

WORKSHOP 5

In deze workshop leer je een platformgame te maken. Dat wil zeggen dat je hoofdpersootje van het ene naar het andere platform kan springen.

Verder leer je in deze workshop werken met zogenaamde *views*.

Een view is een deel van het speelveld. Een grote room kan onderverdeeld worden in zogenaamde views. Een view beweegt met het personage mee. Dat is met een platform wel zo prettig spelen.



INLEIDING

Platformspelen zijn zeer gangbaar, vooral op spelcomputers. In een platformspel kijk je vanaf de zijkant naar het gebeuren. De speler bestuurt een personage dat rondloopt in een wereld die bestaat uit platforms. Het personage kan over deze platforms lopen en van het ene naar het andere platform springen of vallen. Verder kan het in ladders of touwen klimmen om zich te verplaatsen.

Op de platforms moet het personage objecten verzamelen, vijanden vermijden of doden (door schieten of bovenop ze te springen), schakelaars kunnen omzetten om doorgangen te openen, enzovoorts.

In sommige platformspellen zie je het hele level tegelijk, maar in de meeste gevallen zie je slechts een deel van het level rondom het personage. In dat geval is het een extra uitdaging je weg te vinden.

DE BASIS

We beginnen met een heel simpel platformlevel. In ieder platformspel komen drie basisprojecten voor: een linker- en rechterkant van het personage en een blokje dat gebruikt wordt voor de platforms waar het personage op loopt en springt.



Verzamel deze benodigde sprites.

Het is natuurlijk het fraaist als dat hetzelfde personage is als in je doolhoflevels. Daartoe moet je wel nieuwe objecten aan de sprites van het personage koppelen.

Het blokje is solid zonder events en/of actions. Het personage is iets ingewikkelder.

Het is al eerder gemeld, maar zeker hier is het van groot belang.

Begin heel simpel.

Test iedere stap die je maakt.

Ga niet verder voordat hetgeen je hebt gedaan goed werkt.

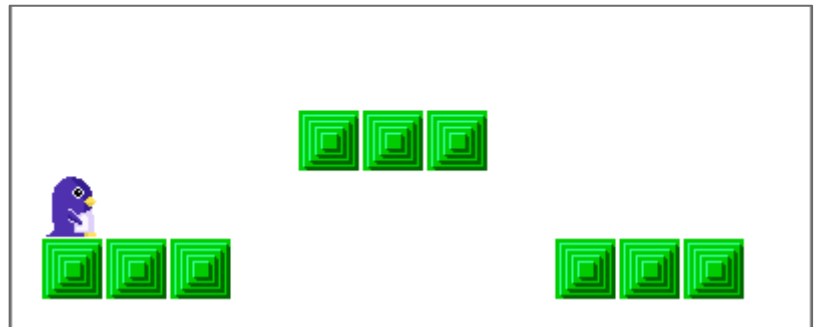
Bewegen en springen van het personage

Het probleem hier bij is dat het personage bovenop alle platforms moet kunnen lopen. Het mag er niet doorheen zakken. Het mag ook niet zoals in een doolhofspel in het luchtledige lopen. Het kan dus alleen horizontaal lopen. Het springt omhoog en valt omlaag. Als het personage valt dan moet het netjes landen op het volgende platform.



Maak om te beginnen een simpel level met een paar platforms.














Zet ergens op een platform de personage die naar rechts kijkt.



LOPEN

In tegenstelling tot het doolhofspel, waar het personage zich voortbeweegt met behulp van een bepaalde snelheid, springt het personage hier steeds 4 pixels vooruit, hetgeen nagenoeg hetzelfde effect geeft.

Instellingen personage-rechts

<p>Name: <input type="text" value="jump_right"/></p> <p>Sprite  <input type="text" value="jumpright"/> </p> <p><input type="button" value="New"/> <input type="button" value="Edit"/></p> <p><input checked="" type="checkbox"/> Visible <input checked="" type="checkbox"/> Solid</p> <p>Depth: <input type="text" value="0"/></p>	<p>Events:</p> <ul style="list-style-type: none"> Step tegel2 jump-basis <Left> <Up> <Right>	<p>Actions:</p> <ul style="list-style-type: none"> Start of a block Change sprite into jumpleft If a position is collision free Jump to position (-4,0) End of a block
---	---	---

- Als de linker pijltjestoets wordt ingedrukt moet de sprite veranderen in personage links.
- Als in de linker richting, op relatieve afstand (-4,0), de positie botsingvrij is ...
- ... dan moet de personage zich 4 pixels naar links bewegen.
Dus $x=-4$, en $y=0$, relatief aangevinkt.



Maak zelf deze instellingen en doe dat ook voor pijltjestoets rechts. Dan $x=4$ natuurlijk. **TESTEN VOOR DAT JE VERDER GAAT. ALS HET PERSONAGE NIET GOED LOOPT GA DAN NOG EENS ALLE INSTELLINGEN NA.**

SPRINGEN

Vervolgens moet het personage ook nog kunnen springen. Dat wil zeggen dat we een verticale beweging in moeten stellen.

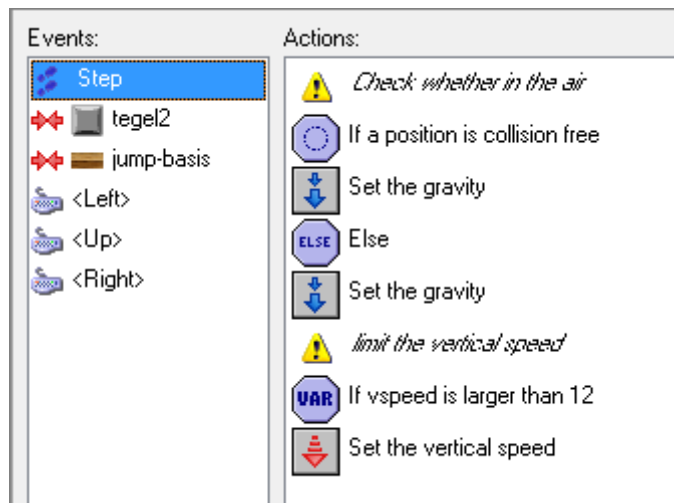
Om het personage te laten vallen hebben we zwaartekracht (*gravity*) nodig. Maar het moet stoppen met bewegen als het een platform of de grond raakt.

Verder moeten we een maximale valsnelheid instellen om te voorkomen dat het personage te snel beweegt. Dat ziet er niet mooi uit.

Controleren

Deze zaken controleren we in een step-event van het personage:

- of de positie onder het personage 'collision free' is. Dwz $(x,y) = (0,1)$. Er wordt dus steeds één pixel naar onder gekeken of dat vrij is.
- als dat zo is, dan 'vliegt' het personage door de lucht en zetten we de zwaartekracht op een positieve waarde (1) in richting (270 = naar beneden)
- anders ...
- zetten we de zwaartekracht op 0
- tevens controleren we de variabele 'vspeed' die de verticale snelheid aangeeft. Op deze manier limiteren we de verticale snelheid tot 12 ...
- als die hoger is dan 12
- zet dan terug naar 12



Landen

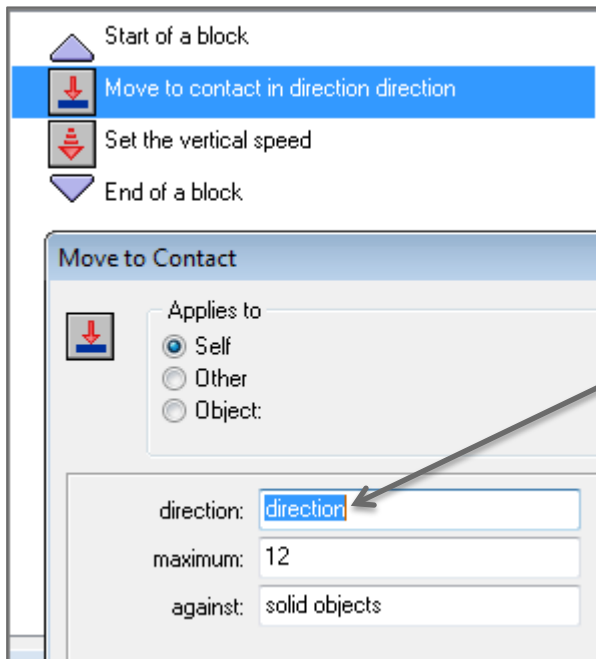
Nu moeten we nog netjes op een platform of op de vloer landen.

De landing vindt plaats op het moment dat het personage botst met het blokje van het platform. Bij deze collision-event zetten we de verticale beweging op 0

Dit kan er echter toe leiden dat het personage net een stukje boven de grond blijft hangen. (dit komt doordat het personage terug geplaatst wordt naar zijn positie net voor de botsing). Daarom willen we het personage verplaatsen naar precies die positie waar de botsing plaatsvond. Hiervoor bestaat een action in GameMaker: **'Move to Contact'**

Met deze action kun je de instantie van een object (in dit geval het personage) in een bepaalde richting bewegen totdat een contactpositie met een ander object is bereikt.

Als er al sprake is van een collision op de huidige positie dan wordt de instantie niet verplaatst.

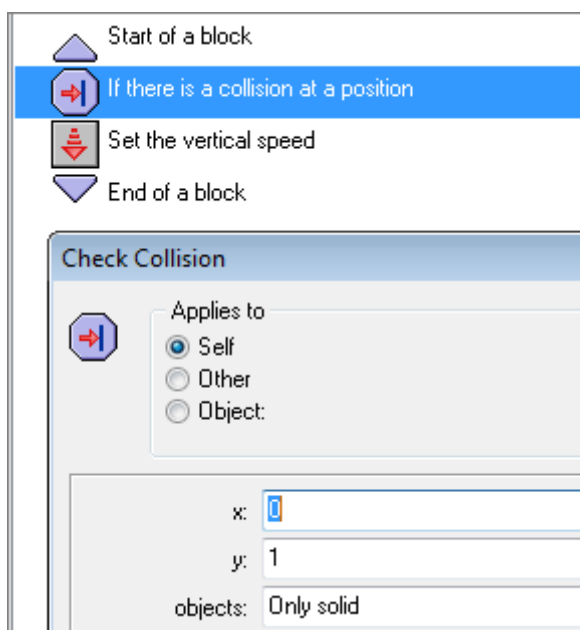


Zo niet, dan wordt de instantie tot stilstand gebracht juist voordat de botsing plaatsvindt. Je kunt de richting instellen maar ook de maximale snelheid van de beweging. Tevens kun je instellen of er sprake moet zijn van een 'solid object' of dat het om alle objecten gaat.

Als richting geven we de variabele richting aan die overeen komt met de huidige richting van de instantie (personage dus)

Als maximale afstand stellen we hier 12 in (hoewel dat hier niet echt nodig is)

Tot slot moeten we er voor zorgen dat het personage springt als we de **<Up>** pijltjestoets wordt ingedrukt. Dit mag echter alleen gebeuren als het personage vaste grond onder de voetjes heeft. Dus platform of grond.



We testen dus eerst of de positie onder het personage een collision vormt ($y=1$, dus wordt er 1 pixel naar beneden gekeken of er ook echt vaste grond is) en als dat zo is, bepalen we de snelheid op -10. (omhoog is negatief)
Relatief

LET OP !

Het kan zijn dat je een beetje moet spelen met de waarde voor de verticale snelheid en die van de zwaartekracht om de beweging te krijgen die naar je zin is.



Als al deze instellingen gedaan zijn is het tijd om te testen of je personage mooi springt en weer landt.

Als het niet goed werkt loop alles dan nog eens na. Voordat je verder gaat moet dit echt eerst heel goed werken, anders gaan de volgende activiteiten ook mis.

KLIMMEN, ladders en touwen.

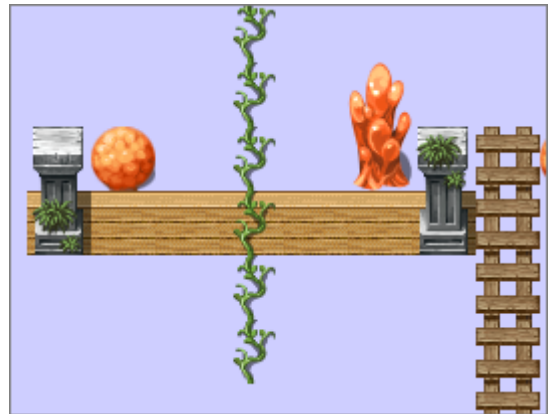
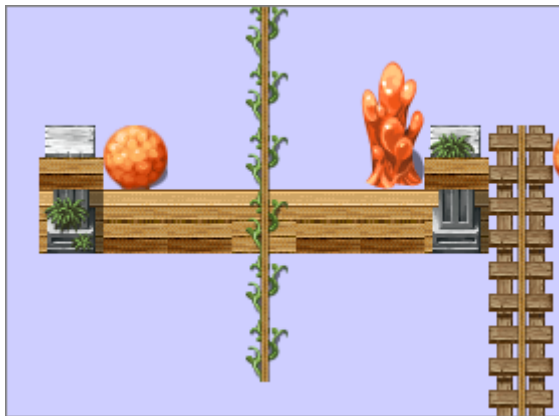
In een beetje platformspel kan het personage natuurlijk op ladders of in touwen klimmen om van het ene naar het andere platform te gaan.

Eerst het volgende:

Je personage moet natuurlijk wel iets hebben om in te klimmen, ladders, touwen, lianen, enz. Sprites zijn hier niet echt handig. Je krijgt dan namelijk het effect dat als je personage in collision komt met het klim-object hij dan gaat klimmen. Dat gebeurt dan vanaf de zijkant en kan er zelfs nog 'lucht' tussen het personage en het klim-object zitten. Dat ziet er heel raar uit.

Je kunt met tilesets een stukje achtergrond maken van klimdingen. Daaroverheen zet je dan *onzichtbare* en *niet solide* smalle verticale tegeltjes.

In dit geval met '**Precise collision checking**'. Als personage dan in aanraking komt met het klim-object lijkt het toch alsof het echt in de touwen klimt.



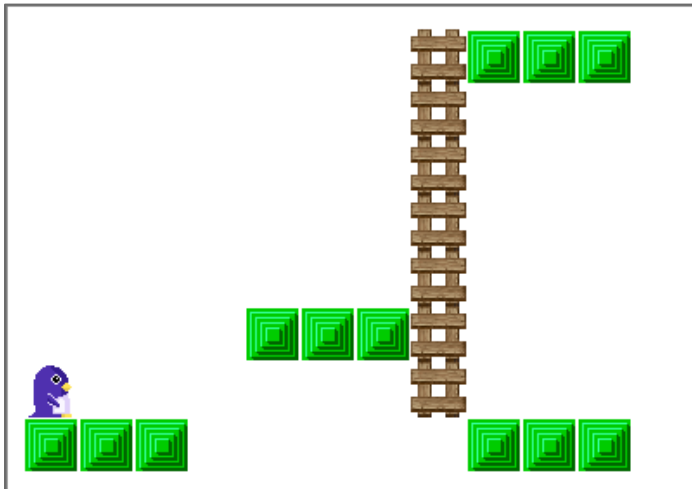
Links:

Achtergrondje van een platform, een lian en een ladder. Over het platform een onzichtbaar, solid horizontaaltegeltje waarover het personage kan lopen. Over de lian en de ladder een smal onzichtbaar en niet solide verticaal blokje waarin het personage kan klimmen.

Rechts:

Zo zal het er in de spelmode uitzien.

Het zal ook leuker zijn als je personage een achterkant heeft als hij klimt.



Dan nu het klimmen.

Voeg op bovenstaande wijze een ladder of touw toe.

Wanneer het personage niet in contact is met de ladder beweegt het zich normaal, maar wanneer het in aanraking komt met de ladder verandert dit.

Om te beginnen mag het personage niet vallen.

Daartoe moeten als **eerste** we in het step-event enkele actions toevoegen.

Events:	Actions:
Step	<ul style="list-style-type: none"> Check whether in the air If a position is collision free Set the gravity ELSE Else Set the gravity limit the vertical speed VAR If vspeed is larger than 12 Set the vertical speed Check whether on a ladder If there is an object at a position <ul style="list-style-type: none"> Start of a block Change sprite into klimmer Set the gravity VAR Set variable vspeed to 0 End of a block

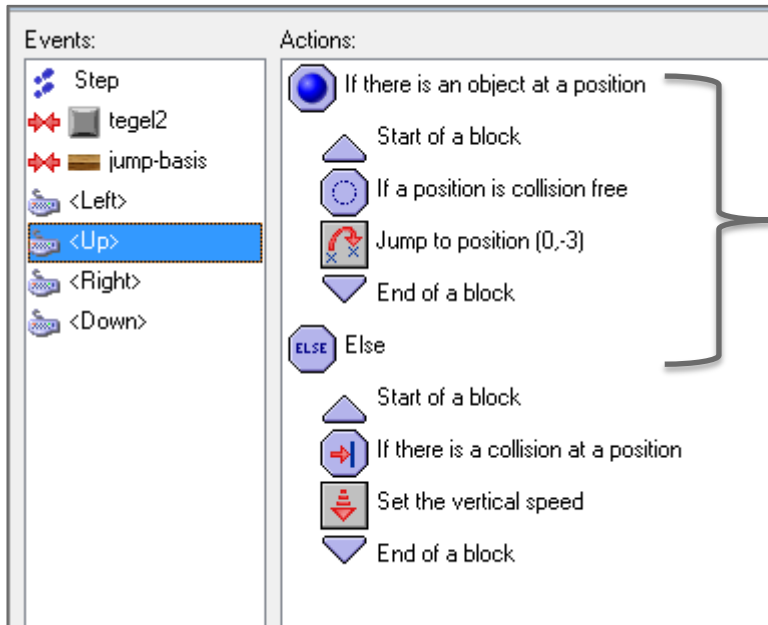
Als het klim-object op relatieve afstand (0,0) is

Zet dan de zwaartekracht op 0

Zet de verticale snelheid op 0

Verander eventueel in een klimpersonage

De tweede noodzakelijke wijziging betreft de event voor de pijtjestoets omhoog. Als het personage bij de ladder is moet deze pijtjestoets het personage omhoog laten klimmen in plaats van springen. Ook hiervoor zijn enkele aanvullende actions nodig.



Er wordt gecontroleerd of het personage in aanraking komt met het klim-object

Als er niets in de weg staat op bijvoorbeeld 3 pixels hoger, dan kan het personage steeds 3 pixels omhoog klimmen.

Het personage moet natuurlijk ook weer omlaag kunnen klimmen. Er moet dus ook een <Down> pijtjestoets komen.

De down-pijl krijgt dezelfde instellingen. Let op, alleen is naar beneden positief.



Laat jouw personage ook omhoog en omlaag klimmen.

Test of het goed werkt. Ga niet verder voordat het goed werkt.



VIJANDEN

In vorige workshops heb je al geleerd hoe je vijanden dood kunt laten gaan door bijvoorbeeld bommen of schieten. Dat kan in een platformspel ook allemaal.

In een platformspel kun je echter ook bovenop een vijand springen om het te vernietigen.

Om een monster te vernietigen door er bovenop te springen moeten we eerst in een 'collision event' van het personage met het monster nagaan of we het monster daadwerkelijk van boven raken. Daartoe moet de volgende test worden uitgevoerd:

Events:	Actions:	
<ul style="list-style-type: none"> Step tegel2 <li style="background-color: #e0e0e0;">spin-obj jump-basis <Left> <Up> <Right> <Down> 	<ul style="list-style-type: none"> ? If an expression is true <code>vspeed>0 && y <other.y+8</code> Start of a block <i>destroy the spider</i> Set the score relative to 10 Change instance into geplet End of a block ELSE Else Start of a block <i>destroy bindy if he has a spiderbite</i> Play sound aagh Set lives relative to -1 Restart the current room End of a block 	<p>(zonder spaties)</p> <p>De rest spreekt voor zich zelf. Daar zit niets nieuws bij.</p>

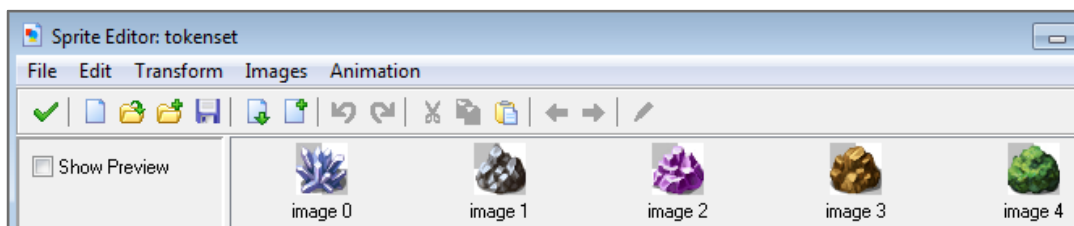
BELONINGEN

In de meeste platformspelen kun je punten verdienen door het verzamelen van allerlei objecten, ofwel tokens. Om daar variatie in aan te brengen kun je ook een set tokens maken.

In de sprite-editor kun je aan een sprite meerdere sprites toevoegen.

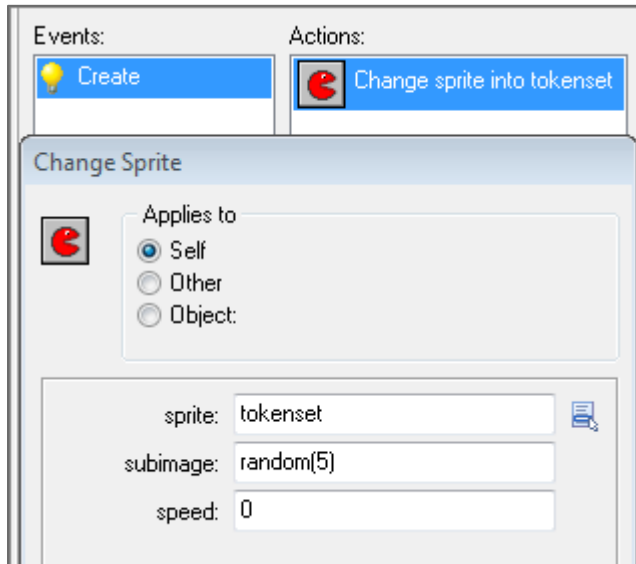
'Add from file' of 'Add from strip'. In een vorige workshop heb je geleerd hoe dat moet.

Onder zie je zo'n setje tokens. In dit geval verschillende kristallen.



De sprite bevat dus enkele sub-images. Vijf in dit geval, meer mag ook.
Zet de tokens in de room waar je ze wilt hebben. Je krijgt steeds dezelfde.

Maar iedere keer als het level in het spel wordt geopend wordt willekeurig een van de andere tokens neergezet. Zo kan het er toch iedere iets keer anders uitzien.




VIEWS

Tot nu toe werd tijdens het spelen de hele room getoond. Bij veel platformspelen is dat echter niet de bedoeling. In plaats daarvan wil je slechts het deel van de room rond het personage zichtbaar maken. Dit maakt het spel uitdagender omdat de speler zijn weg moet vinden langs de platforms. Je kunt dan beloningen verstoppen op moeilijk bereikbare plaatsen of je personage onverwachts in aanvaring laten komen met andere verrassende elementen.

In GameMaker is dit vrij makkelijk te verwezenlijken.

- Creëer om te beginnen een grote room. Bijvoorbeeld 800x576 pixels.
- Klik vervolgens op tabblad 'Views'
- Zet een vinkje in het vakje 'Enable the use of views'
- Selecteer de eerste view
- Zet een vinkje bij 'visible when room starts'.



In het voorbeeld hierboven zie je een deel van een room van 800x576 px. De view is hier 400x288 px. Dat is het omliggende stukje in de room. Dit is het deel dat zichtbaar zal zijn als het spel start.

Omdat we de view het personage laten volgen is het niet nodig de positie van de view in te stellen.

Onderaan bepalen we welk object gevolgd moet worden. We willen voorkomen dat het personage te dicht bij de rand komt tijdens het spelen. Daarom kiezen we Hbor en Vbor 48 pixels. Nu is er tijdens het spelen nooit minder dan 48 pixels rondom het personage zichtbaar.

Dat wil zeggen dat als je dichterbij de rand komt tijdens het spelen je overgaat naar een aangrenzend deel. Ofwel de view verschuift 48 pixels.

Om een vloeiende beweging van de view te krijgen zetten we de max Hspeed en Vspeed van de view op 4.

Het is leuk dat we nu met een view kunnen spelen, maar het maakt het venster waarin de dingen zich afspelen nogal klein. Om dit te voorkomen kun je in de globale game settings bepalen dat het spel een fixed schaal moet hebben van bijvoorbeeld 150- of 200 %.

Kijk hier wel mee uit. Plaatjes worden dan 'opgeblazen.' Dat kan een lelijk grof effect geven. Je moet dus maar een beetje spelen met de instellingen.

TOT BESLUIT

In voorgaande workshops heb je allerlei (basis)technieken geleerd zoals scores in beeld, healthbar, schieten, bommen plaatsen en nog heel veel meer. Al deze dingen zijn ook in een platformspel toepasbaar.










In deze workshop heb je de basisprincipes van een platformspel geleerd.



Nu is het jouw beurt.

Je moet al deze technieken gebruiken en eigen ideeën ontwikkelen om een leuk spel te maken.

Hier komen nog enkele ideeën die je kunt gebruiken:

-  Verschillende monsters, stuiterende ballen, vijanden die schieten of vuurspuwen.
-  Sleutels die gevonden moeten worden om deuren of kisten te openen.
-  De mogelijkheid om bommen te plaatsen die monsters, muren (of per ongeluk het personage) kunnen uitschakelen.
-  Water waarin het personage kan zwemmen. Dit verandert alle bewegingen, verandert de zwaartekracht, geeft beperkte tijd tot de lucht op is, enz.
-  Muren en vloeren die op allerlei manieren vernietigd kunnen worden.
-  Trampolines om alsmaar hoger te kunnen springen.
-  Platforms die verdwijnen of verschijnen
-  Bewegende platforms. Liften
-  En wat je allemaal zelf nog kunt verzinnen.

SUCCES

