A picture containing toy

Description automatically generated

**LEKTIONSPLAN**

**Titel:** Hour of Code 2021 (TimeCraft)

**Alder:** 7+

**Emneområder:** datalogi, matematik, sprog, historie, design, naturfag, teknik

**Det 21. århundredes kompetencer:**kreativitet, kritisk tænkning, problemløsning

**Understøttede sprog:** engelsk (USA), engelsk (UK), tysk, spansk (Spanien), spansk (Mexico), fransk (Frankrig), fransk (Canada), italiensk, japansk, koreansk, portugisisk (Brasilien), portugisisk (Portugal), russisk, kinesisk (forenklet), kinesisk (traditionelt), tyrkisk, polsk, indonesisk, ukrainsk, græsk, bulgarsk, ungarsk, finsk, dansk, svensk, norsk, slovakisk, tjekkisk, nederlandsk

**Multiplayer/singleplayer:** singleplayer

**Nødvendig erfaring med Minecraft (underviser):** begynder

**Lektionsbeskrivelse:** Underholdende og kreativ introduktion til datalogi. Tag på en rejse i tiden, hvor du bruger blokbaseret kode eller Python til at reparere mystiske brud på jordens tidslinje. Du er nødt til at finde spor og bruge kritisk tænkning for at afsløre, hvem der har pillet ved tiden. Du får også mulighed for at kigge nærmere på, hvordan datalogi er integreret i forskellige jobs og personlige interesser.

**Samlet tidsforbrug:** 1 time

**Undervisningsstandarder:**

|  |  |  |
| --- | --- | --- |
| Standarder | | |
| USA | Australien | UK – England |
| Jeg kan opstille modeller for dagligdags handlinger ved at oprette algoritmer, der kan udføre opgaver  **1A-AP-08** | Jeg kan følge, beskrive og vise sekvenser af trin og beslutninger (algoritmer), der kan løse simple problemer  **ACTDIP004** | Jeg kan skrive og debugge simple programmer  **Key Stage 1** |
| Jeg kan udvikle programmer med sekvenser og simple løkker, der kan udtrykke idéer og løse problemer  **1A-AP-10** | Jeg kan definere simple problemer og derefter beskrive og følge sekvenser af trin og beslutninger (algoritmer), der kan løse dem  **ACTDIP010** | Jeg ved, hvad algoritmer er, hvordan algoritmer er implementeret som programmer på digitale enheder, og at programmer kører ved at følge præcise og utvetydige instruktioner  **Key Stage 1** |
| Jeg kan debugge (identificere og rette fejl) algoritmer og programmer, der indeholde sekvenser og simple løkker  **1A-AP-14** | Jeg kan designe, modificere og følge simple algoritmer, der involverer sekvenser af trin, forgreninger og iteration (gentagelse)  **ACTDIP019** | Jeg kan bruge logisk tænkning til at forudse adfærden i simple programmer  **Key Stage 1** |
| Jeg kan sammenligne, hvordan mennesker lever og arbejder før og efter implementering eller indførelse af ny computerteknologi  **1A-IC-16** | Jeg kan designe algoritmer i diagrammer og almindelig tekst og følge algoritmer, så jeg kan forudse output for et givet input og identificere fejl  **ACTDIP029** | Jeg kan bruge logisk tænkning til at forklare, hvordan simple algoritmer fungerer, og til at registrere og rette fejl i algoritmer og programmer  **Key Stage 2** |
| Jeg kan skrive programmer, der inkluderer sekvenser, hændelser, løkker og betingelser  **1B-AP-10** | Jeg kan definere simple problemer og derefter beskrive og følge sekvenser af trin og beslutninger (algoritmer), der kan løse dem  **ACTDIP01** | Jeg kan bruge sekvenser, selektion og gentagelser i programmer og arbejde med variabler og forskellige former for input og output  **Key Stage 2** |
| Jeg kan sammenligne og forbedre flere algoritmer til samme opgave og afgøre, hvilken der er mest hensigtsmæssig  **1B-AP-08** | Jeg kan implementere simple digitale løsninger som visuelle programmer med algoritmer, der involverer forgrening (beslutninger) og brugerinput  **ACTDIP011** | Jeg kan designe, skrive og debugge programmer, der udfører konkrete opgaver, inklusive kontrol og simulation af fysiske systemer og problemløsning ved dekomponering i mindre dele  **Key Stage 2** |
| Jeg kan debugge (identificere og rette fejl) algoritmer og programmer, så de kører som tilsigtet  **1B-AP-15** | Jeg kan implementere digitale løsninger som simple visuelle programmer, der involverer forgrening, iteration (gentagelse) og brugerinput  **ACTDIP020** | Jeg kan forstå adskillige nøglealgoritmer, der afspejler computational thinking, og bruge logisk tænkning til at sammenligne anvendeligheden af alternative algoritmer til samme problem  **Key Stage 3** |
| Jeg kan redegøre for computerteknologier, der har forandret verden, og hvordan de pågældende teknologier påvirkes af kulturelle forhold  **1B-IC-20** | Jeg kan implementere og modificere programmer med brugergrænseflader, der involverer forgrening, iteration og funktioner i et generelt programmeringssprog  **ACTDIP030** | Jeg kan bruge to eller flere programmeringssprog, hvoraf mindst ét er tekstbaseret, til at løse forskellige beregningsmæssige problemer, gøre relevant brug af datastrukturer og designe og udvikle modulære programmer, der anvender procedurer eller funktioner  **Key Stage 3** |
| Jeg kan bruge flowdiagrammer og/eller pseudokode til at håndtere komplekse problemer som algoritmer    **2-AP-10** | Jeg kan indsamle, få adgang til og præsentere forskellige typer data ved at bruge simpel software til at generere information og løse problemer  **ACTDIP009** |  |
| Jeg kan designe og iterativt udvikle programmer, der kombinerer kontrolstrukturer, inklusive indlejrede løkker og sammensatte betingelser  **2-AP-12** |  |  |
| Jeg kan dekomponere problemer og underproblemer i mindre dele for at lette design, implementering og evaluering af programmer  **2-AP-13** |  |  |

**Lærerforberedelse:**

* Installer Minecraft: Education Edition via <https://aka.ms/HourofCode2021>
* Se [introvideoen](https://aka.ms/HOC2021Trailer) om årets Hour of Code-tema
* Se [walkthrough-videoen](https://aka.ms/HOC2021Walkthrough) for at få en bedre forståelse af lektionen, og hvordan du bevæger dig rundt i Minecraft
* Download [underviserguiden](file:///C:\Users\laylahbulman\Downloads\aka.ms\HOC2021educatorguide) med alle de oplysninger, du skal bruge for at hjælpe dine elever til at få udbytte af Hour of Code, inklusive alle eksterne ressourcer og vejledningen til guidet læring, selvstændig læring og hybrid-/fjernundervisning
* Brug den elevrettede [præsentation](C:\\Users\\laylahbulman\\Downloads\\aka.ms\\HOC2021presentationslides) til trinvise instruktioner
* Download en kopi af [kodeløsningerne](https://aka.ms/HOC2021CodingSolutions)
* Mangler du svar på noget? Se [FAQ](https://aka.ms/HOC2021FAQ)

**Temaoversigt:**

Som datalog på Institut for monitorering af tidsfejl er det dit job at rette op på de mystiske tidsforskydninger, der dukker op i historien, og finde ud af, hvem (eller hvad) der forårsager dem.

Vil du hjælpe med at korrigere tidsforskydningerne og redde historien med dine kodemæssige superkræfter?

Kan du finde ud af, hvem eller hvad der er skyld i de vanvittige ændringer i historien?

For at gennemføre din mission i TimeCraft skal du:

* Rejse tilbage til afgørende tidspunkter i verdenshistorien
* Kode din tidsagent til at rette op på tidsforskydningerne
* Finde spor, der kan identificere skurken bag tidsforskydningerne

Opgaven kan løses med blokbaseret kode eller Python efter eget valg.

**Læringsmål:**

* Eleven har en forståelse for vigtigheden og fordelene ved datalogi i alle livets aspekter.
* Eleven kan analysere og løse problemer ved hjælp af algoritmisk tænkning og dekomponering.
* Eleven har benyttet programmeringskoncepter som sekvenser, hændelser, løkker og debugging i praksis.
* Eleven kan skrive kode til løsning af opgaver og problemer.
* Eleven har indblik i den brede vifte af karrieremuligheder, datalogien tilbyder.

**Elevaktiviteter:**

**Intro (5 minutter):**

1. Eleverne reflekterer over de følgende tre spørgsmål:

* Hvad er datalogi?
* Hvordan anvendes datalogi (eller datalogiske færdigheder) i skolen?
* Hvordan anvendes datalogi på arbejdspladsen (i forskellige jobs)?

1. Eleverne gennemgår nøgletermer, der kan give dem en bedre forståelse af læringsaktiviteten.

**Kodeaktiviteter (30-40 minutter)**

Elevernes rejse begynder ved portalen til Ephemera One – det primære rumskib på Institut for monitorering af tidsfejl. De er dataloger og længe ventet af TARRA, en kunstig intelligens med ansvar for rumskibet og de tidsagenter, der rejser i tiden for at rette op på tidsforskydninger i historien med kode.

Eleverne får valget mellem blokke og Python til deres kodeaktiviteter, inden de kan vælge deres tidsagent. Vi anbefaler, at begyndere lægger ud med blokke.

Når de har valgt deres tidsagent, fortsætter eleverne til to introopgaver.

INTROOPGAVER

Opgave 1: Tilkald din tidsagent. Tag imod din kommunikationsenhed fra TARRA, og tilkald din tidsagent til din position.

Opgave 2: Flyt din tidsagent. Flyt din tidsagent fremad, så den står på den grønne blok.

TIDSLINJE FOR KODEUDFORDRINGER

Eleverne sendes i retning af den første udfordring, Bigband Jazz, der er en guidet kodeopgave. Eleverne får to hints til at løse de forskellige kodeaktiviteter, inden de løses for dem. Efter alle kodeaktiviteter får eleverne mulighed for at søge efter spor, der kan afsløre skurken.

Når de har rettet op på den første tidsforskydning: Bigband Jazz, teleporteres eleverne tilbage til lobbyen, så de kan vælge en ny udfordring. Eleverne kan vælge mellem alle tidsforskydningerne, men vi anbefaler, at de retter dem i den rækkefølge, der vises på den store skærm over tidslinjen.

|  |  |
| --- | --- |
| **Tidsforskydning 1**  Bigband Jazz  (Guidet udfordring) | En stor jazzmusiker har mistet sin elskede trompet og erstattet den med en kazoo! Kod din tidsagent til at finde vej igennem labyrinten, så den kan hente musikerens trompet og redde jazzen. |
| **Tidsforskydning 2**  Pyramiderne i Giza | Pyramiderne i Giza er forvandlet til kuber!  Kod din tidsagent til at hjælpe designeren med at udvikle en sikker struktur, der kan holde i årtusinder ... pyramiden. |
| **Tidsforskydning 3**  Mission til månen | Astronauterne har brug for din hjælp! Send din tidsagent igennem en labyrint, så den kan aflevere de beregninger, astronauterne skal bruge for at kunne lande på månen! |
| **Tidsforskydning 4**  Den kinesiske mur | Den kinesiske mur er ikke særlig lang endnu ... Den er faktisk ret kort!  De sultne pandaer æder bambusstilladset, så arbejderne ikke kan bygge muren så lang og høj, som de vil. Kod en bambusskov, der kan aflede pandaernes opmærksomhed! |
| **Tidsforskydning 5**  Mona Lisa | Mona Lisa er holdt op med at smile – hun ser faktisk ret sur ud. Tidsskurken har trampet alle planterne ned i Mona Lisas have, så hun ikke længere har noget at smile af. Kod din tidsagent til at muntre hende op ved at plante flere blomster. |
| **Tidsforskydning 6**  Første flyvning | Skurken har gravet huller i landingsbanen. I stedet for fly har vi nu kun luftballoner. Kod din tidsagent til at reparere landingsbanen, så opfinderne kan gå i luften! |
| **Tidsforskydning 7**  Første datalog | Det første computerprogram spillede en melodi, men skurken har ødelagt koden. Brug din tidsagent til at reparere koden, så musikken spiller! |
| **Tidsforskydning 8**  Menneskets bedste ven | Menneskets bedste ven er ikke længere en hund – det er en bjørn! Brug din tidsagent til at indgå venskab med en hvalp, og før hvalpen tilbage til menneskene, så de kan blive bedste venner igen. |
| **Tidsforskydning 9**  Palæontologisk puslespil | Åh nej! Skurken er rejst tilbage i tiden og har stjålet fossiler, så brachiosaurussen har fået en kort hals. Brug din tidsagent til at erstatte de stjålne fossiler, så palæontologerne kan bygge brachiosaurussen med en lang hals! |
| **Tidsforskydning 10**  Opdagelsens elementer | Skurken har sneget sig ind i laboratoriet og skjult flere af elementerne. Brug din tidsagent til at finde de skjulte elementer, så forskeren kan gøre den store opdagelse! |

Når eleverne har løst tre kodeopgaver, har de gennemført spillets første loop og identificeret skurken. De kan kode videre ved at vælge en ny tidsagent og rette op på yderligere tidsforskydninger.

**Evaluering:**

* Hvad var det bedste ved Hour of Code?
* Hvad var det mest udfordrende ved Hour of Code?
* Hvordan har du brugt datalogiske færdigheder i dag?
* Har du lært noget nyt i dag?
* Hvorfor er datalogi vigtig for alle?
* Har du lyst til at prøve Minecraft: Education Edition igen?

**Differentiering:**

Hour of Code kan benyttes på tre forskellige måder for at give så mange elever som muligt adgang til læringen: i klasseværelset med en lærer, i klasseværelset som selvstudie og som fjernundervisning. De respektive undervisningsmetoder kræver forskellige niveauer af lærerinvolvering og modificering for at give eleverne adgang til Hour of Code.

**Lærerundervisning (ansigt til ansigt)**

***Lærerundervisning giver eleverne den mest omfattende støtte***. Som lærer kan du give eleverne nøjagtige instruktioner og eksempler, inden de introduceres til deres egne kodeopgaver i Hour of Code. Du kan bruge [præsentationen i Hour of Code](https://aka.ms/HOC2021Presentation) til at guide dig selv og eleverne igennem såvel gameplay som kodning og evaluering.

**Selvstudie**

***Selvstudiet giver eleverne den mindst omfattende støtte***. Som lærer skal du først og fremmest sikre dig, at eleverne er i stand til at logge på Minecraft: Education Edition. Du skal også give dem adgang til præsentationen i Hour of Code, så de kan få hjælp til at klare sig igennem gameplayet i deres eget tempo. Præsentationen giver eleverne en nøjagtig vejledning i, hvordan de benytter Hour of Code. Du bør også give eleverne den visuelle ordliste, du finder sidst i underviserguiden. Vi anbefaler, at du udleverer materialerne til eleverne på et digitalt medium eller som udskrevne papirer, så eleverne har nem adgang til dem.

**Fjernlæring**

Fjernlæring kan foregå både synkront og asynkront.

***Hvis eleverne deltager i en synkron (live) virtuel session***, kan du undervise på samme måde, som hvis de var i klasseværelset (med lærerundervisning). Her kan du altså som lærer give eleverne nøjagtige instruktioner og eksempler, inden de introduceres til deres egne kodeopgaver i Hour of Code. Du kan bruge præsentationen i Hour of Code til at guide dig selv og eleverne igennem såvel gameplay som kodning og evaluering. Sørg for, at din virtuelle kommunikationsform (f.eks. Microsoft Teams) er klar til at dele indhold, og at du har tændt for lyden.

***Hvis eleverne deltager i en asynkron (ikke live) virtuel session***, kan du undervise på samme måde, som hvis de skulle i gang med et selvstudie. Som lærer skal du først og fremmest sikre dig, at eleverne har adgang til Minecraft: Education Edition, inden lektionen går i gang. Du skal også give dem adgang til præsentationen i Hour of Code, så de kan få hjælp til at klare sig igennem gameplayet i deres eget tempo. Præsentationen giver eleverne en nøjagtig vejledning i, hvordan de benytter Hour of Code. Du bør også give eleverne den visuelle ordliste, du finder sidst i underviserguiden. Vi anbefaler, at du udleverer materialerne (præsentationen og ordlisten) til eleverne i digital form (f.eks. Microsoft Teams-kanal, OneNote-notesbog, skolens læringssystem etc.).

Hvis du planlægger at afholde lektionen via fjernlæring, bør du overveje følgende:

1. Lektionen er ikke designet til multiplayer. Den enkelte elev skal arbejde i sin egen version af verdenen.

2. Hvis du opdeler eleverne i par eller mindre grupper og sætter dem sammen i breakout rooms, kan de hjælpe hinanden, når kodeopgaverne driller.

3. Da eleverne efter al sandsynlighed har forskellig erfaring med Minecraft og kodning, kan det være en hjælp at udnævne gruppeledere og give dem ansvaret for at hjælpe kammerater, der sidder fast.

Få flere oplysninger om fjernlæring i Minecraft: Education Edition på <https://aka.ms/remote-learning-kit>