A picture containing toy

Description automatically generated

**UNTERRICHTSPLAN**

**Titel:** Stunde des Codes 2021 (TimeCraft)

**Alter:** 7+

**Themenbereiche:** Informatik, Mathematik, Sprachunterricht, Geschichte, Bildende Künste, Wissenschaften, Konstruktion

Fähigkeiten des **21. Jahrhunderts:** Kreativität, Kritisches Denken, Problemlösen

**Unterstützte Sprachen:** Englisch (USA), Englisch (Großbritannien), Deutsch, Spanisch (Spanien), Spanisch (Mexiko), Französisch (Frankreich), Französisch (Kanada), Italienisch, Japanisch, Koreanisch, Portugiesisch (Brasilien), Portugiesisch (Portugal), Russisch, Chinesisch (vereinfacht), Chinesisch (traditionell), Türkisch, Polnisch, Indonesisch, Ukrainisch, Griechisch, Bulgarisch, Ungarisch, Finnisch, Dänisch, Schwedisch, Norwegisch, Slowakisch, Tschechisch, Niederländisch

**Mehrspieler/Einzelspieler:** Einzelspieler

**Nötige Erfahrungsstufe mit Minecraft (Lehrkraft):** Anfänger

**Lektionsbeschreibung:** Ein unterhaltsamer und kreativer Einstieg in die Informatik. Reise durch die Zeit und verwende blockbasierten oder Python-Code, um seltsame Fehler im Zeitverlauf der Erde zu beheben. Sammle Hinweise und setze kritisches Denken ein, um herauszufinden, wer der Übeltäter in diesem Abenteuer durch die Zeit ist. Schülerinnen und Schüler erkunden gleichzeitig die Integration von Information in der Arbeitswelt und im persönlichen Alltag.

**Vorgesehene Gesamtdauer:** 1 Stunde

**Lehrstandards:**

|  |  |  |
| --- | --- | --- |
| Standards | | |
| Vereinige Staaten | Australien | Vereinigtes Königreich – England |
| Ich kann tägliche Prozesse erstellen, indem ich Algorithmen erschaffe und ihnen folge, um Aufgaben abzuschließen.  **1A-AP-08** | Ich kann einer Reihe von Schritten und Entscheidungen (Algorithmen) folgen sowie sie beschreiben und wiedergeben, die für das Lösen von einfachen Problemen notwendig sind.  **ACTDIP004** | Ich kann einfache Programme erstellen und debuggen.  **Key Stage 1** |
| Ich kann Programme mit Sequenzen und einfachen Schleifen erstellen, um Ideen auszudrücken oder Probleme zu lösen.  **1A-AP-10** | Ich kann einfache Probleme definieren sowie eine Reihe von Schritten und Entscheidungen (Algorithmen) beschreiben und ihnen folgen, die für das Lösen dieser Probleme notwendig sind.  **ACTDIP010** | Ich kann verstehen, was Algorithmen sind und wie sie als Programme auf digitalen Geräten umgesetzt werden sowie dass Programme mittels genauer und eindeutiger Anweisungen ausgeführt werden.  **Key Stage 1** |
| Fehler im Algorithmus oder Programm mit Sequenzen und einfachen Schleifen debuggen (finden und beheben).  **1A-AP-14** | Einfache Algorithmen mit Sequenzen von Schritten, Verzweigungen und Iterationen (Wiederholungen) entwerfen und bearbeiten sowie ihnen folgen.  **ACTDIP019** | Ich kann mithilfe von logischem Folgern das Verhalten von einfachen Programmen vorhersagen.  **Key Stage 1** |
| Ich kann einen Vergleich ziehen zwischen der Art, wie Menschen leben und arbeiten, bevor neue Computertechnologien implementiert oder übernommen werden, und wie sie das danach tun.  **1A-IC-16** | Algorithmen entwerfen, die diagrammatisch und auf Englisch gestaltet sind, und Algorithmen folgen, um die Ausgabe für eine bestimmte Eingabe vorhersagen und Fehler finden zu können.  **ACTDIP029** | Ich kann mithilfe von logischem Folgern erklären, wie einfache Algorithmen funktionieren und wie man Fehler in Algorithmen und Programmen aufspürt und behebt.  **Key Stage 2** |
| Ich kann Programme erstellten, die Sequenzen, Ereignisse, Schleifen und Bedingungen enthalten.  **1B-AP-10** | Ich kann einfache Probleme definieren sowie eine Reihe von Schritten und Entscheidungen (Algorithmen) beschreiben und ihnen folgen, die für das Lösen dieser Probleme notwendig sind.  **ACTDIP01** | Ich kann Sequenzen, Auswahlen und Wiederholungen in Programmen verwenden sowie mit Variablen und verschiedenen Eingabe- und Ausgabeformen arbeiten.  **Key Stage 2** |
| Ich kann mehrere Algorithmen für dieselbe Aufgabe vergleichen und verbessern sowie feststellen, welcher am besten geeignet ist.  **1B-AP-08** | Ich kann einfache digitale Lösungen als visuelle Programme mit Algorithmen inkl. Verzweigungen (Entscheidungen) und Benutzereingabe implementieren.  **ACTDIP011** | Ich kann Programme für spezifische Ziele, darunter Steuerung oder Simulation physikalischer Systeme, entwerfen, schreiben und debuggen sowie Probleme lösen, indem ich sie in Teilprobleme aufteile.  **Key Stage 2** |
| Ein Programm oder einen Algorithmus testen und debuggen (Fehler finden und beheben), um zu gewährleisten, dass sie wie vorgesehen funktionieren.  **1B-AP-15** | Ich kann digitale Lösungen als einfache visuelle Programme mit Verzweigungen, Iterationen (Wiederholungen) und Benutzereingabe implementieren.  **ACTDIP020** | Ich kann mehrere Schlüsselalgorithmen verstehen, die das informatische Denken widerspiegeln, sowie mit logischem Folgern die Nützlichkeit von unterschiedlichen Algorithmen für dasselbe Probleme vergleichen.  **Key Stage 3** |
| Ich kann Computertechnologien erläutern, die die Welt verändert haben, und erklären, wie diese Technologien kulturelle Praktiken beeinflussen bzw. wie sie von diesen beeinflusst werden.  **1B-IC-20** | Ich kann Programme mit Benutzeroberflächen in einer Universal-Programmiersprache implementieren und bearbeiten, die Verzweigungen, Iterationen und Funktionen enthalten.  **ACTDIP030** | Ich kann mit zwei oder mehr Programmiersprachen, von denen mindestens eine textbasiert ist, eine Vielzahl von Informatikproblemen lösen, Datenstrukturen sinnvoll einsetzen sowie modulare Programme mit Prozeduren oder Funktionen entwerfen und entwickeln.  **Key Stage 3** |
| Ich kann mit Flussdiagramen und/oder Pseudocode komplexe Probleme in Algorithmenform bringen.    **2-AP-10** | Ich kann mithilfe einfacher Software unterschiedliche Arten von Daten sammeln, abrufen und präsentieren, um Informationen zu vermitteln und Probleme zu lösen.  **ACTDIP009** |  |
| Ich kann Programme entwerfen und iterativ entwickeln, die Steuerstrukturen wie verschachtelte Schleifen oder zusammengesetzte Bedingungen enthalten.  **2-AP-12** |  |  |
| Ich kann Probleme und Teilprobleme in kleinere Teile aufteilen, um den Entwurf, die Implementierung und die Prüfung von Programmen zu erleichtern.  **2-AP-13** |  |  |

**Lehrkraft-Vorbereitung:**

* Installiere Minecraft: Education Edition unter <https://aka.ms/HourofCode2021>
* Sieh dir [das Einführungsvideo](https://aka.ms/HOC2021Trailer) zum Thema der diesjährigen Stunde des Codes an.
* Schau dir [das Walkthrough-Video](https://aka.ms/HOC2021Walkthrough) an, um einen besseren Eindruck von der Lektion zu erhalten und zu lernen, wie du durch die Minecraft-Welt navigieren kannst.
* Lade den [Leitfaden für Lehrkräfte](aka.ms\HOC2021educatorguide) herunter, der dir alle Informationen vermitteln soll, die du benötigst, um deinen Schülerinnen und Schülern dabei zu helfen, eine angenehme und erfolgreiche Stunde des Codes zu erleben. Dazu gehören auch alle externen Ressourcen und empfohlenen Anweisungen für Frontalunterricht, selbstständiges Lernen und hybride bzw. Remote-Lernumgebungen.
* Verwende die schülerorientierten [Präsentationsfolien](aka.ms\HOC2021presentationslides) für Schritt-für-Schritt-Anweisungen.
* Lade eine Kopie der [Programmierlösungen](https://aka.ms/HOC2021CodingSolutions) herunter.
* Hast du weitere Fragen? Schau dir die [FAQ](https://aka.ms/HOC2021FAQ) an.

**Thematischer Überblick:**

Als Informatik-Fachkraft am Institut für extreme Zeitfehler ist es deine Aufgabe, die geheimnisvollen Zeitrisse zu beheben, die in der Geschichte auftreten, und herauszufinden, wer (oder was!) für sie verantwortlich ist.

Wirst du mit deinen Super-Programmierfähigkeiten dabei helfen, die Zeitrisse zu beheben und die Geschichte zu retten?

Wirst du herausfinden, wer – oder was – diese verrückten Änderungen im Verlauf der Geschichte verursacht?

In deiner TimeCraft-Mission musst du Folgendes tun:

* Reise zurück zu aufregenden Zeitpunkten der Weltgeschichte.
* Programmiere deinen Zeit-Agenten, um die Zeitrisse zu beheben.
* Finde mithilfe der Hinweise heraus, wer oder was für die Zeitrisse verantwortlich ist.

Die Schülerinnen und Schüler können blockbasierten oder Python-Code verwenden.

**Lernziele:**

* Die Bedeutung und Vorteile von Informatik in allen Lebensbereichen verstehen.
* Probleme mit algorithmischem Denken und Problemzerlegung analysieren und lösen.
* Programmierkonzepte wie Sequenzen, Ereignisse, Schleifen und Debugging üben.
* Programmierlösungen kreieren, um erfolgreich eine Aufgabe zu bewältigen oder ein Problem zu lösen.
* Die erweiterten Karrieremöglichkeiten erkennen, die sich durch die Informatik bieten.

**Schüleraktivitäten:**

**Einführungsaktivitäten (5 Minuten):**

1. Die Schülerinnen und Schüler denken über die folgenden drei Fragen nach:

* Was ist Informatik?
* Wie wird Informatik (oder werden Informatikfähigkeiten) in der Schule eingesetzt?
* Wie wird Informatik am Arbeitsplatz (in unterschiedlichen Jobs) eingesetzt?

1. Die Schülerinnen und Schüler führen sich wichtige Glossarbegriffe zu Gemüte, die ihnen dabei helfen werden, die Lernerfahrung richtig zu erfassen.

**Programmieraktivitäten (30–40 Minuten)**

Die Schülerinnen und Schüler beginnen ihrer Programmierreise am Portal zur Ephemera One, dem Hauptraumschiff des Instituts für extreme Zeitfehler. Sie sind Informatik-Fachkräfte, die von T.A.R.R.A sehnlichst erwartet werden – dem K.I.-Roboter, der das Schiff und die Zeit-Agenten verwaltet. Zeit-Agenten sind Roboter, die sich mittels Zeitreise an verschiedene Punkte in der Geschichte begeben, um mit Programmierung Zeitrisse zu beheben.

Die Schülerinnen und Schüler werden zu Beginn aufgefordert, sich hinsichtlich der Programmieraktivitäten für Blöcke oder für Python zu entscheiden. Danach müssen sie ihren Zeit-Agenten auswählen. Unsere Empfehlung für Programmierneulinge lautet Blöcke.

Nach der Auswahl der Zeit-Agenten müssen die Schülerinnen und Schüler zwei einführende Programmierherausforderungen erfüllen.

EINFÜHRENDE PROGRAMMIERHERAUSFORDERUNGEN

Herausforderung 1: Rufe deinen Zeit-Agenten herbei. Hole dir den T.A.L.K.-Kommunikationsgerät bei T.A.R.R.A. ab und rufe deinen Zeit-Agenten zu deinem Aufenthaltsort.

Herausforderung 2: Bewege den Zeit-Agenten. Bewege deinen Zeit-Agenten vorwärts, sodass er auf dem grünen Block zu stehen kommt.

ZEITVERLAUF-PROGRAMMIERHERAUSFORDERUNGEN

Die Schülerinnen und Schüler werden zur ersten Herausforderung, Big-Band-Jazz, geführt, bei der er sich um eine angeleitete Programmiererfahrung handelt. Für jede Programmieraktivität können die Schülerinnen und Schüler zwei Tipps erhalten, bevor die Aufgabe für sie gelöst wird. Nach jeder Programmiererfahrung folgt eine kurze Suche nach Hinweisen und Spuren, um den Zeit-Übeltäter zu identifizieren.

Nach Abschluss von "Zeitriss 1: Big-Band-Jazz" werden die Schülerinnen und Schüler zurück in die Hauptlobby teleportiert, wo sie ihre nächste Herausforderung auswählen. Die Schülerinnen und Schüler können frei aus den Zeitrissen auswählen, allerdings ist es empfehlenswert, dass sie den Zeitrissen folgen, die auf dem großen Bildschirm über dem Zeitverlauf angezeigt werden.

|  |  |
| --- | --- |
| **Zeitriss 1**  Big-Band-Jazz  (Das ist die angeleitete Herausforderung.) | Der große Jazz-Musiker hat seine geliebte Trompete verloren und muss stattdessen mit einem Kazoo vorliebnehmen! Programmiere deinen Zeit-Agenten so, dass er durch das Labyrinth geht und die Trompete des Musikers findet, um den Jazz zu retten. |
| **Zeitriss 2**  Die Pyramiden von Gizeh | Die Pyramiden von Gizeh sind jetzt Würfel!  Programmiere deinen Zeit-Agenten so, dass er bei der Gestaltung eines sicheren Bauwerks hilft, welches tausende Jahre überstehen kann: die Pyramide. |
| **Zeitriss 3**  Mondmission | Die Astronauten brauchen deine Hilfe. Durchlaufe mit deinem Zeit-Agenten ein Labyrinth und liefere die Berechnungen ab, um den Astronauten bei der Landung auf dem Mond zu helfen! |
| **Zeitriss 4**  Chinesische Mauer | Die Chinesische Mauer ist noch nicht wirklich beeindruckend ... Tatsächlich ist sie noch sehr kurz geraten!  Die hungrigen Pandabären knabbern ständig am Bambusgerüst, was dazu führt, dass die Arbeiter nicht die großartige Chinesische Mauer errichten können, die sie im Sinn haben! Programmiere einen Bambusgarten, um die hungrigen Pandas abzulenken! |
| **Zeitriss 5**  Mona Lisa | Mona Lisa lächelt nicht mehr – sie ist vielmehr sehr verärgert. Der Zeit-Übeltäter ist zurückgereist und hat Mona Lisas Garten zertrampelt, weswegen ihr berühmtes Lächeln einem finsteren Blick gewichen ist. Programmiere deinen Zeit-Agenten so, dass er ihre Stimmung verbessert, indem er mehr Blumen pflanzt. |
| **Zeitriss 6**  Der erste Flug | Der Zeit-Übeltäter hat die Start- und Landebahn mit lauter Löchern versehen. Statt Flugzeuge haben wir nur noch Heißluftballons. Programmiere deinen Zeit-Agenten so, dass er die Start- und Landebahn repariert, damit die Erfinder ihren ersten Flug unternehmen können! |
| **Zeitriss 7**  Erste Programmiererin | Das erste Computerprogramm spielte ein Lied ab, aber der Zeit-Übeltäter hat den Code ruiniert. Repariere mit deinem Zeit-Agenten den Code, damit die Musik erklingen kann! |
| **Zeitriss 8**  Der beste Freund des Menschen | Der beste Freund des Menschen ist nicht mehr der Hund, sondern ein Bär! Lass deinen Zeit-Agenten sich mit einem Welpen anfreunden und ihn zurück zu den Menschen führen, damit Mensch und Hund wieder beste Freunde werden. |
| **Zeitriss 9**  Urzeit-Puzzle | Oh nein! Der Zeit-Übeltäter ist zurückgereist und hat einige Fossilien gestohlen, sodass der Brachiosaurus jetzt einen ganz kurzen Hals hat. Setzte deinen Zeit-Agenten ein, um die gestohlenen Fossilien zu ersetzen, damit die Paläontologinnen und Paläontologen den Brachiosaurus mit einem langen Hals nachbilden können! |
| **Zeitriss 10**  Entdeckung der Elemente | Der Zeit-Übeltäter ist in das Forschungslabor eingeschlichen und hat einige der Elemente versteckt. Finde mit deinem Zeit-Agenten die versteckten Elemente, damit die Wissenschaftlerin die großartige Entdeckung machen kann! |

Wenn die Schülerinnen und Schüler drei Programmieraktivitäten abgeschlossen haben, werden sie die erste Spielschleife beendet und den Zeit-Übeltäter identifiziert haben. Danach können sie ihre Programmieraktivitäten erweitern, indem sie einen neuen Zeit-Agenten auswählen und weitere Zeitrisse zu beheben versuchen.

**Bewertung:**

* Welcher Teil der Stunde des Codes hat dir am besten gefallen?
* Was war die größte Herausforderung in der Stunde des Codes?
* Wie hast du heute Programmierfähigkeiten eingesetzt?
* Nenne bitte eine neue Sache, die du heute gelernt hast.
* Warum ist Informatik für alle Menschen wichtig?
* Würdest du Minecraft: Education Edition gerne wieder spielen?

**Unterschiedliche Formen:**

Um den Zugang der Schülerinnen und Schüler zur Stunde des Codes zu erleichtern, gibt es drei unterschiedliche Unterrichtsformen für sie: in der Klasse mit einer leitenden Lehrkraft, in der Klasse in Form von Eigenarbeit und in einer (virtuellen) Remote-Lernform. Alle diese Arten weisen unterschiedliche Stufen der Unterstützung durch Lehrkräfte und der Anpassung für den Erfolg der Schülerinnen und Schüler sowie das Mitmachen bei der Stunde des Codes auf.

**Lehrkraft-Leitung (Frontalunterricht)**

***Diese Unterrichtsform bietet den Schülerinnen und Schülern den höchsten Grad an Unterstützung durch die Lehrkraft***. Als Lehrkraft gibst du ihnen genaue Anweisungen und Modelle, bevor du sie selbstständig an den Programmieraktivitäten in der Stunde des Codes arbeiten lässt. Du wirst die ["Stunde des Codes"-Präsentation](https://aka.ms/HOC2021Presentation) verwenden, um dich sowie deine Schülerinnen und Schüler durch die ersten Schritte im Spielen, die Programmierung und die Rückschau auf den Unterricht leiten zu lassen.

**Eigenarbeit**

***Diese Unterrichtsform bietet den Schülerinnen und Schülern den niedrigsten Grad an Unterstützung durch die Lehrkraft***. Als Lehrkraft sorgst du dafür, dass sich die Schülerinnen und Schüler auf der "Minecraft: Education Edition"-Plattform anmelden können. Du stellst ihnen die "Stunde des Codes"-Präsentation zur Verfügung, die sie bei ihrem eigenen Spielerlebnis, dessen Tempo sie selbst bestimmen, unterstützen wird. Die Präsentation gibt den Schülerinnen und Schülern genaue Anweisungen, wie sie an der Stunde des Codes teilnehmen können. Darüber hinaus solltest du den Schülerinnen und Schülern das visuelle Glossar zukommen lassen, das du am Ende dieses Leitfadens findest. Empfehlenswert ist es, dass du diese Materialien deinen Schülerinnen und Schülern auf einem digitalen Medium deiner Wahl oder in Form von Papierausdrucken zur Verfügung stellst, sodass sie ohne Verzögerung verwendet werden können.

**Fernunterricht**

Diese Unterrichtsform kann synchron oder asynchron stattfinden.

***Wenn du die Schülerinnen und Schüler in einer synchronen virtuellen (Live-)Sitzung mitmachen lässt***, dann läuft die Sitzung wie im Frontalunterricht unter Lehrkraft-Leitung ab. Als Lehrkraft gibst du ihnen genaue Anweisungen und Modelle, bevor du sie selbstständig an den Programmieraktivitäten in der Stunde des Codes arbeiten lässt. Du wirst die "Stunde des Codes"-Präsentation verwenden, um dich sowie deine Schülerinnen und Schüler durch die ersten Schritte im Spiel, die Programmierung und die Rückschau auf den Unterricht leiten zu lassen. Vergewissere dich, dass deine virtuelle Kommunikationsplattform (wie z. B. Microsoft Teams) in der Lage ist, Inhalte zu teilen, und dass dein Ton eingeschaltet ist.

***Wenn deine Schülerinnen und Schüler an einer asynchronen virtuellen Session (also nicht live) teilnehmen,*** dann läuft die Sitzung wie in Eigenarbeit in der Klasse ab. Als Lehrkraft sorgst du dafür, dass die Schülerinnen und Schüler vor dem Unterricht Zugriff auf die "Minecraft: Education Edition"-Plattform haben. Du musst ihnen zudem die "Stunde des Codes"-Präsentation zur Verfügung stellen, die sie bei ihrem eigenen Spielerlebnis, dessen Tempo sie selbst bestimmen, unterstützen wird. Die Präsentation gibt den Schülerinnen und Schülern genaue Anweisungen, wie sie an der Stunde des Codes teilnehmen können. Darüber hinaus solltest du den Schülerinnen und Schülern das visuelle Glossar zukommen lassen, das du am Ende dieses Leitfadens findest. Empfehlenswert ist es, dass du diese Materialien (Präsentation und Glossar) deinen Schülerinnen und Schülern auf einem digitalen Medium deiner Wahl zur Verfügung stellst (Beispiele: Microsoft-Teams-Kanal, OneNote-Notizbuch, dein Lernmanagement-System usw.).

Wenn du diese Lektion in Form von Fernunterricht abhalten willst, solltest du diese Tipps beherzigen:

1. Diese Lektion ist nicht als Mehrspieler-Erfahrung konzipiert. Alle Schülerinnen und Schüler sollten in ihrer eigenen Version der Welt arbeiten.

2. Teile die Schülerinnen und Schüler in Paare oder kleine Gruppen in Nebenräumen ein, damit sie sich gegenseitig helfen können, während sie sich an den Programmierlösungen versuchen.

3. Da die Schülerinnen und Schüler mit hoher Wahrscheinlichkeit unterschiedliche Grade an Vorwissen bezüglich Minecraft und Programmieren vorweisen werden, könnte es hilfreich sein, Gruppenleiterinnen bzw. Gruppenleiter unter ihnen festzulegen, die ihren Mitschülerinnen und Mitschülern zur Hand gehen können, falls diese Problem haben sollten.

Weitere Informationen über Fernunterrichtsmöglichkeiten in Minecraft: Education Edition erhältst du unter <https://aka.ms/remote-learning-kit>