine Belohnung für das Bestreiten der vorherigen Curricula) ist, dass es im Gegenzug aber auch ein paar klare Regeln geben muss, damit alle Beteiligten etwas davon haben. Schließlich geht es darum, noch etwas Neues über Programmierung zu lernen!

Lassen Sie die Kinder moderiert selbst wichtige Regeln erarbeiten (siehe unten) und ergänzen Sie dann gegebenenfalls. Am besten schreiben Sie die Regeln gut sichtbar an die Tafel. Weisen Sie darauf hin, dass in einem kleinen Wettbewerb in Teams gearbeitet wird. Wer also eigene Teamkameraden behindert, tut sich selbst damit keinen Gefallen.

Hinweis Scheuen Sie nicht davor, auch klare Konsequenzen bei Nicht-Einhaltung der Regeln zu formulieren, wenn notwendig. Wer immer wieder Unfug treibt oder andere abhält, erhält so z.B. eine Spielauszeit o.ä.

Hier ein Überblick über die wichtigsten Regeln für das Miteinander in Minecraft

- Ich gehe nur an die Turtle meiner Mitschüler, wenn diese es mir erlauben! (Warum? Mehrere Spieler können gleichzeitig auf eine Turtle zugreifen und sich gegenseitig in die Turtle schreiben oder Code löschen)
- Ich behindere meine Mitspieler nicht, indem ich ihnen absichtlich Blöcke abbaue oder platziere. „Griefing“ ist nicht erlaubt! Ich stehle meinen Mitspielern keine Items!
- Wenn meine Teamkameraden nicht weiterkommen oder etwas nicht verstehen, versuche ich ihnen zu helfen, wenn ich kann!
- Wenn wir aufgefordert werden, zuzuhören, nehmen wir die Hände vom Rechner und spielen währenddessen nicht!
- Wer sich auch nach Verwarnung nicht an die Regeln hält, bekommt eine Spielauszeit!

Die aufgestellten Regeln gelten nicht nur für dieses, sondern auch alle nachfolgenden Minecraft Curricula. Sollte sich aus konkretem Spielkontext noch Bedarf an weiteren Regeln zeigen, ergänzen Sie diese einfach im Verlauf.

2.2 Schritt zwei | Gruppen einteilen

Nachdem Sie gemeinsam mit den Kindern die wichtigsten Regeln festgehalten haben, kann es auch schon an die Gruppeneinteilung gehen.

Teilen Sie Ihre Schüler den Teams Blau, Rot, Grün, Violett oder Orange zu, wobei jedes Team bestenfalls aus fünf Schülern besteht. Sollte ein Team weniger als vier Spieler umfassen, stellt dies kein Problem dar, erfordert aber an späterer Stelle noch ein paar zusätzliche Handgriffe Ihrerseits. Wir gehen bei dieser Aufteilung davon aus, dass jedes Gruppenmitglied einen Rechner zur Verfügung hat. Es empfiehlt sich, in jeder Gruppe ein Kind zu haben, dass Minecraft bereits gespielt hat.

Da die Unterrichtseinheiten aufeinander aufbauen, gelten die festgelegten Teams über den gesamten Verlauf der Curricula! Fertigen Sie zur besseren Übersicht eine Teamliste an. Geben Sie nun jeder Gruppe je Teammitglied eine Kopie des Arbeitsblattes „Turtle Trainingscenter Teil 1“ (siehe Materialien). Dieses gilt es über den Verlauf der Unterrichtseinheit auszufüllen, sodass die Kinder schließlich eine kleine Übersicht über die wichtigsten Funktionen und Turtle-Befehle angefertigt haben, auf die sie immer wieder zurückgreifen können. Am Ende der Lektion werden die Arbeitsblätter gemeinsam verglichen.

!! Aufgabe: Füllt als Team das Arbeitsblatt „Turtle Trainingscenter Teil 1“ aus! Hilfestellungen zum Ausfüllen geben euch die Figuren, Turtles und Schilder in der Minecraftwelt!

2.3 Schritt drei | Mit der Server-Welt verbinden

Nachdem die Teams nun entsprechend in Gruppen zusammensitzen, kann es an die Rechner gehen. Sind diese hochgefahren, müssen die Kinder den Technic Launcher öffnen, sich einloggen und das Code your Life Modpack auswählen, wie unter „Minecraft starten und die Code your Life Minecraft Welt betreten“ beschrieben. Am besten gehen Sie die dort beschriebenen Schritte gemeinsam mit den Schülerinnen und Schülern durch. Nutzen Sie an dieser Stelle einen Beamer, damit Sie die Arbeitsschritte besser verständlich machen können.

Bevor die Kinder dem Server beitreten, sollten Sie dringend folgende Hinweise geben:

- Schaut euch beim ersten Betreten der Welt in Ruhe um und nehmt Rücksicht auf Minecraft-Neulinge!
- Ihr landet in der Eingangshalle des Turtle Trainingscenters. Dort befinden sich 5 Sachbearbeiter der Turtle AG, jeweils einer für jedes Team. Sprecht NUR mit Sachbearbeiter, der für euer Team zuständig ist und lasst euch teleportieren.
- Einmal zugeteilt, könnt ihr nicht einfach wieder das Team wechseln! Bei jedem Sachbearbeiter steht, für welches Team er zuständig ist.

Hinweis Wenn die Kinder sich in den folgenden Curricula erneut mit dem Server verbinden, starten sie dort, wo sie das Spiel zuletzt verlassen haben. Daher ist es wichtig, dass die Kinder mit dem jeweils selben Account spielen, mit dem Sie das Curriculum begonnen haben. Fertigen Sie am besten eine Übersicht an, wer welchen Account nutzt, damit es später kein Durcheinander gibt.

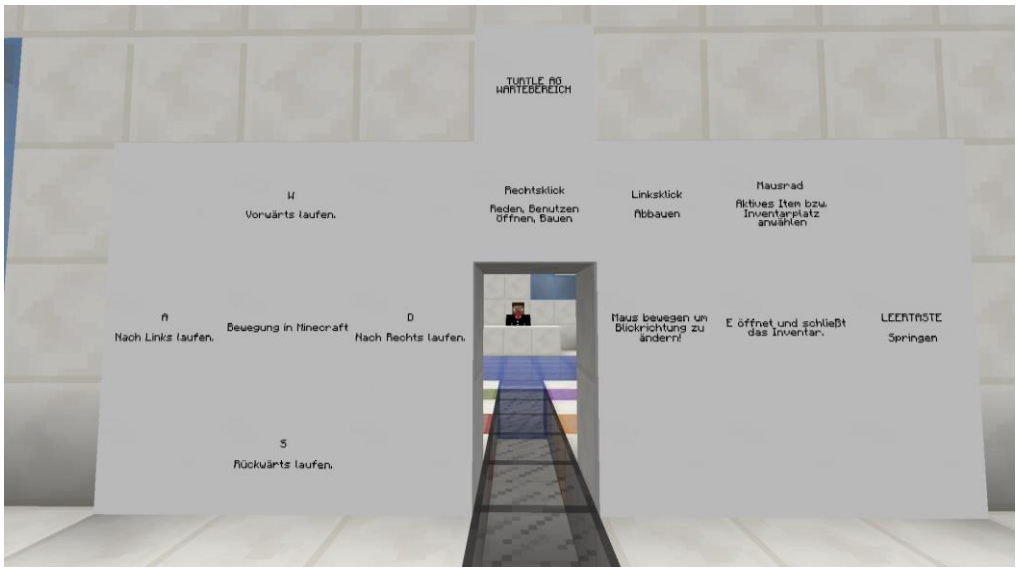
Nun können Sie grünes Licht zum Verbinden auf den Server geben. Vergessen Sie nicht, auch sich selbst zu verbinden. Aus dem Spiel heraus können Sie den Kindern oft am besten helfen. Öffnen Sie das Chatfenster mit „T“ und geben dort Folgendes ein:

/gamemode 1

Somit versetzen Sie sich in den Kreativmodus und können das Geschehen im wahrsten Sinne des Wortes „überfliegen“. Außerdem haben Sie Zugriff auf alle Spielblöcke, die Sie bei Bedarf an die Schüler verteilen können. Mit der Steuerung und Funktionsweise sollten Sie durch ein wenig Übung schon vertraut sein.

Sollte Minecraft nicht im Vollbildmodus ausgeführt werden, fordern Sie Ihre Schüler einmalig dazu auf, die Taste „F11“ auf ihrer Tastatur zu drücken. Das Spiel geht dann in den Vollbildmodus. Einige Texte lassen sich so besser lesen.

Im Eingangsbereich gibt es einige Schilder, die Minecraft-Einsteigern eine erste Hilfe bei der Steuerung geben sollen.



Keine Sorge: Eine gewisse Aufregung ist zu diesem Zeitpunkt ganz normal. Für die Kinder ist es meist etwas völlig Neues, ein Computerspiel in der Schule zu spielen. Im Unterricht! Sollte es zu laut werden, betonen Sie nochmal die Regeln, auf die sich zu Beginn alle geeinigt haben.

Wiederholen Sie noch einmal die Anweisung, dass die Kinder sich von den Sachbearbeitern der Turtle AG in ihren entsprechenden Teambereich teleportieren lassen sollen. Per Rechtsklick lassen sich diese NPCs (englisch: „Non Player Characters“: Nicht-Spieler-Charaktere) ansprechen.

ALLE Erklärungen zu Turtle-Befehlen und anderen Funktionen sind mehr oder weniger offensichtlich über NPCs, Schilder und Turtles in der Minecraftwelt verteilt.

Hinweis Bestärken Sie die Kinder immer wieder darin, sich in der Welt in Ruhe umzusehen und mit allen NPCs zu sprechen, die Sie treffen. Weisen Sie darauf hin, dass es sich definitiv lohnt, alles zu lesen! Wer nicht richtig liest, dem wird es vermutlich schwerfallen, alle Aufgaben vernünftig zu lösen.

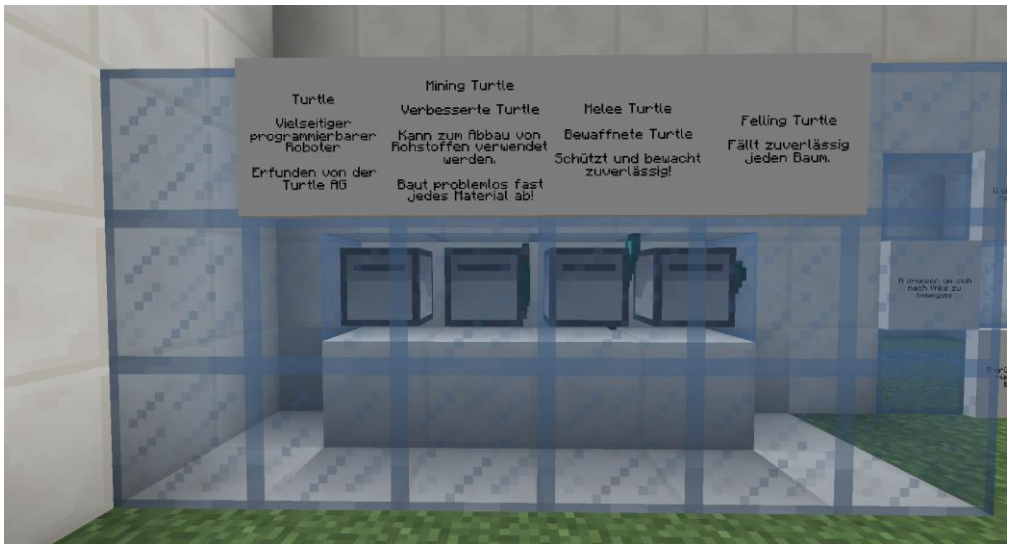
Sobald Sie sich vergewissert haben, dass alle Spielerinnen und Spieler teleportiert wurden, kann es in die eigentliche Arbeitsphase gehen.

Phase 3 | Arbeitsphase

Hauptaufgabe dieser Unterrichtseinheit ist es, die erste Hälfte des Turtle Trainingscenters zu durchlaufen und dabei alle Felder auf dem Arbeitsblatt „Turtle Trainingscenter Teil 1“ auszufüllen. Rufen Sie das Ganze als kleinen Wettbewerb aus. Dem Team, welches den Parcours als erstes erfolgreich absolviert und das Arbeitsblatt vollständig ausgefüllt hat, winkt ein kleiner Bonus!

Jedes Team befindet sich nun in einem eigenen Trainingsbereich, der baugleich zu denen der anderen Teams ist. Im ersten Raum der Anlage befindet sich Trainer Albert, der in einem Gespräch alles Wissenswerte über den anstehenden Fitnesstest und die Turtle-AG berichtet.

In einem Glaskasten gibt es unterschiedliche Minecraft-Turtles mit kurzen Beschreibungen zu sehen.



Außerdem gibt es die Möglichkeit einen Timer zu starten, der die Zeit misst. Fordern Sie dafür ein Mitglied jedes Teams auf, den Computer im neben Trainer Albert rechtszuklicken. Der Timer wird dann automatisch gestartet, sobald die Gruppe den ersten Raum verlässt und gibt beim Erreichen des Ziels eine Zeit aus.

Hinweis Lassen Sie von hier an die Kinder selbst viel ausprobieren. Sollte einmal nicht klar sein, was zu tun ist, um voranzukommen, fragen Sie die Kinder zunächst, ob Sie sich auch wirklich richtig umgeschaut und mit allen NPCs gesprochen haben. Diese geben alle nötigen Hinweise! Bei Bedarf können Sie sich auch mit Ihrer Spielfigur zu einem der Teams teleportieren (siehe „Die wichtigsten Befehle für die Lehrkraft“) und aushelfen, wenn unbedingt notwendig.

3.1 Der Fitnesstest

Nun geht es richtig los! Die erste Aufgabe besteht darin, einen kleinen Fitnesstest zu bestreiten, das heißt, einen kleinen Parcours zu durchlaufen. So wird sichergestellt, dass alle Spieler die grundlegende Bewegungssteuerung (Laufen, Umsehen, Springen, Schwimmen) in Minecraft beherrschen.

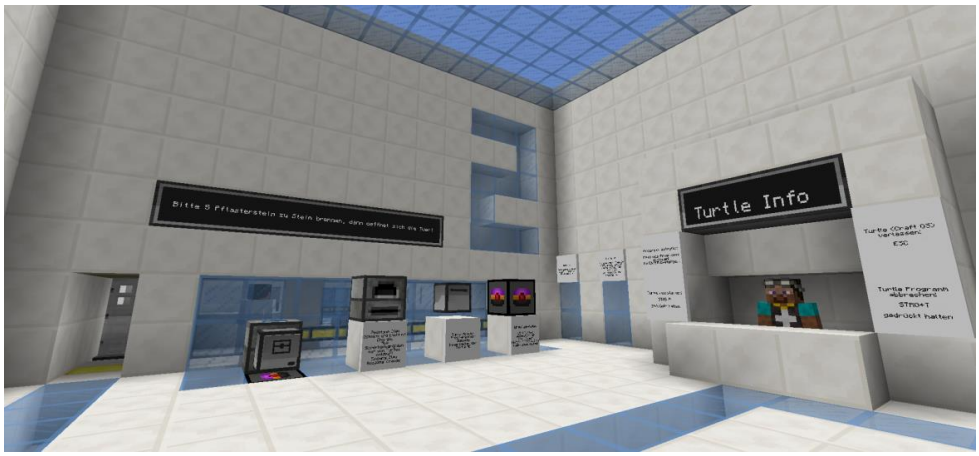
Aufgabe: Sprecht mit Trainer Albert, durchlauft den ersten kleinen Parcours und besteht den Fitnesstest!

Allen Kindern, die Minecraft bereits gespielt haben, wird diese Aufgabe sehr leicht fallen. Wer schon früher fertig ist, kann seine Teammitglieder unterstützen. Wichtig ist, dass wirklich mit Albert gesprochen wird, da sonst der Fitnesstest nicht richtig abgeschlossen werden kann!

Über den Parcours sind bereits einige Turtles verstreut, die Bruchstein in ihrem Inventar aufbewahren. Dieser lässt sich herausnehmen und für die nächste Aufgabe verwenden.

3.2 Minecraft Turtles bedienen und Programme ausführen

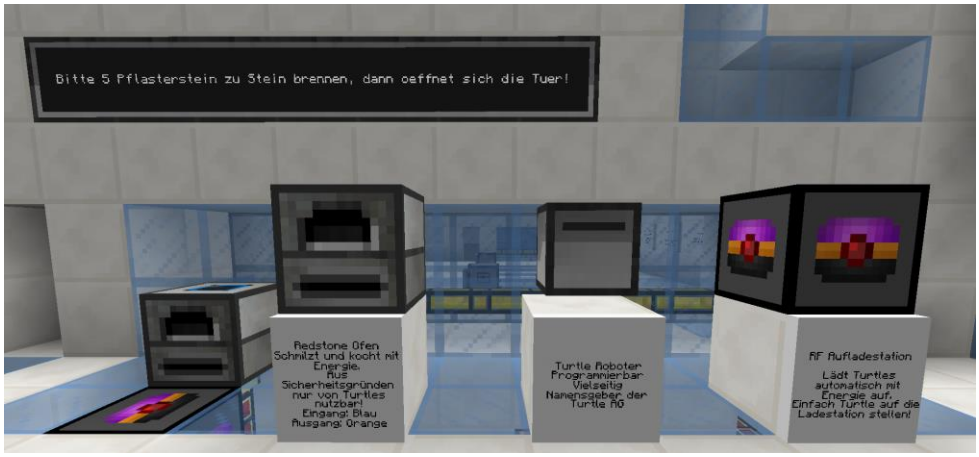
Im nächsten Abschnitt wartet Mechaniker Ben an der Turtle Info auf die Spieler.



Auch er kann per Rechtsklick angesprochen werden und gibt Auskunft über folgende Fragen:

- Was ist eine Turtle?
- Wie rufe ich ein Programm auf?
- Wie schreibe ich ein Programm?
- Wie lade ich die Turtle auf?
- Wie kann die Turtle den Ofen bedienen?

Außerdem werden Transportrohre, Energieleitungen und Energiezellen in Glasschaukästen vorgestellt.



Die Kinder erfahren hier alles Notwendige, um in den nächsten Raum zu gelangen. Die Aufgabe:

!! Aufgabe: Brennt 5 Bruchsteine zu Stein, dann öffnet sich die Tür! Nutzt dafür die Bruchsteine aus den Turtles und das Programm „ofen_fuellen“ auf der Turtle, die ihr von Mechaniker Ben bekommt!

Lösungsbeispiel

- Die Kinder müssen zunächst 5 Bruchsteine aus dem Inventar der Turtles aus dem Fitnessparcours sammeln, wenn nicht schon geschehen.

- Anschließend müssen sie eine Turtle, die sie von Mechaniker Ben im Gespräch erhalten, auf der RF Aufladestation vor dem Redstone Ofen platzieren. Es ist wichtig, dass sie dabei in Richtung des Ofens schaut.

- Nun wird die Turtle per Rechtsklick geöffnet und gestartet.
- Die 5 gesammelten Bruchsteine müssen unten rechts in den ersten Slot des Inventars der Turtle gelegt werden.
- In den Editor muss jetzt nur noch „ofen_fuellen“ eingegeben werden und die Turtle gibt die Bruchsteine in den Ofen, in dem sie gebrannt und in das Transportsystem (gelbe Rohre) gegeben werden.

- Nach kurzer Zeit öffnet sich die Tür in den nächsten Raum, den die Kinder nun betreten können.

Hinweis: Das Abbauen und Verarbeiten von Ressourcen ist in der Code you Life Minecraft-Welt nur wirklich effektiv mit Turtles. Darauf weisen auch die NPCs hin. Das Abbauen von Blöcken mit anderen Werkzeugen nimmt unnötig viel Zeit in Anspruch. Ist ein Block platziert, lässt er sich, bis wenige Ausnahmen nicht mehr oder nur schwer wieder abbauen! Weisen Sie die Kinder darauf hin! Turtles können aber jeder Zeit mit einer Spitzhacke wieder abgebaut werden.

3.3 Mit Turtles craften

Hinweis: Jedes Mal beim Betreten eines neuen Abschnitts im Turtle Trainingscenter werden die Spielinventare der Kinder, wie auf den Schildern über den Türen beschrieben, automatisch geleert! Weisen Sie die Kinder nochmals darauf hin!

Im nächsten Raum geht es um das „Crafting“ mit Turtles, d.h. das Weiterverarbeiten von Gegenständen mit Turtles. Normalerweise können in Minecraft Blöcke auf eine Werkbank gelegt und dort von den Spielern neukombiniert werden. Das ist hier so nicht möglich. Einzige legitime Crafting-Werkzeuge sind die Turtles. Die Turtle-AG hat diese Vorschrift zum Schutz der Mitarbeiter erlassen.

Darüber werden die Schülerinnen und Schüler hier von Forscher Caesar aufgeklärt, sobald sie mit ihm sprechen. Außerdem beantwortet er folgende Fragen:

- Wie benenne ich eine Turtle?
- Wie crafte ich mit einer Turtle?
- Wie navigiere ich im CraftOS (dem Betriebssystem) der Turtle?
- Wie erfahre ich, welche Programme sich auf der Turtle befinden?

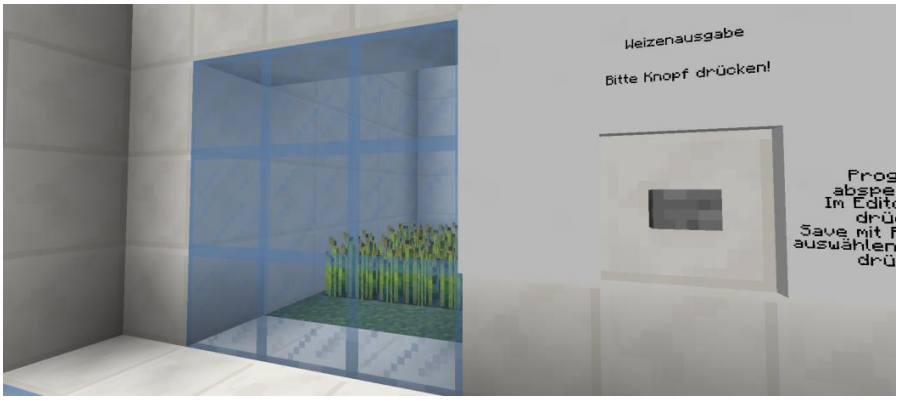
Um in den nächsten Abschnitt zu gelangen, müssen die Schüler nun zusammen 64 Brote herstellen und in die Kiste rechts neben der nächsten Tür legen. Wer „Minecrafter“ ist, weiß es schon: Das Craften in Minecraft funktioniert über Rezepte. Das heißt, um ein Brot herzustellen, muss ich in einem Crafting-Feld, in diesem Fall im Inventar der Turtle, drei Weizen nebeneinander auslegen, so wie es die visuelle Darstellung an einer der Wände des Raumes zeigt.

Damit die Turtle craftet, muss man ihr einen entsprechenden Befehl geben bzw. ein kleines Programm schreiben, so wie in unserem Lösungsbeispiel unten.

!! Aufgabe: Craftet 64 Brote mit Hilfe eurer Turtles! Legt die Brote in die Kiste rechts neben der nächsten Tür!

Lösungsbeispiel

- Nachdem sie mit Forscher Caesar gesprochen haben, können die Kinder von Forscher Demetrius eine Turtle erhalten.
- Durch Rechtsklicken des Knopfes an der Wand mit der Überschrift „Weizenausgabe“ erhalten sie pro Klick 32 Weizen in ihr Inventar.



- Die Kinder platzieren die Turtle auf einer Ladestation und öffnen sie per Rechtsklick.
- Den Weizen verteilen sie nun gleichmäßig, entsprechend dem Rezept für Brot im Inventar der Turtle.
- Nun geben die Spieler der Turtle einen Namen, indem sie „label set meineturtle“ in den Editor eingeben und mit der „Entertaste“ bestätigen. Anstelle von „meineturtle“ kann jeder beliebige Name eingegeben werden (Achtung: keine Leerzeichen):

- Als nächstes wird ein einfaches Crafting Programm geschrieben. Wir haben bereits von Mechaniker Ben erfahren, dass wir dafür „edit programmname“ eingeben müssen. Für „programmname“ können die Kinder wieder einen beliebigen Namen eingeben, am besten einen, der etwas mit der auszuübenden Funktion zu tun hat, also wählen sie beispielsweise:

- Nachfolgend gelangen die Kinder in den Programmeditor und tippen dort einfach „turtle.craft()“ ein:

- Jetzt muss das Programm noch durch Drücken der Taste „Strg“ und dem anschließenden Betätigen der „Entertaste“ abgespeichert.

- Um das Programm auszuführen, verlassen die Kinder die Bearbeitung, indem sie auf „Strg“ drücken, mit den Pfeiltasten „Exit“ wählen und mit der „Entertaste bestätigen.

- Zurück im Hauptbildschirm muss jetzt nur noch der Name des Programms eingetippt und bestätigt werden. Wenn alles richtiggemacht wurde, sollte die Turtle nun Brot angefertigt haben, dass sich die Kinder nun nehmen und wie gefordert in die Kiste legen können.

- Auch hier geht nach einer kurzen Wartezeit die Tür auf.

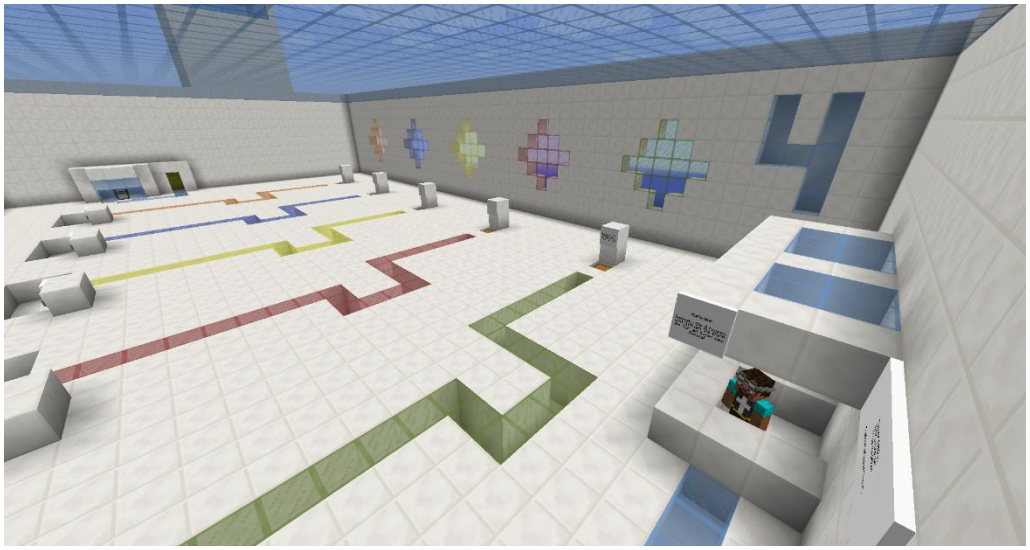
Die Kinder haben nun ihr erstes kleines Minecraftturtle-Programm geschrieben.

Hinweis: Sollte das Programm nicht richtig funktionieren, liegt vielleicht ein Tippfehler vor. Meistens wird ein derartiger Fehler schon durch eine Fehleranzeige signalisiert. In so einem Fall muss das Programm einfach erneut über „edit programmname“ editiert und abgespeichert (!) werden. Die Kinder sollten außerdem darauf achten, dass der Weizen richtig platziert wurde.

3.4 Die Turtles bewegen und Signale ausgeben lassen

Zum Abschluss der Unterrichtseinheit wird es ein wenig kniffliger. In diesem Abschnitt ist es besonders wichtig, auf das Vorwissen aus den Curricula zu „Programmieren mit Logo“ zurückgreifen zu können, insbesondere auf das Prinzip der for-Schleife.

Die Kinder sind nun beim Turtle-Bewegungstest angelangt. Sie müssen mit ihrer Turtle durch einen kleinen Pfad navigieren und am Ende des Weges ein Redstone-Signal, also eine Art Stromsignal, ausgeben lassen. Erst dann wird in der jeweiligen Kiste am Ende eine farbige Kugel erscheinen. Diese können sie sich aus der Kiste nehmen und in das Inventar der Türöffner-Turtle auf der anderen Seite der Halle legen. Sobald ein Team insgesamt mindestens vier Kugeln gesammelt und in die Türöffner-Turtle gelegt hat, kann auf dieser auch das Programm „tueroeffner“ gestartet werden. Dieses wiederum öffnet die Tür zum nächsten Abschnitt.



Da die Kinder idealerweise zu fünft in einem Team sind, gibt es den identischen Turtle-Pfad fünf Mal in der Halle, sodass theoretisch jedes Teammitglied genügend Raum hat, um eine eigene Turtle zu programmieren. Lösungsstrategien gibt es natürlich verschiedene.

Aufgabe: Sammelt mit Hilfe der Turtles 4 Kugeln aus den Kisten in dieser Halle und betätigt damit das Türöffner-Programm!

In diesem Bereich erklärt Techniker Erik:

- Was ist eine for-Schleife? (in Lua)
- Wie bewege ich meine Turtle?
- Wie gebe ich ein Redstone Signal mit der Turtle aus?

Hinweis Von Techniker Franz nebenan bekommen die Kinder bei Bedarf Ersatzturtles und Spitzhacken. Denn eine Besonderheit gibt es: Ist die Turtle erstmal unterwegs, gibt es kein Zurück mehr! Ist das Programm unzureichend geschrieben, um die Turtle bis ans Ende zu bringen, gibt es keine Möglichkeit, von außen auf die Turtle zuzugreifen. Den Kindern bleibt dann nichts Anderes übrig, als den Reset-Knopf am Anfang des Weges zu betätigen und die Turtle aus dem Tunnel zu löschen. Sie müssen dann eine neue Turtle auf den Anfangsspot stellen und eine neues Programm schreiben!

Die Turtle-Bewegungen und Signalausgabe

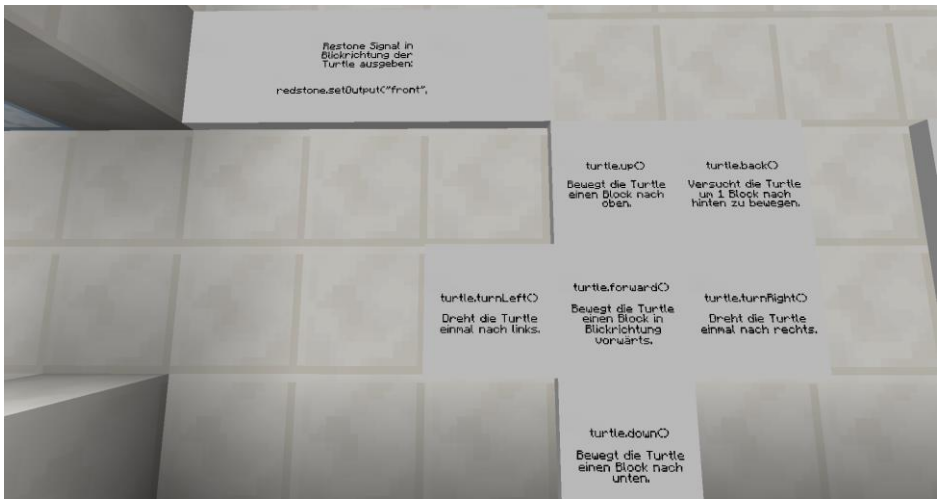
Um eine Minecraft-Turtle zu bewegen, werden in Lua ganz ähnliche Befehle genutzt, wie sie die Kinder schon vom Programmieren mit Logo kennen:

- Vorwärts: turtle.forward()
- Rückwärts: turtle.back()
- Nach oben: turtle.up()
- Nach unten: turtle.down()
- Nach links drehen: turtle.turnLeft()
- Nach rechts drehen: turtle.turnRight()

Um ein Stromsignal an den Kasten am Ende des Tunnels auszugeben gibt es folgenden Befehl:

- Redstone-Signal nach vorn ausgeben: redstone.setOutput(„front“, true) (Achtung: eine häufige Fehlerquelle ist das Vergessen der Anführungszeichen vor und nach front)

Diese Befehle werden nicht nur von Techniker Erik erklärt, sondern sind zusätzlich auf Schildern an der Wand einsehbar. Auch auf dem Arbeitsblatt werden diese Befehle abgefragt.



Hinweis: Sind die Kinder in diesem Abschnitt angekommen, ist dies ein guter Zeitpunkt, um sie erneut an das Ausfüllen des Arbeitsblattes zu erinnern! Können sich die Kinder mal nicht an einen Befehl erinnern, müssen sie so nicht andauernd wieder zu den NPCs laufen oder Schilder lesen, sondern finden die wichtigsten Befehle zusammengefasst auf ihrem Arbeitsblatt.

Die for-Schleife in Lua

Das Konzept der for-Schleife sollte den Kindern schon bekannt sein. Nehmen Sie sich ruhig fünf Minuten Zeit, die Funktionsweise noch einmal gemeinsam mit den Teams zu rekapitulieren, wenn das Prinzip nicht mehr so ganz klar ist.

- Was macht eine for-Schleife? (Antwort: Einen Befehlsblock wiederholen, ohne dass dieser immer wieder neu geschrieben werden muss.)

Ein Programm für eine Turtle, die sich 6 Schritte nach vorn bewegen soll, könnte in Lua z.B. so aussehen:

```
for i=1, 6 do
turtle.forward()
end
```

Auch an einer der Wände in diesem Abschnitt ist ein for-Schleifen-Beispiel abgebildet.

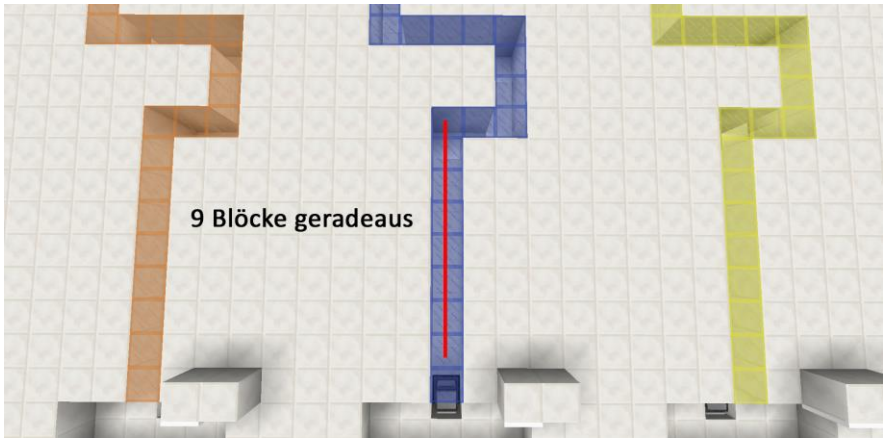
Hinweis: Die Vermutung liegt nahe, dass sich anstelle dieser Schleife auch einfach der Befehl „turtle.forward(6)“ nutzen ließe. Nach diesem Prinzip funktionieren die Turtles in Minecraft jedoch nicht!

Der Turtle-Tunnel kann natürlich auch ohne for-Schleife, dafür aber mit signifikant mehr Einzelbefehlen durchlaufen werden. Die Eingabe all dieser Befehle braucht ihre Zeit. Je mehr Code ich habe, umso mehr Tippfehler sind auch möglich. Weisen Sie die Kinder nochmals darauf hin, lassen Sie aber prinzipiell alle Lösungswege zu!

Lösungsbeispiel

Sehen wir uns nun exemplarisch einen Lösungsweg mit for-Schleife an:

- Sollte noch keine Turtle auf der Lade Station am Anfang des Tunnels stehen, platzieren wir dort unsere Turtle, sodass sie in Richtung Tunnel schaut.
- Wir öffnen die Turtle per Rechtsklick und erstellen über „edit wegfinder“ ein neues Programm mit dem Namen „wegfinder“.



- Durch abzählen der Blöcke sehen wir, dass die Turtle zuerst 9 Schritte vorwärtsgehen muss.
- Für den ersten Teil unseres Programms tippen wir:

```
for i=1,9 do
  turtle.forward()
end
```

- Als nächstes muss sich die Turtle nach rechts drehen, zwei Schritte nach vorn gehen und wieder nach links drehen. Die Befehle fügen wir unter der for-Schleife ein:

```
turtle.turnRight()
turtle.forward()
turtle.forward()
turtle.turnLeft()_
```

- Es folgen weitere 3 Vorwärtsschritte und eine Linksdrehung:

```
turtle.forward()
turtle.forward()
turtle.forward()
turtle.turnLeft()
```

- Nun bietet sich für die nächsten 4 Schritte wieder eine for-Schleife an:

```
for i=1,4 do
  turtle.forward()
end
```

- Wir geben den Befehl zum Rechtsdrehen und eine weitere for-Schleife für die letzten 5 Schritte ein:

```
turtle.turnRight()

for i=1,5 do
  turtle.forward()
end
```

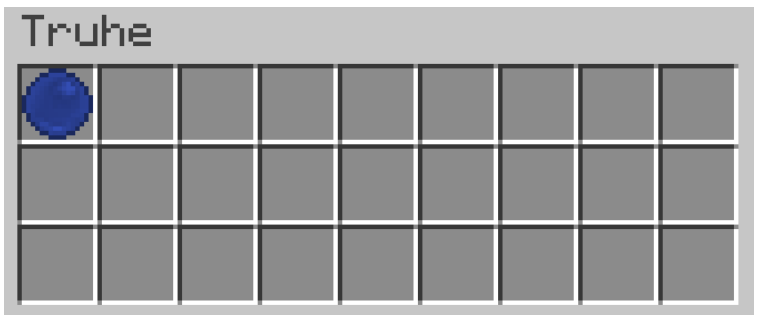
- Ganz wichtig: Nicht das Redstone-Signal vergessen, sonst öffnet sich die Kiste nicht! Am Ende unseres Programms geben wir auch diesen Befehl noch ein:

```
redstone.setOutput("front",true)
```

- Wir speichern das Programm über „Strg“ und „Save“. Der komplette Programmcode sieht so aus:

```
for i=1,9 do
  turtle.forward()
end

turtle.turnRight()
turtle.forward()
turtle.forward()
turtle.turnLeft()

turtle.forward()
turtle.forward()
turtle.forward()
turtle.turnLeft()

for i=1,4 do
  turtle.forward()
end

turtle.turnRight()

for i=1,5 do
  turtle.forward()
end

redstone.setOutput(„front“, true)
```

- Nach dem Speichern des Programms, verlassen wir es über „Strg“ und Exit.
- Nun geben wir nur noch „wegfinder“ ein und bestätigen mit Enter.
- Die Turtle, hat den Parcours erfolgreich durchlaufen und ein Signal ausgegeben. Nach kurzer Zeit erscheint in der Kiste am Ende des Tunnels eine farbige Kugel, die wir in unser Inventar nehmen.



Hinweis: Achten Sie auf Groß- und Kleinschreibung, da die Befehle sonst nicht richtig ausgeführt werden können! Sollten die Kinder eine Fehlerausgabe bekommen, handelt es sich oft um Tippfehler. Mal wird vergessen, eine Klammer zu schließen, mal fehlt ein Punkt oder es gibt einen klassischen Buchstabendreher.

Die Teams müssen auf obige oder ähnliche Weise insgesamt 4 Kugeln aus den Kisten sammeln, damit es weitergehen kann. Natürlich kann jedes Kind zunächst für sich mit der Turtle experimentieren. Wer bereits eine Kugel erhalten hat, sollte seine Teammitglieder unterstützen und gegebenenfalls den eigenen Lösungsweg erklären.

Hinweis – kleinere Gruppe: Ist es aufgrund der Aufteilung so, dass weniger als vier Kinder Teil eines Teams sind, entsteht für diese hier normalerweise ein Mehraufwand. Lassen Sie die Kinder daher zunächst ihrer Anzahl entsprechend Kugeln sammeln und geben sie die fehlenden Kugeln dann über den Kreativmodus manuell. Teleportieren Sie sich dafür zum entsprechenden Team, öffnen Sie im Kreativmodus das Menü und geben dort „Kugel“ in das Suchfeld ein. Wählen Sie die Kugeln aus, die den Kindern noch fehlen und werfen Sie die Kugeln dann aus Ihrem Inventar. Die Kinder brauchen diese nur noch einzusammeln und können jetzt das Türöffnerprogramm ausführen.

Das Türöffnerprogramm

Haben die Teams jeweils vier Kugeln gesammelt, müssen sie diese nun nur noch in die obersten vier Inventar-Slots der Türöffner-Turtle legen, „tueroeffner“ in das System eingeben und bestätigen. Die Tür wird sich nach kurzer Zeit öffnen.

Variante: Besonders schlau ist, wer auf die (an dieser Stelle eher unkonventionelle) Idee kommt, das Türöffnerprogramm selbst zu editieren und sich so sehr viel Zeit spart. Auch bei dieser Lösung handelt es sich um einen legitimen Weg, das erste Curriculum zu beenden. Dafür muss zu Beginn des Programms nur eine einzige Zeile Code hinzugefügt und das Programm dann abgespeichert werden:

```
redstone.setOutput(„back“, true)
```

So wird bei einer Ausführung des Programms durch die Turtle ein Redstone-Signal in die Leitung ausgegeben, die für die Öffnung der letzten Tür verantwortlich ist. Das Sammeln der farbigen Kugeln wird damit überflüssig.

Großartig! Die erste Hälfte des Turtle Trainingcenters haben die Kinder erfolgreich hinter sich gebracht! Dazu gratuliert auch der Turtle-AG Junior Manger im letzten Raum.

Phase 4 | Nachbereitung

Animieren Sie die Teams, die bereits fertig sind, den anderen Teams entweder zu helfen, oder ruhig noch etwas mit der Turtle zu experimentieren, sofern noch Zeit zur Verfügung steht.

Spätestens 20 Minuten vor Ende der Unterrichtseinheit, sollten Sie jedoch einen Schnitt machen und zum Vergleich des Arbeitsblattes übergehen. Das Arbeitstempo der Gruppen kann sehr unterschiedlich ausfallen. In der angegebenen Zeit ist der erste Teil des Turtle Trainingscenters jedoch in Regel gut bestreitbar.

Verschaffen Sie sich Gehör und bitten Sie die Kinder für den Moment, die Hände vom Rechner zu nehmen. Die Kinder haben jetzt eine ganze Weile gespielt. Fragen Sie erst einmal, wie es ihnen beim Spielen ergangen ist und gehen Sie dann zum Vergleich des Arbeitsblattes über.

4.1 Das Arbeitsblatt vergleichen

Alle Felder auf dem Arbeitsblatt „Turtle Trainingscenter Teil 1“ lassen sich mit Hilfe der Schilder, Turtles und NPCs ohne Probleme ausfüllen. Gehen Sie das Blatt mit den Kindern Schritt für Schritt durch. Am besten melden sich Ihre Schüler und Sie nehmen sie dann nacheinander dran.

Vergleichen Sie die Ergebnisse der Kinder mit der ausgefüllten Version des Arbeitsblattes aus dem Materialanhang der Handreichung (siehe Materialien).

Stellen Sie Fragen wie:

- Wo habt ihr den Befehl gefunden? (Antwort: NPC mit Namen, Abschnitt, Schild, usw.)
- Haben alle verstanden, wie der Befehl funktioniert?

Bitten Sie bei Unklarheiten die Kinder, sich zunächst gegenseitig aufzuklären. Bei Bedarf ergänzen Sie.

4.2 Präsentation der Lösungswege zum Turtle-Tunnel

Die letzten Minuten gehören wieder ganz den Kindern, denn nun sollen sie ihre Programme vorstellen. Es ist sehr wahrscheinlich, dass unterschiedliche Versionen zum Durchlaufen des Turtle-Tunnels im letzten Abschnitt geschrieben wurden.

Sofern Sie nicht sowieso schon in diesem Bereich stehen, begeben Sie sich in die besagte Halle, entweder per Flugfunktion im Kreativmodus, oder Sie teleportieren sich einfach zu einen der Kinder. Wählen Sie nun exemplarisch einen Tunnel aus und löschen Sie gegebenenfalls noch darin vorhandene Turtles (Reset-Knopf). Nehmen Sie sich eine neue Turtle und platzieren diese auf der Aufladestation mit Blick Richtung Tunnel.

Fragen Sie, welches der Kinder gern zeigen möchte, wie sein Team die Turtle programmiert hat und wählen Sie einen Freiwilligen aus. Per Beamer können alle anderen Schülerinnen und Schüler das Geschehen mitverfolgen.

Hinweis: Bitten Sie die restlichen Kinder, während der kleinen Demonstration nicht weiterzuspielen, sondern ihrer Mitschülerin oder ihrem Mitschüler aufmerksam zuzuhören und bei Bedarf Fragen zu stellen oder auszuhelfen.

Fragen Sie die anderen Kinder, was sie anders oder genauso gemacht haben und warum. Diskutieren Sie die Lösungswege.

Sollte noch genug Zeit zur Verfügung stehen, lassen Sie ruhig noch ein Kind aus einem anderen Team seinen Lösungsweg zeigen.

4.3 Gewinner der Challenge

Welches Team war nun beim Durchlaufen des Parcours und dem korrekten Ausfüllen des Arbeitsblattes das schnellste?

Küren Sie das Gewinnerteam. Es hat ein kleines Paket mit Einrichtungsgegenständen (Sofa, Tisch, ...) zur Ausstattung der späteren Filiale gewonnen! Dieses bekommt es aber erst in der dritten Unterrichtseinheit. Notieren Sie sich deshalb unbedingt den Namen des Gewinnerteams!

Phase 5 - Ausblick

Das erste Minecraft-Curriculum ist geschafft! Bitten Sie die Kinder unbedingt, ihre ausgefüllten Arbeitsblätter auch zu den nächsten Unterrichtseinheiten wieder mitzubringen! Die Übersichten werden ihnen eine große Hilfe sein.

Auch das nächste Curriculum wird noch im Turtle Trainingscenter stattfinden. In der zweiten Hälfte werden wir weitere wichtige Turtle-Befehle besprechen und auf if-then-Bedingungen und while-Schleifen eingehen.

CURRICULUM 2 – Turtle Trainingscenter 2

Der zweite Part des Turtle Trainingscenters wartet! Eine Minecraft-Turtle kann sich nicht nur bewegen, sondern auch Blöcke erkennen, platzieren und abbauen. Sogar Kisten lassen sich von Turtles leeren und befüllen. Das werden die Kinder bald mit eigenen Augen sehen.

Die Unterrichtseinheit „Turtle Trainingscenter 2“ vermittelt den Kindern die restlichen, notwenigen Grundlagen, um später auf der Turtle Insel mit dem Bau einer eigenen Turtle-AG Filiale zu beginnen. Sie lernen die Turtle als Brückenbauer und Transporthelfer kennen.

Außerdem erfahren die Kinder, wie sie ihre Turtles mit while-Schleifen und if-then-Bedingungen noch effektiver programmieren können. Auch diesmal gibt es wieder einen kleinen Wettbewerb, in dem das Arbeitsblatt „Turtle Trainingscenter 2“ ausgefüllt und die anstehenden Abschnitte schnellstmöglich durchlaufen werden müssen.

Überblick

Kompetenzen

Zeitaufwand	**90 Minuten**
Technik	Laptops / Standrechner, je Gerät eine Mouse, LAN/WLAN, Beamer
Methoden	Gruppenarbeit, Simulation, Frage und Antwort, Arbeitsblatt
Vorkenntnisse	Programmieren mit Logo, Turtle Trainingscenter 1

Die Schülerinnen und Schüler …

- lernen, die Minecraft-Turtles Blöcke platzieren und abbauen zu lassen.
- werden if-then-Bedingungen in Lua programmieren.
- Lernen die while-Schleife in Lua kennen.
- lassen die Turtles überprüfen, ob ihnen Hindernisse im Weg stehen.
- machen die Turtles zum Transporthelfer.

Phase	Aufgabe	Methode	Zeit
Reflexion	Wiederholung: Was haben wir über Minecraft-Turtles gelernt?	F & A, Arbeitsblatt	5´
Vorbereitung	Freischalten des zweiten Trainingsabschnittes		5´
Arbeitsphase	Die Turtle als Bauarbeiter	Gruppenarbeit, Simulation und Arbeitsblatt	60´
	Blöcke erkennen mit der Turtle		
	Die Turtle als Transporthelfer		
	Das kaputte Rohrsystem		
Nachbereitung	Besprechung des Arbeitsblattes	F & A, Arbeitsblatt	15‘
	Präsentation der Lösungswege	Simulation	
Ausblick	Eine Filiale auf der Turtle Insel		5‘

Unterrichtsverlauf

Vorbemerkung

In diesem Curriculum wird es nun schon um Einiges kniffliger. Alle Informationen, welche die Kinder benötigen, um die Aufgaben zu lösen, lassen sich in der Minecraft-Welt selbst finden. Da die Aufgaben ein wenig komplexer werden, kann es eine Vielzahl unterschiedlicher Lösungswege für die einzelnen Bereiche geben. Machen Sie während der Arbeitsphase im Spiel immer wieder von Ihrer Teleportfunktion und dem Kreativmodus Gebrauch, um die Aufgaben bei Bedarf mit einzelnen Teams „vor Ort“ durchzugehen und Lösungsmöglichkeiten anzudeuten. Nutzen Sie dazu auch den Beamer, damit die Kinder gegebenenfalls einzelne Schritte besser nachvollziehen können!

Phase 1 | Reflexion

Nutzen Sie den Anfang der Stunde, um die letzte Unterrichtseinheit mit den Kindern zusammenzufassen und sich das Setting erneut ins Gedächtnis zu rufen.

- Was haben wir in Minecraft gemacht? Warum sind wir im Turtle Trainingscenter der Turtle-AG? (Antwort: Wir machen eine Mitarbeiterschulung. Wir sollen später eine neue Filiale der Turtle-AG aufbauen. Wir müssen lernen mit Turtles umzugehen.)

- Wichtig: Welche Regeln haben wir für das gemeinsame Spiel in Minecraft aufgestellt? (siehe Regeln Curriculum Turtle Trainingscenter 1)

Überlegen Sie gemeinsam mit den Kindern:

- Was könnte eine Minecraft-Turtle noch für uns tun? (Antwort: bauen, abbauen, Blöcke transportieren, …)

Sie können nun schon verraten, dass die Kinder ihrer Turtle heute sicherlich einige dieser Dinge beibringen werden.

Phase 2| Vorbereitung

2.1 Schritt eins | In Gruppen zusammenfinden und mit dem Server verbinden

Bitten Sie nun die Kinder, sich wieder genauso in den Gruppen zusammenzufinden, in denen sie auch das letzte Mal gespielt haben. Außerdem sollten alle das Arbeitsblatt „Turtle Trainingscenter 1“ parat haben. Teilen Sie zusätzlich das Arbeitsblatt „Turtle Trainingscenter 2“ aus (siehe Materialien). Dieses gilt es über den Verlauf der Unterrichtseinheit wieder auszufüllen.

Sollten Sie den Server noch nicht gestartet haben, tun Sie dies und verbinden sich im Spiel mit der Minecraftwelt. Auch die Kinder sollten sich wieder über die Accounts einloggen, die sie das letzte Mal genutzt haben. So wird garantiert, dass jeder dort wieder einsteigt, wo sie oder er das Spiel zuletzt verlassen hat.

Nachdem sich alle verbunden haben, bitten Sie die Kinder einen Moment zu warten. Sie müssen nun den nächsten Abschnitt freischalten, damit es weitergehen kann.

Hinweis: Sollte eines der Kinder das vorherige Mal noch nicht mit dabei gewesen sein, teilen Sie es einem der bestehenden Teams zu. Sie oder er muss sich jetzt nur noch vom passenden NPC in den entsprechend farbigen Teambereich teleportieren lassen und zum Rest der Gruppe laufen. Bitten Sie die restlichen Teammitglieder, ihren Neuzugang zu unterstützen, so gut es geht.

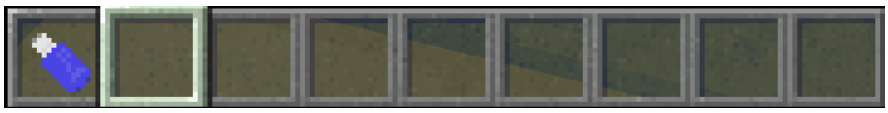
2.2 Schritt zwei | Den zweiten Abschnitt des Trainingscenters freischalten

Beginnen Sie mit den folgenden Anweisungen erst, wenn sich alle mit dem Server verbunden haben und alle Kinder im letzten zugänglichen Raum beim Turtle-AG Junior Manger stehen. Bitten Sie die Kinder, dort stehen zu bleiben.

- Öffnen Sie ihre Chatkonsole durch Drücken von „T“
- Geben Sie folgendes ein: /tp -59 86 469 (Bestätigen Sie mit der Entertaste)

- Somit teleportieren Sie sich zu einem speziellen Lehrerbereich (beliebig oft wiederholbar).

- Schauen Sie sich um: In Ihrer Nähe befindet sich eine Wand mit mehreren Knöpfen. Suchen Sie das Schild mit der Aufschrift „Bitte Knopf drücken, um Curriculum 2 zu starten!“ und betätigen Sie den Knopf darunter mit einem Rechtsklick!



- Alle Schüler sollten nun einen USB-Stick in ihrem Spielinventar haben.

Fragen Sie die Kinder, ob sie den Stick auch alle erhalten haben. Wenn nicht betätigen Sie den Knopf einfach erneut.

Hinweis: Damit der USB-Stick in das Inventar eines Spielers gelangen kann, muss dieser mindestens einen freien Slot in seinem Inventar zu Verfügung haben! Sollten Sie den Knopf, der die Vergabe der USB-Sticks aktiviert, aus Versehen zerstört haben, nehmen Sie sich über das Kreativmenü einfach einen neuen Knopf und platzieren ihn dort, wo der alte angebracht war.

Wenn Sie (erneut) den in der Welt integrierten Timer die Zeit nehmen lassen möchten, bitten Sie ein Kind aus jedem Team, den Computer im letzten Raum mit einem Rechtsklick zu aktivieren. Der Timer wird dann gestartet.

Mit dem USB-Stick im Inventar können die Kinder jetzt mit dem Turtle-AG Junior Manager sprechen. Dieser teleportiert sie dann automatisch in den nächsten Abschnitt des Trainingscenters, in dem es weitere knifflige Aufgaben zu lösen gilt.

Phase 3 | Arbeitsphase

Bevor die Kinder richtig loslegen, formulieren Sie noch einmal das Ziel dieser Unterrichtseinheit:

!! **Aufgabe:** Durchlauft als Team den zweiten Teil des Turtle Trainingcenters und füllt dabei das Arbeitsblatt „Turtle Trainingcenter 2“ vollständig aus! Hilfe beim Ausfüllen geben euch die NPCs, Schilder und Turtles in der Minecraft-Welt!

Erwähnen Sie noch einmal deutlich, dass auch dieses Mal wieder im schon bekannten Wettbewerbsformat gespielt wird. Das Team, das die Aufgabe als erstes löst, erhält am Ende wieder ein kleines Bonuspaket für das dritte Curriculum!

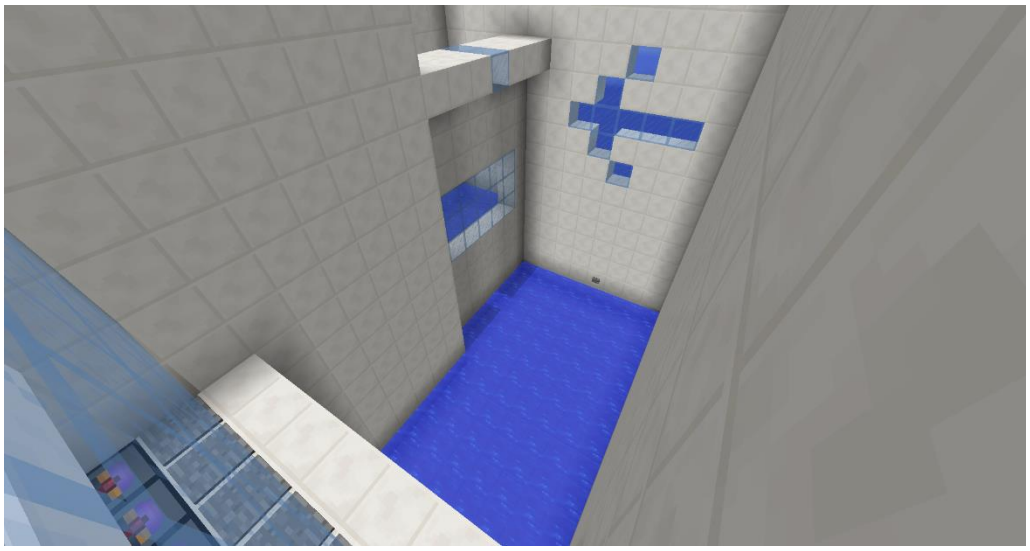
Erinnern Sie daran, dass es wichtig ist, eigene genutzte Turtles zu benennen, bevor größere Programme geschrieben oder eine Turtle wieder abgebaut wird! Eine Turtle ohne Namen verliert sämtliche selbstgeschriebene Programme, wenn sie abgebaut wird!

3.1 Die Turtle als Bauarbeiter

Im ersten neuen Abschnitt finden sich die Kinder in einer Halle mit Steinboden wieder. Ingenieur Gunnar beantwortet vor Ort folgende Fragen:

- Wie rüste ich die Turtle mit einem Werkzeug aus?
- Wie baue ich mit der Turtle Blöcke ab?

Nach kurzem Umsehen, werden die Kinder merken, dass sie einen größeren Abgrund überqueren müssen, um weiterzukommen.



Schnell wird den meisten klar: eine Art Brücke muss her!

Aufgabe: Gelangt mit Hilfe der Turtles auf die andere Seite des Abgrunds! (Tipp: Lasst die Turtles dafür Stein abbauen und eine Brücke errichten!)

Laufen die Kinder über die Druckplatten vor dem Abgrund, wird ihr Inventar gelöscht. Dies soll es erschweren, selbst mit abgebautem Stein eine Brücke zu bauen.

Hinweis: Einen kleinen Trick gibt es dennoch, um Gegenstände mit hinter die Druckplatten zu nehmen. Es kann passieren, dass Kinder auf die Idee kommen, ihre Gegenstände aus dem Inventar hinter die Druckplatten zu werfen, über die Druckplatte gehen und die Gegenstände hinter der Druckplatte wiederaufnehmen. Blöcke können dann auch per Hand platziert werden. Ganz verhindern lässt sich dies nicht. Weisen Sie die Kinder nicht auf diese Möglichkeit hin, verbieten Sie diesen Lösungsweg aber auch nicht, wenn er sich einmal in der Umsetzung befindet. Machen Sie nur klar, dass die hier vermittelten Turtle-Befehle später definitiv gebraucht werden!

Wer in den Abgrund fällt, landet in einem Wasserbecken. Am Ende des Beckens befindet sich ein Knopf, der die Kinder per Rechtklick wieder nach oben in die Halle teleportiert.

Um eine Brücke zu bauen, benötigen die Teams zunächst Baumaterial. Am besten eignen sich dafür die Steine aus dem Steinboden. Von Ingenieur Hans bekommen die Kinder bei Bedarf eine Turtle und eine Diamantspitzhacke. Letztere müssen sie der Turtle in unbenutztem Zustand ausrüsten, damit die Turtle auch vernünftig Stein abbauen kann.

Bevor die Kinder die Turtle ausrüsten, macht es Sinn, den kleinen Roboter zuerst auf eine der Aufladestationen an der Wand zu stellen und ihr mit „label set“ einen Namen zu geben, so wie es in der letzten Unterrichtseinheit erklärt wurde.

Die Turtle mit der Spitzhacke ausrüsten

Um die Turtle mit der Diamantspitzhacke auszurüsten, muss die Hacke erst einmal in den oberen linken (aktiven) Inventar der Turtle gelegt werden.

Zusätzlich muss die Turtle noch den passenden Befehl ausführen. Wir empfehlen, dafür ein simples Programm zu schreiben. Ein Beispiel: Über „edit werkzeug“ öffnen wir die Bearbeitung eines neuen Programms mit dem Namen „werkzeug“ und tippen einen der beiden folgenden Befehle ein:

- turtle.equipRight() rüstet das Werkzeug an der rechten Seite der Turtle aus.
- turtle.equipLeft() rüstet das Werkzeug an der linken Seite der Turtle aus.

Welchen der Befehle wir nehmen, ist uns überlassen, wir müssen uns nur für einen entscheiden. Eine Turtle kann bis zu zwei verschiedene Werkzeuge auf einmal ausgerüstet haben. Auf welcher Seite der Turtle ein Werkzeug ausgerüstet ist, hat jedoch keinen Einfluss auf die Funktionalität.

Nachdem das Programm abgespeichert und die Bearbeitung geschlossen wurde, können wir nun einfach „werkzeug“ eingeben und bestätigen. Und siehe da, die Turtle hat nun eine Spitzhacke an ihrer Seite.

Hinweis: Die Turtle kann nur Diamantwerkzeuge ausrüsten und auch nur, wenn diese sich in unbeschädigtem Zustand befinden! Sobald eine Spitzhacke einmal von einem Spieler benutzt wurde, ist sie bereits etwas beschädigt und lässt sich von einer Turtle nicht mehr ausrüsten. Sollten die Kinder ihre Spitzhacke zwischendurch aus Versehen bereits manuell benutzt haben, müssen sie das Werkzeug auswählen, mit „Q“ aus ihrem Inventar werfen und sich von Ingenieur Hans eine neue Hacke geben lassen.

Die Turtle Steine abbauen lassen

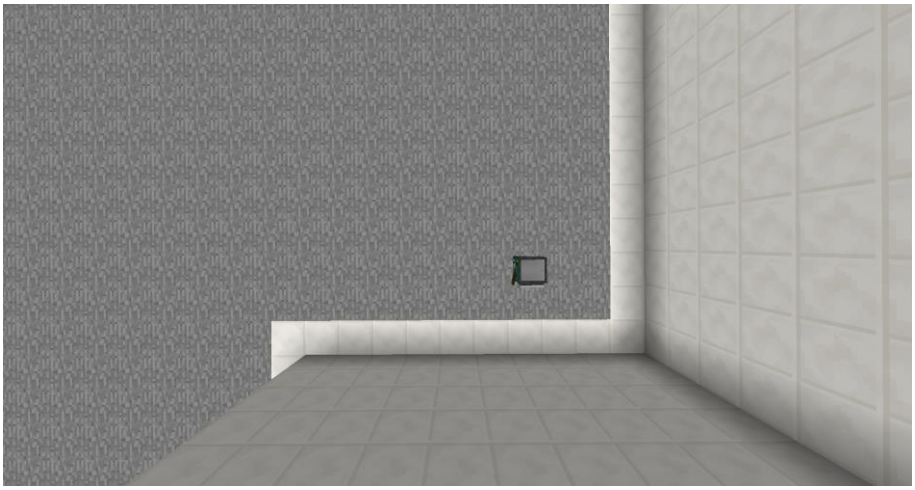
Die Kinder können auch versuchen, die Steine selbst mit der Spitzhacke abzubauen, sonderlich schnell voran kommen werden sie jedoch dabei nicht. Einen Steinblock manuell abzubauen dauert in dieser Minecraft-Welt fast 2 Minuten! Eine Turtle hat den Stein hingegen mit einem Schlag abgebaut und kann somit wesentlich schneller und sinnvoller zur Lösung der aktuellen Aufgabe beitragen.

Die Befehle zum Abbauen eines Blocks sehen für die Turtle in Lua wie folgt aus:

- turtle.dig() baut den Block direkt vor der Turtle ab.
- turtle.digUp() baut den Block über der Turtle ab.
- turtle.digDown() baut den Block unter der Turtle ab.

Lösungsbeispiel zum Abbau von Stein

- Nachdem wir unsere Turtle auf der Aufladestation aufgeladen, ihr den Namen „Bob“ gegeben und sie ausgerüstet haben, platzieren wir sie an folgendem Ausgangspunkt:



- Ausgehend von dieser Position öffnen wir über „edit abbauen“ die Bearbeitung eines neuen Programms mit dem Namen „abbauen“.

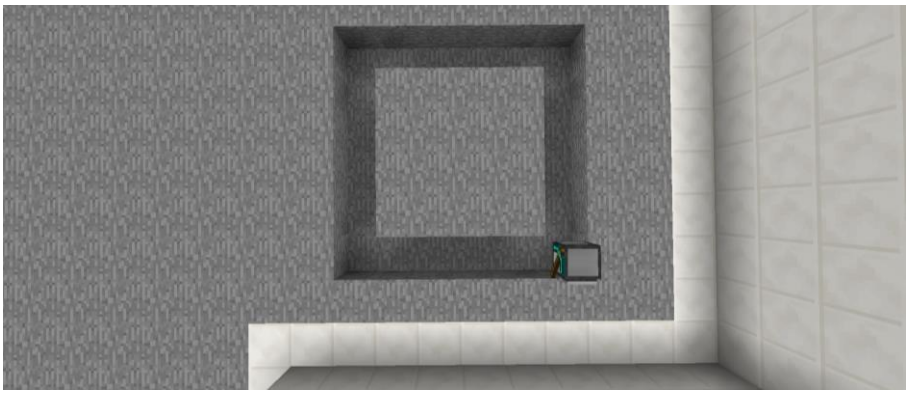
- Per Augenmaß schätzen wir, dass 20 Steine in etwa ausreichen sollten, um über den Abgrund zu gelangen, also sollte Bob ein entsprechendes Feld abbauen.

- Wir denken uns, dass sich diese Aufgabe schnell mit einer verschachtelten for-Schleife lösen lässt und geben folgenden Code ein:

```
for i=1,4 do

  for j=1,5 do
    turtle.forward()
    turtle.digDown()
  end

  turtle.turnLeft()

end
```

- Wir speichern das Programm ab, verlassen die Bearbeitung und führen das Programm „abbauen“ aus. Was geschieht nun?

- Bob baut den äußeren Rand eines 4x5 Blöcke großen Quadrates ab! Genau 20 Steine sollten sich nun im Inventar der Turtle befinden.



Aber was passiert beim Ausführen des Programms genau? Schauen wir uns zuerst die innere for-Schleife an. Alles was in der Schleife steht, soll fünfmal hintereinander ausgeführt werden. In unserem Fall bedeutet dies, dass Bob fünf Schritte nach vorn geht und dabei jeweils den Block unter sich abbaut. Somit hätten wir aber nur eine Seite des Quadrates abgebaut. Damit die insgesamt vier Seiten des Quadrates abgebaut werden, muss dieser Vorgang viermal wiederholt werden. Dafür nutzen wir eine weitere for-Schleife, welche die erste beinhaltet. Um die Turtle aber nicht nur viermal geradeaus laufen zu lassen, fügen wir am Ende noch eine Linksdrehung ein. Übersetzt wird viermal die Befehlskette „5er Streifen abbauen und nach links drehen“ durchlaufen, sodass Bob schließlich wieder auf ihrem Ausgangspunkt landet.

Hinweis: Hierbei handelt es sich um nur einen von sehr vielen Lösungswegen. Genauso gut könnten die Kinder ein kleines Programm schreiben, dass immer nur einen Block abbaut und einen Schritt vorwärtsgeht. Dieses Programm müssten sie dann nur 20 Mal immer wieder ausführen, um an die 20 Steine zu gelangen. Viele Wege führen zum Ziel. Natürlich muss auch nicht eine Turtle allein allen Stein sammeln. Die Teammitglieder können mehrere Turtles gleichzeitig arbeiten lassen.

Eine Brücke bauen

Nun, da genügend Steine vorhanden sind, gibt es wiederum eine Vielzahl an Möglichkeiten fortzufahren.

Lösungsbeispiel zum Bauen der Brücke

- Wir entscheiden uns im Beispiel dafür, unsere Turtle „Bob“ wieder abzubauen und auf eine der Aufladestationen am Abgrund zu platzieren. Beim Abbauen der Turtle, werden ihr die gesammelten Steine aus dem Inventar fallen. Wir sollten also nicht vergessen, die Steine mitzunehmen! Aber Vorsicht: Wir müssen aufpassen, dass wir nicht aus Versehen, über eine der Druckplatten am Abgrund laufen, denn dann ist

auch der ganze Stein (und im schlimmsten Fall unsere Turtle) aus unserem Inventar verschwunden und wir dürfen uns erneut in den Steinbruch begeben!

- Sobald wir die Turtle entsprechend platziert haben, legen wir ihr den Stein wieder in den ersten Slot ihres Inventars und schauen in Ruhe, was als nächstes zu tun ist.

- Eines der Schilder über dem Abgrund gibt uns die Information, dass Turtles nicht direkt über Druckplatten laufen können, sondern sich zuerst einen Block darüber begeben müssen.

- Daher starten wir mit „edit bruecke“ die Bearbeitung unseres neuen Brückenbauprogramms und geben als erstes „turtle.up()“ ein.

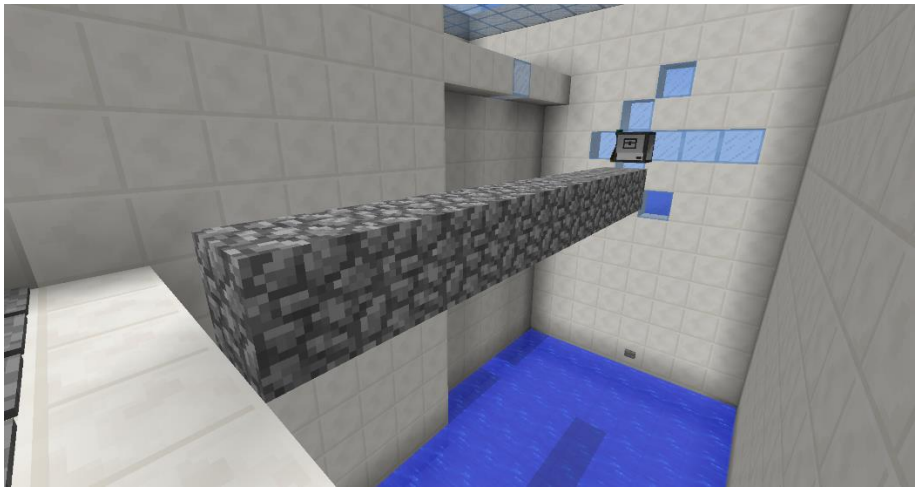
- Am einfachsten wird uns die Überquerung fallen, wenn wir Bob die Brücke direkt an der Kante des Abgrunds ansetzen lassen. Also bringen wir unsere Turtle mit dem nachfolgenden Code in Position:

```
turtle.up()

turtle.forward()
turtle.forward()
turtle.forward()
```

- Da wir schlecht abschätzen können, wie weit Bob die Brücke bauen soll, entscheiden wir uns, mit Hilfe einer for-Schleife erst einmal einen Steg von 9 Blöcken Länge bauen zu lassen:

```
for i=1,9 do
  turtle.placeDown()
  turtle.forward()
end
```

- Um besser einsehen zu können, wir am besten fortfahren, speichern wir das Programm, schließen die Bearbeitung und führen das Programm aus.

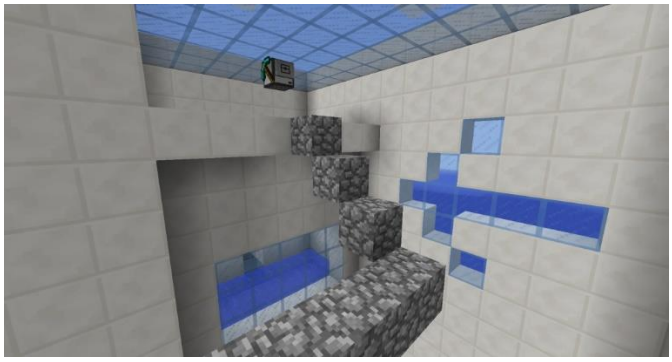


- Der erste Schritt zu Überquerung ist geschafft! Wir klettern auf die Brücke und folgen Bob bis zum Ende des Steges. Ein Blick nach links zeigt uns, dass wir noch nicht ganz am Ziel angekommen sind.

- Kurzerhand entscheiden wir uns, Bob über „edit treppe“ ein zusätzliches kleines Treppenprogramm zu verpassen:

```
turtle.turnLeft()

for i=1,4 do
  turtle.placeDown()
  turtle.up()
  turtle.forward()
end

turtle.forward()
```

- Wir speichern das Programm, verlassen die Bearbeitung und führen „treppe“ aus.



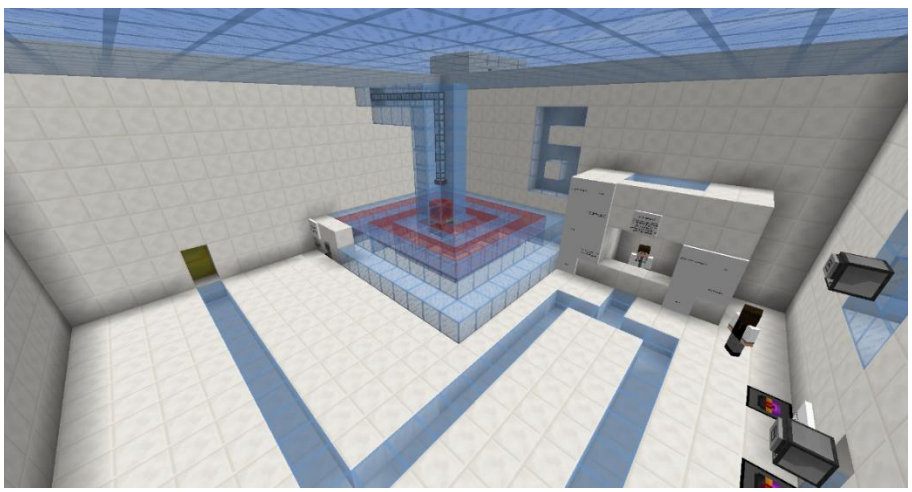
- Das sieht gut aus! Jetzt nur noch die Treppe hochspringen und wir haben unser Ziel erreicht. Wir bedanken uns bei Bob für seine gute Arbeit und springen weiter in den nächsten Abschnitt.

Hinweis: Auch hierbei handelt es sich nur um einen Lösungsweg von vielen. Hat die Turtle erst einmal etwas von der Brücke gebaut, können die Kinder beispielsweise auch die restlichen Steine aus dem Inventar der Turtle nehmen und sich selbst zum anderen Ende des Abgrundes hochbauen.

Geben Sie auch hier den Hinweis, das Ausfüllen des Arbeitsblattes über all das fleißige Programmieren und Bauen nicht zu vergessen!

3.2 Blöcke erkennen mit der Turtle

Nach einem kurzen Sprung ins kühle Nass finden sich die Kinder nun im nächsten Abschnitt wieder. Hier dreht sich alles um das Erkennen von Blöcken und den Einsatz von while-Schleifen und if-Bedingungen in einem Lua-Turtle-Programm.



Während Coderin Johanna hier mit einer Turtle samt Spitzhacke aushilft, erklärt Coderin Ines die nächste Aufgabe und beantwortet zum Thema passende Fragen:

- Wie schreibe ich eine while-Schleife in Lua?
- Wie schreibe ich eine if-then-Bedingung?
- Wie gebe ich ein Redstone-Signal nach oben aus?
- Wie kann die Turtle erkennen, ob sich direkt vor ihr ein Block befindet?

Um den Weg in den nächsten Bereich freizumachen, müssen die Kinder ihre Turtle dieses Mal einen schneckenhausähnlichen Tunnel durchlaufen und am Ende ein Redstone-Signal nach oben ausgeben lassen.

!! **Aufgabe:** Lasst eure Turtle den Schneckenhaus-Tunnel durchlaufen und am Ende ein Redstone-Signal an den Kolben über ihr abgeben!

Natürlich lässt sich der ganze Weg mit Einzelbefehlen bestreiten. Viel schneller und effizienter ginge es jedoch, wenn die Turtle eine Möglichkeit hätte, zu erkennen, wann sich vor ihr ein Block befindet, damit sie weiß, wann sie eine Rechtsdrehung vornehmen und weiterlaufen muss. Kein Problem, denn dafür gibt es in Lua folgende Befehle für unsere Minecraft-Turtles:

- turtle.detect() überprüft, ob sich direkt vor der Turtle ein Block befindet.
- turtle.detectUp() überprüft, ob sich direkt über der Turtle ein Block befindet.
- turtle.detectDown() überprüft, ob sich direkt unter der Turtle ein Block befindet.

All diese Befehle geben den Wert „true“ („Ja, da ist ein Block“) an die Turtle zurück, wenn sich ein Block in der entsprechenden Richtung anschließt und „false“ („Nein, da ist kein Block“), wenn der Weg frei ist.

Um etwas Produktives mit diesen Werten anfangen zu können, formulieren wir am besten passende Bedingungen dafür, was die Turtle tun soll, wenn sie einen Block vor sich findet.

if-then-Bedingungen

Es wäre toll, wenn wir der Turtle sagen könnten: „WENN du einen Block direkt vor dir siehst, DANN tue dieses und jenes“. Dafür gibt es in Lua, aber auch in andern Programmiersprachen, die if-then-Bedingung (die „wenn-dann-Bedingung“).

Wie so etwas im Lua-Code aussieht, zeigt das Beispielprogramm „blockdetector“ auf der gleichnamigen Turtle in diesem Abschnitt. Mit „edit blockdetector“ können wir uns den Programmcode ansehen:

```
if turtle.detect() then

  print("Block vor mir!")
  turtle.up()
else

  print("Kein Block vor mir!")
  turtle.down()
end
```

Übersetzt heißt das so viel wie: „Wenn die Blockdetector-Turtle vor sich einen Block findet, dann soll sie eine entsprechende Meldung ausgeben (print) und einen Block weiter nach oben wandern“. Das Wörtchen „else“ („anderenfalls“) gibt an, was passieren soll, wenn diese Bedingung NICHT erfüllt ist. In diesem Fall soll die Turtle im CraftOS einfach die Meldung „Kein Block vor mir!“ ausgeben und sich wieder einen Block nach unten bewegen.

Wenn wir das Programm ausführen, werden wir sehen, dass es genau so funktioniert. Natürlich ist es auch möglich, den Spieß umzudrehen und der Turtle zu sagen, sie soll eine Tätigkeit solange ausführen, wie eine Bedingung NICHT zutrifft. Dafür stellen wir einfach das Wort „not“ hinter „if“.

if not turtle.detect() then würde also den anschließenden Befehlsblock nur ausführen, wenn die Bedingung NICHT zutrifft! Wichtig ist es außerdem, immer ein „end“ ans Ende eines if-then-Blocks zu stellen, damit klar ist, wo die Bedingungsüberprüfung endet und das Programm richtig funktioniert.

Die while-Schleife

Im Gegensatz zu einer if-then-Bedingung, wird die festgelegte Bedingung in einer while-Schleife nicht nur einmal überprüft. Stattdessen wird der darunter stehende Befehlsblock ausgeführt, solange (englisch: „while“) die Bedingung im Schleifenkopf zutrifft.

Schauen wir uns dazu das Beispielprogramm „hoch“ der Turtle „hochgeher“ vor Ort an:

```
while turtle.detect() do
_ turtle.up()
end
```

Die Übersetzung: „Solange die Turtle einen Block vor sich sieht, soll sie sich einen Block nach oben bewegen“. Dieser Befehlsblock wird ausgeführt und danach erneut überprüft, ob die Bedingung noch erfüllt ist. Führen wir das Programm aus, werden wir sehen, dass die Turtle genau dies tut. Kann sie keinen Block mehr vor sich finden, bleibt sie stehen und das Programm ist beendet.

Der Schneckenhaus-Tunnel

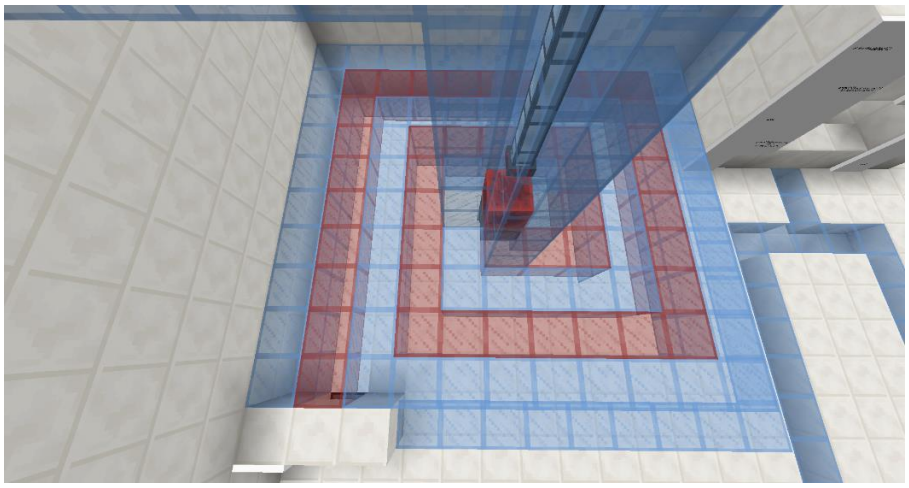
Animieren Sie die Kinder dazu, sich auch in diesem Raum in Ruhe umzusehen. Alle zuvor beschriebenen Informationen erhalten sie zum einen im Gespräch mit Coderin Ines und durch das Lesen der Beispiele auf den Schildern, zum anderen können sie sich auch die Beispielprogramme auf den Turtles ansehen.

Hinweis: Ist ein Team bei diesem Aufgabenbereich angelangt und weiß nicht so recht, wie es eine Lösung für den Schneckenhaustunnel finden soll, gehen sie die Beispiele wie oben beschrieben gemeinsam Schritt für Schritt durch. Wir haben die Erfahrung gemacht, dass es sehr hilfreich sein kann, erst einmal in Worten und auf Papier zu formulieren, was die Turtle tun soll. So lässt sich anschließend viel einfacher eine „Übersetzung“ in den Lua-Programmcode formulieren.

Achtung! Auch bei diesem Tunnel gilt wieder: Steckt die Turtle erst einmal fest, helfen nur noch der Reset-Knopf und eine neue Turtle!

Lösungsbeispiel

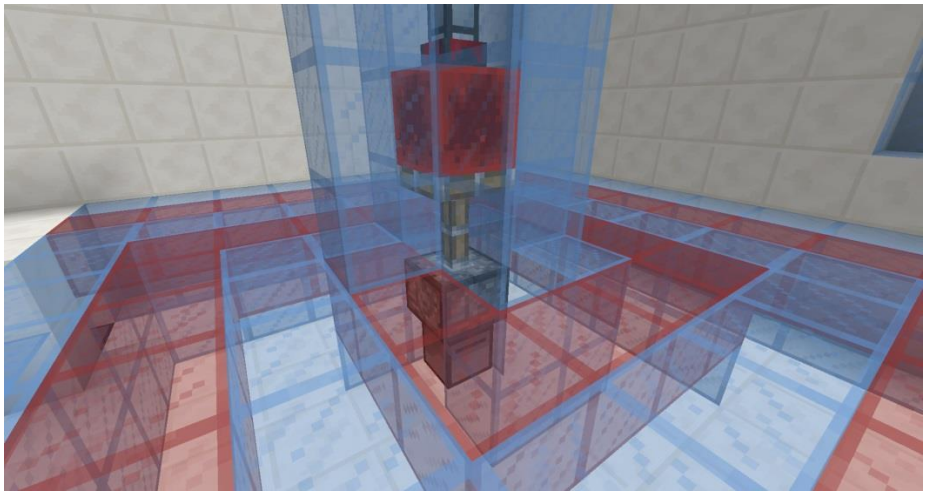
- Nachdem wir unsere Turtle mit Namen „Finn“ auf der Ladestation am Eingang des Tunnels platziert haben (mir Blick Richtung Tunnel), klettern wir zunächst auf das Glasdach, um uns eine grobe Übersicht zu verschaffen.



- Anschließend versuchen wir die Bearbeitung des Problems in Worte zu fassen: „Solange die Turtle KEINEN Block vor sich hat, soll sie einen Schritt vorwärtsgehen. Wenn sie jetzt einen Block vor sich hat (also auf eine Wand trifft), dann soll sich die Turtle nach rechts drehen“. Der Vorgang soll wiederholt werden, bis die Turtle das Ende erreicht hat. Wenn sie hier nämlich auf eine Wand trifft und sich dreht, hat sie direkt wieder eine Wand vor sich und kann nicht mehr weitergehen. Dann fehlt nur noch das Redstone-Signal, das die Turtle nach oben ausgeben muss.

- Nun übersetzen wir unsere Überlegungen in den passenden Programmcode:

```
while not turtle.detect() do

  turtle.forward()
  if turtle.detect() then
    turtle.turnRight()
  end

end

  redstone.setOutput("top", true)
```



- Speichern wir das Programm ab und führen es aus, werden wir feststellen, dass alles so funktioniert, wie wir es uns vorgestellt haben! Toll, durch das Verwenden der while-Schleife und der if-then-Bedingung haben wir nicht nur Zeit, sondern auch eine Menge Codezeilen gespart!

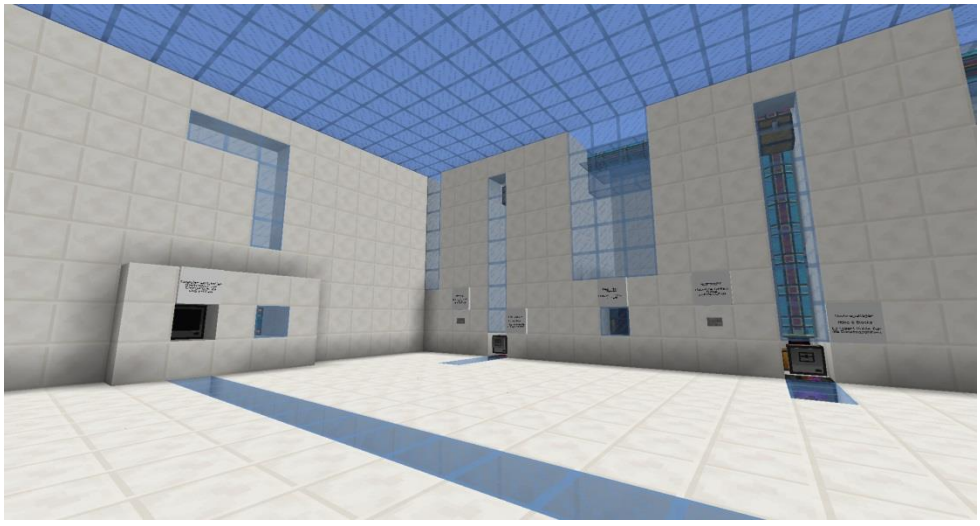
Hinweis: Allein mit while-Schleifen und if-then-Bedingungen lässt sich hier eine Vielzahl an Lösungen finden. Natürlich lassen sich auch for-Schleifen einbinden. Sogar ganz ohne Schleifen und Bedingungen ist eine Lösung möglich, wenn auch mit signifikant größerem Zeitaufwand. Alle Lösungen, die zum Ziel führen sind legitim. Hat sich ein Team in einem eher umständlichen Lösungsansatz verfangen, weisen Sie auf effizientere Alternativen hin, zwingen Sie die Kinder aber nicht dazu, einen speziellen Lösungsweg zu gehen!

Prima! Auch dieser Abschnitt ist geschafft. Mit dem Wissen über while-Schleifen und if-then-Bedingungen lassen sich viele zukünftige Aufgaben in der Code your Life Minecraft-Welt lösen.

3.4 Die Turtle als Transporthelfer

Schon zuvor haben die Kinder von den NPCs erfahren, dass sich einige Maschinen aus Sicherheitsgründen nur von Turtles bedienen lassen. In diesem Bereich müssen die Turtles Kohle aus einem Hochregallager (einer Kiste) holen und in eine Dampfmaschine füllen, die dann wiederum Energie produziert. Diese Energie wir durch die Rohre in einen Energiespeicher geleitet. Es werden mindestens 400000 RF (Redflux ist hier die Stromeinheit) benötigt, um die Tür zum letzten Aufgabenbereich zu öffnen. Der Energiestand kann jederzeit an einem Computer am Ende der Halle abgefragt werden.

Logistikerin Katrin erklärt an ihrem Stand die zu lösende Aufgabe.



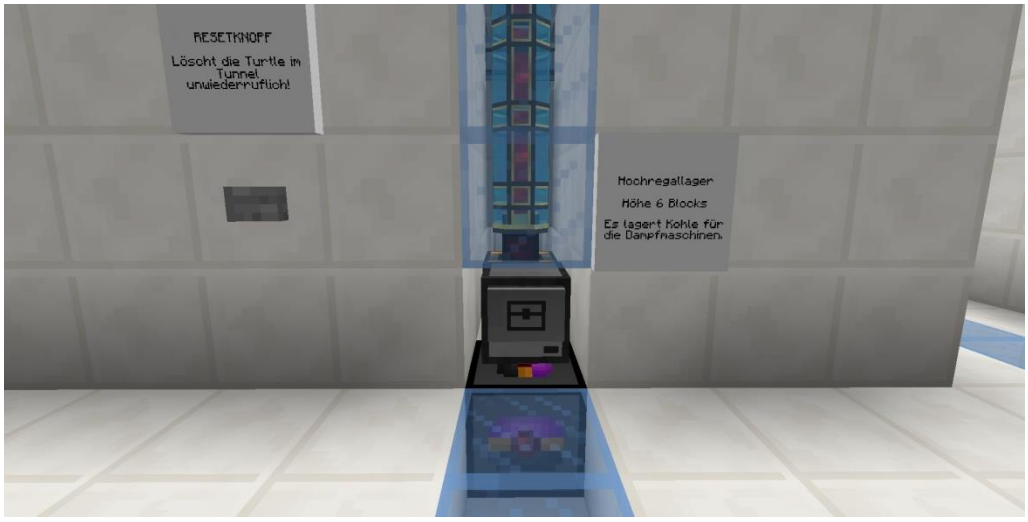
Aufgabe: Sammelt mit Hilfe eurer Turtles ausreichend Energie, um in den nächsten Abschnitt zu gelangen! Nutzt eure Turtles, um Kohle von den Hochregallagern in die Dampfmaschinen zu geben!

Am Stand von Katrin gibt es außerdem neue Turtle-Befehle auf Schildern zu entdecken:

- turtle.suck() versucht, Gegenstände aus einem fremden Inventar (Kiste, Ofen, usw.) direkt vor sich in das eigene Inventar zur ziehen.
- turtle.suckUp() versucht, Gegenstände aus einem fremden Inventar direkt über sich in das eigene Inventar zu ziehen.
- turtle.suckDown() versucht, Gegenstände aus einem fremden Inventar direkt unter sich in das eigene Inventar zu ziehen.
- turtle.drop() wirft Gegenstände aus dem aktiven Slot des eigenen Inventars direkt vor sich aus.
- turtle.dropUp() wirft Gegenstände aus dem aktiven Slot des eigenen Inventars direkt über sich aus.
- turtle.dropDown() wirft Gegenstände aus dem aktiven Slot des eigenen Inventars direkt unter sich aus.

Mit Hilfe dieser Befehle lässt sich unsere Aufgabe einfach lösen. Von Logistikerin Lisa erhalten wir bei Bedarf wieder Turtle und Spitzhacke. Insgesamt gibt es fünf Hochregalschächte, in die die Kinder ihre Turtles schicken können. So hat auch bei fünf Teammitgliedern jeder die Gelegenheit, seine eigene Turtle zu benutzen

Hinweis: Je mehr der fünf Dampfmaschinen befüllt werden, umso schneller schreitet die Energieproduktion voran. Weisen Sie die Kinder darauf hin! Zwar genügt es auch, lediglich eine Dampfmaschine zu befüllen, bis schließlich der benötigte Energiestand erreicht ist, dauert es aber dann wieder seine Zeit. Wer im Team arbeitet ist auch hier wieder klar im Vorteil!



Lösungsbeispiel

- Wir stellen unsere Turtle mit Namen „Porti“ auf die Aufladestation vor einem der Hochregalschächte.
- Eines der Schilder informiert uns darüber, dass der Schacht sechs Blöcke hoch ist.
- Mit „edit lager“ starten wie die Bearbeitung des Porgramms „lager“.
- Dieses Mal entscheiden wir uns, wieder zwei for-Schleifen zu Hilfe zu nehmen. Wir schreiben folgenden Code:

```
turtle.forward()

for i=1,6 do
  turtle.up()
end
turtle.suckUp()
for i=1,6 do
  turtle.down()
end

turtle.drop()
```

- Wir schicken Porti zunächst einen Schritt nach vorn, damit sie über sich freie Bahn hat. Mit Hilfe der ersten for-Schleife klettert die Turtle den Schacht rauf, nimmt sich dann mit dem neu Erlernten turtle.suckUp() alle Items aus der Kiste über sich, steigt mit der zweiten for-Schleife wieder hinab und befüllt schließlich die Dampfmaschine direkt vor sich per turtle.drop().
- Nun nur noch schnell das Programm abspeichern und ausführen!
- Super, ein Blick auf den Statuscomputer zeigt, dass die erste Energie schon in den Speicher befördert wird! Jetzt fix schauen, ob die Teammitglieder auch gut zurechtkommen.

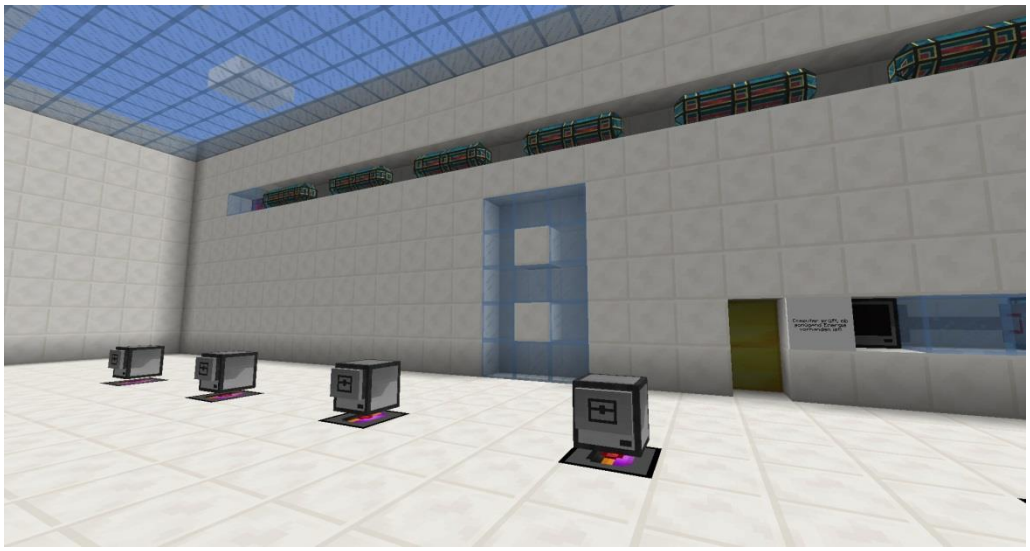
```
15680 Energie gespeichert!
Benoetige 400000 RF !
15760 Energie gespeichert!
Benoetige 400000 RF !
15840 Energie gespeichert!
Benoetige 400000 RF !
15920 Energie gespeichert!
Benoetige 400000 RF !
16000 Energie gespeichert!
Benoetige 400000 RF !
16080 Energie gespeichert!
Benoetige 400000 RF !
16160 Energie gespeichert!
Benoetige 400000 RF !
16240 Energie gespeichert!
Benoetige 400000 RF !
16320 Energie gespeichert!
Benoetige 400000 RF !
```

Es liegt nahe, dass es auch hier wieder verschiedene Lösungswege gibt. Ohne die neuen Befehle funktioniert es jedoch nicht. Erinnern Sie die Kinder ruhig noch einmal an das Ausfüllen des Arbeitsblattes.

3.5 Das kaputte Rohrsystem

Fast geschafft! Eine kleine Aufgabe gilt es nun noch zu bewältigen, bevor die Kinder zu vollwertigen Mitarbeitern der Turtle-AG befördert werden. Im nächstem Raum fällt das Problem recht schnell ins Auge: Die Energieleitung, die den Öffnungsmechanismus an der nächsten Tür bedient, ist unterbrochen! Wie erfreulich, dass auch hier die Turtles zur Stelle sind, um auszuhelfen.

Zu Beginn stehen bereits fünf Turtles auf einer Ladestation. Die Besonderheit: Jede Turtle verfügt bereits über ein Energie-Transportrohr in ihrem Inventar.



Zwar können die Kinder die Rohre durch umständlichere Tricks auch ohne Turtle-Programmierung reparieren, schneller geht es jedoch immer noch mit den Turtles. Die place-Befehle, mit denen die Turtle einen Block platziert, sollten die Kinder noch vom Brückenbau zu Beginn des Curriculums kennen. Sie sind hier aber auch noch einmal auf entsprechenden Schildern abgebildet.

!! **Aufgabe:** Repariert die Energieleitung!

Elektrikerin Magda gibt uns hier bei Bedarf eine neue Turtle samt Diamantspitzhacke.

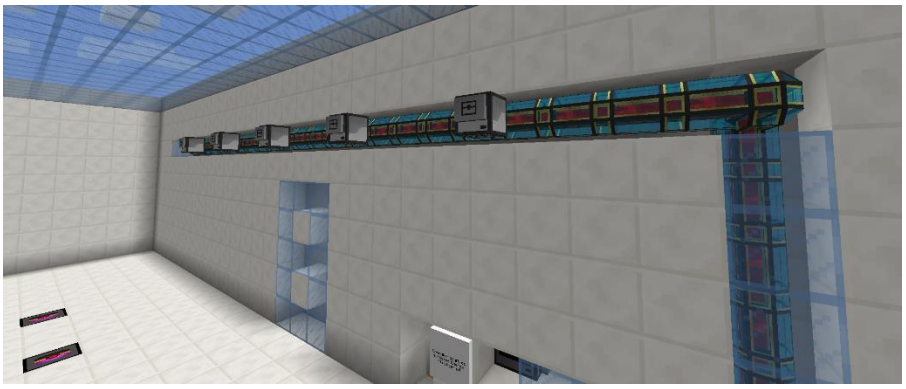
Lösungsbeispiel

- Von der Ladestation aus muss unsere Turtle vier Schritte nach vorn gehen, bis zur Lücke in der Energieleitung aufsteigen und das fehlende Rohr platzieren.

- Nachdem wir uns vergewissert haben, dass ein Rohr im aktiven Inventarslot der Turtle liegt, schreiben wir folgendes kleines Programm mit dem Namen „rohrsetzen“:

```
for i=1,4 do
  turtle.forward()
end

while turtle.detect() do
  turtle.up()
end

turtle.place()
```

- Wir speichern und führen das Programm aus. Auf gleiche Art und Weise lassen wir die übrigen Lecks in der Energieleitung reparieren.



- Dann noch schnell den Computer unten neben der Tür rechtsklicken, um den Energiefluss zu überprüfen.

- Die Tür öffnet sich! Geschafft!

Hinweis: Sollten die Kinder die Rohre aus Versehen falsch setzen, sodass sie nicht mehr erreichet werden können oder über die Druckplatten laufen, sodass die Rohre verschwinden, begeben Sie sich in den Kreativmodus, suchen Sie nach „Kryotheum-Stabilisierter Energiedukt“, nehmen Sie sich ein paar Exemplare ins Inventar und geben Sie diese anschließend den Kindern. Teleportieren Sie sich zum entsprechenden Team und werfen sie die ausgewählten Rohre mit „Q“ aus ihrem Inventar, damit die Kinder sie einsammeln können.

Hinter der Tür wartet der Turtle-AG Manager auf die Kinder und gratuliert ihnen zur erfolgreich durchlaufenen Turtle-AG-Mitarbeiterschulung. Anschließend werden die Kinder wieder in die Empfangshalle teleportiert.

Phase 4 | Nachbereitung

Gestatten Sie den Teams, die bereits fertig sind, entweder noch ein wenig frei zu spielen, oder motivieren Sie sie dazu, auch die anderen Teams zu unterstützen.

In den letzten 20 Minuten sollten Sie jedoch, genau wie im Curriculum zuvor, zum Vergleich des Arbeitsblattes übergehen.

4.1 Das Arbeitsblatt vergleichen

Auch alle Felder auf dem Arbeitsblatt „Turtle Trainingscenter Teil 2“ lassen sich mit Hilfe der Schilder, Turtles und NPCs problemlos ausfüllen. Gehen Sie das Blatt mit den Kindern wieder schrittweise durch.

Vergleichen Sie die Ergebnisse der Kinder mit der ausgefüllten Version des Arbeitsblattes aus dem Materialanhang der Handreichung (siehe Materialien).

Stellen Sie erneut Fragen wie:

- Wo habt ihr den Befehl gefunden? (Antwort: NPC mit Namen, Abschnitt, Schild, usw.)
- Haben alle verstanden, wie der Befehl funktioniert?

4.2 Präsentation der Lösungswege

Bevor es an die Auszeichnung des diesmaligen Gewinnerteams geht, sollen die Kinder nun noch einmal die Gelegenheit bekommen, ein paar ihrer Lösungswege zu demonstrieren. Sofern es die Zeit zulässt, können Sie alle Bereiche nochmal über den Beamer gemeinsam mit den Kindern durchgehen.

Lassen Sie verschiedene Teams die einzelnen Aufgaben und ihre Lösungen vorstellen. Regen Sie zur Diskussion unter den Kindern an:

- Wie haben die anderen Teams, die Aufgaben gelöst?
- Welcher Code ist der effizienteste?

4.3 Gewinner der Challenge

Auch dieses Mal soll das Team gekürt werden, das die Aufgaben und das Arbeitsblatt am schnellsten gelöst hat. Der Preis: ein weiteres Paket mit Dekorationsgegenständen für die eigene Filiale, die es ab Curriculum 3 zu bauen gilt. Notieren Sie sich am besten wieder den Namen des Teams, damit sie nicht vergessen, wem sie das Paket im nächsten Curriculum zukommen lassen müssen.

Phase 5 - Ausblick

Erst einmal dürfen Sie den Teams nun gratulieren. Die Kinder haben sich in der Mitarbeiterschulung der Turtle-AG intensiv mit den wichtigsten Befehlen der Turtle-Programmierung auseinandergesetzt und sollten fit genug sein, nun auf die Turtle Insel entlassen zu werden.

Dort müssen sie in der nächsten Unterrichtseinheit mit dem Bau einer eigenen Filiale beginnen, natürlich mit tatkräftiger Unterstützung der kleinen Turtles.

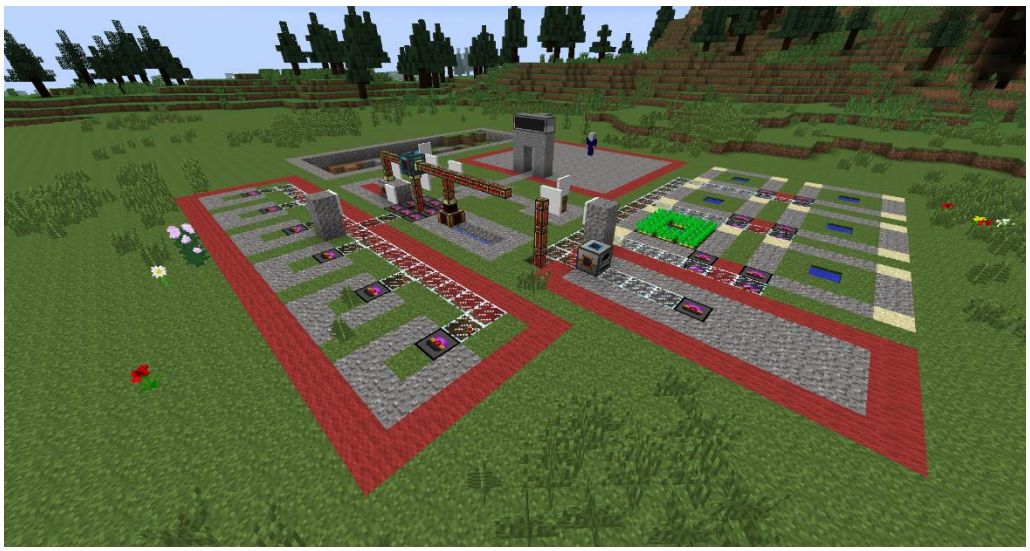
Hinweis: Um schnell den ein oder anderen Befehl nachschlagen zu können, ist es wichtig, dass die Kinder beide Arbeitsblätter auch zur nächsten Unterrichtseinheit wieder mitbringen!

CURRICULUM 3 – Eine Filiale ensteht

Endlich ist es soweit – Die Kinder haben die Turtle-AG-Mitarbeiterschulung durchlaufen und können sich nun der spannenden Aufgabe widmen, in Teams eigene Filialen auf der Turtle Insel zu errichten!

In der dritten Unterrichtseinheit erhält jedes Team ein Stück Land. Mit dem Wissen aus den beiden vorherigen Curricula gilt es zunächst, den lokalen Steinabbau mit Turtles anzukurbeln, um Baumaterialien für ein hübsches erstes Filialgebäude zu erhalten. Wie genau sie diese Aufgabe bewältigen, ist dabei den Kindern überlassen. Sie können alle ihnen zur Verfügung stehenden Ressourcen nutzen.

Wer baut die schönste Filiale? Wer programmiert die findigsten Turtles? Gemeinsam finden es, dies herauszufinden!



Überblick

Kompetenzen

Zeitaufwand	**90 Minuten**
Technik	Laptops / Standrechner, je Gerät eine Mouse, LAN/WLAN, Beamer
Methoden	Gruppenarbeit, Simulation, Frage und Antwort, Arbeitsblatt
Vorkenntnisse	Programmieren mit Logo, Turtle Trainingcenter 1 & 2

Die Schülerinnen und Schüler …

- entdecken gemeinsam die Turtle Insel.
- müssen ihr Programmierwissen problemlöseorientiert anwenden.
- bauen ein Turtle-AG Filialgebäude.

Ablauf

Phase	Aufgabe	Methode	Zeit
Reflexion	Was haben wir über Minecraft-Turtles gelernt? Wie können uns die Turtles auf der Insel helfen?	F & A	5´
Vorbereitung	Orientierung: Das Startgelände		10´
	Vorstellung der Aufgabe und Verteilung der Teams auf der Insel		
Arbeitsphase	Freiarbeit: Bearbeitung der Bau-Challenge	Gruppenarbeit, Simulation und Arbeitsblatt	50´
Nachbereitung	Präsentation der Filialgebäude und Turtles	Präsentation, Simulation, Arbeitsblatt	15‘
Ausblick	Regenerative Energieversorgung		5‘

Unterrichtsverlauf

Vorbemerkung

Von hier an gestalten sich die Curricula etwas freier, da die Kinder nicht mehr in einem abgegrenzten Gebiet spielen, sondern sich ohne größere Einschränkung auf der Turtle Insel bewegen können. Dementsprechend sind Lösungswege auch noch einmal variabler. Exemplarisch skizzieren wir im Curriculum einen Weg, auf welchem sich die Challenge bestreiten lässt. So haben Sie bei Bedarf eine gute Orientierungshilfe zur Hand.

Wichtig für die Lehrkraft:

Öffnen Sie Ihre Chat-Konsole durch Drücken von „T“ und geben Sie

/tp -59 86 469

ein, um sich zu einem speziellen Lehrerbereich zu teleportieren. Vermutlich werden Sie diesen Befehl im Verlauf der nächsten Curricula mehrmals nutzen, um von diesem Areal aus Gegenstände an die Teams auszugeben und schneller zwischen den Teams hin und her zu teleportieren.

Phase 1 | Reflexion

Zu Beginn fassen Sie noch einmal gemeinsam mit den Kindern zusammen, was diese in den letzten beiden Unterrichtsstunden gelernt haben.

Frage: Was kann eine Turtle alles für uns tun?

Antworten:

- sich bewegen
- Blöcke abbauen
- Blöcke platzieren
- Blöcke erkennen
- Gegenstände aus einem Inventar (Kiste, Ofen) holen
- Gegenstände in ein Inventar (Kiste Ofen) legen
- Signale ausgeben

Die wichtigsten Befehle und Prinzipien (for-/while-Schleife, if-then-Bedingung) haben die Kinder übersichtlich auf ihren Arbeitsblättern „Turtle Trainingscenter 1“ und „Turtle Trainingscenter 2“ zusammengetragen. Eine große Hilfe für die anstehenden Curricula! Empfehlen Sie den Kindern, die Blätter griffbereit zu halten.

Phase 2 | Vorbereitung

Bevor es nun ins Spiel geht, sollten Sie sich gemeinsam anschauen, was die Kinder auf der Insel erwartet.

2.1 Orientierung: Das Startgelände

Starten Sie den Server, loggen Sie sich ein und gehen Sie ins Spiel. Wechseln Sie in den Kreativmodus (/gamemode 1) und teleportieren Sie sich mit /tp -59 86 469 zum separaten Lehrerbereich. Dieser Bereich sollte Ihnen noch aus Curriculum 2 bekannt sein. Suchen Sie hier den Knopf mit der Überschrift „Lehrer Teleport zur Insel“ und betätigen Sie diesen per Rechtsklick.

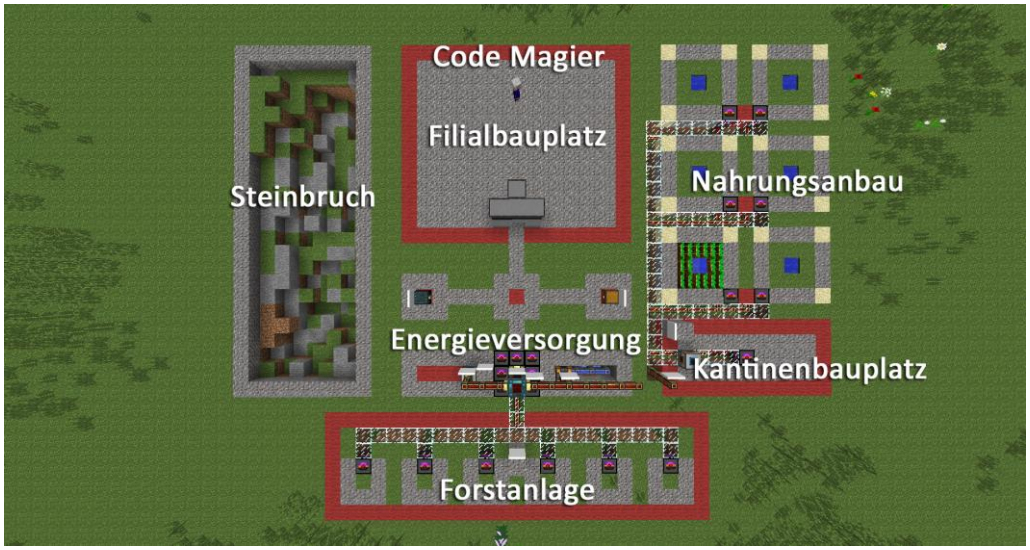
Sie landen nun in einer Halle mit zwei NPCs, den Turtle-AG Supervisors. Lassen Sie sich vom NPC mit der Beschreibung „Teleportiert zu den Teams“ zu einem Teambereich Ihrer Wahl, z.B. „Team Rot“, teleportieren. Dazu müssen Sie ihn lediglich per Rechtsklick ansprechen und eine entsprechende Auswahl treffen.

Sie sollten jetzt in einem der Teamareale stehen. Erklären Sie den Kindern über den Beamer die einzelnen Bestandteile des Startgeländes, wie nachfolgend beschrieben.

Hinweis: Die Kinder sollten zu diesem Zeitpunkt noch nicht mit dem Spiel verbunden sein! Gehen Sie die einzelnen Elemente des Startgeländes zuerst in Ruhe mit den Kindern durch. Sind die sie erst einmal im Spiel, ist es schwieriger bei all der Begeisterung eine entsprechende Aufmerksamkeit zu generieren. Es ist wichtig, dass die Teams ihre Ausgangssituation erfassen, um die nachfolgende Aufgabe anzugehen.

Übersicht

Fliegen Sie im Kreativmodus, wie oben auf der Grafik abgebildet, so über das Startgelände, dass Sie alle Strukturen gut einsehen können. Die Startareale der Teams sind bis auf die unterschiedlichen Farbmarkierungen baugleich. Jeder startet mit den gleichen Voraussetzungen. In dieser Unterrichtseinheit sind vor allem der Steinbruch und der Filialbauplatz interessant. In späteren Curricula werden aber auch die Forstanlage, der Nahrungsanbau und der Kantinenbauplatz relevant.



Hinweis: All die zur Verfügung gestellten Anfangsstrukturen sollen lediglich als Orientierungshilfe dienen. Prinzipiell kann jedes Team auf der Insel bauen, wie und wo es möchte. Wir empfehlen, den Kindern dennoch das Nutzen der vorhandenen Anlagen an Herz zu legen.

Steinbruch

Stein zählt auf der Insel tatsächlich zu den wichtigsten Ressourcen. Diesen benötigen die Teams zum einen, um ihre Filiale aufzubauen, zum anderen dienen gebrannter Stein und Bruchstein als eine Art Tauschwährung, mit der bei NPCs später Werkzeuge, Maschinen, Nahrung und Saatgut gekauft werden kann. Der angedeutete Steinbruch ist eine gute Anlaufstelle für die Turtles der Kinder, um mit dem Abbau zu beginnen. Natürlich lässt sich der Steinbruch erweitern bzw. Stein auch anderen Orts ausfindig machen.

Filialbauplatz

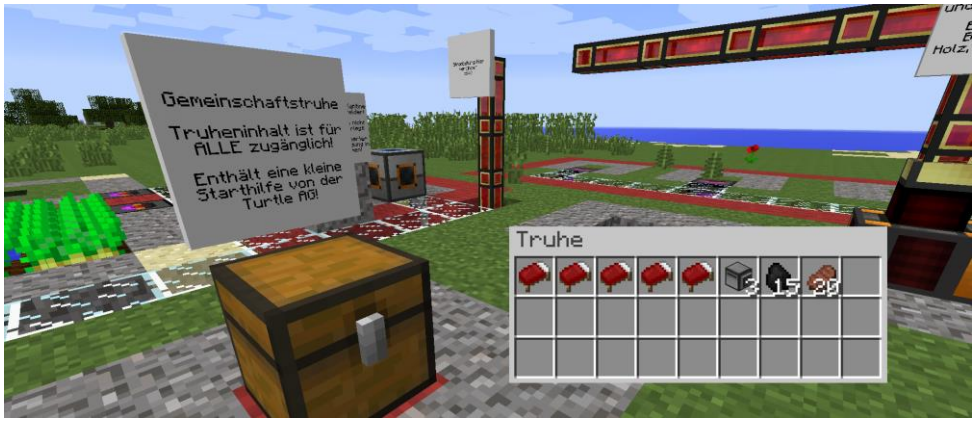
Hier sollen die Kinder idealerweise ihre Filiale errichten. Natürlich kann das Gebäude auch größer oder an anderer Stelle aufgestellt werden. Alle Kriterien, die es beim Bau zu erfüllen gilt, sind auf dem Arbeitsblatt „Auftrag 1: Die Filiale errichten!“ (siehe Materialanhang) für die Teams zusammengefasst.

Energieversorgung

Weiterhin befindet sich auf dem Startgelände eine kleine Energieversorgungsanlage, bestehend aus einer Dampfmaschine, einer (leeren!) Energiezelle, einer angeschlossenen Turtle Ladestation und einem Schmelzofen, in dem Materialien weiterverarbeitet werden können. Die einzelnen Komponenten sind, bis auf den Redstone-Ofen, bereits durch Energieleitungen (Restone-Energiedukte) miteinander verbunden und können, wie in der Mitarbeiterschulung erklärt, aus Sicherheitsgründen nur von Turtles bedient werden.

Truhen

Zu Beginn wird den Teams jeweils eine Gemeinschaftstruhe mit fünf Betten, drei Crafty Turtles (nur diese Turtles können komplexere Rezepte craften!), etwas Kohle für die Dampfmaschine, einigen Energietransportrohren und zwanzig Steaks zur Verfügung gestellt. Außerdem gib es gegenüber noch eine unzerstörbare Privat-Endertruhe. Dort hat jeder nur Zugriff auf die Gegenstände, die er oder sie selbst dort hineingelegt hat.



Forstanlage

Im vierten Curriculum werden die Kinder die Aufgabe bekommen, eine kleine Baumfarm mit Hilfe ihrer Turtles zu realisieren, um zum Teil auf eine regenerative Energieversorgung mit Holzkohle umzusteigen. Wie genau das funktionieren soll, dazu an späterer Stelle mehr. Empfehlen Sie den Kindern vorerst, den dafür vorangelegten Bereich nicht abzubauen, verbieten Sie es aber auch nicht!

Nahrungsanbau

Im letzten und abschließenden Curriculum sollen die Teams eine autonome Nahrungsversorgung ihres Filialgebietes umsetzen, indem sie ihre Turtles automatisiert Kartoffeln anbauen, ernten und in einer Kantine zu Ofenkartoffeln verarbeiten lassen. Auch hierzu an später mehr. Bis dahin ist von einem Um- oder Abbau der Anlage abzuraten, denn die angedeuteten Strukturen können später noch eine große Hilfe sein.

Der Code Magier

Auf dem Filialbauplatz jedes Teams steht ein Code Magier NPC, der in einem Gespräch nützliches Zusatzwissen preisgibt. Wer aufmerksam liest, erhält so die ein oder andere Info, um das Team noch schneller voran zu bringen und die eigene Filiale ansprechender zu gestalten. Der Code Magier beantwortet Fragen zu:

- der Verwendung von Funktionen in Lua.
- dem Benutzen von Computern und Monitoren.
- dem genauen Identifizieren von Blöcken mit Turtles.
- Variablen.
- Endlosschleifen.
- dem Kopieren von Programmen.

Ein Gespräch mit dem Code Magier ist jetzt noch nicht zwingend notwendig. Fast alle Aufgaben lassen sich auch ohne seine Tipps bewältigen. Wer bei der Bewertung der eigenen Filiale besonders punkten will, sollte sich jedoch anhören, was der Magier zu sagen hat und seinen Rat in die Tat umsetzen. Sehr wichtig wird das Code Magier Wissen im nächsten Curriculum.

Hinweis: Die Spielfiguren der Kinder werden im Verlauf des Curriculums Hunger bekommen und an Gesundheit verlieren, wenn sie nichts essen! Zu erkennen ist dies an den Fleischkeulen und Herzen über der Gegenstandsauswahlleiste. Die zwanzig Steaks aus der Gemeinschaftstruhe sollten für diese Unterrichtseinheit zwar genügen, wer dennoch mehr Nahrung benötigt, kann sich aber bei den Turtle-AG Verkäufern im Turtle-AG Turm für Stein Kartoffeln kaufen. Um zu essen, muss der Spieler zunächst Nahrung wie andere Blöcke „ausrüsten“ und dann die rechte Maustaste gedrückt halten, bis die Portion verzehrt wurde. Je nachdem, was gegessen wird, regeneriert sich die Hungerleiste wieder. Um eine automatisierte Nahrungsversorgung kümmern wir uns in Curriculum 5.

Der Turtle-AG Turm

Ein Blick zur Mitte der Insel führt unweigerlich zum zentralen Turtle-AG Turm. Begeben Sie sich auch hierhin und zeigen Sie den Kindern die beiden Verkäufer-NPCs der Turtle-AG.

Die NPCs verkaufen wichtige Ressourcen im Tausch gegen abgebauten und gebrannten Stein. Dazu gehören Maschinen, Kisten, Werkzeuge, Nahrung, Saatgut, Dünger und mehr. Um einen Gegenstand kaufen zu können, genügt es, die geforderte Anzahl an Stein im eigenen Inventar liegen zu haben.

Hinweis: Fassen Sie noch einmal für die Kinder zusammen, dass es sich wirklich lohnen kann, die vorhandenen Strukturen für die Bearbeitung der Aufgaben zu nutzen, dass sie aber auch selbst bestimmen können, wo und wie sie bauen, solange sie sich dabei auf das Lösen der Aufgaben konzentrieren und die anderen Teams nicht stören.

2.2 Vorstellung der Aufgabe und Verteilung der Teams auf der Insel

Sobald Sie mit der Vorstellung der einzelnen Areale im Teambereich fertig sind, teilen Sie das Arbeitsblatt „Auftrag 1: Die Filiale errichten!“ (siehe Materialanhang) aus. Das Arbeitsblatt ist in Form einer Mitarbeiteranweisung der Turtle-AG geschrieben und gibt den Spielerinnen und Spielern die Aufgabe, ein erstes Filialgebäude zu errichten. Außerdem wird eine Empfehlung für das Vorgehen beim Lösen der Aufgabe gegeben.

!! **Aufgabe:** Errichtet ein ansprechendes Filialgebäude in eurem Teambereich! Beachtet dabei die Arbeitsanweisungen der Turtle-AG auf dem Arbeitsblatt! Sammelt zusätzlich so viel Bruchstein und Stein in der Gemeinschaftskiste wie möglich!

Die Mindestkriterien für ein ordentliches Filialgebäude lauten wie folgt:

- Das Gebäude sollte nicht aus Bruchstein oder Erde gebaut werden! Als Baumaterial eignen sich gebrannter Stein, besser aber noch Ziegelstein, Holzbretter oder schönere Blöcke.
- An jeder Seite des Gebäudes befindet sich mindestens ein Fenster aus Glas.
- Das Filialgebäude sollte mindestens 5 Blöcke hoch sein. Mindestens 5 Betten sollten im Gebäude problemlos Platz finden.

Zum Ende der Unterrichtseinheit sollen die Kinder ihre Bauergebnisse präsentieren. Animieren Sie die Kinder, nicht nur die Mindestanforderungen für den Bau des Filialgebäudes zu erfüllen, sondern ästhetisch ansprechend zu bauen. Sie werden am Ende gemeinsam über die besten Umsetzungen abstimmen, sowohl was die bauliche Gestaltung als auch die technische Programmierung der Turtles angeht. Außerdem wird am Ende verglichen, wer den meisten Bruchstein und gebrannten Stein gesammelt hat. Weisen Sie die Kinder darauf hin, dass dabei nur die Materialien gezählt werden, die am Ende auch in den Gemeinschaftskisten liegen!

Hinweis: Ist die erste Gemeinschaftstruhe voll, können sich die Teams zusätzliche Truhen bei den Verkäufern im Turtle-AG Turm kaufen. Die neuen Kisten sollten sich im Filialgebäude oder zumindest in Nähe der ersten Kiste befinden. Die Kisten anderer Teams zu leeren, ist verboten!

Hinweis: In unserer Code your Life Minecraft-Welt ist die Modifikation „Not Enough Items“ vorinstalliert. Dies ermöglicht es den Spielerinnen und Spielern, nach Minecraft-Rezepten für das Craften von Blöcken zu suchen, während sie sich in ihrem Inventar befinden. Wenn die Kinder z.B. nicht wissen, wie sie eine Treppenstufe herstellen können, müssen sie einfach nur „Treppe“ in das untere schwarze Suchfeld eingeben. Auf der rechten Seite werden nun gefiltert alle Blöcke angezeigt, die das Wort „Treppe“ enthalten. Per Linksklick auf einen dieser Blöcke wird das entsprechende Rezept dazu angezeigt. Machen Sie die Kinder auf diese Möglichkeit der Rezeptrecherche aufmerksam!

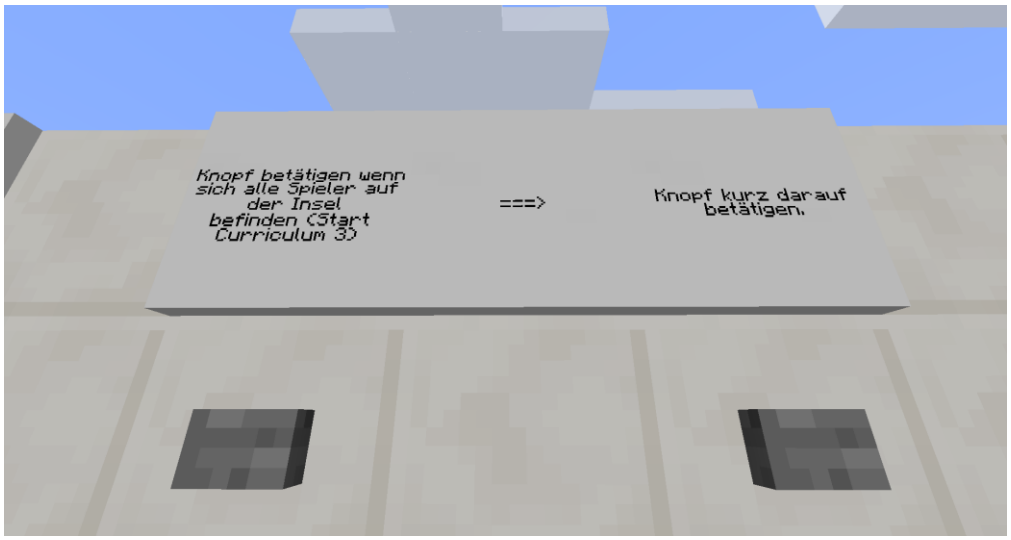
Verteilung der Teams auf der Insel

Nun geht es wieder ins Spiel. Bitten Sie die Kinder, sich wie gewohnt mit der Minecraft-Welt zu verbinden. Sollten die Teams noch nicht in der Empfangshalle stehen, in welcher sie gelandet sind, als sie sich das erste Mal mit der Welt verbinden haben (Curriculum 1), fordern Sie die Kinder auf, sich vom Turtle-AG Manager am Ende des Turtle Trainingcenters wieder in die Halle teleportieren zu lassen.

Hinweis: Sie können sich auch selbst in die Empfangshalle begeben und einzelne Kinder per Teleportbefehl (siehe Curriculum 1) direkt zu sich teleportieren, sollte es Probleme geben.

Anschließend müssen die Kinder in der Halle wieder mit dem für ihr Team zuständigen Sachbearbeiter sprechen und sich von diesem auf die Insel teleportieren lassen.

Sobald sich alle Teams auf der Insel befinden, müssen Sie noch zwei Einstellungen vornehmen und es kann losgehen. Teleportieren Sie sich über /tp -59 86 469 wieder in den speziellen Lehrerbereich und aktivieren Sie den Knopf mit der Überschrift „Knopf betätigen, wenn sich alle Spieler auf der Insel befinden (Start Curriculum 3)“.



Im Anschluss müssen Sie nur noch, wie auf den Schildern beschrieben, den Knopf rechts daneben rechtsklicken. Dadurch bekommen alle Kinder eine Turtle und einen Satz Werkzeuge in ihr Inventar gelegt.

Verteilung der Gewinnerpakete

Nun ist es auch an der Zeit, den Wettbewerbsgewinnern aus den ersten beiden Curricula ihren kleinen Bonus zukommen zu lassen. Neben der Mauer im Lehrerbereich befinden sich zwei Kisten, eine mit der Aufschrift „Gewinnerteam Curriculum 1“ und eine andere mit der Aufschrift „Gewinnerteam Curriculum 2“. Nehmen Sie die dort enthaltenen Einrichtungsgegenstände in Ihr Inventar und übergeben Sie diese im Spiel den entsprechenden Teams. Die Namen sollten Sie sich zuvor notiert haben.

Über den Knopf am äußeren rechten Rand der Mauer im Lehrerbereich können Sie sich bei wieder auf die Insel in den Turtle-AG Turm teleportieren, um von dort aus zu den einzelnen Teams zu gelangen.

Phase 3 | Arbeitsphase

Jetzt kann es endlich losgehen! Wir empfehlen den Lehrenden, während der Arbeitsphase im Spiel immer wieder bei den einzelnen Teams vorbeizuschauen und sich zu erkundigen, wie es vorangeht oder ob es Probleme gibt. Über den Beamer bekommen die Kinder immer wieder einen Einblick in den Fortschritt der anderen Teams.

Hinweis: Weisen Sie die Kinder darauf hin, dass sie per Rechtsklick zwar ohne Probleme Blöcke platzieren können, dass die meisten sich jedoch nur mit Turtles wieder einigermaßen schnell abbauen lassen. Daher ist es sinnvoll im Voraus zu planen, wo welcher Block wirklich hingesetzt werden soll.

Im Anschluss skizzieren wir nun einen möglichen Ansatz von vielen, die aktuelle Aufgabe zu bewältigen. Wichtig ist es zunächst, sich die übergeordnete Aufgabe in kleinere Teilaufgaben zu zerlegen. Das Arbeitsblatt zum Curriculum gibt mit seinen Anweisungen bereits eine gewisse Strukturierungshilfe.

Lösungsbeispiel

Wir überlegen zuerst, was alles getan werden muss, um mit dem Bau des Filialgebäudes zu beginnen.

- Wir wollen das Grundgerüst unseres Filialgebäudes aus Stein bauen.
- Stein bekommen wir am schnellsten, wenn wir unsere Turtles für uns in den Steinbruch schicken.
- Unsere Turtles wiederum müssen wir mit Energie aufladen und mit Werkzeugen ausrüsten.
- Maschinen können nur von Turtles bedient werden. Wenn wir Stein und andere Materialien brennen wollen, brauchen wir eine Turtle, die den Ofen bedient.
- Um während der Arbeit nicht zu verhungern, benötigen wir Nahrung. Diese kaufen wir vorerst bei einem Turtle-AG Verkäufer. Auch dafür benötigen wir aber Stein als Tauschwährung.
- Wir müssen das Gebäude schließlich bauen.

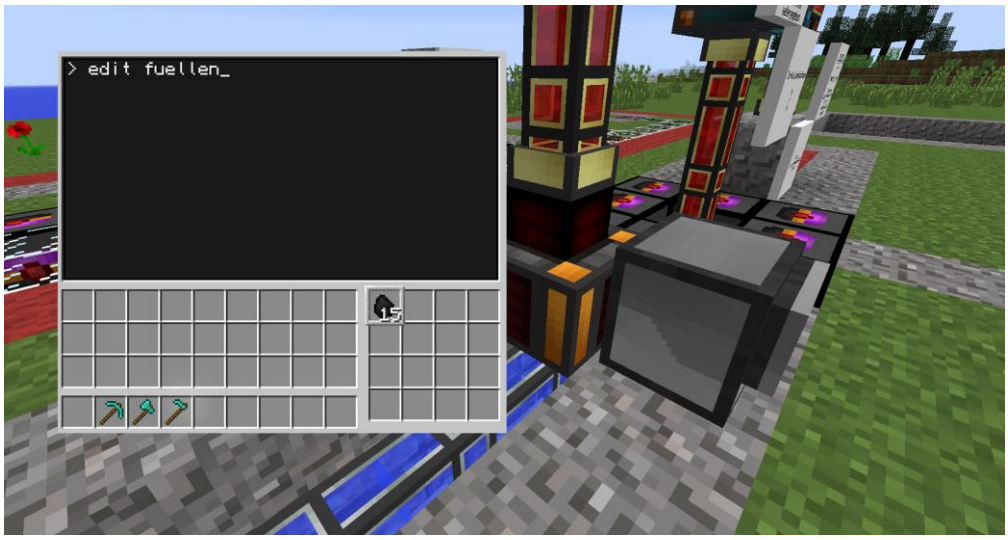
Hinweis: Sollten die Kinder bei der Bearbeitung ins Stocken geraten, kann es hilfreich sein, obige Überlegungen gemeinsam mit ihnen zu erarbeiten und schriftlich auf einem Blatt Papier oder an der Tafel festzuhalten.

Es gibt also einiges zu tun! Nun da wir unsere Aufgabe etwas strukturiert haben, sollten uns die Bearbeitung und eine gewisse Arbeitsteilung etwas leichter fallen.

Energieversorgung

Damit unsere Turtles mit ihrer Arbeit im Steinbruch beginnen können, gilt es als erstes für ein wenig Energie in den Ladestationen der Turtles zu sorgen. Ein Teammitglied platziert seine Turtle direkt vor der Dampfmaschine auf dem Startgelände und gibt ihr zunächst mit „label set Lucky“ einen Namen („Lucky“), damit später beim Abbau der Turtle keine Programme und

keine Energie verloren gehen können. Außerdem wird ein wenig Kohle aus der Gemeinschaftskiste in den oberen linken Inventarslot der Turtle gelegt.



Wir wollen, dass Lucky die Kohle aus ihrem Inventar in die Dampfmaschine füllt, daher öffnen wir über „edit fuellen“ die Bearbeitung des Programms „fuellen“. Hier geben wir „turtle.drop()“ ein, speichern ab und verlassen die Bearbeitung.

```
turtle.drop()
```

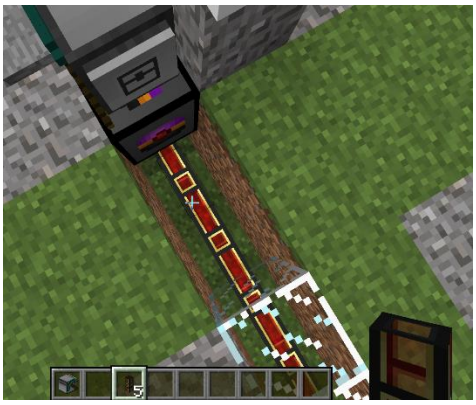
Anschließend lassen wir Lucky das Programm „fuellen“ ausführen und siehe da – die Dampfmaschine produziert Energie, die nun genutzt werden kann. Wir entschließen uns, Lucky zunächst an Ort und Stelle zu lassen und nun eine separate Mining-Turtle („Abbau-Turtle“) vorzubereiten. Kurzum wird eine neue Turtle auf die Ladestation neben der Dampfmaschine platziert. Wir geben ihr den Namen „Minnie“ und rüsten sie, wie wir es in der Schulung gelernt haben, über ein kleines Werkzeug-Programm mit einer unbeschädigten Diamantspitzhacke aus. Haben wir das gemacht, sollte Minnie bereits genug Energie gesammelt haben, um mit dem ersten Abbau beginnen zu können. Wir bauen Minnie ab und nehmen sie mit nach nebenan zum Steinbruch.

Hinweis: Die Kohle aus der Startausrüstung gibt bereits sehr viel Energie. Sollte den Kindern irgendwann dennoch keine Energie mehr zur Verfügung stehen, was eher unwahrscheinlich ist, da sie auch neue Kohle abbauen werden, können Sie ihnen im Notfall (!) einzelne Kohlestücke manuell über den Kreativmodus geben.

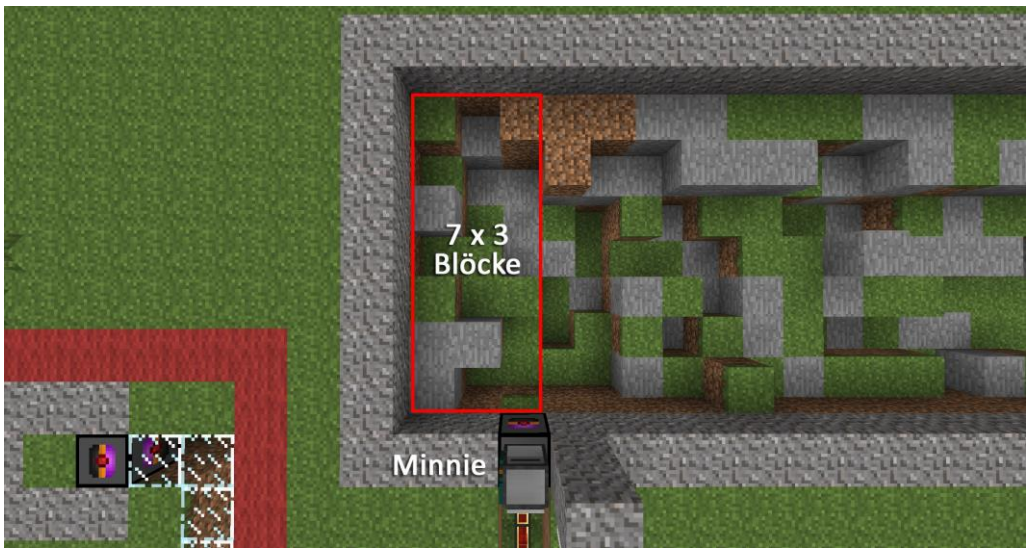
Steinabbau

Am Steinbruch angekommen, schauen wir uns um. Auch hier gibt es eine Ladestation direkt am äußeren Rand. Bei genauerem Hinsehen fällt jedoch auf, dass diese noch nicht mit dem

Rest des Energienetzes verbunden ist. Auch das Schild am Steinbruch weist darauf hin. Wir gehen an die Gemeinschaftskiste und nehmen uns ein Energietransportrohr (gehört zur Startausrüstung der Teams), das wir anschließend in die Lücke einbauen. Die Ladestation ist nun an das Energienetz angeschlossen und wir können unsere Turtle bei Bedarf auch direkt vor Ort aufladen, sofern genügend Energie vorhanden ist.



Auch wenn der Steinbruch noch ein wenig größer ist, entscheiden wir uns, zunächst einen kleineren Bereich abzubauen, da wir noch nicht genau wissen, wie weit wir mit unserem derzeitigen Energieniveau kommen. Um eine konstante Energieversorgung kümmern wir uns später.



Wir platzieren Minnie auf der Ladestation und planen ein Programm für einen Schacht, der zunächst die Ausmaße 7x3x15 haben soll. Später können wir die Turtle immer noch umprogrammieren oder an anderer Stelle neu ansetzen. Außerdem wäre es gut, wenn Minnie nach getaner Arbeit wieder an die Oberfläche in ihre Ausgangsposition zurückkehrt, damit wir auch an die wertvollen Ressourcen in ihrem Inventar kommen, denn so ein Schacht kann

schnell gefährlich in die Tiefe gehen. Wer tief fällt, verliert wohlmöglich sein Minecraft-Leben und alle Gegenstände aus dem eigenen Inventar!

Nun schreiben wir unser Programm.

- Über „edit schacht“ öffnen wir die Bearbeitung des neuen Programms „schacht“.
- Für die Umsetzung werden wir verschachtelte for-Schleifen nutzen. Unsere Turtle soll 15 Mal einen Bereich von 7x3 Blöcken abbauen, nachdem sie sich in den Steinbruch bewegt hat. Also beginnen wir wie folgt:

```
turtle.forward()

for i=1,15 do
```

- Anschließend folgen weitere for-Schleifen für die besagte 7x3 Fläche:

```
  for i=1,3 do

    for i=1,7 do
      turtle.digDown()
      turtle.forward()
    end

    for i=1,7 do
      turtle.back()
    end
```

- Somit sorgen wir dafür, dass Minnie dreimal einen Streifen von 7 Blöcken abbaut (sofern, Blöcke vorhanden sind) und sich wieder an den Anfang des Streifens begibt. Damit sie sich dabei jedes Mal einen Block nach links bewegt und zum Abbau eines neuen Streifens ansetzt, fügen wir noch folgenden Code unten an und schließen die Dreier-Schleife mit einem end:

```
    turtle.turnLeft()
    turtle.forward()
    turtle.turnRight()

  end
```

- Nun soll sich Minnie wieder an ihre Ausgangsposition begeben, um von dort aus die nächste Ebene auszuheben.

```
  turtle.turnRight()

  if not turtle.back() then
    for i=1,2 do
      turtle.forward()
    end
    else
    for i=1,4 do
       turtle.forward()
    end
  end
```

- Mit dieser if-then-Bedingung stellen wir sicher, dass die Turtle wieder auf der richtigen Ausgangsposition landet, egal, ob sie nach dem Abbau einer Ebene auf ein Hindernis gestoßen ist oder nicht.

- Schließlich muss Minnie nur noch gedreht werden und sich einen Block nach unten bewegen, um mit dem Abbau einer neuen Ebene beginnen zu können. Hier befindet sich auch das Ende der ersten 15er-Schleife, weswegen wir mit einem „end“ die Schleife schließen.

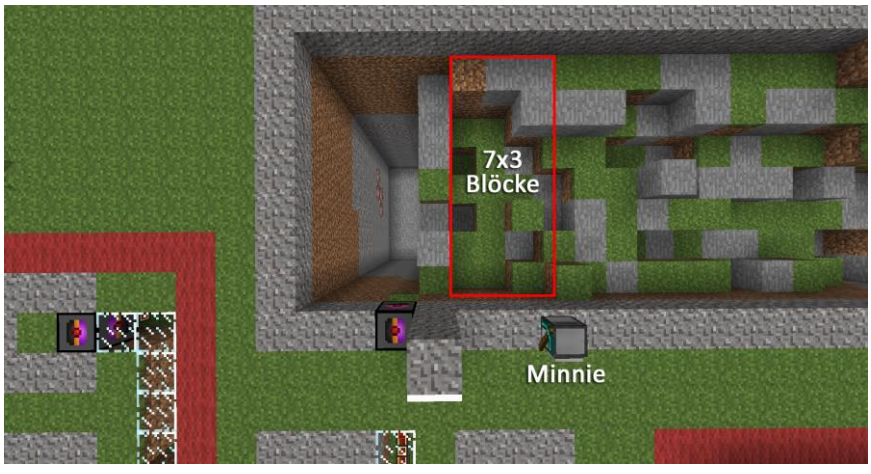
```
  turtle.turnLeft()
  turtle.down()

  end


for i=1,15 do
  turtle.up()
end

turtle.back()
```

- Wir zuvor gewünscht, wollen wir nun noch erreichen, dass Minnie wieder an die Oberfläche kommt und sich in ihre Ausgangsposition begibt. Damit wir leicht an die gesammelten Materialien gelangen können, nutzen wir eine weitere Schleife und ein turtle.back(), um unseren fleißigen Helfer zurückzubringen (siehe oben).

- Wir speichern das Programm ab und führen es aus. Nun heißt es warten, bis Minnie von ihrer Abbau-Tour zurück ist.

- Ein Blick in das Inventar der Turtle zeigt, dass Minnie tatsächlich einiges an Bruchstein abgebaut hat. Zusätzlich ist sie hier und da noch auf andere Materialien gestoßen.

- Super, unser Programm scheint zu funktionieren! Wir nehmen uns die Ressourcen in das eigene Inventar, laden Minnie ausreichend über die Ladestation auf und platzieren sie ein wenig weiter rechts erneut am Rand des Steinbruchs, sodass sie über unser Programm „schacht“ den nächsten Bereich ausheben kann.



Hinweis: Wir könnten das Programm an dieser Stelle auch so editieren, dass Minnie bei der Ausführung einen größeren Bereich aushebt, also z.B. einen Kubus von 7x5x20 Blöcken. Alles was wir dafür ändern müssten, wären die Zahlenwerte in den Schleifenköpfen. Aber dies sei jedem selbst überlassen. Schließlich gibt es auch völlig andere Wege, den Steinabbau umzusetzen. Wir entscheiden uns dafür, den derzeitigen Code beizubehalten. Hat Minnie nicht zu große Areale abzubauen, geht ihr auch weniger schnell die Energie aus!

Weiterverarbeitung - Stein

Sobald Minnie uns mit genügend Bruchstein versorgt hat, nehmen wir diesen und befördern ihn, mit Hilfe einer Turtle, in unseren Redstone-Ofen. Wie praktisch, dass wir für Lucky schon ein Programm geschrieben haben („fuellen“), das auch hier wieder zum Einsatz kommen kann!

Wir bauen Lucky ab und setzen ihn vor den Ofen (blauer Anschluss). In den oberen linken Inventarslot der Turtle legen wir einen Stapel Bruchstein (64 Stück). Aber Moment: Ohne Energie kann der Ofen nicht arbeiten. Schauen wir auf die Energieleitungen, fällt auf, dass auch hier ein Rohr fehlt, um Ofen und Energieproduktion miteinander zu verbinden. Auch hier besorgen wir uns schnell ein Energietransportrohr und beheben das Problem.

Jetzt müssen wir nur noch unser „füllen“-Programm auf Lucky ausführen. Mit der zugeführten Energie wird nun Bruchstein zu (gebranntem) Stein umgewandelt. Letzteren bekommen wir aber nur aus dem Ofen gezogen, indem wir wieder eine Turtle nutzen.

Wir benennen eine neue Turtle „Daisy“ und platzieren sie genau auf der anderen Seite des Ofens (oranger Anschluss). Äquivalent zum Füllprogramm, brauchen wir nun ein Programm, das den verarbeiteten Stein aus dem Ofen herauszieht, also öffnen wir über „edit stein“ eine neue Bearbeitung und halten dort den Befehl „turtle.suck()“ fest.

Wir speichern, führen das Programm aus und Daisy erhält gebrannten Stein. Da der Ofen nicht alles auf einmal brennen kann, erhalten wir nicht immer sofort den gesamten Stein. Sollte noch etwas fehlen (wir müssen genau so viel Blöcke bekommen, wie wir hineingegeben haben), führen wir das Programm etwas später einfach nochmals aus.

Hinweis: Da Lucky und Daisy sich in diesem Beispiel nicht von der Stelle bewegen und nur mit dem Block direkt vor sich interagieren, benötigen sie zum Ausführen ihrer Programme keine Energie! Das heißt, wir können auf ein vorheriges Aufladen der beiden in diesem Fall verzichten.

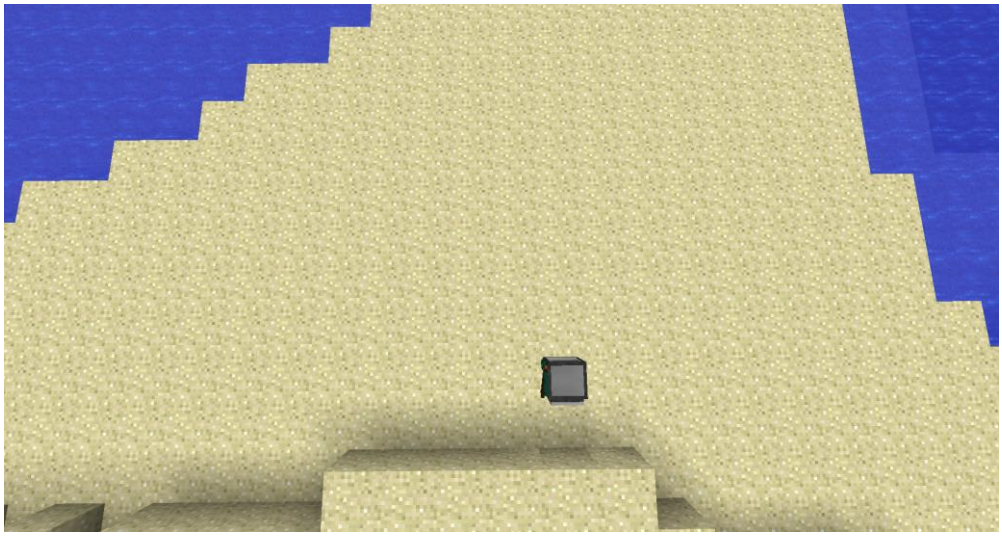
Weiterverarbeitung - Glas

Laut Anweisung der Turtle-AG benötigen wir an jeder Seite unseres Filialgebäudes später mindestens ein Glasfenster. Doch woher bekommen wir Glas? Über die Suchfunktion in

unserem Inventar („Not enough Items“) geben wir „Glas“ ein und erfahren bei einem Klick auf den Glasblock rechts, dass wir Glas erhalten, wenn wir Sand im Ofen schmelzen.

Alles, was uns also fehlt, ist der Sand. Natürlich könnten wir eine extra Turtle programmieren, die für uns am Strand Sand abbaut. Wir entscheiden uns aber dazu, Minnie einen kleinen Strandausflug zu spendieren, schließlich besitzt sie bereits ein passendes Programm („schacht“)!

Sobald Minnie von ihrer letzten Abbau-Tour wieder an die Oberfläche zurückgekehrt ist, entnehmen wir ihr alle Gegenstände, bauen die Turtle ab und laden sie vorsichtshalber noch einmal auf. Da wir uns auf einer Insel befinden, ist auch der gesuchte Strand nicht weit. Wir wählen ein passendes Stück Strand aus und platzieren Minnie.

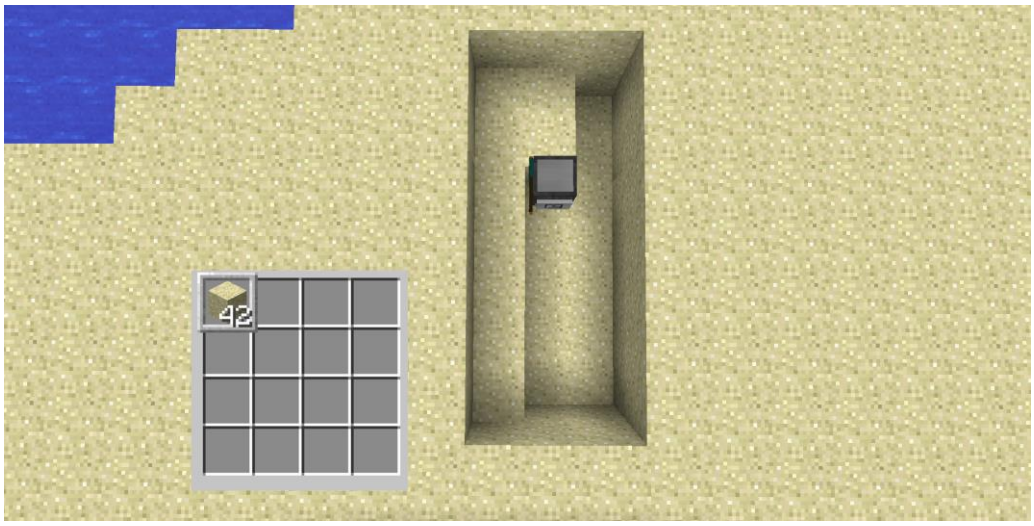


Da wir längst nicht so viel Sand benötigen, wie Stein, ist es auch nicht notwendig 15 Blöcke in die Tiefe zu gehen. Wir modifizieren das Programm „schacht“ für den Moment so, dass nur zwei Ebenen Sand ausgehoben werden. Das heißt, wir ändern die Werte in der for-Schleife am Anfang und die der for-Schleife am Ende von i=1,15 in i=1,2 :

```
turtle.forward()
for i=1,2 do
  for i=1,3 do
    for i=1,7 do
      turtle.digDown()
      turtle.forward()
    end
```

```
for i=1,2 do
  turtle.up()
end
turtle.back()
```

Wir speichern, beenden die Bearbeitung und lassen Minnie das modifizierte Programm ausführen. Nach kurzer Zeit dürfen wir uns über eine ordentliche Menge Sand freuen.

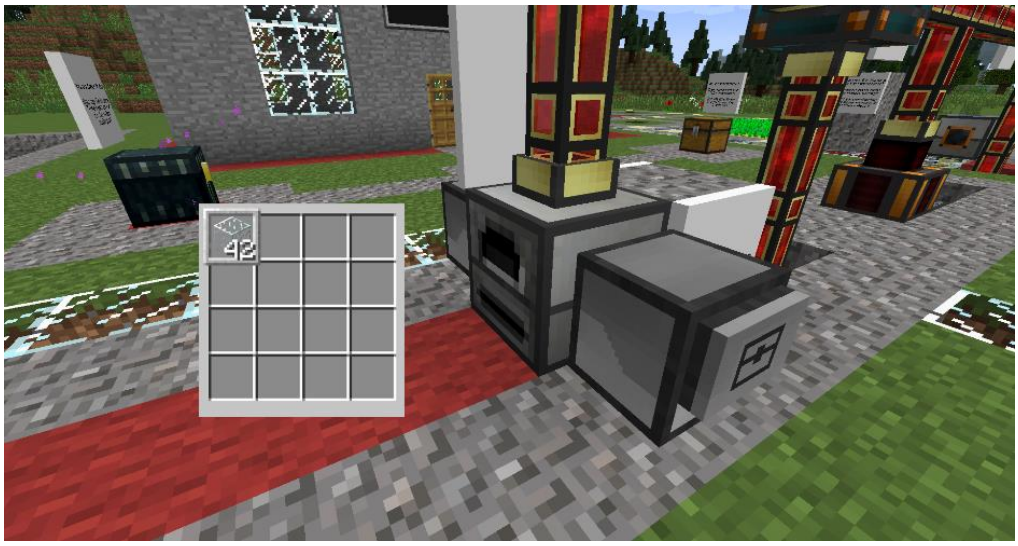


Wir nehmen uns den Sand aus Minnies Inventar, bringen sie wieder zurück zum Steinbruch und passen das Abbau-Programm je nach Belieben wieder an. Minnie soll nun einfach weiter Bruchstein abbauen.

Wir gehen zu Lucky und legen den Sand in seinen oberen linken Inventarslot. Falls sich darin noch Bruchstein befindet, tauschen wir diesen einfach kurz mit dem Sand. Über „fuellen“ lassen wir Lucky den Sand in den Redstone-Ofen geben.

Achtung: Dies ist nur möglich, wenn sich keine anderen Blöcke mehr im Ofen befinden! Sollten sich noch andere Materialien in der Verarbeitung befinden, müssen wir letztere erst abschließen und die verarbeiteten Blöcke mit Daisy aus dem Ofen ziehen.

Sobald sich der Sand erst einmal im Ofen befindet, müssen wir nur einen Moment warten und können im Anschluss, das fertige Glas mit Daisy und ihrem Programm „stein“ aus dem Redstone-Ofen extrahieren.



Großartig, jetzt kann es endlich an den Bau des Filialgebäudes gehen! Mit Glas und Stein im Gepäck geht es nach nebenan zum Filialbauplatz!

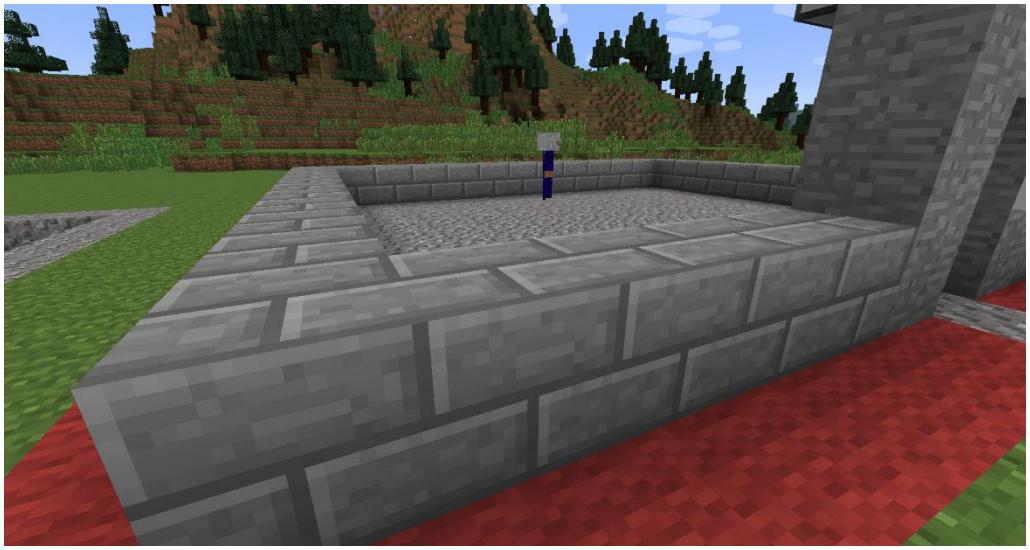
Bau

Wir rufen uns noch einmal die Mindestanforderungen der Turtle-AG für das Filialgebäude in Erinnerung:

- Als Baumaterial keinen Bruchstein und keine Erde verwenden.
- An jeder Seite mindestens ein Glasfenster einbauen.
- Fünf Betten müssen problemlos Platz finden.
- Das Gebäude muss mindestens fünf Blöcke hoch sein.

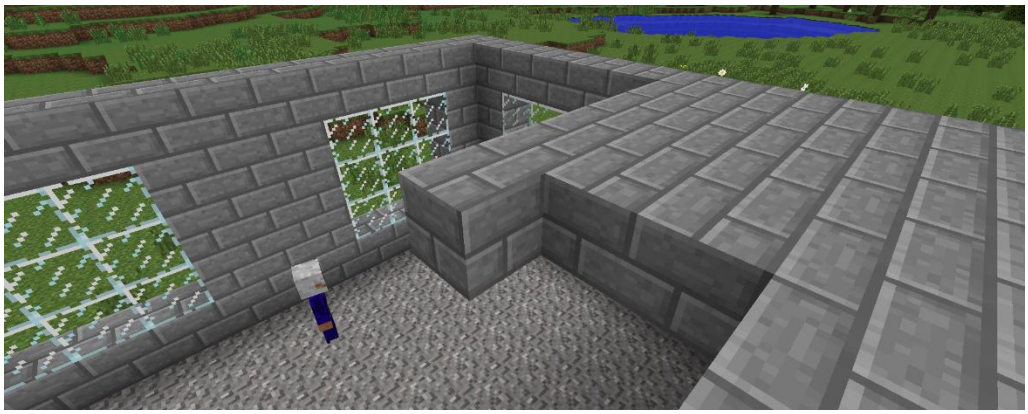
Normalen Stein haben wir genug. Aber wir wollen noch ein wenig hübscher bauen und entscheiden uns dazu, Steinziegel herzustellen. Über die Suchfunktion im Inventar (Suchwort: „Steinziegel“) erfahren wir, dass wir einfach nur vier Steine in einem 2x2 Feld auslegen müssen, um vier Steinziegel zu erhalten. Das können wir auch in unserem eigenen Inventar tun und benötigen nicht zwingend eine Crafty Turtle.

Wir craften einige Steinziegel und fangen an zu bauen. Per Rechtsklick platzieren wir unsere Blöcke. Mit den ersten Steinziegeln legen wir einen Grundriss fest.



Anschließend legen wir fest, wo unsere Fenster hinkommen und setzen Glas an die entsprechenden Stellen. Da wir genug Glas haben, können wir sogar mehr als vier Fenster bauen.

Nach und nach füllen wir den Grundriss mit entsprechenden Materialien auf. Wir müssen dabei allerdings aufpassen, dass wir nicht herunterfallen, denn ohne uns kleine Hilfstreppen zu bauen, kommen wir anschließend nicht wieder so leicht nach oben! Vorsichtig platzieren wir Steinziegel für Steinziegel. Schließlich fehlt nur noch das Dach.



Schon bald können wir uns über das fertige Filialgebäude freuen. Um eine Tür kümmern wir uns im nächsten Curriculum. Innen platzieren wir noch unsere fünf Betten aus der Gemeinschaftstruhe. Platz genug haben wir ja. Das Filialgebäude steht und wir haben unsere Hauptaufgabe erfüllt!

Bonusziele

Auf dem Arbeitsblatt „Auftrag 1: Eine Filiale errichten!“ gibt es noch einige Bonusziele.

Eines davon ist es, den Monitor am Filialbauplatz den Namen der Filiale anzeigen zu lassen. Wie das geht, verrät der Code Magier auf dem Filialbauplatz.

Auf dem Computer über dem Eingang befindet sich bereits ein passendes Beispielprogramm, was wir uns über „edit beispiel“ ansehen können. Hinter mon.write müssen wir nur den Text in den Klammern und Anführungszeichen in den gewünschten Anzeigetext ändern, in unserem Beispiel hier wählen wir „Turtle-AG Team Rot“.

```
mon = peripheral.wrap("back")

mon.clear()
mon.setCursorPos(7,3)
mon.setTextScale(1)
mon.write("Turtle-AG Team Rot")
```

Die Teams können hier natürlich ihre Wunschnamen eintragen. Sobald wir das Programm gespeichert haben, führen wir nur noch „beispiel“ aus und erhalten am Monitor an der Frontseite des Gebäudes die gewünschte Anzeige.

Weitere Bonusziele sind es, möglichst hochwertige Materialien zu verwenden und ein Schrägdach für die Filiale zu bauen. Natürlich können die Kinder ihr Filialgebäude darüber hinaus noch weiter verschönern. Der Fantasie sind keine Grenzen gesetzt.

Etwa 20 Minuten vor Ende der Unterrichtseinheit ist es Zeit, gemeinsam zu schauen, welchen Fortschritt die Kinder gemacht haben. Wer hat den meisten Stein gesammelt? Welches Gebäude ist das schönste? Wie wurden die Turtles programmiert?

Phase 4 – Nachbereitung

Nun sollen die Teams ihre Ergebnisse natürlich noch präsentieren. Geben Sie jedem Team etwa drei Minuten, um seine Filiale und seine Turtles bei Ihnen am Rechner und über den Beamer vorzustellen. Die anderen Teams können währenddessen Fragen stellen.

Hinweis: Bitten Sie die Kinder, nicht mehr auf ihren Rechnern weiterzuspielen, während die anderen Teams präsentieren. Zum einen ist es nicht fair, noch weiter Ressourcen zu sammeln, während die anderen keine Gelegenheit mehr dazu haben, zum anderen sollen die Präsentierenden auch genügend Aufmerksamkeit für ihre Arbeit bekommen.

Gehen Sie mit jedem Team die Checkliste auf dem Arbeitsblatt „Auftrag 1: Eine Filiale errichten!“ durch.

- ¬ Wurden die Mindestkriterien für den Bau eingehalten?
- ¬ Wurden Bonusziele bearbeitet?
- ¬ Wie wurden die Turtles eingesetzt?
- ¬ Wieviel Stein und Bruchstein hat das Team in der Gemeinschaftskiste gesammelt? (Achtung es zählen nur die Materialien in der Kiste! Halten Sie die Ergebnisse der Teams am besten in einer Übersicht an der Tafel fest.)

Vergleichen Sie abschließend und diskutieren Sie gemeinsam mit den Kindern, welche Umsetzungen besonders gelungen scheinen und warum. Sicherlich kann sich der ein oder andere hier noch Anregungen für das nächste Curriculum holen.

Phase 5 – Ausblick

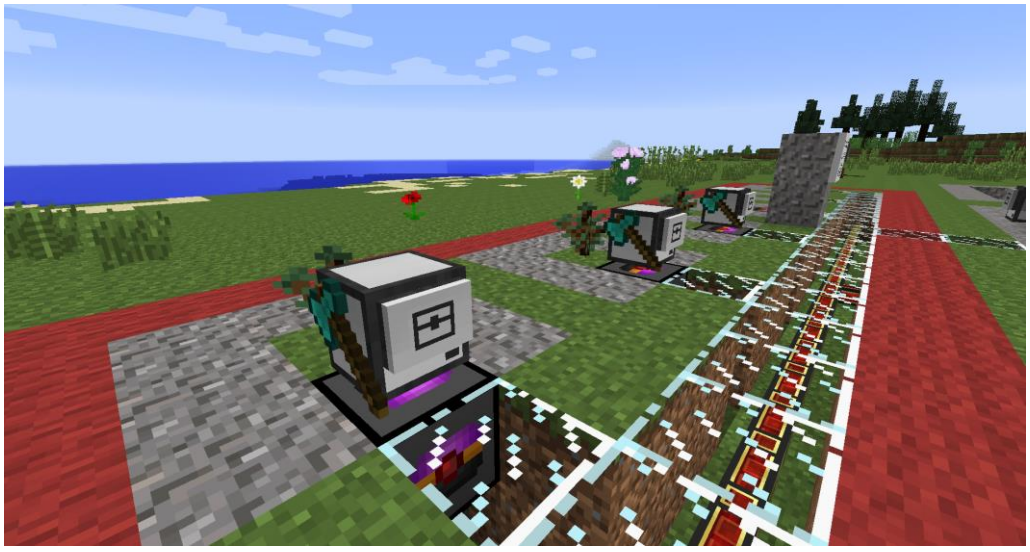
In dieser Unterrichtseinheit wurde den Teams zu Beginn noch etwas Kohle zur Verfügung gestellt, um eine erste Energieversorgung gewährleisten zu können und die Maschinen und Turtles am Laufen zu halten.

Im nächsten Curriculum wird es die Hauptaufgabe sein, selbst für ausreichend Nachschub an Energie zu sorgen und zwar über das regenerative System einer kleinen automatisierten Forstwirtschaft.

CURRICULUM 4 – Regenerative Energieversorgung

Ein erstes Filialgebäude steht und der Steinabbau läuft. Doch die Energiereserve geht zur Neige. Die Teams müssen nun dafür sorgen, dass genügend Brennstoff nachkommt. Dampfmaschinen funktionieren nicht nur mit normaler, sondern auch mit Holzkohle. Zeit, mit den Turtles eine kleine Forstwirtschaft aufzubauen!

In dieser Unterrichtseinheit sollen die Kinder Baumsetzlinge pflanzen, wachsen lassen, die ausgewachsenen Bäume fällen und das so erhaltene Holz im Redstone-Ofen zu Holzkohle verarbeiten. Letztere kann dann in der Dampfmaschine wieder zu Energie umgewandelt werden. Ohne die Hilfe der Turtles lässt sich diese Aufgabe kaum realisieren. Versuchen die Kinder nämlich, die Bäume selbst zu fällen, erhalten sie anstelle von verwertbaren Holzblöcken lediglich ein paar Stöcke. Eine große Unterstützung bei der Automatisierung des Forstbetriebs kann der Turtle-Befehl turtle.inspect() sein, über den der Code Magier einiges zu berichten weiß. Natürlich gilt es auch weiterhin, möglichst viel Stein zu sammeln!



Überblick

Kompetenzen

Zeitaufwand	**90 Minuten**
Technik	Laptops / Standrechner, je Gerät eine Mouse, LAN/WLAN, Beamer
Methoden	Gruppenarbeit, Simulation, Frage und Antwort, Arbeitsblatt
Vorkenntnisse	Programmieren mit Logo, Turtle Trainingcenter 1 & 2, Eine Filiale entsteht

Die Schülerinnen und Schüler …

- lernen Endlosschleifen kennen.
- machen Bekanntschaft mit dem Befehl turtle.inspect().
- errichten einen kleinen Forstbetrieb.
- setzen eine regenerative Energieversorgung um.
- müssen ihr Programmierwissen problemlösungsorientiert anwenden.

Ablauf

Phase	Aufgabe	Methode	Zeit
Reflexion	Wie steht es um die Filialen der Teams? Was braucht die Filiale, um weiter zu gedeihen?	F & A	5´
Vorbereitung	Vorstellung der Aufgabe: Stabile Energieversorgung		5´
Arbeitsphase	Freiarbeit: Bearbeitung der Challenge	Gruppenarbeit, Simulation und Arbeitsblatt	55´
Nachbereitung	Präsentation der Ergebnisse	Präsentation, Simulation, Arbeitsblatt	20‘
Ausblick	Nachhaltige Nahrungsproduktion		5‘

Unterrichtsverlauf

Vorbemerkung

In diesem Curriculum ist es wichtig, die Kinder immer wieder auf den Code Magier zu verweisen, sollten sich Probleme bei der Bearbeitung der Aufgabe, insbesondere was die Automatisierung des Betriebs angeht, ergeben. Der Code Magier hält alle nötigen Informationen zum wichtigen turtle.inspect()-Befehl bereit, der genutzt werden kann, um zu überprüfen, ob beispielsweise ein Baum gewachsen ist, oder nicht. Zudem erklärt er die Umsetzung von Endlosschleifen, die es gestatteten, ein Programm ständig ausführen zu lassen, sodass eine manuelle Aktivierung nicht mehr notwendig ist.

Phase 1 | Reflexion

Fassen Sie noch einmal den Inhalt des letzten Curriculums zusammen. Überlegen Sie gemeinsam mit den Kindern:

- Was haben wir das letzte Mal auf der Turtle Insel gemacht? (Antwort: mit den Turtles Stein abgebaut und eine erste Filiale errichtet)
- Wo kam die Energie für die Turtles und die Maschinen das letzte Mal her? (Antwort: aus dem Vorrat in der Gemeinschaftskiste / Turtles haben Kohle gefunden und abgebaut)

Die Kinder sollen darüber nachdenken, welche Energiequellen noch erschlossen werden könnten. Weisen Sie darauf hin, dass die Dampfmaschinen in Minecraft auch mit Holzkohle Energie produzieren können.

Hinweis: Sofern gewünscht, können Sie hier sogar zu einer kleinen Diskussion über regenerative Energie in der realen Welt anregen. Wo liegen die Vor- und Nachteile einer solchen Energieversorgung?

Phase 2 | Vorbereitung

Teilen Sie das Arbeitsblatt „Auftrag 2: Stabile Energieversorgung“ aus und gehen Sie mit den Kindern die einzelnen Punkte der Aufgabe durch.

Vorstellung der Aufgabe: Stabile Energieversorgung

Die Turtle-AG wünscht in ihrem neuen Auftrag die Umsetzung einer automatisierten Brennstoffproduktion aus Holz. So sollen sich die Filialen nun weitestgehend selbst mit Energie versorgen, indem sie die Turtles einen kleinen Forstbetrieb führen und die Dampfmaschinen mit Holzkohle versorgen lassen. Überschüssige Energie wird in einer Energiezelle gespeichert.

Aufgabe: Pflanzt und fällt automatisiert Bäume mit den Turtles und verarbeitet die gewonnenen Holzstämme zu Holzkohle weiter! Holzkohle soll dann in der Dampfmaschine zur Stromerzeugung genutzt werden.

Am Ende der Unterrichtseinheit, wird erneut verglichen: Wer hat die meiste Energie gespeichert (Rechtsklick auf die Energiezelle)? Wieviel Bruchstein, Stein und Holzkohle liegen in der Gemeinschaftskiste? Wie wurden die Turtles programmiert? Außerdem gibt es wieder einige Bonusziele.

Sehr wichtig ist das Gespräch mit dem Code Magier! Wer diesen nicht konsultiert, wird es nicht einfach haben, seinen Betrieb zu automatisieren.

Hinweis: Geben Sie den Kindern den Hinweis, dass sie sich gerne auch Notizen zu dem machen können, was der Code Magier ihnen mitteilt, z.B. auf der Rückseite des Arbeitsblattes.

Die Kinder sollten sich wie gewohnt einloggen und mit der Code your Life Minecraft-Welt verbinden, nachdem Sie den Server gestartet haben. Jeder steigt wieder dort ein, wo er oder sie das Spiel zuletzt verlassen hat.

Phase 3 | Arbeitsphase

Im Folgenden werden wir wieder einen möglichen Lösungsweg skizzieren. Das Arbeitsblatt hilft mit seinen Anweisungen bereits bei einer Vorstrukturierung der Gesamtaufgabe.

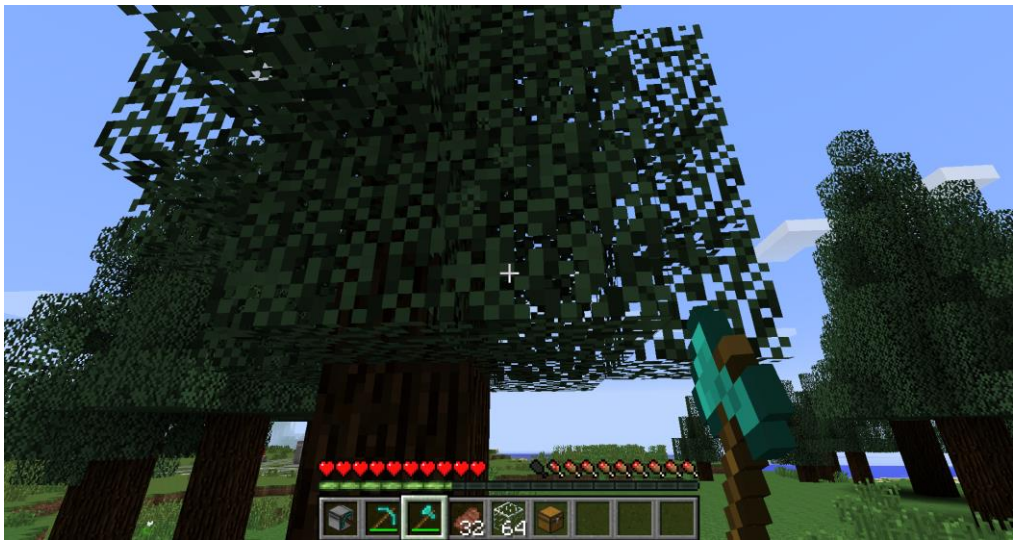
Lösungsbeispiel

Wir überlegen zunächst, was wir alles vorbereiten müssen, um mit der Programmierung unserer Holzfäller-Turtles beginnen zu können. Wir müssen:

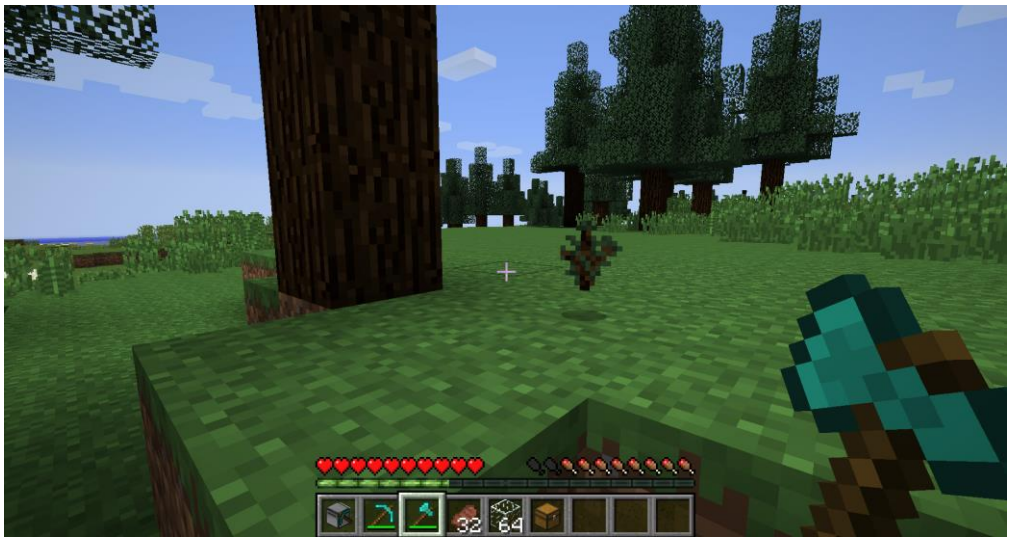
- ausreichend Baumsetzlinge sammeln. Für den Beginn sollten 32 Fichtenholzsetzlinge genügen.
- Ladestationen des Forstbereiches an das Energienetz anschließen, damit wir unsere Turtles nicht immer manuell aufladen müssen.
- mindestens eine Turtle mit einer Axt ausrüsten, die den Holzfäller-Job übernehmen kann.
- mit dem Code Magier über Variablen, Endlosschleifen & Co. sprechen.

Baumsetzlinge

Ohne Setzlinge, können wir nichts pflanzen. Mit einer Axt begeben wir uns ins Umland und halten nach Fichtenbäumen Ausschau. Natürlich könnten wir auch andere Baumsorten nutzen, denn für den Verarbeitungsprozess in Holzkohle gibt es keinen Unterschied zwischen den einzelnen Holzsorten.

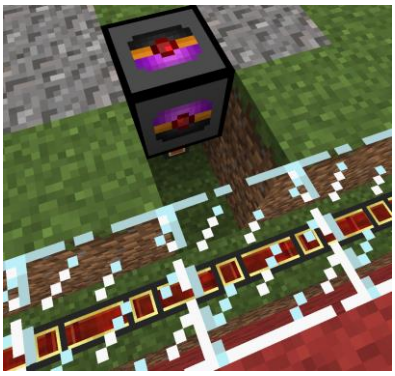


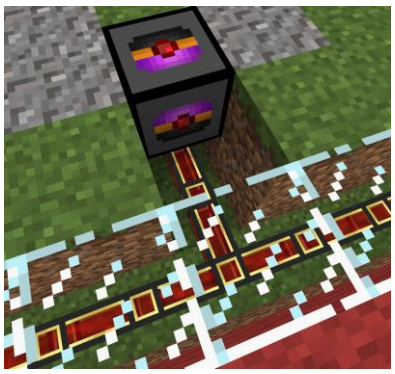
Sobald wir passende Bäume gefunden haben, entfernen wir mit der Axt das Laub. Oft werden wir hier gar nichts erhalten, aber ab und zu kommt dadurch doch ein Fichtenholzsetzling zum Vorschein, den wir unserem Inventar hinzufügen können. Auf diese Art und Weise sammeln wir zunächst 32 Setzlinge. Soll es schnell gehen und haben wir viel gebrannten Stein, können wir uns aber auch bei einem der Verkäufer-NPCs im Turtle-AG Turm Setzlinge kaufen.



Ladestationen im Forstbereich

Im zu Beginn vordefinierten Forstbereich der Filiale befinden sich bereits einige Ladestationen, über welche unsere zukünftigen Holzfäller-Turtles Energie beziehen können. Die Stationen müssen jedoch erst wieder mit dem Hauptstromnetz verbunden werden. Wir beschaffen uns ein Energietransportrohr und schließen zunächst nur eine Station an, denn wir wollen unseren Code erst einmal nur prototypisch an einer einzelnen Turtle testen und solange den Strom für die anderen Ladestationen sparen.

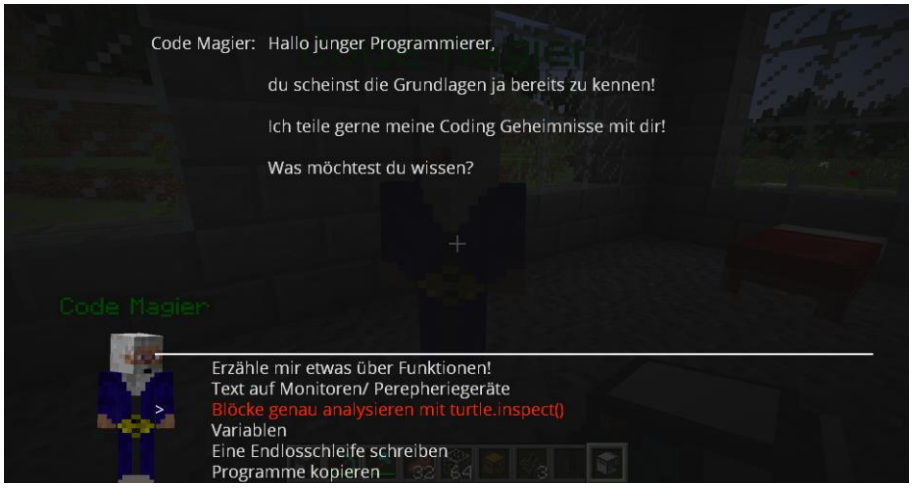




Turtle vorbereiten und mit dem Code Magier sprechen

Wir platzieren unsere Turtle auf der Ladestation, die wir eben an das Energienetz angeschlossen haben, geben ihr einen Namen und rüsten sie wie gewohnt aus, dieses Mal jedoch mit einer Diamantaxt. Die gesammelten Setzlinge legen wir der Turtle in den oberen linken Inventarslot.

Beim Code Magier auf dem Filialgelände erkundigen wir uns über Variablen, Endlosschleifen und den für unsere Aufgabe so wichtigen turtle.inspect()-Befehl.



Hinweis: Sollten die Kinder keine unbeschädigten Äxte mehr im Inventar haben, müssen sie sich mit Stein eine neue Axt bei einem der NPC-Verkäufer im Turtle-AG Turm kaufen.

Die Turtle programmieren

Wie genau setzen wir das Wissen des Code Magiers nun in die Tat um? Schauen wir uns eine von vielen Möglichkeiten an.

Hinweis: Die Lehrkraft kann die folgend vorgestellte Turtle mit dem Namen „Holzfaeller“ bei einem der Turtle-AG Supervisors im Turtle-AG Turm erhalten. Das enthaltene Programm „faellen“ lässt sich editieren und bei Bedarf vorführen. Lediglich die benötigten Ressourcen müssen der Turtle noch in die passenden Inventarslots gelegt werden.

Unsere Turtle mit dem Namen „Holzfäller“ haben wir bereits ausgerüstet und auf der Ladestation platziert. Setzlinge liegen ebenfalls bereits in ihrem Inventar.

- Über „edit faellen“ öffnen wir die Bearbeitung unseres neuen Programms.
- Da wir unser Programm nicht nur einmal, sondern endlos ausführen lassen wollen, beginnen wir mit der Erzeugung einer Endlosschleife, also „while true do“. So hat es uns der Code Magier erklärt. Später dürfen wir nicht vergessen, die Schleife noch mit einem end zu schließen.

```
while true do
```

- Sobald die Turtle vor sich einen Holzblock erkennt, der Baum also ausgewachsen ist, soll sie mit dem Abholzen beginnen. Dies setzt jedoch voraus, dass die Turtle den Holzblock auch als Holzblock identifizieren kann. Es könnte sich z.B. auch ein Baumsetzling vor der Turtle befinden. Diesen wollen wir aber wachsen lassen und nicht abbauen. Mit turtle.detect() können wir lediglich feststellen, ob sich vor der Turtle ein Block befindet oder nicht. Um welchen Block es sich handelt, lässt sich so aber nicht entschlüsseln.
- Der Code Magier empfiehlt für einen solchen Fall den Befehl turtle.inspect(). Das Kommando turtle.inspect() gibt zwei Werte zurück: Der erste Wert gibt mit „true“ (wahr) oder „false“ (falsch) an, ob überhaupt ein Block inspiziert werden konnte. Im zweiten Wert wird eine Art Datenfeld übergeben, in dem je nach Block dessen verschiedene Eigenschaften gespeichert sind, wie z.B. auch der Name des Blocks, über welchen sich eine eindeutige Identifizierung vornehmen lässt.

- Genau diesen Umstand können wir nutzen, um unsere Holzfäller-Turtle zwischen einem ausgewachsenen Baum und einem Setzling unterscheiden zu lassen. Damit wir vernünftig mit den Werten aus turtle.inspect() arbeiten können, lesen wir die Werte zunächst wie folgt in die Variablen „erfolg“ und „daten“ ein:

```
erfolg, daten = turtle.inspect()
```

Hinweis: Variablen sollten den Kindern bereits aus dem Curriculum „Programmieren mit Logo“ bekannt sein. Sie dienen als eine Art „Gefäß“ für verschiedenste Werte, wie einzelne Zahlen, Text, Datenfelder oder Wahrheitswerte. Definieren lassen sie sich in Lua wie oben gezeigt: Auf der linken Seite stehen ein oder mehrere Variablennamen und ein „=“ sorgt für die Zuweisung der auf der rechten Seite stehen die Werte an diese Variablen.

```
if erfolg then
  print(daten.name)

  if daten.name == "minecraft:log" then
    turtle.dig()
    turtle.digUp()
    turtle.up()
```

- Anschließend prüfen wir, ob die Turtle einen Block vor sich identifizieren kann, also ob erfolg auch true (wahr) ist. Wenn dem so ist, soll die Turtle über den print-Befehl den internen Minecraft-Namen des Blocks in ihrer Konsole ausgeben, damit wir sehen, um welchen Block es sich handelt. Über das .name hinter daten wird sozusagen das Namensfeld unserer Variable daten angesprochen

- Eine weitere if-then-Bedingung prüft anschließend, ob dieser Name „minecraft:log“ entspricht. Das „minecraft:log“ steht dabei für den Namen eines Holzblockes in Minecraft. Das wissen wir, da der Code Magier die internen Namen der wichtigsten Blöcke für uns aufgelistet hat.

Hinweis: Wichtig ist hierbei, das ein doppeltes „=“ hinter daten.name gesetzt wird, da hier kein Wert neu zugeteilt wird (einfaches „=“), sondern nur überprüft wird, ob der Wert auf der rechten Seite („minecraft:log“) dem entspricht, was momentan in daten.name gespeichert ist. Diese Abfrage wird Lua durch ein doppeltes „=“ realisiert.

- Sollten die Werte übereinstimmen, soll unsere Holzfäller-Turtle anschließend den Block vor sich abbauen, den Block über sich abbauen (sollte sich z.B. Laub im Weg befinden) und sich einen Block nach oben bewegen.

```
elseif daten.name == "minecraft:sapling" then
  turtle.select(13)
  turtle.place()
  turtle.select(1)
```

- Was aber soll die Turtle tun, wenn sie keinen Holzblock vor sich identifiziert, sondern einen Setzling? Wäre es nicht gut, wenn sie diesen, wenn möglich, gleich düngt, damit der Baum schneller wächst? Für diesen Fall formulieren wir mit einem „elseif“ eine alternative Vorgehensweise. Sollte die Turtle vor sich einen Block mit dem Namen „mineacraft:sapling“ (entspricht dem Namen eines Setzling-Blockes in Minecraft) identifizieren, soll sie in unserem Beispiel mittels „turtle.select(13)“ zuerst auf den Inventarslot Nummer 13 wechseln (Slot unten links). Hier legen wir später unseren Dünger, das Knochenmehl, hinein. Durch das anschließende „turtle.place()“ soll dieser Dünger nun „platziert“, also benutzt werden. Nachfolgend soll wieder auf den ersten Inventarslot der Turtle gewechselt werden.

```
elseif daten.name == "minecraft:leaves" then
  while turtle.down() do

  end

  turtle.place()

end
```

- Ein weiteres „elseif“ fügen wir für den Fall ein, dass die Turtle vor sich Laub („minecraft:leaves“) identifiziert. Das ist nämlich meist der Fall, wenn die Turtle die Baumkrone erreicht hat und kein Holzblock mehr zu finden ist. Sollte dieses Szenario eintreffen, soll sich die Turtle solange nach unten bewegen, wie es geht, bis sie sich also wieder auf dem Boden befindet.

- Anschließend soll sie noch etwas platzieren und zwar einen weiteren Setzling. Dafür ist es wichtig, dass sich der Setzling auch im oberen linken (ersten) Slot befindet. Findet die Turtle nichts, was sie platzieren kann, ignoriert sie den Befehl. Mit einem „end“ beenden wir die if-then-Abfragen zur Namensüberprüfung des identifizierten Blockes.

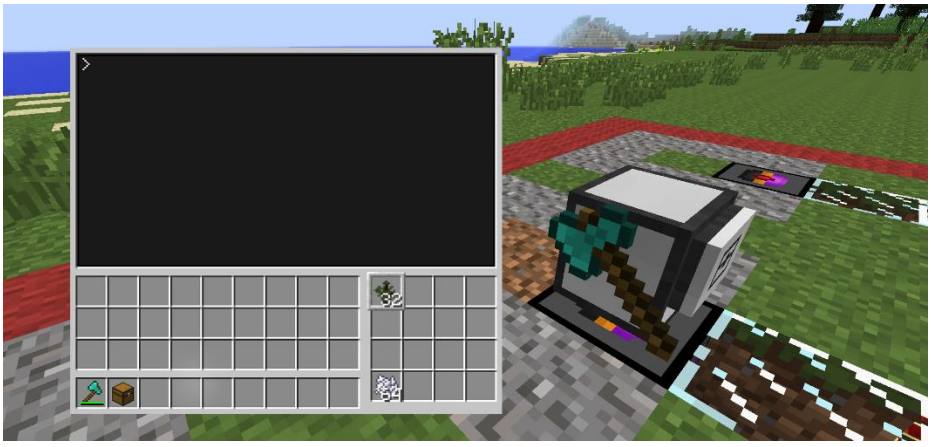
- Außerdem müssen wir auch die übergeordnete if-then-Bedingung noch mit einem „end“ schließen, die abfragt, ob die Turtle überhaupt einen Block vor sich identifizieren konnte („if erfolg then“). Bevor wir das tun, fügen wir jedoch noch ein „else“ mit einer entsprechenden Handlungsalternative ein:

```
  else

    print("runter")
    while turtle.down() do
    end
    turtle.place()

  end

end
```

- Auch diese Vorgehensweise befiehlt der Turtle, sich solange nach unten zu bewegen, bis sie sich wieder auf dem Boden befindet. Darauffolgend soll sie hier ebenso einen Setzling platzieren. Zusätzlich soll sie „runter“ in ihrer Konsole ausgeben, um anzuzeigen, dass sie sich nach unten bewegt. Da es in seltenen Fällen auch vorkommen kann, dass sich an der Spitze der Baumkrone kein Laub mehr befindet, müssen wir diesen Befehlsblock nochmals hinzufügen, denn auch dann wollen wir, dass die Turtle wieder zum Boden zurückkehrt.
  Ein weiteres „end“ schließt das „while true do“ – unsere Endlosschleife – und bildet damit gleichzeitig das Ende unseres Programmcodes.

- Wir speichern das Programm und schließen die Bearbeitung.

- Nun gehen wir noch einmal sicher, dass sich alle benötigten Gegenstände an ihrem Platz befinden.



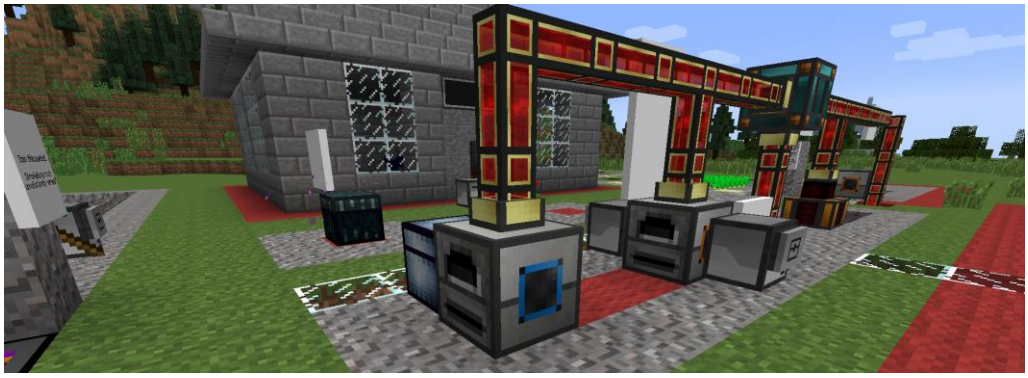
- Noch einmal: Die Setzlinge müssen sich, unserem Code-Beispiel entsprechend, unbedingt im oberen linken Slot befinden. Soll auch noch gedüngt werden, sollte sich das Knochenmehl im unteren linken Slot befinden. Knochenmehl lässt sich bei den NPC-Verkäufern im Turtle-AG Turm gegen Stein erwerben.

- Nun führen wir unser Programm aus.

- Großartig! Unser kleiner Holzfäller platziert Setzlinge, düngt diese sogar (Bonusziel!), fällt anschließend den gewachsenen Baum und beginnt wieder von vorn. Da er immer wieder auf der Ladestation landet, tankt er in regelmäßigen Abständen auch Energie auf. Bei Bedarf müssen wir nur ab und an neue Setzlinge und neues Knochenmehl hinzufügen.

Mehr Automatisierung

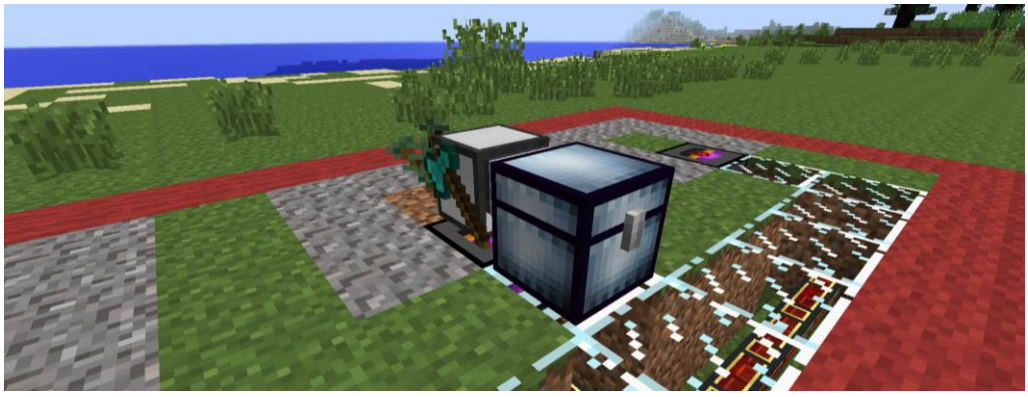
Den größten Teil der Aufgaben haben wir bereits erledigt. Nun müssen noch die Dampfmaschine und der Ofen automatisch befüllt werden, um die Holzkohle und Energieproduktion am Laufen zu halten.



Damit unsere Steine parallel weitergebrannt werden können, entscheiden wir uns, bei einem der NPC-Verkäufer im Turtle-AG Turm einen neuen Redstone-Ofen, ein paar Energietransportrohre und eine Tresortruhe zu kaufen, die wir zusätzlich an unser Energienetz anschließen (siehe oben). Wenn wir die Truhe direkt an einer Ausgabeseite (orange) des Ofens aufstellen, wird die Holzkohle, die der Ofen produziert, direkt in die Truhe abgegeben. Diese müssen wir dann nur noch wieder in die Dampfmaschine geben.

Achtung: Die Seite, an der z.B. eine Turtle Holz in den Ofen hineingeben soll, muss einen blauen Schacht dort haben. Blau steht an den Öfen immer für eine Eingabe, Orange für die Ausgabe!

Doch zunächst müssen wir dafür sorgen, dass das Holz in den Ofen kommt. Das heißt, unsere Holzfäller-Turtle sollte das gesammelte Holz am besten auswerfen. Kein Problem, dafür gibt es ja „turtle.drop()“ ! Aber damit die Turtle das Holz nicht einfach wild in die Welt wirft und die Blöcke nach einer Weile verschwinden, stellen wir auch hinter dieser Turtle noch eine Kiste auf.



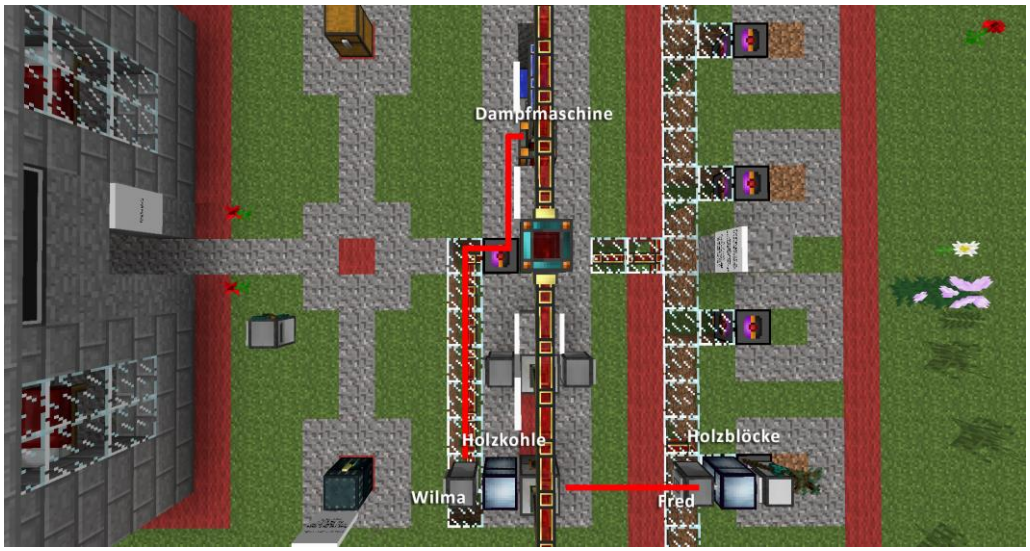
Nun öffnen wir unser Programm „faellen“ und fügen im letzten elseif-Block noch folgendes hinzu:

```
elseif daten.name == "minecraft:leaves" then
  while turtle.down() do

  end

  turtle.place()

  turtle.select(2)
  turtle.turnLeft()
  turtle.turnLeft()
  turtle.drop()
  turtle.select(1)
  turtle.turnLeft()
  turtle.turnLeft()

end
```

So wird unsere Turtle jedes Mal, wenn sie von ihrer Fäll-Routine zurückgekehrt ist, das Holz in ihrem Inventar auswählen (das standardmäßig in den zweiten Slot gelegt werden sollte), sich zur Truhe drehen, das Holz dort hineinlegen, wieder auf den ersten Inventarslot und sich zurückdrehen, bevor sie erneut mit ihrem Standardprogramm beginnt.

Hinweis: Sollte das Programm „faellen“ gerade noch ausgeführt werden, weil es sich in einer Endlosschleife befindet, halten Sie „Strg“ und „T“ solange gedrückt, bis das laufende Programm abgebrochen wird und sie es über Eingaben wieder editieren können.

Nun brauchen wir nur noch ein paar Helfer-Turtles, welche die Ressourcen zwischen den verschiedenen Stationen hin und her transportieren und in die Verarbeitung geben. Schauen wir uns die Anlage einmal von oben an.

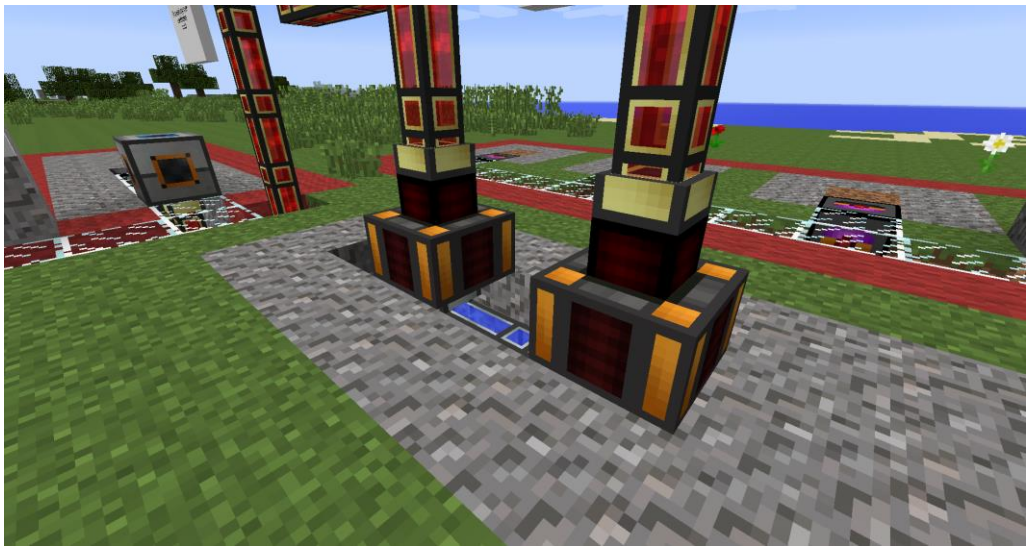


Im oberen Beispiel haben wir die Turtle „Fred“ dafür eingeteilt, Holzblöcke aus der Kiste, die unsere Holzfäller-Turtle befüllt, zu entnehmen, links in den Ofen zu geben, wieder in die Ausgangsposition zu fahren und erneut Materialien aus der Kiste zu holen. Im Ofen wird aus den Holzblöcken Holzkohle hergestellt und automatisch in die Kiste hinter sich ausgegeben, da diese an einem orangenen Ausgabeslot angeschlossen ist. Auf der linken Seite soll „Wilma“ nun wiederum die Kohle aus der Kiste nehmen, zur Dampfmaschine transportieren und diese mit der Kohle befüllen. Auch sie begibt sich anschließend wieder zurück in ihre Ausgangsposition und beginnt von vorn. Mittels Endlosschleifen, turtle.suck() und turtle.drop() und simplen Turtle-Bewegungen lassen sich diese kurzen Wege einfach realisieren und sollen an dieser Stelle nicht detaillierter aufgezeigt werden.

Bonusziele

Unsere Energieproduktion haben wir damit zu einem großen Teil automatisiert. Eines der Bonusziele haben wir mit der automatischen Düngung der Setzlinge durch unsere Holzfäller-Turtle sogar schon umgesetzt. Auch einen weiteren Ofen nutzen wir im Beispiel oben bereits, um parallel Stein brennen zu können und Holzkohle zu produzieren.

Auch zusätzliche Dampfmaschinen können wir uns bei Bedarf im Turtle-AG Turm kaufen. Diese sollten wir dann auf einem der Wasserrohre neben der bereits installierten Dampfmaschine platzieren, da das Gerät zusätzlich zum Brennstoff Wasser benötigt, um Energieproduzieren zu können.



Natürlich können wir auch noch weitere Holzfäller-Turtles an den Ladestationen installieren, die dann auch andere Baumsorten anpflanzen. Letztere erhalten wir bei den NPC-Verkäufern im Turm.

Zu guter Letzt wollen wir unserer Filiale noch eine Tür einsetzen. Über die Suchfunktion in unserem Inventar erfahren wir das Rezept für eine Holztür:

Wir benötigen also 6 Holzbretter. Diese wiederum erhalten wir, wenn wir einfach nur Holzblöcke weiterverarbeiten:

Mit einer Crafty Turtle lassen wir uns über ein Programm, das wir „tuer“ nennen, schließlich eine Tür craften …

und setzen diese in unsere Filiale ein.

Sollten die Teams noch Lust und Zeit haben, können sie natürlich auch ihre Filiale noch ausbauen und verschönern.

Phase 4 | Nachbereitung

Bitten Sie die Kinder etwa 25 Minuten vor Ende der Unterrichtseinheit wieder, mit dem Spielen aufzuhören und ihre Ergebnisse über Ihren Laptop am Beamer zu präsentieren. Dabei können wie gewohnt Fragen gestellt werden.

Gehen Sie mit jedem Team die Checkliste auf dem Arbeitsblatt „Auftrag 2: Stabile Energieversorgung“ durch.

- ¬ Wie bauen die Turtles Holz ab?
- ¬ Ist der Ablauf automatisiert, oder müssen Programme manuell ausgeführt werden?
- ¬ Wurden Bonusziele erreicht?
- ¬ Wie viel Bruchstein, Stein, Holzkohle und Energie hat das Team gesammelt?

Besonders interessant ist in dieser Einheit natürlich die gesammelte Menge an Energie. Der Energiestand der Energiezelle lässt sich mit einem Rechtsklick auf die Energiezelle überprüfen. Ein Hovern über die aufgeploppte Energiesäule gibt die exakte Menge an RF (Redflux) an, die momentan gespeichert ist.

Notieren Sie wieder die gesammelten Ressourcen der einzelnen Teams und diskutieren Sie im Anschluss gemeinsam über die besten Umsetzungen. Gab es größere Probleme bei der Bewältigung der Aufgabe? Zeigen Sie in diesem Fall gerne den ein oder anderen Codeschnipsel unserer Holzfäller-Turtle, um den Kindern einen Lösungsansatz aufzuzeigen.

Phase 5 | Ausblick

Im finalen Curriculum dieser Handreichung wird die Nahrungsversorgung der Filiale im Vordergrund stehen. Zum nächsten Mal können die Kinder gerne schon ein paar Ideen dazu sammeln, wie sie ihre Turtles dazu nutzen können, eine kleine Feldwirtschaft aufzubauen.

CURRICULUM 5 – Nachhaltige Nahrungsproduktion

In unserem letzten Curriculum müssen die Teams sich über eine nachhaltige Nahrungsproduktion in ihrer Filiale Gedanken machen. Selbst der fleißigste Filialmitarbeiter bekommt einmal Hunger und letzterer führt auch in Minecraft über irgendwann zum Leistungsverlust bei der eigenen Spielfigur.

Bisher haben die Kinder sich wahrscheinlich nur über den Vorrat aus der Gemeinschaftskiste und hier und da ein gekauftes Brot versorgt. Auch die Nahrungsproduktion soll nun aber von programmierten Turtles übernommen werden. Hauptaufgabe wird es sein, das Bepflanzen und Abernten von Kartoffelfeldern zu automatisieren und die geernteten Produkte in nahrhafte Ofenkartoffeln weiterzuverarbeiten. In unserem Lösungsbeispiel zeigen wir dieses Mal, wie die Verwendung von Funktionen bei der Strukturierung des eigenen Programms helfen kann. Natürlich gilt es auch in dieser Unterrichtseinheit wieder, möglichst effizient Ressourcen und Energie zu produzieren.

Überblick

Kompetenzen

Zeitaufwand	**90 Minuten**
Technik	Laptops / Standrechner, je Gerät eine Mouse, LAN/WLAN, Beamer
Methoden	Gruppenarbeit, Simulation, Frage und Antwort, Arbeitsblatt
Vorkenntnisse	Programmieren mit Logo, Turtle Trainingcenter 1 & 2, Eine Filiale errichten, Regenerative Energieversorgung

Die Schülerinnen und Schüler …

- können ihre Lua-Programme mit Hilfe von Funktionen strukturieren und effizienter gestalten.
- setzen eine autonome Nahrungsversorgung ihrer Filiale um.
- müssen ihr Programmierwissen problemlöseorientiert anwenden.

Phase	Aufgabe	Methode	Zeit
Reflexion	Wo stehen wir? Woher beziehen wir unsere Lebensmittel in der Minecraft-Welt?	F & A	5´
Vorbereitung	Vorstellung der Aufgabe: Nachhaltige Nahrungsproduktion		5´
Arbeitsphase	Freiarbeit: Bearbeitung der Challenge	Gruppenarbeit, Simulation und Arbeitsblatt	55´
Nachbereitung	Präsentation der Ergebnisse	Präsentation, Simulation, Arbeitsblatt	15‘
Ausblick	Wie könnte es weitergehen?		5‘

Unterrichtsverlauf

Vorbemerkung

Auch in diesem Curriculum kann es hilfreich sein, sich anzuhören, was der Code Magier zum Thema „Funktionen“ zu berichten hat. Zwar ist dies nicht zwingend notwendig, um die neue Aufgabe erfolgreich bearbeiten zu können, dennoch kann das Aufstellen von Funktionen sehr hilfreich sein, um repetitives Codeschreiben zu verhindern, Zeit zu sparen und Programme besser zu strukturieren.

Phase 1 | Reflexion

Wie immer sollten Sie zu Beginn gemeinsam mit den Kindern zusammenfassen, was in der letzten Unterrichtseinheit geschehen ist.

- ¬ Woher beziehen eure Filialen Energie?
- ¬ Was macht der Befehl „turtle.inspect()“?
- ¬ Wozu lassen sich Endlosschleifen nutzen?
- ¬ Was sind Variablen?
- ¬ Woher habt ihr in den letzten Unterrichtseinheiten eure Lebensmittel bezogen?
- ¬ Wie könnte man mit Hilfe der Turtles eine unabhängigere Lebensmittelversorgung gewährleisten? (Antwort: die Turtles Felder bewirtschaften lassen.)

Hinweis: Besteht noch Klärungsbedarf zu den im letzten Curriculum vorgestellten Befehlen und Prinzipien wie turtle.inspect(), der Endlosschleife oder Variablen, gehen Sie am Ende der Vorbereitungsphase noch einmal das Beispielprogramm unserer Holzfäller-Turtle aus der letzten Unterrichtseinheit durch und erklären die einzelnen Schritte.

Phase 2 | Vorbereitung

Bevor Sie den Server starten und die Kinder sich mit dem Spiel verbinden, teilen Sie das Arbeitsblatt „Auftrag 3: Nachhaltige Nahrungsversorgung“ an die Teams aus und besprechen gemeinsam die einzelnen Teilaufgaben.

Aufgabe: Errichtet eine automatisierte Nahrungsmittelproduktion und versorgt die Belegschaft eurer Filiale ausreichend mit Backkartoffeln. Nutzt eure Turtles, um die Kartoffeln zu pflanzen, zu ernten und zu verarbeiten!

Wichtig für alle Minecraft-Neulinge ist es zu wissen, dass sowohl die Spieler selbst, als auch die Turtles Hacken benötigen (keine Spitzhacken!) um einen Erd- oder Grasblock zu beackern. Das Beackern ist notwendig, um anschließend Kartoffeln oder anderes Saatgut in das Feld setzen zu können. Kartoffeln erhalten die Spieler, falls noch nicht abgeerntet, vom Beispielfeld im vordefinierten Nahrungsanbau-Bereich oder aber bei den NPC-Verkäufern im Turtle-AG Turm. Diamanthacken können den Turtles genauso ausgerüstet werden wie andere Werkzeuge.

Vergessen Sie auch hier nicht, den Kindern wieder ein Gespräch mit dem Code Magier ans Herz zu legen, sollten diese sich noch nicht mit ihm zum Thema „Funktionen“ unterhalten haben. Weisen Sie die Teams darauf hin, dass sich mit Hilfe von Funktionen viel Zeit ersparen lässt, verraten Sie aber noch nicht allzu viel über deren Funktionsweise. Teil der Challenge ist es, sich diese Informationen selbst zu erschließen!

Falls noch nicht geschehen, starten Sie den Server und weisen Sie die Kinder an, sich wieder mit der Minecraft-Welt zu verbinden. Diese können dann direkt mit der Bearbeitung der neuen Aufgabe beginnen. Wie immer gilt es außerdem, so viele Ressourcen wie möglich in den Gemeinschaftskisten zu sammeln. Dieses Mal müssen neben Bruchstein, Stein, Holzkohle und Energie zusätzlich Backkartoffeln aufbewahrt werden!

Hinweis: Die Programme der Turtles müssen nach einem Serverstart ebenfalls neu gestartet werden! Machen Sie die Kinder darauf aufmerksam!

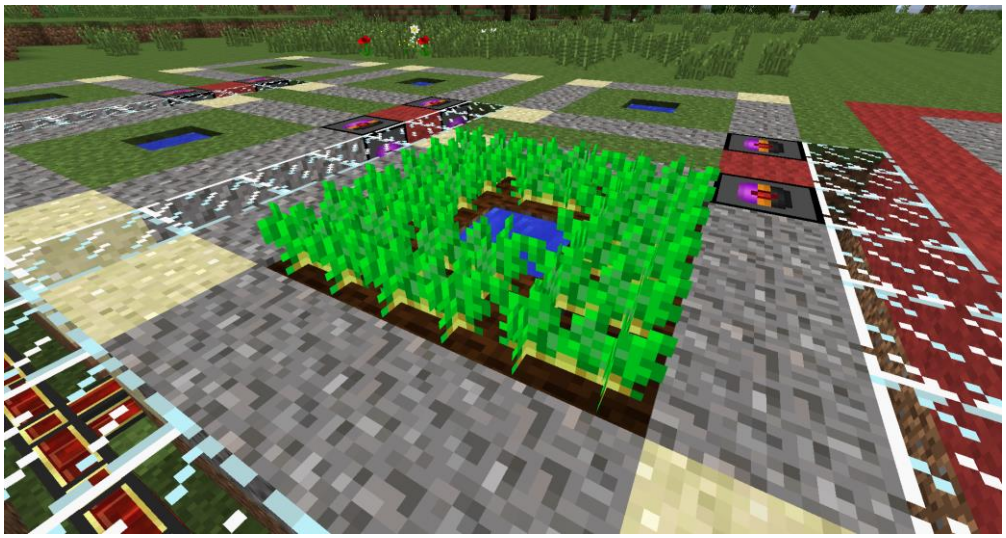
Phase 3 | Arbeitsphase

Nachfolgend skizzieren wir wieder einen möglichen Lösungsweg.

Lösungsbeispiel

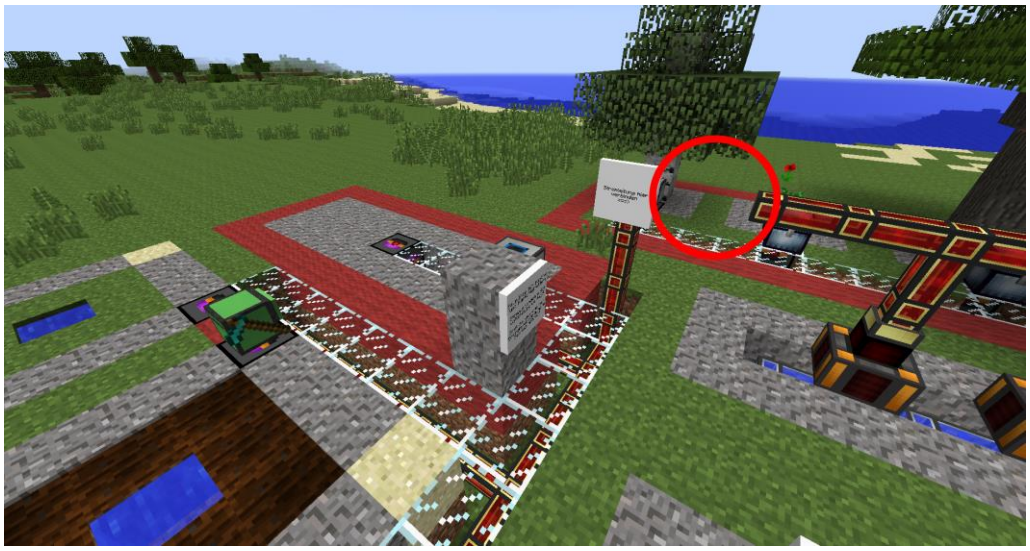
Wie immer überlegen wir zunächst, welche Vorbereitungen notwendig sind, bevor wir mit der Programmierung unserer Farmer-Turtle beginnen können.

- Wir benötigen Kartoffeln als Saatgut für die Felder. Acht Stück sollten für den Anfang genügen.
- Wir brauchen eine neue Turtle, die wir aufladen und mit einer Hacke ausrüsten müssen.
- Wir benötigen einen Bereich, in dem wir unsere Kartoffeln anpflanzen. Wir nutzen in diesem Fall, den vordefinierten Nahrungsanbaubereich auf dem Filialgelände (siehe Übersicht Curriculum 3).
- Wir sollten uns anhören, was der Code Magier zum Thema „Funktionen“ zu erzählen hat.



Die Kartoffeln sollten uns zum jetzigen Zeitpunkt keine Probleme mehr bereiten. Gegen gebrannten Stein, können wir uns diese bei einem der NPC-Händler im Turtle-AG Turm beschaffen. Sofern diese noch nicht abgeerntet wurden, können wir aber auch die Kartoffeln aus dem vordefinierten Bereich auf dem Filialgelände nehmen. Per Linksklick auf die Pflanzen lässt sich das Gemüse abernten.

Eine neue Turtle erhalten wir ebenfalls von den Verkäufern, sollten wir keine mehr vorrätig haben. Wir platzieren die Turtle auf einer der Ladestationen im Anbaubereich, geben ihr den Namen „Farmer“ und rüsten sie per Befehl mit einer Diamanthacke aus. Wichtig ist, dass wir die Turtle in Richtung des roten Wollblockes (siehe Bild oben) platzieren, damit unser Beispielcode anschließend vernünftig funktioniert. Außerdem platzieren wir noch eine Truhe auf dem roten Wollblock, in welche ein Teil der geernteten Kartoffeln gelegt werden soll.



Damit unser kleiner Farmer auch über die Ladestation aufgeladen werden kann, auf der er momentan steht, müssen wir den Anbaubereich noch mit dem Energienetz verbinden. Was hier zu tun ist, wissen wir inzwischen aus den vorherigen Curricula. Wir nehmen uns ein Energietransportrohr (falls nicht mehr vorrätig, ebenfalls bei den Verkäufern zu erhalten) und koppeln Pflanzbereich und Energieproduktion mit wenigen Handgriffen.

Schließlich statten wir auch dem Code Magier nochmal einen Besuch ab und reden mit ihm über Funktionen in Lua. Wir erfahren, dass es sich bei einer Funktion um ein Programmkonstrukt handelt, in dem wir Quellcode festhalten können, der sich später durch Aufruf der Funktion immer wieder ausführen lässt. Für einen solchen Aufruf müssen wir lediglich den Namen der Funktion und eventuell geforderte Parameter in unser Programm schreiben. Das kann uns viel Zeit sparen und zusätzlich immens bei der Strukturierung komplexerer Programme helfen!

Farmer-Turtle

Wie hilft uns dieses neugewonnene Wissen nun aber bei der Programmierung unserer Farmer-Turtle? Schauen wir uns eines der vordefinierten Kartoffelfelder einmal von oben an:

In der Mitte unser Anbaufläche von acht Erdblöcken befindet sich ein Wasserfeld. Dieses hilft den Kartoffeln beim Wachsen und verhindert, dass die Kartoffeln nach einer gewissen Zeit wieder aus der Erde springen. Außerdem ist es wichtig, dass tagsüber genug Licht auf unsere Felder trifft. Unter Bäumen, Vorsprüngen oder ähnlichem wachsen die Pflanzen nur, wenn sie entsprechend ausgeleuchtet sind.

Hinweis: Auch die hier vorgestellte Turtle „Farmer“ kann die Lehrkraft bei den Supervisors im Turtle-AG Turm erhalten und sich den Quellcode im Spiel selbst ansehen und ausprobieren. Wir weisen ausdrücklich darauf hin, dass es sich hierbei um eine recht anspruchsvolle Umsetzung handelt. Natürlich können die Aufgaben auch kleinschrittiger und weniger komplex bearbeitet werden, sodass die Turtle z.B. nur einen Feldstreifen immer wieder aberntet, ohne sich um Ecken bewegen zu müssen. Mit unserem Beispiel wollen wir einen Einblick geben, was noch möglich ist. Die Kinder werden ihre eigenen Wege haben, an die Aufgabe heranzugehen.

Dem Setup oben entsprechend soll unsere Turtle acht Felder bewirtschaften. Sie startet auf der Ladestation, denn um sich bewegen zu können, benötigt sie Energie. Um unser Programm zu planen, zerlegen wir dieses zunächst in logische Einzelabschnitte:

- Die Turtle soll sich nach links drehen.
- Die Turtle soll sich nach vorn bewegen und nach links Richtung Erdblock drehen.
- Die Turtle soll überprüfen, ob etwas geerntet und/oder gepflanzt werden soll und dann eine entsprechende Aktion vornehmen.
- Die Turtle soll sich wieder nach rechts drehen.

Damit haben wir die Behandlung eines Anbaufeldes geregelt. Im Prinzip soll die Turtle alle Aktionen, außer dem ersten Linksdrehen, nun noch sieben Mal wiederholen - einmal für jedes Feld. Einziger Knackpunkt sind die Ecken, an denen die Turtle sich noch einmal nach links bewegen muss. Woher weiß die Turtle nun aber, wann sie sich drehen muss? Genau für diesen Zweck bestehen die vordefinierten Eckfelder hier aus Sand. Mittels des hilfreichen turtle.inspect()-Befehls, kann unser kleiner Farmer erkennen, ob er sich auf einem Sandblock befindet. Diesen Umstand können wir nutzen, um der Turtle zu sagen: „Wenn der Block unter dir aus Sand besteht, wende dich nach links!“ Der Rest des Weges besteht aus Kies. Sofern die Turtle Kies unter sich erkennt, soll sie einfach mit dem Bewegen, Ernten und Pflanzen fortfahren.

Wie können uns Funktionen nun helfen, das Programm effizienter zu gestalten? Anstatt sieben Mal immer wieder zu schreiben, dass die Turtle sich bewegen und eventuell etwas ernten, anpflanzen oder pflügen soll, können wir genau diese beiden Schritte in Funktionen packen. Diesen geben wir die Namen „bewegen“ und „pflanzen“. Wichtig ist, dass wir diese definieren, bevor wir sie im eigentlichen Programm ausführen. Dafür müssen wir sie nun einmal festhalten. Wir öffnen unseren „Farmer“ und starten über „edit anbauen“ ein neues Programm mit dem Namen „anbauen“.

Die Drehung an den Ecken lassen wir zunächst außen vor und kümmern uns erst einmal um das Bewirtschaften der Felder.

```
function bewegen()
```

Über „function bewegen()“ geben wir an, eine neue Funktion für das Programm zu definieren, die den Namen „bewegen“ hat. Alle Befehle, die nun folgen, werden sozusagen in dieser Funktion konserviert.

```
  erfolg, daten = turtle.inspectDown()
  if erfolg then
    if daten.name == "minecraft:gravel"
      turtle.forward()
```

Mit „turtle.inspectDown()“ lesen wir die Daten des Blocks unter der Turtle ein und legen diese, wie auch schon in der letzten Unterrichtseinheit, in die Variablen „erfolg“ und „daten“. Erkennt unser Farmer unter sich einen Block Kies („minecraft:gravel“), soll er sich um einen Schritt vorwärts bewegen.

```
    else
      if turtle.detect() then
      turtle.select(2)
      turtle.drop(10)
      turtle.select(1)
      end
```

Wenn dies nicht der Fall ist („else“), soll mittels „if turtle.detect()“ die weitere Überprüfung vorgenommen werden, ob die Turtle vor sich einen Block erkennen kann. Dies würde nämlich zutreffen, sobald die Turtle die Kiste vor sich sieht, die wir zuvor auf dem roten Wollblock (siehe oben) platziert haben. Dann soll unser Farmer seinen zweiten Inventarslot auswählen, zehn der dort befindlichen Gegenstände in die Kiste geben und wieder auf den ersten Slot zurückwechseln. Auf dem ersten Slot sollten die Kartoffeln liegen, die zum Anpflanzen genutzt werden. Der zweite Slot wird automatisch befüllt, sobald der erste Slot vollständig belegt ist, das heißt, wenn sich dort mindestens 64 Kartoffeln befinden. Indem wir unsere Turtle nur die Kartoffeln aus dem zweiten Slot in die Kiste legen lassen, gewährleisten wir, dass im ersten Slot immer noch genügend Kartoffeln zum Anpflanzen vorhanden sind.

```
    turtle.turnLeft()
    turtle.forward()
    end
  end
end
```

Außerdem soll sich unser kleiner Farmer auf jeden Fall noch nach links drehen und einen Schritt vorwärtsgehen, wenn er zuvor keinen Kies unter sich identifiziert hat. Da nur die Eckfelder nicht aus Kies bestehen, ist also klar, dass sich die Turtle genau auf einem solchen Feld befindet und nun abbiegen muss.

Mit einem weiteren „end“ schließen wir zunächst unsere Funktion „bewegen“. Hier haben wir nun festgelegt, nach welchen Regeln sich die Turtle jeweils um ein Feld fortbewegen soll.

Hinweis: Weisen Sie die Kinder darauf hin, dass sie bei verschachtelten Programmen (mehrere if-then-Bedingungen ineinander usw.) darauf achten sollen, das Setzen der „ends“ nichts zu vergessen. Zu jedem if, zu jedem while, jeder for-Schleife und jeder function gehört am Ende ein „end“! Manch einem kann es z.B. helfen, das „end“ bereits zu Beginn an die entsprechende Stelle zu setzen. So kann es später nicht vergessen werden.

```
function pflanzen()
  turtle.turnLeft()
```

Widmen wir uns nun der Definition der Funktion „pflanzen“. Mit „function pflanzen()“ markieren wir den Kopf unserer Funktion. Da wir davon ausgehen, dass die Turtle sich, durch die vorherige Bewegung, rechts neben einem Pflanzfeld befindet, soll sie sich zunächst in jedem Fall nach links drehen, damit sie das entsprechende Pflanzfeld im Fokus hat.

```
  erfolg, daten = turtle.inspect()

  if erfolg then
    print(daten.name)
    print(daten.metadata)
```

Anschließend lesen wir hier wieder die Daten des Blocks vor der Turtle in die Variablen „erfolg“ und „daten“ ein. Um auch ein visuelles Feedback dafür zu haben, welcher Block gerade von der Turtle eingelesen wurde, lassen wir über den print-Befehl den Namen und die Metadaten des Blocks in der Turtle-Konsole ausgeben. Die Metadaten eines Kartoffelfeldes geben das aktuelle Wachstumsstadium der Pflanze wieder.

```
    if daten.metadata == 7 then
      turtle.dig()
      turtle.place()
```

Die Metadaten können wir nutzen, um herauszufinden, ob die Pflanze ausgewachsen ist und geerntet werden kann. Das machen wir in der nächsten if-then-Abfrage. Wir überprüfen, ob der Wert, der in „daten.metadata“ steht, dem Zahlenwert „7“ entspricht. Wie wir vom Code Magier wissen, ist dies der Fall, wenn die Pflanze ihr volles Reifestadium erreicht hat. Mit „turtle.dig()“ und „turtle.place()“ sorgen wir in diesem Szenario dafür, dass die Turtle die Kartoffeln erntet und im Anschluss das Feld direkt neu bepflanzt.

```
    else
      turtle.select(13)
      turtle.place()
      turtle.select(1)

    end
```

Anderenfalls („else“) wollen wir den Wachstumsprozess beschleunigen, indem wir die Pflanze wieder mit Knochenmehl düngen, genauso, wie wir es schon mit den Bäumen im Holzfäller-Beispielprogramm getan haben (Curriculum 4). Damit dies funktionieren kann, ist es wichtig, dass eventuell vorhandenes Knochenmehl in Inventarslot 13 liegt, also dem Slot ganz unten

links, denn über „turtle.select(13)“ wählt unser kleiner Farmer dann genau diesen Slot, um den dort befindlichen Gegenstand mit „turtle.place()“ auf das Feld vor ihm zu platzieren. Schließlich befehlen wir der Turtle noch, wieder auf den ersten Slot, den Slot mit den Pflanzkartoffeln, zurück zu wechseln. Mit einem „end“ schließen wir die Abfrage, ob die Kartoffeln bereits ausgewachsen sind oder nicht.

```
  else
    turtle.dig()
    turtle.place()
  end
```

Was passiert nun aber, wenn die Turtle vor sich erst gar keine Daten einlesen kann, weil dort noch nichts gepflanzt ist? Auch dafür müssen wir mit einem „else“ noch eine alternative Handlungsweise festlegen. Sollte die Turtle auf dem Feld vor sich nichts erkennen, ist dies ein Zeichen dafür, dass hier noch gar nichts gepflanzt oder sogar noch gar nicht gepflügt wurde. Das holen wir ganz einfach mit einem „turtle.dig()“ und einem „turtle.place()“ nach.

```
  turtle.turnRight()

end
```

Egal, ob nun geerntet, gepflügt, gepflanzt oder gedüngt wurde. Am Ende soll die Turtle sich wieder nach rechts drehen, um weiter um das Feld patrouillieren zu können. Mit einem „end schließen wir die Funktion „pflanzen“, sollten wir dies noch nicht getan haben.

Kaum zu glauben, aber mit all dem Code, den wir bisher geschrieben haben, passiert beim Ausführen des Programms momentan: gar nichts. Wir haben zwar die Funktionen „bewegen“ und „pflanzen“ definiert, diese aber noch nicht einmal wirklich aufgerufen. Das eigentliche Programm lässt sich nun sehr kurz schreiben.

```
while true do
  bewegen()
  pflanzen()
  bewegen()
  pflanzen()
  bewegen()
end
```

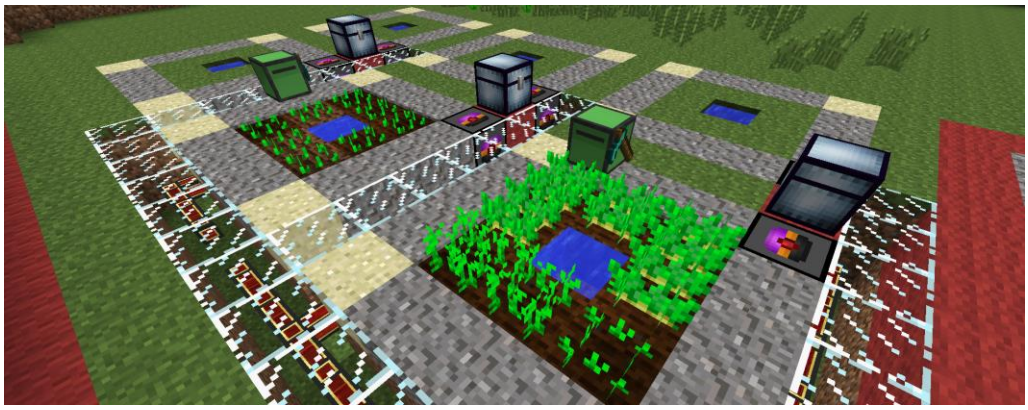
Mit „while true do“ starten wir unter den Funktionen eine Endlosschleife, damit unser Farmer das Programm später auch stetig ausführt. Nun müssen wir zwischen dem while und seinem end nur noch Folgendes einfügen:

Durch das bloße Schreiben von „bewegen()“ und „pflanzen()“ werden diese Funktionen nun nacheinander mehrmals aufgerufen. Für jede Seite des Kartoffelfeldes wird die while-Schleife dabei einmal durchlaufen. Daher rufen wir fünfmal die bewegen-Funktion (die Turtle muss sich fünf Blöcke fortbewegen) und dreimal die pflanzen-Funktion (pro Seite gibt es drei Feldblöcke zu bewirtschaften) auf.

Nun ist es allerhöchste Zeit, das Programm abzuspeichern, falls noch nicht geschehen! Über einen Blick in das Inventar der Turtle versichern wir uns, das die entsprechenden Hilfsmittel richtig verteilt sind. Im ersten oberen linken Slot sollten sich mindestens acht Kartoffeln befinden, damit der kleine Farmer genügend Pflanzmaterial zur Verfügung hat. In Slot Nummer 13 sollten wir, sofern vorrätig, wieder etwas Knochenmehl packen, damit die Turtle, den Wachstumsprozess im besten Fall etwas beschleunigen kann.

Sobald wir alles überprüft haben, kann es losgehen. Wir speichern, schließen die Programmbearbeitung und lassen über „anbauen“ unser Programm ausführen.

Wenn wir alles richtiggemacht haben, marschiert unsere kleine Turtle nun unermüdlich um das Kartoffelfeld, kümmert sich um die Pflanzen und gibt überschüssige Kartoffeln in die von uns aufgestellte Truhe! Ab und zu sollten wir etwas Knochenmehl nachfüllen, damit es mit der Ernte etwas schneller geht.



Prima, jetzt müssen wir die Kartoffeln nur noch aus der Kiste holen und von einer Koch-Turtle verarbeiten lassen.

Koch-Turtle

Der größte Teil der Aufgabe ist geschafft. Eine Turtle zu automatisieren, welche die Kartoffeln aus der Kiste holt und in den Ofen gibt, sollte mit dem bis hier erreichten Wissensstand keine größeren Probleme mehr bereiten. Mittels Endloschleife, den Turtle-Bewegungsbefehlen und den Variationen von „turtle.suck()“ und „turtle.drop()“ lässt sich einfach eine Koch-Turtle realisieren, die für die Verarbeitung der Kartoffeln sorgt.

Eine Ladestation und ein Ofen stehen dafür schon auf dem vordefinierten Kantinengelände bereit. Wir entscheiden uns lediglich noch dazu, eine weitere Kiste hinter dem Ofen aufzustellen, in welche die fertigen Backkartoffeln automatisch ausgegeben werden.

Hinweis: Eine fortgeschrittene Beispiel-Koch-Turtle erhalten Sie bei einem der Turtle-Supervisors im Turtle-AG-Turm. Das Besondere an dieser Turtle ist, dass Sie ihren Nutzer nach der gewünschten Anzahl an Backkartoffeln fragt und eine entsprechende Eingabe verlangt, genau diese Menge aus der Kiste holt, verarbeitet und dem Nutzer ausgibt, wenn dieser das Programm „kochen“ ausführt. Über den edit-Befehl können Sie sich das Programm bei Interesse anschauen und den Kindern bei Bedarf vorstellen. Dies empfehlen wir aber nur bei einer fortgeschrittenen Gruppe.

Bonusziele

Eines der Bonusziele haben wir in unserem Beispiel bereits erfüllt. Sofern unserem kleinen Farmer Knochenmehl zur Verfügung steht, düngt er die Kartoffeln automatisch.

Bei den anderen Bonuszielen handelt es sich weitestgehend um Bauaufgaben. Einerseits gilt es ein kleines Kantinengebäude zu bauen, andererseits soll eine Art Gewächshaus um die Felder gebaut werden. Inzwischen sollten wir genügend Materialien haben, um auch diese Aufgaben ohne Probleme lösen zu können.

Wichtig ist es nur, darauf zu achten, dass genug Licht an die Kartoffeln kommt, denn sonst wachsen diese nur mit Hilfe von Knochenmehl. Daher entscheiden wir uns in unserem Beispiel für ein Glasdach.

Phase 4 | Nachbereitung

Je nach Größe der Gruppen ist es spätestens 25 Minuten vor Ende der Unterrichtseinheit wieder Zeit, die Teams ihre Ergebnisse über den Beamer präsentieren zu lassen. Gehen Sie wie gewohnt gemeinsam die einzelnen Aufgabenpunkte auf dem Arbeitsblatt „Auftrag 3: Nachhaltige Nahrungsversorgung“ durch.

- Wie wurden die Programme für die Pflanz-Turtles umgesetzt?
- Ist der Ablauf automatisiert, oder müssen Programme manuell ausgeführt werden?
- Wurden Bonusziele erreicht?
- Wie viel Bruchstein, Stein, Holzkohle, Energie und Backkartoffeln hat das Team gesammelt?

Notieren Sie wieder die gesammelten Ressourcen der einzelnen Teams und diskutieren Sie im Anschluss gemeinsam über die besten Umsetzungen. Da es sich hierbei um das letzte Curriculum dieser Handreichung handelt, lassen Sie alle Unterrichtseinheiten gemeinsam Revue passieren. Die Kinder haben gelernt mit den Computercraft Turtles umzugehen, kleine Programme zu schreiben und diese problemlöseorientiert einzusetzen. Wie sehen die Filialplätze inzwischen aus? Was hat sich alles entwickelt und was könnte in Zukunft noch entstehen?

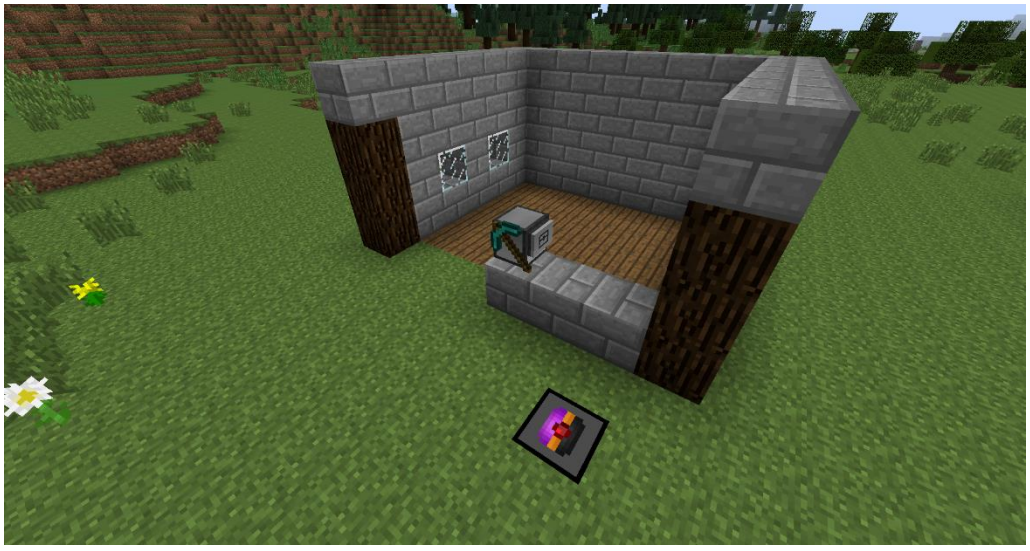
Phase 5 | Ausblick

Sie können den Teams gratulieren! In fünf Unterrichtseinheiten haben sie die Curricula der Turtle-AG durchlaufen und viel Neues gelernt!

Natürlich muss Ihre Arbeit mit Minecraft hier noch lange nicht enden. Gerne können Sie die von uns zur Verfügung gestellte Welt weiternutzen und neue Projekte mit ihren Kindern umsetzen. Sicherlich haben ihre Schülerinnen und Schüler schon jetzt viele Ideen, wie sie mit der Welt weiterverfahren könnten. Überlegen Sie gemeinsam und sammeln Sie Ideen an der Tafel.

Hilfreiche, weiterführende Informationen zu Computercraft und seinen Turtles finden Sie beispielsweise unter:

- http://ftb.gamepedia.com/Getting_Started_(ComputerCraft)/de (eine Einführung in Computercraft)
- http://computercraft.info/wiki/Turtle_(API) (englisch, wichtigste Turtle-Befehle in einer Übersicht)



Ausblick: Die Turtle baut ein Haus nach Plan

Wie wäre es z.B. ein Programm zu schreiben, nach welchem die Turtle allein ein Haus baut? Sozusagen eine Art Bauplan, mit dessen Hilfe die Turtle ein (im Programm) vorgezeichnetes Haus baut. Eine spannende neue Aufgabe für die fortgeschritteneren Teams!

Wie so etwas aussehen kann, können Sie sich gemeinsam ansehen, wenn Sie sich die Turtle mit dem Namen „Bauarbeiter“ von einem der Turtle-AG-Supervisors aus dem Turm holen und die Turtle das Programm „haus“ auf einer ebenen Fläche ausführen lassen. Wichtig ist jedoch,

dass Sie zuvor entsprechende Materialien in den Inventarslots der Turtle verteilen, wie nachfolgend gezeigt.

Auch hier können Sie sich das Programm natürlich über den edit-Befehl ansehen.

Schlusswort

Lernen mit Minecraft? Ist das die Zukunft? Selbstverständlich braucht es insgesamt mehr als ein Spiel, um den Herausforderungen des 21. Jahrhunderts hinsichtlich einer sinnvollen, zeitgemäßen Lernkultur zu begegnen. Projekte mit Minecraft können dabei nur einer von vielen möglichen Ansätzen sein, dieser anspruchsvollen Aufgabe zu begegnen.

> „We live in the 21st century. Why shouldn't we learn in 21st century ways?" Marc Prensky (Lehrer, Autor, Unternehmer)

Auf dem Gebiet des Digital Game-based Learning gibt Minecraftein Paradebeispiel ab. Zum einen handelt es sich hier um ein Spiel, das nicht nur theoretische Konzepte, sondern bereits international und national erfolgreich durchgeführte Praxisprojekte vorweisen kann, zum anderen sind deren Ergebnisse nicht nur von wissenschaftlichen Experten, sondern in erster Linie von den Lehrern selbst dokumentiert und befürwortet, wie sich inzwischen in vielen Communities im Netz, Fortbildungen etc. gezeigt hat.

Die offene Spielwelt Minecrafts erlaubt es, zahlreiche Modifikationen vorzunehmen und sich, im Wahrsten Sinne des Wortes, zu einem hohen Maß ein digitales Lernsetting so zusammen zu bauen, wie der Lehrer es haben will. Gewalttätige Monster lassen sich bei Bedarf „ausschalten", Schüler können „unsterblich" gemacht werden und bestimmte Bereiche nach Belieben unzerstörbar oder unbetretbar. Andere kommerzielle Spiele sind dahingehend weniger flexibel und eignen sich für einen Einstieg in das Digital Game-based Learning vermutlich weniger als Minecraft, was jedoch nicht generell gegen einen möglichen Einsatz spricht.

Natürlich lassen sich derartige Konzepte, je nach Thematik und Schwerpunkt, auch mit anderen Spielen umsetzen. Digital Game-based Learning unterstützt nicht nur den für die Schüler natürlicheren Zugang zu schulischen Inhalten, sondern fördert überdies viele in unserer

Gesellschaft und Wirtschaft so wichtige und immer mehr an Bedeutung gewinnende Soft Skills, wie Teamfähigkeit, Selbstorganisation, Problemlösekompetenz und mehr. Zudem macht es ein Projekt orientiertes und interdisziplinäres Arbeiten möglich, so wie es auch in beruflichen Einrichtungen später üblich ist.

Durch das Spielsetting und den Lehrer lancierte Fragestellungen zu behandelten Thematiken, werden Schüler zusätzlich zu kritischem, Problem fokussiertem Denken angeregt. Digital Game-based Learning kann sich, wenn richtig eingesetzt, als mächtiges Werkzeug für den Lehrer erweisen und ihm in der heutigen Zeit die Aufgabe erleichtern, seine Schüler zu ganzheitlich und eigenständig denkenden, verantwortungsbewussten und selbstständig handelnden Persönlichkeiten heranzubilden.

Anhang – Arbeitsblätter aus dem Curriculum

Im Anhang dieses Curriculums finden Sie sortiert alle Arbeitsblätter, welche Sie bei der Umsetzung der Unterrichtseinheiten benötigen. Es sind sowohl die Blnako-Arbeitsblätter gelistet, als auch eine jeweilige Lösungsvorlage. Im Detail sind dies:

Aufgabenblatt Minecraft Steuerung (Test 1)

Lösungsblatt Minecraft Steuerung (Test 1)

Aufgabenblatt Turtle Befehle (Test 5/6/7/8)

Lösungsblatt Turtle Befehle (Test 5/6/7/8)

Auftrag 1: Die Filiale errichten!

Auftrag 2: Stabile Energieversorgung

Auftrag 3: Nachhaltige Nahrungsproduktion

Aufgabenblatt Minecraft Steuerung (Test 1)

Minecraft Steuerung (Test 1)

Inventar Auf/Zu

Benutzen/Reden/Bauen

Abbauen/Werkzeuge

Inventar auswählen

Springen

Turtle Basiswissen (Test 2/3/4)

Turtle neu starten

Turtle verlassen

Turtle Programm abbrechen

LUA Konsole öffnen

LUA Konsole schließen

Programm aufrufen

Neues Programm/ Programm editieren

Programm im Editor speichern

Editor verlassen

Turtle benennen

Turtle bewegen

Vorwärts

Rückwärts

Nach oben

Nach unten

Links drehen

Rechts drehen

Navigieren im Craft OS

Verzeichnisinhalte anzeigen

Verzeichnis wechseln

for Schleife

Redstone Signal ausgeben

Turtle Energiestand abfragen

Lösungsblatt Minecraft Steuerung (Test 1)

Minecraft Steuerung (Test 1)

	Vorwärts W		Inventar Auf/Zu	E	Inventar auswählen	Mausrad	Springen Space
A Links		D Rechts	Benutzen/Reden/Bauen	Rechtsklick			
	S Rückwärts		Abbauen/Werkzeuge	Linksklick			

Turtle Basiswissen (Test 2/3/4)

Turtle neu starten	STRG+R	Turtle Programm abbrechen	STRG+T
Turtle verlassen	ESC	LUA Konsole öffnen	lua
		LUA Konsole schließen	exit()

Programm aufrufen	Programmname in der Craft OS Konsole eingeben
Neues Programm/ Programm editieren	edit <Programmname>
Programm im Editor speichern	STRG, dann „Save“ wählen, mit ENTER bestätigen
Editor verlassen	STRG, dann „Exit“ wählen, mit ENTER bestätigen
Turtle benennen	label set <Turtlename>

Turtle bewegen

Vorwärts	turtle.forward()
Rückwärts	turtle.back()
Nach oben	turtle.up()
Nach unten	turtle.down()
Links drehen	turtle.turnLeft()
Rechts drehen	turtle.turnRight()

Navigieren im Craft OS

Verzeichnisinhalte anzeigen	ls
Verzeichnis wechseln	cd

for Schleife

```
for i=<Anfangswert>,<Endwert>,<Intervall> do
    <Code>
end
```

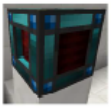
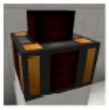
	Seite	An/Aus
Redstone Signal ausgeben	redstone.setOutput(“*<top/back/front/left/right/bottom>*”,true/false)	
Turtle Energiestand abfragen	turtle.getFuelLevel()	

Aufgabenblatt Turtle Befehle (Test 5/6/7/8)

Turtle Befehle (Test 5/6/7/8)

	vor der Turtle	über der Turtle	unter der Turtle
Block abbauen			
Block platzieren			
Block einsaugen			
Block fallenlassen			
Block erkennen			

Turtle ausrüsten		Inventarplatz auswählen

			
Blockname			
Funktion			

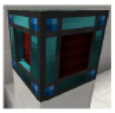
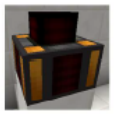
while Schleife

if Bedingung

Lösungsblatt Turtle Befehle (Test 5/6/7/8)

Turtle Befehle (Test 5/6/7/8)

	vor der Turtle	über der Turtle	unter der Turtle
Block abbauen	turtle.dig()	turtle.digUp()	turtle.digDown()
Block platzieren	turtle.place()	turtle.placeUp()	turtle.placeDown()
Block einsaugen	turtle.suck()	turtle.suckUp()	turtle.suchDown()
Block fallenlassen	turtle.drop()	turtle.dropUp()	turtle.dropDown()
Block erkennen	turtle.detect()	turtle.detectUp()	turtle.detectDown()
Turtle ausrüsten	turtle.equipLeft() turtle.equipRight()	Inventarplatz auswählen	turtle.select(1..16)

			
Blockname	Redstone Ofen	Energiezelle	Dampmaschine
Funktion	brennt Stein/ Erze zu Barren kocht Nahrung	Speichert Energie	Erzeugt aus Kohle Energie

while Schleife

```
while <Bedingung> do
   <Code>
end
```

Kopfgesteuerte Schleife,
wird ausgeführt solange die
Bedingung wahr (TRUE) ist.

if Bedingung

```
if <Bedingung> then
  elseif <Bedingung>
  else
end
```

Auftrag 1: Die Filiale errichten!

Errichten Sie auf ihrem Bauplatz ein Filialgebäude, das folgende Kriterien erfüllt:

- [ ] Der Bau muss größtenteils aus Steinblöcken/ Steinziegeln errichtet werden!
- [ ] Der Bau enthält kein Gras, keine Erde und keinen Bruchstein!
- [ ] Der Bau muss mindestens 5 Blöcke hoch sein!
- [ ] Der Bau muss mindestens 5 Betten problemlos Platz bieten.
- [ ] Der Bau muss pro Seite mindestens ein Glasfenster besitzen.

Programmieren Sie die Turtles dafür so, dass:

- [ ] sie automatisch einen mit Länge, Breite und Tiefe definierten Bereich abbauen.
- [ ] sie den Abbau nur starten wenn, sich mindestens 2000 RF im Energiepseicher befinden.
- [ ] sie abgebauten Bruchstein automatisch im Ofen zu Stein brennen.
- [ ] sie fertig gebrannten Stein in die Gemeinschaftstruhe legen.

Bonusziele:

- [ ] Lassen Sie den Monitor an der Vorderseite den Namen ihrer Filiale anzeigen.
- [ ] Benutzen Sie hochwertige Materialien für den Bau und gestalten Sie ihre Filiale von aussen und innen möglichst ansprechend!
- [ ] Bauen Sie ein Schrägdach für Ihre Filiale!

Wettbewerbsziel: Rohstoffe sammeln

Sammeln Sie so viel Bruchstein und Stein in ihrer Gemeinschaftskiste wie möglich!
Mengen am Ende hier eintragen:

Bruchstein	Stein

Auftrag 2: Stabile Energieversorgung

Pflanzen und fällen Sie automatisiert Bäume mit den Turtles und verarbeiten Sie die gewonnenen Holzstämme zu Holzkohle weiter. Holzkohle soll dann in der Dampfmaschine zur Stromerzeugung genutzt werden.

- [ ] Sammeln Sie ausreichend Baumsetzlinge!
- [ ] Rüsten Sie Turtles mit Äxten aus!
- [ ] Erkundigen Sie sich beim Code Magier über Variablen, Endlosschleifen und die Möglichkeit, Blöcke genau analysieren zu lassen!

Programmieren Sie die Turtles so, dass:

- [ ] sie die Baumsetzlinge automatisch pflanzen!
- [ ] sie die gewachsenen Bäume automatisch fällen!
- [ ] sie den Ofen automatisch mit Holzstämmen befüllen!
- [ ] sie die Dampfmaschine automatisch mit Holzkohle befüllen!

Bonusziele:

- [ ] Turtles düngen die Baumsetzlinge automatisch mit Knochenmehl.
- [ ] Pflanzen Sie 3 verschiedene Baumsorten an!
- [ ] Nutzen Sie mehrere Öfen und Dampfmaschinen!
- [ ] Setzen sie in den Eingang ihrer Filiale eine Tür ein!

Wettbewerbsziel: Rohstoffe sammeln

Sammeln Sie so viel Bruchstein, Stein und Holzkohle in ihrer Gemeinschaftskiste wie möglich! Speichern Sie auch so viel Energie wie möglich! Mengen am Ende hier eintragen:

Bruchstein	Stein	Holzkohle	Energie

Auftrag 3: Nachhaltige Nahrungsproduktion

Damit nicht nur die Turtles optimale Arbeit leisten können, müssen Sie eine automatisierte Nahrungsmittelproduktion errichten. Ihr Auftrag ist es, die Belegschaft Ihrer Filiale mit ausreichend gebackenen Kartoffeln zu versorgen. Die Turtles übernehmen dabei Pflanzung, Ernte und Zubereitung!

- [ ] Beschaffen Sie ausreichend Kartoffeln zum einpflanzen!
- [ ] Rüsten Sie Turtles mit Hacken aus!

Programmieren Sie die Turtles so, dass:

- [ ] sie automatisch unbeackertes Gras/Erde in beackertes Land umharken!
- [ ] sie automatisch Kartoffeln in beackertes Land pflanzen!
- [ ] sie automatisch ausgewachsene Kartoffeln ernten!
- [ ] sie geerntete Kartoffeln im Ofen zu Backkartoffeln verarbeiten!

Bonusziele:

- [ ] Turtles düngen die Kartoffeln automatisch mit Knochenmehl.
- [ ] Bauen Sie um die Felder ein Gewächshaus!
- [ ] Bauen Sie eine optisch ansprechende Kantine in der Turtles bei Bedarf Backkartoffeln ausgeben!

Wettbewerbsziel: Rohstoffe sammeln

Sammeln Sie so viel Bruchstein, Stein, Holzkohle Backkartoffeln in Ihrer Gemeinschaftskiste wie möglich! Speichern Sie auch so viel Energie wie möglich!
Mengen am Ende hier eintragen:

Bruchstein	Stein	Holzkohle	Energie	Backkartoffeln