Programmeren met GameMaker

Samenvatting

Leeftijd
12 - 14 jaar
Vaardigheden
abstraheren
algoritme en procedure
decompositie van het probleem
Totale tijdsduur
135minuten

Wil je graag je eigen computerspel maken? Tijdens deze workshop leren leerlingen om een

computerspel helemaal zelf te ontwerpen en zo te programmeren dat het spel werkt zoals ze zelf

willen. Ze leren een eerste, eenvoudig, spel maken met ruimte voor eigen ontwerp. Met deze basis

kunnen ze achteraf ook thuis verder werken aan hun droomspel!

Context

Programmeren en computationeel denken worden steeds belangrijker. Via GameMaker worden deze onontbeerlijke vaardigheden op een speelse manier aangeleerd.

Doelstellingen

VVKSO

Secundair onderwijs (A-stroom)

Kerncomponenten van techniek

De leerlingen kunnen

1 verschillende onderdelen en deelsystemen in een technisch systeem onderzoeken: de functies en de relaties ertussen toelichten;

9 met concrete voorbeelden uit techniek de rol illustreren van sturingen en regelsystemen in technische systemen;

10 technische systemen, het technisch proces, hulpmiddelen en keuzen herkennen in verschillende toepassingsgebieden uit de wereld van techniek waaronder energie, informatie en communicatie, constructie, transport en biochemie.

Techniek als menselijke activiteit

De leerlingen kunnen

17 hulpmiddelen kiezen en inzetten in functie van het doel en het gebruik;

20 technische systemen realiseren in verschillende toepassingsgebieden uit de wereld van techniek waaronder energie, informatie en communicatie, constructie, transport en biochemie.

Secundair onderwijs (B-stroom)

Kerncomponenten van techniek

De leerlingen kunnen:

1 verschillende onderdelen in een eenvoudig technisch systeem onderzoeken: de functies en de relaties ertussen toelichten;

2 onderzoeken hoe het komt dat een zelf gebruikt technisch systeem niet of slecht functioneert;

6 technische systemen, het technisch proces, hulpmiddelen en keuzen herkennen in verschillende verkenningsgebieden1 uit de wereld van techniek: informatie- en communicatietechniek, verzorging, voeding, bouw, elektriciteit, hout, metaal, kunststoffen, schilder- en grafische technieken, mode, tuinbouw.

Techniek als menselijke activiteit

De leerlingen kunnen

8 een eenvoudig ontwerp aanvullen uitgaande van de vooropgestelde vereisten;

18 technische systemen realiseren in verschillende verkenningsgebieden1 uit de wereld van techniek: informatie- en communicatietechniek, verzorging, voeding, bouw, elektriciteit, hout, metaal, kunststoffen, schilder- en grafische technieken, mode, tuinbouw.

Techniek en samenleving

De leerlingen kunnen

19 aan de hand van voorbeelden illustreren dat een technisch systeem ontworpen en gemaakt is om aan behoeften te voldoen;

Methodologie

Part	Beschrijving	Timing
1	**Inleiding en voorbereiding** In deze stap wordt de omgeving waarin het spel geprogrammeerd wordt, voorbereid. *Bijhorende schermafdrukken zijn terug te vinden bij de downloads.* **Inleiding** De lesgever laat een voorbeeldspel zien en vertelt dat de leerlingen dit gaan maken. Gedurende de hele workshop is het belangrijk het computationeel denken te benadrukken. De leerlingen moeten zelf het probleem opsplitsen in kleinere problemen om zo te komen tot de oplossing van het probleem. Tijdens de workshop worden de leerlingen door de lesgevers begeleid naar een trap maar de lesgevers gaan niet mee de trap op. Er is dus geen unieke oplossing van de oefeningen, elk kind kan op een andere manier	30 minuten

het probleem aanpakken.

Kort overlopen wat de leerlingen gaan maken, kom samen met de leerlingen tot volgende punten:

· Zelf een doolhof bouwen

· Hierin loopt een mannetje in rond

· Je kan punten verdienen en verliezen bij voorwerpen bommen

Software openen:

· Game Maker: logo met groene G

· Er komt een groen scherm open met reclame, dit mag weggeklikt worden.
(Dit is voor de betalende versie met onder andere 3D spelletjes, maar de gratis versie is ook heel uitgebreid)

Wat hebben we allemaal nodig in deze software om te kunnen bouwen? Begeleid leerlingen naar onderstaande antwoorden (bijv. wat zie je allemaal, wat is nog niet aanwezig in jouw 'wereld'?,...)

· Een mannetje om te lopen

· Muren/blokjes om het doolhof te bouwen

· Bommen

· Voorwerpen om punten te verdienen

· Een achtergrond

Deze 5 dingen gaan wij aanmaken: 1 background en 4 sprites

Sprites aanmaken

Laat leerlingen in het voorbeeld zelf zien dat sprites enkel afbeeldingen zijn en dat er daarmee alleen geen spel gebouwd kan worden.

Sprites zijn dus de afbeeldingen die we nodig gaan hebben.

· Klik bovenaan op de knop 'Create a sprite' (rood pacmanmannetje)

· Verander de naam bij 'Name' in bijvoorbeeld 'Mannetje'.

· Klik op 'Load Sprite' om een sprite te kiezen op de pc.
Deze staan in de Drive bij TWA Computers > Workshops > Game Maker > de aangemaakte groepsmap > Sprites
Hier kunnen de leerlingen zelf een sprite kiezen.

· De rest is al juist ingesteld, dus 'OK' drukken.

Op deze manier kan je 1 of 2 sprites samen maken met de leerlingen, de rest doen ze zelf.

De 4 sprites die aangemaakt moeten worden:

· ‘Mannetje’: Een mannetje om te lopen

· ‘Blokje’: Muren/blokjes om het doolhof te bouwen

· ‘Bommetje’: Bommen, waarbij je punten af krijgt

· ‘Voorwerpje’: Voorwerpen om punten te verdienen

Belangrijk is dat ze de verkleinwoorden gebruiken (zie later)

Als deze af zijn kan er links op het scherm nagekeken worden of in het mapje ‘Sprites’ de 4 aangemaakte sprites zitten met juiste afbeelding.
Indien er een afbeelding vergeten is: dubbelklikken op de sprite en opnieuw kiezen voor ‘load sprite’.
Indien er een sprite teveel gemaakt is: rechts klikken op de sprite en ‘delete’ kiezen.

Background aanmaken

Werkwijze identiek aan sprites aanmaken, alleen moet je nu bovenaan klikken op ‘Create a Background’

Room aanmaken

Deze stap kan ook omgewisseld worden met de volgende stap (objects aanmaken).
Als je aan de leerlingen vraag wat de volgende stap is, antwoorden ze meestal dat we nu alles moeten samenzetten. Dat is deze stap. Als ze antwoorden dat we nu die afbeeldingen (sprites) moeten gebruiken om echte voorwerpen te maken, kan je kiezen om eerst die stap te doen. Op die manier geef je hen nog meer het gevoel dat ze mee zijn.

Het aanmaken van de room is eigenlijk een ruimte maken waarin gespeeld wordt.
Wij maken voorlopig 1 room, dit wordt dan ook het eerste level.

· Klik bovenaan op ‘Create a room’ (wit vierkantje)

· Namen instellen, doe je door op ‘Settings’ te klikken

o Name: Dit is de naam van de room, hier kan je best kiezen voor ‘Level 1’ (deze is niet zichtbaar tijdens het spel, vergelijkbaar met de namen van de sprites)

o Caption for the room: Dit is de titel die zichtbaar is tijdens het spel, dit kunnen de leerlingen dus kiezen.

· Achtergrond kiezen, doe je door op ‘Backgrounds’ te klikken. Hier heb je 2 keuzes:

o Bij ‘Color’ kan je de grijze kleur veranderen in een achtergrondkleur naar keuze.

o Onderaan kan je ‘no background’ vervangen door de eigen gekozen achtergrond door er op te klikken. Indien je deze kiest kan je ook kiezen om ‘stretch’ aan te duiden of niet. (de afbeelding wordt dan uitgerokken ofwel verschillende keren naast elkaar geplaatst indien hij te klein zou zijn)

Objects aanmaken

Aangezien de sprites enkel afbeeldingen zijn, moeten we hier eerst nog echte voorwerpen van maken om ze te kunnen plaatsen in de room en daarna te programmeren. Dit zijn de objects.
Van elke sprite moeten we dus een object maken.

· Klik bovenaan op ‘Create an object’ (blauwe bal)

· Bij ‘Name’ vul je bij het eerste object ‘Man’ in, dit is dus het object dat we maken van de sprite ‘Mannetje’. We mogen niet ‘Mannetje’ kiezen omdat we namen nooit 2 keer mogen gebruiken. Daarom dat we eerder de verkleinwoorden gebruikten voor de sprites.

· Net daaronder veranderen we ‘no sprite’ naar (in dit geval) ‘Mannetje’. Bij deze is ons eerder gemaakte sprite nu gelinkt aan dit object, waardoor het dit uiterlijk krijgt.

· We moeten nog 2 dingen instellen alvorens op te slaan:

o Het vakje ‘Visible’ staat automatisch aangevinkt. Dit wil zeggen dat dit object zichtbaar zal zijn. Dit is bij alle 4 objecten nodig, dus dit mag blijven staan. Je zou dit wel kunnen uitvinken als je bijvoorbeeld een geheime muur wilt bouwen.

o Het vakje ‘Solide’ staat automatisch uitgevinkt. Dit wil zeggen dat het object niet vast is. Hiermee bedoelen we dat dit object nooit in aanraking kan komen met ander objecten (botsen tegen een muur, een bom aanraken,…) Onze 4 objecten moeten dus wel ‘Solide’ zijn, dus we mogen niet vergeten dit aan te vinken.

Opnieuw kan er 1 object samen gemaakt word en de andere alleen.

De 4 objecten die gemaakt moeten worden zijn dus

· Man met als sprite mannetje

· Bom met als sprite bommetje

· Voorwerp met als sprite voorwerpje

· Blok met als sprite blokje

Als dit af is, kan opnieuw nagekeken worden of alles klopt door links in de mapjes te kijken of er nu 10 onderdelen aangemaakt zijn (4 sprites, 1 background, 4 objects en 1 room).

Als dit klopt, kan het spel ook al een eerste keer opgeslagen worden:

· Links bovenaan kiezen voor ‘File’

· Dan kiezen voor ‘Save as’

· Dan naar het aangemaakte groepsmapje gaan:
Bureablad > Drive > Workshops > aangemaakte groepsmap

· De leerlingen geven hun spel hun naam. Handig is om te vragen hun voornaam er in te zetten.

· De leerlingen slaan op. De lesgever slaat nog onmiddellijk niet op, op die manier verschijnen de spelletjes van de leerlingen op het grote scherm. Zo kunnen ze nagaan of alle spelletjes goed zijn opgeslagen.

Objecten in room plaatsen

Nu alle onderdelen gemaakt zijn, kunnen we alles gaan samenvoegen: we gaan level 1 bouwen door alle objecten in onze room te plaatsen.

· We dubbelklikken op 'Level 1' om terug in onze room te zitten

· We gaan ons beeld mooi zetten om makkelijker te kunnen bouwen

o Door te scrollen kan je het speelveld kleiner of groter zetten

o Door te slepen of te klikken in het groot vak links onder kan je het speelveld verplaatsen

o Belangrijk is dat alle vakjes van het raster goed zichtbaar zijn.

· Dan klikken we bovenaan (binnen het scherm van de room) op objects (als je daar nog niet zat) om de objecten te gaan plaatsen:

o Als je klikt op het veld, zie je dat automatisch het laatst aangemaakt object verschijnt. Doe je dit per ongelijk, kan je het terug verwijderen door rechts te klikken en te kiezen voor delete. Als je meerdere keren geklikt hebt in het zelfde vakje, moet je ook meerdere keren deleten om alles weg te krijgen. (of kiezen voor 'Delete all objects under cursor').

o Je kan van object wisselen door links op het object te klikken en te kiezen welke je wilt plaatsen.

o We beginnen met het plaatsen van de muren:

§ Alle buitenste vakjes gaan we vullen met een object 'Blok', om een buitenmuur te krijgen zodat de andere objecten niet uit het speelveld kunnen uiteindelijk.

§ Daarna kunnen we de binnenmuren bouwen door daar ook objecten 'Blok' te plaatsen.

§ Let wel dat je tussen 2 muren altijd 2 vakjes openlaat. Dit omdat een object meestal 1 vakje breed is en er anders heel moeilijk tussen geraakt.

o Daarna plaatsen we ook 1 keer het object 'Man', 3 keer 'Voorwerp' en 5 keer 'Bom'. Dit worden de startposities van deze objecten.

Spel testen

Als ons spel volledig gebouwd is, gaan we het al een eerste keer 'testen'.

Dit doen we door bovenaan op 'Run the game' (groene driehoek, play symbool) te klikken.

Het spel opent zich, maar kan nog niet gespeeld worden.
Want we hebben nog niets geprogrammeerd.

2	**Programmeren spel** **Inleiding** Om het spel ook te laten werken, moeten we verschillende handelingen gaan programmeren: · Ons mannetje moet kunnen bewegen in het spel om zich te verplaatsen · Als we bij een muur komen moeten we daar niet doorlopen, maar terugkaatsen · Als we bij een voorwerp komen moeten er 2 dingen gebeuren: o We krijgen punten bij o Het voorwerp verplaatst, zodat we dit opnieuw kunnen gaan zoeken · Als we bij een bom komen, moeten er ook 2 dingen gebeuren: o We krijgen punten af o Niet de bom, maar wijzelf verplaatsen Deze 4 handelingen gaan we een voor een programmeren. Na elke stap kan er getest worden of het werkt. Het programmeren gebeurt bij de objecten, daar vinden we 2 grote kolommen terug: 'Events' en 'Action', belangrijk is dat de leerlingen het verschil tussen deze 2 goed kennen alvorens te programmeren: · 'Event': dit zijn gebeurtenissen in het spel, de momenten waarvoor je iets wilt programmeren. Als je op het knopje 'pijl naar boven' drukt, wil je bijvoorbeeld dat er iets gaat gebeuren. Het drukken op dat knopje is in dit voorbeeld dan het event. Een ander voorbeeld van een event kan de botsing van de man met een blok zijn. Want je wilt dat daar iets op volgt, dit is de gebeurtenis (het event) waarvoor we iets willen programmeren. · 'Actions': dit zijn de acties die volgen op de event. Dit moet er dus geprogrammeerd worden om te gebeuren nadat een event plaatsvindt. Als je op een pijltje gedrukt hebt, wil je bijvoorbeeld dat de man ook naar die richting beweegt. Het bewegen naar die richting is dan de actie. Kort samengevat: ALS …. gebeurt (=event), DAN moet …. gebeuren (=actie). Om te bepalen bij welk object we programmeren, moeten we de vraag stellen welk object de actie (niet het event!) moet uitvoeren.	15 minuten
3	**Programmeren spel** **Bewegen in het spel** We willen dat de man in 1 van de 4 richtingen beweegt als we op 1 van de 4 pijltjes drukken, we hebben dus 4 events en 4 acties nodig: · Event: het drukken op een knop op het toetsenbord · Actie: de man beweegt in die richting	20 minuten

· Gekozen object: de man, aangezien hij de actie (het bewegen) uitvoert

We doen 1 event samen:

· Dubbelklikken op ‘man’ om dit object te openen

· Drukken op ‘new event’, onderaan de kolom ‘events’

· Een volgend scherm vraagt welk soort event, we kiezen voor ‘keyboard’ omdat dit een event is waarbij het toetsenbord gebruiken

· Nog een volgend scherm vraagt welke toets op het toetsenbord, we kiezen ‘up’ om als eerste de beweging naar boven te programmeren

Bij dit event hoort natuurlijk ook een actie:

· Helemaal rechts staan allemaal blokjes om te programmeren, wij kiezen voor het eerste: ‘Move fixed’ (pijltjes naar verschillende richtingen).

· Dit blokje slepen we naar de kolom ‘actions’

· Er verschijnt een scherm om dit in te stellen:

o Bij ‘Applies to’ staat ‘Self’ aangeduid, dit wil zeggen dat het toegepast wordt op dit object zelf. Dit is bij ons altijd van toepassing, dus hier hoeven we niets te veranderen

o Daaronder kunnen de we richting aanduiden bij ‘directions’, we kiezen voor het pijltje naar boven

o Ook kunnen we hieronder de ‘speed’ aanduiden, dit is de snelheid. 5 is een gemiddelde snelheid, de leerlingen kunnen zelf kiezen.

o Ten slot is er nog een vakje ‘relative’, dat je kan aanduiden. Dit hoeven we hier niet te doen, maar komt later wel terug

Op deze manier kunnen de leerlingen ook de andere 3 events en acties aanmaken: links (left), rechts (right) en beneden (down).

Als ze dit gaan testen, gaan ze zien dat je wel nog door muren en ander objecten loopt. Dit moeten we nog verder programmeren. Ook kunnen ze nu nagaan of hun gekozen snelheid goed was en eventueel aanpassen.

4	**Programmeren** **Terugkaatsen bij muren** Als ‘man’ bij een ‘blok’ (een muur dus) komt, moet hij terugkaatsen. · Event: de botsing tussen man en blok · Action: man kaatst terug · Gekozen object: man, want hij kaatst terug en voert dus de actie uit Dit event en actie maken we samen: · Open object ‘man’ · ‘new event’ · Nieuw scherm met vraag welk soort event: dit is een ander soort event, een botsing. We kiezen voor ‘Collision’ (2 pijlen die botsen). · Nieuw scherm met de vraag waarmee er gebotst moet worden: ‘blok’. · Ook een andere actie als daarnet: ‘Bounce’ (pijl die terugkaatst). · Hier hoeven we niets verder in te stellen. Dit kan opnieuw getest worden.	20 minuten
5	**Programmeren** **Acties bij gevonden voorwerpen** Als we bij het voorwerp komen, willen we dat er 2 dingen gebeuren: · Er komen punten bij · Het voorwerp verplaatst naar een willekeurige plaats zodat we er opnieuw achteraan kunnen gaan We moeten nu dus 2 acties gaan programmeren bij 1 event! · Event: de botsing tussen man en voorwerp · Actions: o Punten bij o Voorwerp verplaats naar willekeurige positie · Gekozen object: voorwerp, aangezien hij de actie uitvoert. Dus niet man, hij voert geen actie uit in dit geval. Deze doen we opnieuw samen, dit is de laatste. Hierna gaan de leerlingen de acties bij de bom zelf programmeren. Als dit goed benadrukt wordt, weten de leerlingen dat ze bij deze goed moeten opletten, want dit is vergelijkbaar met de bom.	20 minuten

	· Het object ‘voorwerp’ openen · ‘new event’ · Opnieuw kiezen voor ‘collision’ · Kiezen voor botsing met ‘man’ (Dus niet ‘voorwerp’, want dan zeg je eigenlijk: “het voorwerp botst met het voorwerp”) · Als eerste actie hebben we het blokje ‘Jump to random’ (pijltje naar een vraagteken) nodig. Dit zorgt ervoor dat het object naar een willekeurige plaats springt. Hier hoeven we verder niets in te stellen. · De tweede actie is niet zichtbaar bij de blokjes die je rechts ziet. Hiervoor moeten we rechts eerst een ander tabblad aanklikken: ‘score’ · Hier kiezen we voor het eerste blokje, ‘set score’, om een score in te stellen. · In het scherm dat verschijnt kan je een nieuwe score intypen. Dit kunnen de leerlingen zelf kiezen. · Heel belangrijk is dat nu wel het vakje ‘relative’ wordt aangeduid. Dit zorgt ervoor dat de nieuwe score niet zomaar verandert in het gekozen getal, maar dat het wordt opgeteld met de vorige score. De nieuwe score is dus relatief ten opzichte van de vorige, er wordt vergeleken met wat de score was. (Eerder kozen we om dit niet aan te duiden bij de speed van beweging, omdat dat ervoor zou zorgen dat de man sneller en sneller zou gaan elke keer je op dat knopje drukt) Dit kan opnieuw getest worden, als een voorwerp bereikt wordt, verandert de score bovenaan het scherm (naast de naam van het spel) en verplaatst deze naar een andere plaats.	
6	**Programmeren** **Acties bij bom** Deze doen de leerlingen dus zelf, het is vergelijkbaar met het vorige. Tips die je kan geven: · Denk goed na wie de actie moet uitvoeren en waar je dus moet gaan programmeren · Om punten af te trekken heb je een negatief getal nodig, dus een min ervoor zetten	20 minuten

7	**Einde workshop en extra's** Extra’s Als je merkt dat sommige of alle leerlingen al volledig klaar zijn en er is nog tijd over, kan je volgende onderdelen kort aanhalen waardoor de leerlingen nog wat kunnen uitbreiden: · De room kan eventueel aangepast worden indien ze nodig vinden (muren anders zetten, meer bommen, andere startposities,…) · De room kan ook groter gemaakt worden. Dit doe je door bij settings de breedte en hoogte aan te passen. · Eventueel kunnen geluiden toegevoegd worden, hiervoor moet je nieuwe onderdelen aanmaken (sounds) en deze ook programmeren (in het tabblad ‘main 1’ vind je een blokje ‘play sound’) · Er kan een extra object aangemaakt worden met ander knoppen om te bewegen, zodat je met 2 tegelijk kan spelen op 1 toetsenbord. Einde van de workshop Op het einde van de workshop slaan de leerlingen hun spelletjes nogmaals goed op. Daarna krijgen ze de aangemaakte groepsmap mee naar huis. Indien ze zelf een USB-stick meebrengen kan je de map samen met hun kopiëren op hun USB-stick, anders kan je een e-mailadres vragen om het door te mailen via Wetransfer.	10 minuten

Organisatie

Materialen en voorbereiding

Computer met GameMaker (https://www.yoyogames.com/gamemaker)

Map aanmaken

Er wordt een mapje aangemaakt waarin alles staat wat de leerlingen nodig hebben en waar de leerlingen ook hun spelletjes in gaan opslaan. Deze mag gekopieerd worden en de naam aangepast naar de groep die komt.

In dit mapje zit/komt te zitten:

· Sprites

· Backgrounds

· Voorbeeld

· Software: Game Maker Installer

· Link naar website: Game Maker Handleiding

· De spelletjes van de leerlingen

Dit mapje krijgen de leerlingen ook achteraf mee naar huis (ofwel via USB-stick ofwel mailen via wetransfer)

Computers aanzetten en nakijken

Per kind wordt 1 pc aangezet en nagekeken of Google Drive nog is aangemeld en dus synchroniseert. (belangrijk voor het opslaan van de spelletjes)

Ook de lesgever pc en de beamer wordt aangezet.

Begeleiding