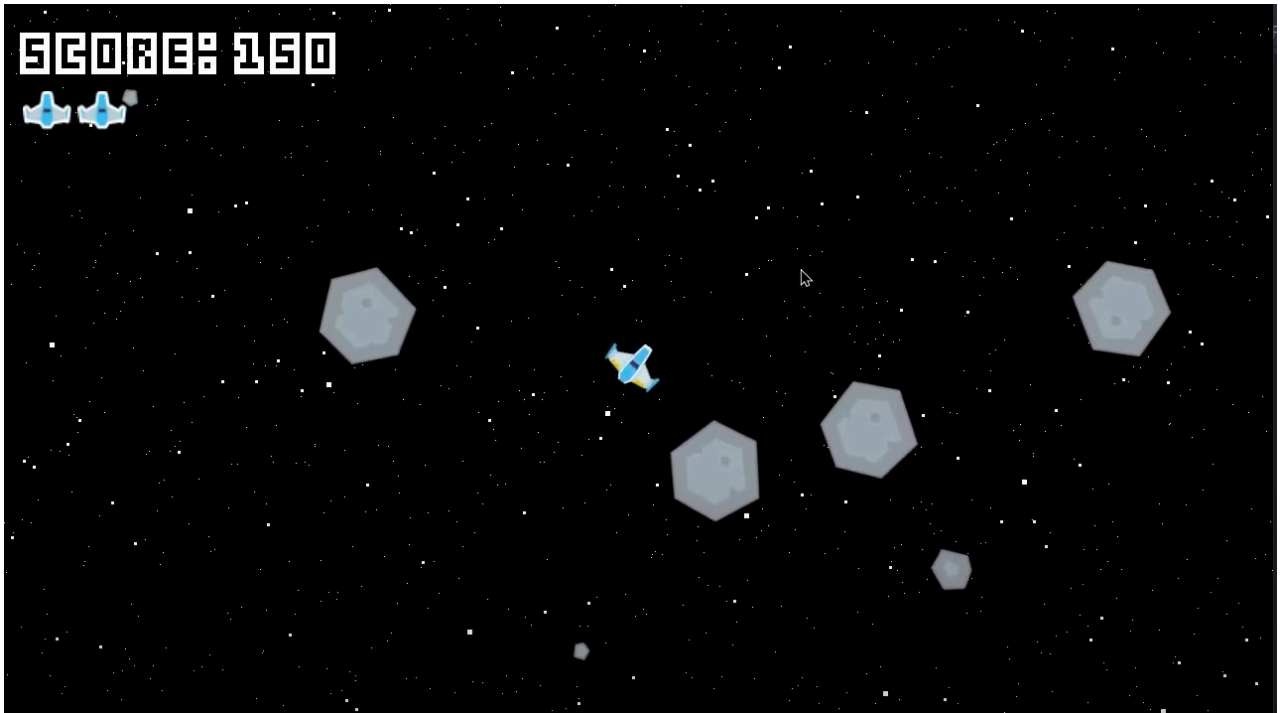


HET MAKEN VAN EEN VIDEOGAME



Door Emma van Gerven

Groep 53

Docent: B. De Kort

Inleverdatum: 21-02-2024

Samenvatting

In dit verslag worden de technische en sociale aspecten, de opbouw, het proces en de toekomst van gameontwikkeling besproken. Door middel van onderzoek en toepassing van theorie wordt er een game gemaakt. Door onderzoek te doen naar verschillende programma's waarmee games gemaakt kunnen worden, is erachter te komen welke het beste te gebruiken is voor dit project. Hieruit zal blijken dat hoewel Unity en Unreal zeer goede Game Engines zijn, en Gamemaker een van de oudste, Godot toch het beste middel is om het praktische deel van dit verslag uit te voeren.

Door de sociale aspecten van games en de populariteit te analyseren, zullen de elementen die het beste in de game geplaatst kunnen worden, naar voren komen. Wat maakt de game interessant? De antwoorden zijn voornamelijk: een verhaal en de mogelijkheid om online te kunnen spelen. Verhalen trekken mensen aan en geven een gevoel van betrokkenheid. Online kunnen spelen geeft een mogelijkheid om nieuwe mensen te leren kennen of om met mensen die ver weg wonen toch te kunnen spelen.

Met deze informatie wordt er verder gewerkt aan hoe men begint met het proces van een videogame maken. De wereld opbouw, het onderwerp en verhaal zijn allemaal aspecten die hierin voorkomen. Door te werken met beide modellen van Bizzocchi (2006) zal er een layout gemaakt worden. Deze wordt passend gemaakt en geïmplementeerd in een scene tree.

Hierna wordt onderzoek gedaan naar hoe de game daadwerkelijk te maken is. Hoe zorg je voor structuur in de verhaallijn? Hoe zorg je ervoor dat bepaalde dingen gespeeld worden door de speler zonder dat het tegendraads werkt met het verhaal?

Omdat gameontwikkeling nog altijd verandert en bevordert, moeten we ook naar de nieuwe ontwikkelingen in de bijpassende

technologie kijken. Hierbij wordt een analyse gedaan van de verschillende technologische vooruitgang die in de game industrie van hulp kunnen zijn. Vooral VR en AI zijn ontwikkelingen die de game industrie ver kunnen brengen. Dit zijn dan ook de onderdelen waarin de meeste ontwikkeling mogelijk is.

Er zal uit komen dat er verschillende dingen belangrijk zijn om een game te maken. Een in dit geval wordt er gebruikgemaakt van een engine. Godot, uit onderzoek is gebleken dat voor dit project Godot de beste game engine is, het is beginner vriendelijk maar heeft genoeg functies om het een uitdagende engine te maken. Om een game interessant te maken moet er een verhaal in zitten, dit blijkt uit onderzoek waarin gamers van verschillende leeftijdsgroepen een aantal vragen hebben beantwoord. Ook geeft de groep aan dat games waarin er een beïnvloedbaar verhaal is, deze beter ervaren worden dan verhalen waarbij dat niet is. Uit de enquête is gebleken dat spellen waar je samen online kan spelen een van de populairste spellen zijn. De beste manier om deze dingen te implementeren is door het model van Bizzocchi (2006) te gebruiken, het eerste model is om de wereld en verhaallijn goed te doordenken, maar om het een beïnvloedbaar verhaal te maken moet er gebruik gemaakt worden van het tweede model dat meer controle geeft aan de speler. Om dit dan te maken moet men denken aan verschillende dingen. Tijdens het onderzoek naar de ontwikkeling van de game zal er gedacht moeten worden aan dingen zoals sprites en de visuele aspecten van de game, maar ook de ontwikkeling van het script is belangrijk. Ook de connecties van levels zijn belangrijk om te implementeren, het verhaal moet kunnen kloppen en makkelijk te spelen en te begrijpen zijn.

Om echt te weten te komen wat er allemaal betrokken is bij games moeten we kijken naar een kleine geschiedenis en de mogelijke toekomst ervan. AI ontwikkelt zich steeds verder en zal ook in games veel meer voor komen. In de geschiedenis was namelijk goed te zien dat een hoop nieuwe technologie snel ook

in games werd gebruikt. De opkoming van AI en de nieuwe ontwikkeling van Visual reality kunnen zorgen voor ongelooflijke dingen. Er zou zeer mogelijk een volledige Virtual Reality in de gamewereld kunnen komen, waarin je als speler sensoren op je lichaam plakt, een bril op doet en kan verdwijnen in de wonderen van de gamewereld.

Inhoudsopgave

Samenvatting	1
Inhoudsopgave	4
Inleiding	6
Hoofd- en Deelvragen	7
Welke Game Engine kan je het beste gebruiken als beginner?	8
GameMaker	11
Godot	13
Hoe maak je een game interessant genoeg om te spelen?	15
Hoe relevant zijn deze vragen met betrekking tot dit project?	15
Wat is nou eigenlijk het verschil tussen een goede en een slechte game?	18
Hoe begin je met het ontwerpen van een game?	21
Wat moet er allemaal gedaan worden om een werkende game te hebben?	30
De visuele opbouw	30
Het verbinden van levels	31
De narratieve aspecten	33
De opstelling:	34
De complicatie:	35
De ontwikkeling:	37
De oplossing:	37
Hoe ziet de technische toekomst van videogames eruit?	41
Conclusie	44
Literatuur	47
Bijlage	51
Bijlage 1	51
Bijlage 2	71
Bijlage 3	81
Bijlage 4	93
Bijlage 5	94
Bijlage 6	96
Bijlage 7	97

Inleiding

Ik heb besloten om voor mijn profielwerkstuk het proces van het maken van een videogame te gaan onderzoeken. Het initiële idee was om te focussen op het maken van een videogame met een kleine complete videogame als eindresultaat. Na overleg met de begeleidende docent meneer De Kort, was het duidelijk dat dit geen goed doel was voor mijn PWS. Meneer De Kort heeft het advies gegeven om een verslag te maken over het maken van een videogame, hierdoor kan er aan bepaalde eisen voldaan worden en kan er aan een videogame gewerkt worden.

Het gekozen vak is Informatica, nadat ik per ongeluk het vak had gekozen in het derde jaar heb ik een passie voor coderen gevonden. Toen er werd verteld dat we een profielwerkstuk moesten maken wist ik gelijk dat ik het vak informatica hiervoor wilden gebruiken.

Het onderwerp video games is geïnspireerd door de hbo opleiding "[Creative Media and Game Technologies](#)¹". Deze opleiding is gericht op het leren maken van video games. Na meerdere open dagen en een oriëntatie dag hebben meegemaakt ben ik gefocust op het binnenkomen bij deze opleiding. Met dit verslag hoop ik onder andere extra kans te maken op toegelaten te worden tot deze opleiding om leren games te programmeren. Ik hoop algemene kennis op te doen over het maken van video games voor zowel een voorstudie voor de opleiding als voor mijn interesse voor hoe een videogame in elkaar zit.

¹ Creative Media and Game Technologies: Een hbo opleiding over gameontwikkeling. Deze opleiding heeft vier courses: 3D Visual Art, Design & Production, Programming en Film Visual Arts.

Hoofd- en Deelvragen

Het doel van dit profielwerkstuk is om erachter te komen hoe je (als beginner) het beste een game kunt maken.

Wat is er allemaal betrokken bij het maken van een game?

1. Welke Game Engine² kan je het beste gebruiken als beginner?
2. Hoe maak je een game interessant genoeg om te spelen?
3. Hoe begin je met het maken van een game?
4. Wat moet er allemaal gedaan worden om een werkende game te hebben?
5. Hoe ziet de technische toekomst van videogames eruit?

² Game Engine: Een stuk software dat een groot deel van de technische last overneemt, zodat gameontwikkelaars zich daadwerkelijk kunnen richten op het ontwikkelen van een game.

Welke Game Engine kan je het beste gebruiken als beginner?

De meest bekende game engines zijn Unity en Unreal, deze engines zijn helaas niet heel gebruiksvriendelijk voor beginners. Unity is de meest bekende engine, wat ervoor zorgt dat er veel video's en tutorials zijn wat zorgt voor een snelle start wel degelijk mogelijk is via Unity. Unity zelf heeft een kanaal waar veel video's over de nieuwste versie te vinden zijn. Als je compleet nieuw bent is het echter geen goed idee om met Unity te beginnen, ook al is er zo veel informatie over te vinden. Als je een beginner bent in programmeren en gameontwikkeling dan moet je eerst gewend raken aan de structuur en het logisch denken dat erbij komt kijken. Om Unity te gebruiken moet je deze logica al kunnen toepassen. Dankzij het vak informatica dat ik de afgelopen zes jaar heb gevolgd is dit voor mij geen obstakel, dit is niet voor iedereen het geval.

Scratch is een heel goed voorbeeld om een begin te maken in informatica en het logisch verband van codeer talen te begrijpen. In Scratch kan je goede acties toevoegen, d.m.v. plaatjes in plaats van letters en tekens is het makkelijker om de logica van code te begrijpen zonder code taal te leren.

Unreal heeft minder tutorials dan Unity maar heeft meer voordelen voor geavanceerde codeerders, zoals grafische afbeeldingen van hoge kwaliteit, geavanceerde natuurkunde simulatie en snelle prototype ontwikkeling. Als je al bekend bent met gameontwikkeling dan is Unreal de beste optie. De kennis die ik heb met gameontwikkeling is op beginnersniveau, Unreal is voor dit project dus geen goede keuze om mee te werken.

GameMaker is ook een redelijk bekende game engine maar heeft de laatste tijd weinig populariteit. Dit zorgt voor minder informatie over de huidige versies die er zijn. GameMaker

heeft zelf net als Unity en Unreal een eigen kanaal op YouTube met veel tutorials over de nieuwste versies. Ook zijn de meeste video's van oudere versies nog relevant aangezien er vaak niet al te veel wordt aangepast aan hoe alles al werkt maar eerder dingen worden toegevoegd. Om GameMaker te gebruiken moet men echter de GML (Game Maker Language) leren. De tot nu toe genoemde engines maken gebruik van de programmeertaal C#, een zeer bekende taal in de game-industrie aangezien deze taal een gebruiksvriendelijkere versie is van C++. Met de taal C# in combinatie met een game engine, is er minder code om zelf te schrijven, waardoor je je beter kan focussen op andere dingen zoals de visuele aspecten. De GML is echter wel gebaseerd op meerdere veelgebruikte code talen, waaronder C# en C++. De taal GML zou in theorie makkelijk aan te leren zijn als men al kennis heeft van deze twee code talen. Ondanks dat GameMaker niet zeer populair meer is, is het wel degelijk een goede optie voor een 2D game. Helaas is 2D echter wel de enige optie voor deze, aangezien GameMaker geen 3D ondersteunt. Voor dit project is dat geen hindernis aangezien de game in 2D zal zijn. Gamemaker is dus een optie ondanks de limieten, van codetaal en dimensies van de game.

Als laatste is de in populariteit opkomende game engine Godot. Deze game engine heeft de laatste tijd steeds meer populariteit, ook al is de game engine in 2014 uitgebracht. Deze nieuwe surge in populariteit kan komen dankzij de prijsverandering voor game developers die werken met Unity. Veel mensen zijn het er niet mee eens en zoeken naar nieuwe goede Game Engines om mee te werken. Dankzij deze verandering heeft Godot een nieuwe surge naar de top gemaakt. Veel populariteit betekent dat er naast de officiële hulpmiddelen genoeg video's en tutorials over de laatste versie zijn. Ook maakt Godot net als de meeste engines, gebruik van C# en C++. Het gebruiken en leren van Godot is dus geen groot obstakel om de game engine te gebruiken.

Nu is de vraag: GameMaker of Godot? GameMaker heeft een eigen taal die geleerd moet worden, hoewel deze berust op meerdere

populaire talen kan dit een valkuil zijn. GameMaker maakt wel voor een reden gebruik van GML, de taal eigen aan deze game engine. Deze taal zou de programmeerder de last moeten minderen van de meeste technische delen van het programmeren. Met minimale tot geen kennis van programmeren kan GameMaker een fijne game engine zijn om mee te starten, maar geen goede introductie in de rest van de talen van de game wereld. Wel heeft GameMaker vele tutorials ondanks de dalende populariteit. Godot maakt daarentegen wel gebruik van C# en of C++. Ook zijn er over Godot genoeg tutorials voor de nieuwe versies dankzij de stijgende populariteit. Puur door te kijken naar de populariteit, zou Godot de beste keuze zijn, GameMaker is al rond 1999 een keuze voor een game engine, en Godot pas sinds 2014. Ondanks de tijd dat GameMaker heeft gehad om te verbeteren en populair te worden, heeft Godot binnen 10 jaar een grotere hoeveelheid gebruikers dan dat GameMaker heeft op dit moment. Maar er is natuurlijk een tijdperk geweest waarin GameMaker ook zeer populair was, dus over de huidige populariteit kunnen we niet spreken als het gaat over welke engine beter is.

Om te onderzoeken welke de beste van de twee is, moeten we in beide een game analyseren. Asteroids: By The Numbers³, is een spel met eenvoudige sprites⁴ en achtergrond, en dus een goede optie om te focussen op de gameplay en het maken van de game zelf, zonder dat je moet letten of het er goed uit ziet. Met deze gekozen sprites is een redelijk makkelijk spel te maken, zo kan er gefocust worden op kennismaken met de game engine. Het hoeft natuurlijk niet precies op de echte Asteroids te lijken, maar een basis ervan. Door in beide engines deze game te maken kunnen we erachter komen welke de beste keuze is voor het maken van de game. Asteroids is in de game gemeenschap een bekende game aangezien het spel al decennialang bestaat. Mensen die veel ervaring hebben of zelf hun eigen games maken,

³ Asteroids: By The Numbers: Een arcade spel uitgebracht in 1979. Een van de eerste games met van toen complexe berekeningen, gebruikt voor de gameplay van het spel.

⁴ Sprite: Een afbeelding om in een game te gebruiken wordt vaak ook wel een sprite genoemd.

maken tutorials of video's over game engines met als introductie een video over hoe het spel Asteroids hierin gemaakt kan worden.

In Bijlage 1 zijn schermafbeeldingen van de game opbouw en code zowel als een ingame foto te zien.

GameMaker

Zelf heb ik GameMaker al een keer eerder gebruikt voor een simpel schoolproject. Ik heb hierbij gemerkt dat het makkelijk programma is, en zeer gebruikersvriendelijk. De nieuwste versie is zeer anders dan de versie waar toen meegewerkt is. Toch is het snel te merken dat er, ondanks de updates en toevoegingen, een zeer gebruikersvriendelijk programma is. Het verschil in de hoeveelheid werk dat je moet doen valt al snel op. De tutorial voor GameMaker was maar een kwartier lang en die duurde voor Godot twee uur. Nou heeft de tutorial van Godot een paar extra toevoegingen, maar dit haalt maximaal een half uur er van af als je deze niet volgt. GameMaker is dus veel makkelijker en sneller om mee te werken.

Het is heel overzichtelijk, en je kan ervoor kiezen of je code of 'visual' wilt gebruiken. Visual geeft de optie om zonder code je game te maken. Er is in dit geval gekozen voor de GML code. Een object aanmaken is zeer simpel, je klikt op 'add object' en je voegt er een sprite⁵ aan toe. Aan dit object voeg je een paar lijntjes code toe zodat hij beweegt en dan heb je meer dan de helft van de game Asteroids al. Met één lijntje code zorg je ervoor dat je schip aan de andere kant van het scherm terugkomt als deze van het scherm af gaat. Voeg toe dat als je klikt er sprites van de kogel 'uit het schip komen'. Voeg een collision actie toe, schrijf één lijn code dat zowel de kogel als de asteroïde vernietigd wordt of kleiner wordt. En dan ben je al klaar.

⁵ Sprite: een afbeelding om te gebruiken in een game.

Het is allemaal dus net iets te simpel, voor een profielwerkstuk over games is GameMaker geen geschikt middel. Het is gebruiksvriendelijk en er is snel veel mee te doen, maar het is te makkelijk. Een van de redenen dat er in dit verslag geen Unity of Unreal wordt onderzocht is omdat deze op een te hoog niveau liggen. Deze engines zijn voor mensen die er al ervaring mee hebben. GameMaker is voor zeer nieuwe beginners, voor mensen die geen verstand hebben van een game engine en het willen leren. Een te hoog niveau is onhaalbaar maar een te laag niveau is onvoldoende om mee te werken.

Godot

Bij het openen van een project in Godot heb je de optie om je project in een map met andere objecten te plaatsen. Als je dit doet dan krijg je de melding dat de map niet leeg is en als je doorgaat, de bestanden automatisch als objecten worden aangemaakt om mee te werken in je game. Heb je mapjes binnen de map, dan worden deze overgenomen en ook de bestanden binnen die map worden objecten gemaakt. Dit kan veel tijd schelen, het plaatsen van objecten in de applicatie van de game en mapjes maken kan alleen als je ze ook daadwerkelijk op je computer hebt staan. Je moet dus twee keer de objecten opzoeken en plaatsen, één keer naar je computer en één keer naar de applicatie van de game. De optie om bestanden per direct bij het aanmaken naar objecten te veranderen is dus een zeer fijne optie. Ook na het aanmaken van de applicatie gebeurt dit, plaats je een nieuw bestand in de map waar de applicatie zich bevindt, dan wordt deze automatisch ook een object. Wil je niet dat dit gebeurt? Heel simpel, kies een lege map en maak daar je applicatie in. Dit is natuurlijk een klein onderdeel, maar het geeft wel een gebruikersvriendelijk gevoel.

Na het gebruik van Godot is nog beter te merken hoe makkelijk GameMaker eigenlijk is. Godot heeft handige toepassingen zoals de @export om variabelen buiten de code te plaatsen en buiten een script er toegang tot hebben. Maar er moet veel meer

gecodeerd worden. Het plaatsen van objecten en sprites en het koppelen hiervan is net als in GameMaker makkelijk, er moet alleen veel meer code gekoppeld worden aan de sprites, objecten en dingen zoals collision shapes⁶. Verdere uitleg over de code is te vinden in bijlage 1 onder 'Godot'.

Zelf vind ik de UI⁷ heel fijn. Het is netjes geordend wat een fijn overzicht geeft tijdens het werken. Bij het toevoegen van objecten zoals een Sprite of Area onthoud het programma welke objecten je veel toevoegt. Deze staan dan in een lijstje naast de lijst van de opties. Ook kun je makkelijk een actie maken voor de invoer van een toets, deze kun je weer ophalen in je code. Er zijn veel kleine dingen zoals deze die het naar mijn gevoel heel gebruiksvriendelijk maken. Ja, Godot vergt meer code en acties, maar de hulpmiddelen zijn zeer fijn en zorgen ervoor dat het geen moeizaam proces is. Er is makkelijk te switchen tussen de scripts en de 2D view van de uiteindelijke gameplay.

Na dit onderzoek naar deze engines heb ik geconcludeerd dat Godot de beste keuze is, het is uitdagend maar niet onhaalbaar om mee te kunnen werken. GameMaker is te simpel en Unity en Unreal zijn voor dit project geen slimme keuzes vanwege de kennis die je al moet hebben om deze te gebruiken.

Voor een echte beginner in de game ontwikkeling, is GameMaker een perfecte start om je vaardigheden te ontwikkelen, kan je geen code dan kun je eerst leren hoe de samenhang werkt met GML visual, heb je dit onder de knie dan kun je beginnen met de kennis die je hebt en hierop code toepassen. Je kunt met GameMaker eerst de basics leren voordat je verder gaat met gevorderde dingen.

⁶ Collision shape(s): Geeft het gebied aan dat het gekoppelde object in beslag neemt. Hiermee kunnen botsingen worden gecreëerd.

⁷ UI: User Interface, de visuele vorm waarin de gebruiker het product of de dienst te zien krijgt, en de tools waarmee hij er gebruik van maakt.

Hoe maak je een game interessant genoeg om te spelen?

Om dit te kunnen beantwoorden moeten we weten wat ervoor zorgt dat mensen games spelen. Om hierachter te komen is er een enquête gemaakt en deze door mensen uit verschillende groepen laten beantwoorden.

Deze enquête is gemaakt deels in opdracht voor maatschappijleer, een deel van de vragen zijn dus niet relevant omdat ze moesten voldoen aan de eisen van het project van dit vak, daarom missen er vragen. De enquête is samen met medestudent Evy van Keulen gemaakt en is te vinden onder Bijlage 1- Enquête.

Uit de enquête blijkt dat de groep die we hebben bereikt vooral jongeren zijn, voornamelijk jongeren spelen videogames dus we kunnen er van uit gaan dat de goede doelgroep is bereikt.

Hoe relevant zijn deze vragen met betrekking tot dit project?

Om een game interessant te maken moet je weten wat de doelgroep van je game interessant vindt, door middel van deze enquête is erachter te komen wat men interessant zou vinden. De leeftijd, gender en afkomst kunnen hier een grote rol in spelen. Jongeren hebben bijvoorbeeld meer behoefte aan snelle games zoals platformers en shooter games. En ouderen bijvoorbeeld meer aan een simpele mystery game. Natuurlijk zijn hier uitzonderingen voor maar over het algemeen zijn dit de voorkeuren.

Met deze informatie en het feit dat we weten dat jongeren meer videogames spelen, kunnen we herleiden dat snelle games een groter publiek zullen hebben. Een groot publiek kan zorgen

voor populariteit, waardoor de game als interessant beschouwd kan worden.

Een snelle game is natuurlijk geen goede uitleg voor iets wat interessant is. Snelheid van het plot en dingen zoals actie zijn wel belangrijk, De meest populaire games zoals de game series Call of Duty en Overcooked, zijn spellen met een snel tempo. De spellen zijn zeer verschillend, Call of Duty is een first person shooter game waarin je alleen of samen meerdere missies moet doen. Deze missies vertellen het verhaal van de game die je speelt. Net als in het echt, is er veel stress en een tijdnood tijdens de missies. Overcooked daarentegen heeft uiteindelijk maar één missie: de titel chef-kok behalen. In dit spel ben je een chef die zo snel mogelijk samen met de andere chefs eten moet bereiden voor de gasten. Dit gaat met een snel tempo en hier ligt een tijdsdruk achter.

De games Call of Duty 1 en Overcooked 1 waren zo populair dat er meerdere versies werden uitgebracht. De Call of Duty serie vertelt het verhaal van Taskforce 141 terwijl de Overcooked series focust op het spelen van een chef in de keuken, er is ook een plot aanwezig. Dit plot is echter onderliggend. Het worden van een chef-kok door te werken in meerdere keukens met slechte omstandigheden om te trainen een oud eetbaar kwaad te kunnen overwinnen. Om het spel te spelen en ervan te genieten is er niet per se een plot nodig, dus waarom is deze wel aanwezig?

Verhalen vertellen is al sinds de oertijd een deel van ons leven, eerst in de vorm van grotschilderingen die op hun eigen manier het verhaal van een gebeurtenis vertellen. Of daarna via liederen en daarna dankzij de boekdrukkunst ook veel via boeken. Al deze manieren van verhalen vertellen we nog steeds; strips, cartoons, liedjes, poëzie, boeken en dankzij de camera ook films. Dankzij de uitvinding van de camera hebben we ook films waarmee we verhalen vertellen. Veel games maken gebruik van al deze dingen om een verhaal te vertellen, er moet wat op het scherm komen, daar komen de 'tekeningen' in beeld. Het

spelen van de game moet natuurlijk niet saai zijn, daar komen de liedjes in beeld, nummers hoeven geen tekst te hebben om een verhaal te vertellen, denk maar aan klassieke muziek. Boeken zijn geschreven verhalen, veel games maken gebruik van tekst in de game, deze tekst is relevant voor het plot, dit volgt bijvoorbeeld de conversaties tussen twee personages in het spel. Een game maakt dus gebruik van vrijwel alle basismethoden om een verhaal te vertellen. Deze manieren vinden wij als mensen al snel geweldig, het zijn immers manieren die we al eeuwenlang gebruiken. Een verhaal in een game kan dus helpen met de aantrekkelijkheid van een game, een mens houdt nou eenmaal van een goed verhaal, dat hoort bij wie we zijn. De veelvoud van manieren hoe het verhaal verteld wordt kan helpen met het bereiken van meerdere mensen. Niet iedereen vindt dezelfde methode van verhalen vertellen even leuk. Dankzij al de gecombineerde methodes kan een game een zo groot mogelijk publiek bereiken.

Wat is nou eigenlijk het verschil tussen een goede en een slechte game?

We hebben dankzij de enquête vernomen dat naast Minecraft, games zoals shooters (snelle games) een groot publiek hebben. En we weten dat verhalen een belangrijk deel van een game zijn, maar wanneer is een game een interessant spel en wanneer is het simpelweg saai? Om dit zo goed mogelijk te kunnen beantwoorden hebben we mensen nodig om dit soort vragen te beantwoorden. In bijlage 2 is de vragenlijst te vinden die naar verschillende mensen is gestuurd die zeer regelmatig gamen. Mensen vanaf 15 tot 43 hebben de vragenlijst ingevuld, de meeste mensen die regelmatig games spelen zitten ook rond deze leeftijd, het juiste publiek is dus bereikt. Zoals verwacht heeft vrijwel iedereen ingevuld elke dag games te spelen, ook hiermee is dus het goede publiek bereikt. Het ging er namelijk om dat mensen die bekend zijn met videogames en het spelen ervan, degene waren die de vragenlijst invullen. De vragenlijst is naar zowel mannen en vrouwen gestuurd, het gender van de persoon had hier echter geen invloed op de uitkomst, het gaat namelijk over wat een game interessant maakt, en niet wat er per gender interessant te beschouwen is.

Gelijk is er bij vraag 5 is er te merken dat een verhaal inderdaad een belangrijk onderdeel van een game is. De vraag over een verhaal in een video game is nog niet geweest en mensen hebben het over de rol van een verhaal in een game. Dit komt overeen met het eerdere onderzoek naar verhalen (in video games) en de grote rol ervan. Een goed verhaal in een game is een belangrijk deel, dit komt ook naar voren bij vraag 6.

Ook wordt er vernomen dat hoewel een spel met een verhaal belangrijk is, dat het maar ligt aan het genre. Sommige games hebben geen verhaal nog en sommige zoals een RPG⁸ juist wel. Dit kan te maken hebben met het feit dat een RPG juist draait om het verhaal van een karakter, je speelt de rol van dit

⁸ RPG: Role Playing Game, een spel waarin je de rol van een personage speelt die op avontuur uitgaat.

personage, als het verhaal dan slecht is, dan is de game zelf, ondanks de goede UI, ook snel slecht.

Een veel voorkomend favoriet is World of Warcraft, een MMO⁹ spel dat constant wordt uitgebreid. Omdat je met zo veel mensen kunt spelen is dit voor velen een fijn spel, om met vrienden te spelen of nieuwe online vrienden te maken tijdens het spelen. Ook is het feit dat er om de zoveel tijd nieuwe uitbreidingen komen, zeer geliefd. Het is veel terug te zien in populaire games die samen te spelen zijn met anderen. Denk maar aan de zeer bekende game Fortnite die na korte tijd snel naar de top is geklommen. Dit spel heeft regelmatig uitbreidingen en is met meerdere mensen te spelen, zowel vrienden als vreemden, ook deze online video game valt sinds kort onder het MMO genre.

⁹ MMO: Massive Multiplayer Online, staat voor een video game waarin zeer veel mensen tegelijk samen online kunnen spelen.

Een game is interessant te maken door er een verhaal aan toe te voegen. Verhalen zijn volgens meerdere gamers een belangrijk deel. Voor sommigen is het niet belangrijk en vinden games zoals shooters fijner, wat de meeste in gemeen hebben is het online co-op¹⁰ genre. Verbinding vinden is iets wat veel gamers doen, dit komt naar voren uit onderzoek aangezien een groot deel van de gevraagde (bijlage 3) als favoriete game WoW¹¹. Dit spel heeft zowel meerdere verhaallijnen als ondersteuning om online met vrienden het spel te spelen.

Ook komt eruit, dat zelf je eigen verhaallijn van de game te kunnen beïnvloeden, een grote bijdrage kan geven aan hoe interessant de game is. Meerdere mogelijkheden zorgen ervoor dat er meer te doen is. Het geeft je het gevoel dat je in de wereld van de game zit en dat je eigen keuzes invloed hebben.

Snelle games zorgen ervoor dat men de aandacht erbij moet houden en dat het niet saai wordt. Games zoals deze worden sneller populair dan de meeste.

¹⁰ Online co-op: Online Cooperation, online kunnen spelen met meerdere mensen.

¹¹ WoW: World of Warcraft, een video game.

Hoe begin je met het ontwerpen van een game?

Om te beginnen moet je weten hoe, alleen code of met een engine? In dit geval beginnen we met een engine: Godot, aangezien dit na onderzoek voor dit project uit kwam als meest geschikte engine.

Om een game te maken heb je hier een idee voor nodig. Het gekozen onderwerp zal het verhaal van een astronaut zijn. Dit verhaal zal worden verteld door middel van een platformer spel. Om een aantrekkelijke game te maken moet je een verhaal hebben, is gebleken uit deelvraag 2: "Hoe maak je een game interessant?". Het hoeft niet altijd aanwezig te zijn maar is voor de meeste een zeer belangrijk aspect van een game. Dan is de volgende vraag natuurlijk; hoe maak je een interessant verhaal? Hiervoor moeten we kijken of verhaal voldoet aan de 'regels' van de verhaalvertelling in een game? Volgens Bizzocchi (2006) heeft een game verschillende stages, de 'Narrative Arc'.

- De *opstelling* introduceert de personages en de verhaalwereld waarin ze leven.
- De *complicatie* introduceert een uitdaging die moet worden overwonnen.
- De *ontwikkeling* is de lange fase die het grootste deel van het vertellen van verhalen domineert, hierin werkt de hoofdpersoon aan haar doel.
- De *oplossing* of climax is het hoogtepunt van de strijd van de ontwikkelingsfase, vaak resulterend in een vorm van overwinning of nederlaag.
- De *ontknoping* of vallende actie verbindt de losse eindjes van het verhaal en maakt het mogelijk om de verhaal ervaring gracieus te eindigen.

De Opstelling - In het begin worden de personages geïntroduceerd, de speler en de vijanden. De speler wordt

geïntroduceerd over de situatie van een afgezonderde astronaut in een verlaten ruimtestation. Het station is verlaten en werkt maar half.

De Complicatie - De speler merkt snel dat het ruimtestation waar hij zich op bevindt, geïnfecteerd is met aliens die het ruimtestation afbreken. Om te blijven leven en contact op te nemen met de aarde moet hij eerst de aliens verslaan.

De Ontwikkeling - De speler vecht de aliens en repareert het schip, om contact met aarde op te kunnen nemen zodat hij gered kan worden en terug naar de aarde kan.

De Oplossing - De speler komt erachter dat zolang hij de koningin van de aliens niet verslaat er nieuwe aliens zullen blijven komen, hij gaat af op de alien koningin en verslaat haar.

De Ontknoping - De speler verslaat de eindbaas en kan eindelijk het schip onderbroken repareren. Hij neemt na het fixen van het schip contact op met aarde om gered te worden

Er is in dit model echter een probleem, voor een interactief spel moet een deel van de controle over het verhaal aan de speler moeten worden gegeven, om ze het gevoel te laten krijgen dat zij zelf het verhaal in de hand hebben, dat zij zelf in het verhaal zitten. Door deze controle aan de speler te geven werkt de 'Narrative Arc' niet zoals bedoeld.

Door een videogame interessant te houden is de mogelijkheid om op het verhaal zelf invloed te hebben een goed middel om de speler aandacht erbij te hebben. Zo komen er dus verschillende eindjes van een verhaal.

Bizzocchi (2006) introduceert daarom een nieuw schema waarin er controle aan de speler gegeven kan worden, zowel als redelijke controle hebben over de game als ontwikkelaar. Hij noemt dit de 'Narrative Framework'.

Dit schema kan gevolgd worden bij een spel met interactieve eigenschappen.

- *Verhaalwereld* - Wat is de omgeving waarin het spel zich ontvouwt?
- *Karakter* - Wie zijn de wezens die deze spelwereld bevolken?
- *Emotie* - Zowel de emoties die worden getoond door de personages in het spel als de emoties die worden opgewekt in de game speler.
- *Narratieve Interface* - Hoe worden narratieve gevoeligheden geconstateerd in het uiterlijk en de functionaliteit van het interface ontwerp.
- *Micro-Narratief* - Kleinere momenten van de verhaalstroom en samenhang die plaatsvinden binnen een bredere context van het spelen de game.

Als dit schema toegepast wordt geeft het een overzichtelijke layout van wat er allemaal gedaan moet worden. Zie hieronder een voorbeeld.

De Verhaalwereld - Het verhaal zal zich afspelen in de ruimte, op een verlaten ruimtestation. Nadat de astronaut ineens wakker geworden is op een voorheen vol ruimtestation moet hij in een verlaten en dim verlicht station taken doen om contact op te nemen met de aarde.

Karakter - De astronaut zelf en groen en oranje aliens, samen met de alien koningin, de eindbaas. De astronaut is iemand die op het station leefde en werkte totdat hij ineens alleen wakker werd. De aliens zijn de wezens die de astronaut moet vechten om zijn doel te behalen en zijn taken te completeren.

Emotie - Het is een kleine game dus veel emotie zal er niet zijn. De ruimte is redelijk donker verlaten en zal enige spanning aangeven.

Narratieve Interface - In de game zullen meerdere momenten van uitleg over de game gegeven worden. Dingen zoals hoe je je personage beweegt tot uitleg over taken en het doel van de game, het verslaan van de uiteindelijke eindbaas. Hiermee zal uiteindelijk het verhaal worden verteld, samen met de vormgeving van de game.

Micro-Narratief - Zoals eerdere gezegd wordt dit een kleine game, dus heel veel kleine momenten van tekst zullen er niet zijn. Momenten zoals een gecompleteerde taak zullen kleine teksten hebben om het een groter betrokkenheidsgevoel te geven.

Dit model is meer bedoeld voor games zoals Detroit: Become Human, een spel uit 2018 waarin de speler door middel van keuzes een heel grote invloed heeft op de verhaallijn van drie karakters. Het spel volgt drie androids met Artificial Intelligence. Er wordt een verhaal verteld van drie verschillende androids die alle drie te maken hebben van 'system distress'. Deze 'distress' is in de werkelijkheid de evolutie van de AI die zich ontwikkelt om eigen gevoelens en

beslissingen. Dit gebeurt op basis van de nieuwe ontwikkelingen. Elke beslissing die je maakt heeft een grote invloed op de verhaallijn en de evolutie van de AI binnen de android.

Ook Baldur's Gate 3 (2023), een nieuw en zeer populair spel, kan geen gebruik maken van alleen het eerste model van Bizzocchi (2006). Dit spel bootst een 'Dungeons and Dragons'¹² spel na. Na volledig zelf een personage hebben mogen maken ga je na de opening uit in een open wereld waarin je karakters ontmoet en relaties mee kan vormen. Elk karakter voegt toe aan je verhaal, en elke bevordering in relatie met deze karakters, verandert het verhaal weer. In dit spel is er nog meer controle aan de speler gegeven. Dit komt omdat het een open wereld game is. De hele wereld is één geheel, het bestaat niet uit levels. Doordat de verhaalwereld één grote kaart is, zijn er talloze opties waar je naar toe kan, natuurlijk stuurt het spel je wel de goede kant op. Dit is door middel van tekst en conversaties met personages in de game.

Dit zijn mooie voorbeelden van spellen die berusten op dit model. De verhaalwereld, de karakters, de emotie, de narratieve en de micro narratieve interface zijn hier allemaal zwaar van belang. Om zo veel mogelijk controle te geven aan de speler wordt er dus vooral op deze punten gelet.

In het praktische deel van dit verslag, zoals eerder al verteld is, wordt onderzoek gedaan naar hoe een game praktisch gezien gemaakt wordt. Het tweede model van Bizzocchi (2006) is een model waarop een game zoals Detroit: Become Human of Baldur's Gate 3, berust. Dit model zal veel helpen met het focussen op de belangrijkste dingen van het ontwikkelen van een beïnvloedbare game.

¹² Dungeons and Dragons: Ook wel DnD (of D&D), is een fantasie rollenspel. Een spel waarin je eigen fantasy personages maakt en hiermee speelt, onder leiding van de spelleider. Deze vertelt het verhaal waarin de personages zich bevinden.

Voor dit onderzoek volgen we dus vooral het eerste model, gecombineerd met het tweede model. Hierboven zijn beide modellen uitgewerkt. Het probleem bij de huidige toepassingen is dat er geen ander einde en dus bijna geen invloed op de game mogelijk is. Om dit op te lossen voegen we bij de ontknoping twee mogelijkheden toe.

Ontknoping 1 - De speler verslaat de eindbaas en kan eindelijk het schip onderbroken repareren. Hij neemt na het fixen van het schip contact op met aarde om gered te worden

Ontknoping 2 - De speler heeft de eindbaas verslagen, maar in het proces heeft zij een deel van het schip verwoest, de satelliet. Dankzij de vernietiging van de satelliet kan de astronaut geen contact met aarde opnemen en zal hij alleen op het ruimtestation blijven.

Ontknoping 3 - De speler verliest van de bazin, er komt 'Game Over' op het scherm.

Er zijn nu gaten in het huidige verhaal: Hoe integreren deze ontknopingen zich in het verhaal? Hoe heeft de speler invloed op het verhaal? Dit is onderdeel van de opstelling en de complicatie. De opstelling wordt dan het volgende:

De Opstelling - In het begin worden de personages geïntroduceerd, de speler en de vijanden. Deze personages bevinden zich op een ruimtestation, het is verlaten en werkt maar half. De speler komt er al snel achter dat deze aliens vijanden zijn en verslagen moeten worden. In de introductie van het spel komt de speler er ook achter dat het schip gemaakt moet worden, als hij contact wil met de aarde.

De integratie is nu bijna klaar. Door de complicatie passend te maken bij de opstelling wordt het compleet.

De Complicatie - De speler merkt snel dat het ruimtestation waar hij zich op bevindt, afgebroken wordt door de kwaadaardige

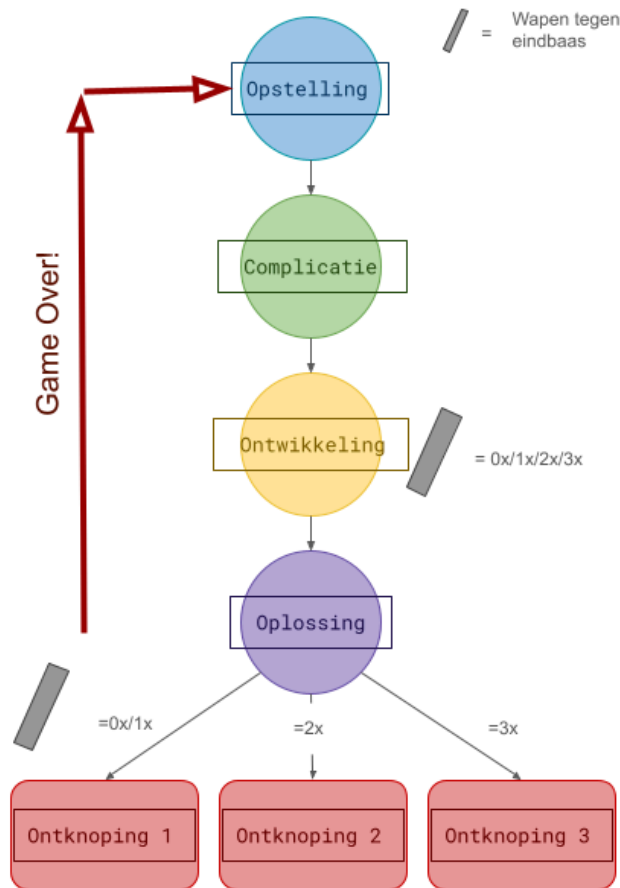
aliens. Dan zal er de eerste reparatie plaatsvinden. De speler wordt volledig geïntroduceerd over de situatie van een afgezonderde astronaut in een verlaten ruimtestation. Bij deze reparatie merkt de speler dat er onderdelen zijn die te gebruiken zijn bij het vechten en het verslaan van de aliens. Dit kan hij echter alleen doen als hij tijdens het vechten het schip repareert.

Om het verhaal mooi dicht te maken moeten er details worden toegevoegd aan zowel de ontwikkeling als de opstelling.

De Ontwikkeling - De speler repareert het schip, om contact met aarde op te kunnen nemen zodat hij gered kan worden en terug naar de aarde kan. *Tijdens het repareren vindt hij grote onderdelen die als wapens gebruikt kunnen worden.*

De Oplossing - De speler komt erachter dat zolang hij de koningin van de aliens niet verslaat er nieuwe aliens zullen blijven komen, hij gaat af op de alien koningin en probeert haar te verslaan met de gevonden onderdelen.

Veel is er niet toegevoegd, wel is hiermee de opbouw compleet gemaakt. Om het overzichtelijk te maken tijdens het maken van de game, worden deze gebeurtenissen in een scene tree gezet. Een scene tree is een overzichtelijk schema van scènes die vertakken naar andere scènes. Tijdens de game zal één voor één een scène van de lijst worden gemaakt. Door te werken met een scene tree zal de het proces van de game overzichtelijker worden.



Bijlage 5

Met deze scene tree is de game overzichtelijk te maken. Het tweede model wordt bij het maken van de game geïmplementeerd, de sprites, de geluiden en de gameplay is wat voor de details zorgen. Dit wordt na de volledige lay-out van de game toegevoegd en zijn te vinden in bijlage 4.

Het is dus belangrijk om te weten hoe een verhaal in een game in elkaar zit. Door de opbouw op te schrijven en het model van Bizzocchi te gebruiken is er eenvoudig een verhaal te creëren. Dankzij een scene tree is het gemaakte verhaal overzichtelijk te implementeren in de game en kan er één voor één aan de onderdelen worden gewerkt. Hiermee is er een goede start van de game gemaakt en kunnen we verder met het praktische onderzoek.

Wat moet er allemaal gedaan worden om een werkende game te hebben?

De visuele opbouw

Als eerste moet er beslist worden hoe de game eruit moet gaan zien. Hiervoor heb je tenminste een vaag idee nodig waar de game over zal gaan. Als idee gaan we uit van het voorbeeld van het vorige hoofdstuk. Met het idee van de astronaut gestrand op een ruimtestation kan er gezocht worden naar passende sprites. De sprites zijn te vinden in Bijlage 5. De sprite sheet is een van de belangrijkste dingen omdat deze de hele omgeving van de game zal vormen. Bij een idee zoals deze kan er niet zomaar een sprite sheet gekozen worden. Een achtergrond met groen gras en blauwe lucht geeft geen sfeer van een ruimtestation. Om het gevoel te geven van een ruimtestation moeten de tegels, de muren en de UI lijken op de sfeer van een ruimtestation. Bij de gekozen sprite sheet bevinden zich minimalistische grijze tegels, groene tegels om het sci-fi gevoel te geven en verschillende soorten tegels met schroeven en ventilatie schaften.

Het speelbare karakter is net zo belangrijk als de achtergrond. Hier ligt namelijk bijna alle focus op omdat dit hetgene is wat de speler het hele spel door controleert. Hierbij moet de stijl van de achtergrond meegenomen worden met de beslissing. Is dit in pixel art? Dan moet het character dat ook zijn. Het ideaal is als ze zelf gemaakt worden of als je een samengesteld sprite sheet kan vinden, de stijlen komen dan mooi overeen.

Met al dit kun je de animaties gaan toevoegen zoals in bijlage 5 te zien is. Door de sprite sheet van het personage te knippen en verschillende hokjes te selecteren komt er een animatie uit. Ook de achtergrond sprite sheet is door te knippen en plakken te gebruiken in de game engine.

Het verbinden van levels

In het vorige hoofdstuk is de scene tree gemaakt, een mooi overzicht van wat er gebeurt wanneer in de game. Dit moet echter worden verwerkt in de game. Wanneer ben je verder in een level? Hoe kom je erachter wat in dit geval de complicatie is? Als we kijken naar vele oudere spellen kunnen we makkelijk zien dat de opstelling en complicatie levels worden geïntroduceerd d.m.v. dialoog. De speler wordt geïntroduceerd aan de controls, omhoog, omlaag, naar rechts en naar links, springen en kruipen. Zodra de speler het onder controle heeft wordt de complicatie geïntroduceerd, de speler moet vechten tegen de gekozen vijand en krijgt hier uitleg over waarom. Deze twee scènes kunnen in dit geval gekoppeld worden door een aantal kleine taken te moeten voltooien voordat er deuren open gaan. Als de speler deze voltooid en door de deur heen loopt kan het dialoog deel beginnen, een narratief van de spelers gedachten en wat het karakter denkt als het door het verlaten ruimtestation loopt. Dit narratief kan het volgende level, de ontwikkeling, koppelen. De ontwikkeling begint in een waarneming van een groep vijandelijke aliens, wat de speler introduceert aan het vecht aspect van de game. Tijdens het vechten zullen er dialoogvensters komen voor de speler, kleine directies om te laten merken wat de speler moet doen om te winnen van deze aliens

Voordat de taken beginnen moet de speler te weten komen dat er een alien koningin is en dat de aliens blijven komen totdat zij verslagen is. We kunnen dit invoegen door de reparaties plaats te laten vinden in de controlekamer, een ruimte waarin zich camera's bevinden. Door een kleine animatie scène toe te

voegen is er te laten zien dat er in een van de kamers een grote alien koningin is die ervoor zorgt dat de aliens blijven komen.

De speler moet nu geïntroduceerd worden aan de 'task at hand'¹³: het repareren van het schip om contact op te nemen met de aarde. Hierbij moet de speler eerst een proef reparatie maken. De speler moet namelijk geïntroduceerd worden aan de actie van de reparaties. Hierna zullen de andere drie plekken waarin de reparaties moeten gebeuren oplichten, of een duidelijke omlijsting krijgen.

Als de speler de reparaties maakt en de onderdelen verzamelt heeft, zal hij verder moeten kunnen naar het volgende level, de oplossing. Maar hoe valt dit te implementeren? Als we de deur pas open laten gaan als de speler alle drie de onderdelen heeft, is er geen mogelijkheid voor verschillende eindes. Ook moet er voorkomen worden dat de speler meteen door de deur loopt.

Door de taken te verdelen in twee delen kunnen we dit verbeteren. Elke taak zal twee aspecten moeten krijgen. Uitgaand van het feit dat de speler de kleine tutorial aan het begin goed heeft gevolgd, zal de speler weten dat er bij een taak een 'wapen' te verkrijgen is. Door reparaties te laten maken en los hiervan erin de objecten verstoppen kunnen we zorgen dat de speler de volle ervaring krijgt maar toch nog enigszins zijn eigen invloed heeft op het spel. Door drie verschillende taken te laten doen waarin er met een klik van de muis het wapen kan gepakt worden, mits de speler het gevonden heeft, kunnen we de voortgang van het spel redelijk controleren. De speler moet zelf de nodige onderdelen voor het effect op het verhaal verzamelen. Na alle taken moeten de deuren open gaan, niet eerder. Zo gaat het door naar het volgende level: de ontknopingen.

¹³ The task at hand: De taak waar nu gefocust op moet worden, de belangrijkste taak.

Afhankelijk van het aantal wapens dat de speler heeft meegebracht zullen er aan het einde van het gevecht verschillende mogelijkheden uitspelen. Met 0 of 1 wapens verslaat de alien koningin jou en kom je niet meer terug, game over. Met 2 wapens heeft de speler de koningin langzaam verslagen, niet snel genoeg om de onomkeerbare technologie schade te voorkomen. De schade die aan de main control room is gedaan is zo veel dat de speler niet meer teruggaat naar zijn thuisplaneet, de aarde.

Bij drie wapens moet de speler een scène te zien krijgen waarin de alien koningin verslagen wordt en de astronaut contact krijgt met de aarde.

De narratieve aspecten

Te veel tekst maakt het saai, maar te weinig zorgt voor verwarring. Om in de game te zorgen dat de speler het verhaal goed begrijpt moet er een goede combinatie van tekst en visuele aspecten te zien krijgen. In het begin moet er natuurlijk redelijk wat tekst zijn om de controls en het plot van het spel uit te leggen. Maar tijdens de rest van het spel moet hier strategisch over nagedacht worden. De speler vindt een van de eerste extra mogelijke wapens, hoe weet hij/zij dat dit ook daadwerkelijk een extra wapen is, bedoeld voor de alien koningin? Hiervoor is een script nodig.

Dit script kunnen we onderverdelen in 5 subscripts: de opstelling, de complicatie, de ontwikkeling, de oplossing en de ontknoping. Deze delen zijn in het vorige hoofdstuk behandeld. Aangezien games een universeel bekend ding is, is er een goede bekende taal voor nodig. De tekst zal daarom in het Engels zijn.

"tekst dat op beeld komt"

Dit moet geanimeerd zijn

Dit gebeurt als de speler de game speelt

A LONE ASTRONAUT

De opstelling:

Na op start hebben gedrukt zal het scherm zwart worden en dan langzaam weg vagen om een gang in de ruimte.

"Where am I?"

Het scherm draait rond en laat een ruimtestation zien, donker en afgelegen.

"What?"

Het scherm beweegt zodat het duidelijk wordt dat het karakter opstaat. Het beeld vult het scherm en de speler wordt geïntroduceerd aan het spel. Het is nu in derde persoon.

De speler krijgt een introductie aan de controls en kan dit oefenen.

De aliens komen op beeld.

"What are those!?"

De speler krijgt instructies over hoe hij/zij deze kan verslaan en voltooit dit.

"Well aliens most certainly are not what I expected to see here... or at all for that matter."

Het is even stil en het scherm schakelt laat een leeg station zien, waar de astronaut rond loopt.

"Where is everyone?"

De astronaut loopt meer rond

"Am I the only one on board?"

De astronaut loopt naar een control paneel en merkt dat het ruimtestation het niet doet, en gewoon zweeft in de ruimte.

"Ughh engineering has never been my strong suit..."

"Seems like there's no choice but to fix the ship if I want to know what's going on..."

"Maybe someone on earth can tell me, I'd have to find the control room first though"

De complicatie:

Onderweg naar de control room ziet de astronaut dat de aliens langzaam het schip afbreken.

"What are they even doing?"

Bij elke ontmoeting van een groep aliens moet de speler deze verslaan.

"These things just keep coming!"

De speler rent naar de control room.

De speler moet hier eerst alle aliens verslaan om verder te kunnen

De astronaut sluit de deur en doet deze op slot en is veilig voor de aliens.

"Finally some peace."

"I'll check the cameras see where these things are coming from"

Het camera paneel licht op.

De speler moet hiernaar toe lopen en ermee interageren.

Het beeld laat verschillende camerabeelden zien en eentje wordt vergroot.

"This isn't like that one movie is it?"

Het beeld laat een hele grote alien zien die ervoor zorgt dat de aliens blijven komen. Het beeld slaat over naar de deur waarin aliens op slaan om binnen te willen komen.

"I guess it's like Aliens the movie... no wait, that's another film..."

"Alright, let's get to work, om a little rusty but it'll have to do."

Het beeld slaat over naar het normale overall view scherm en één paneel licht op.

De speler loopt ernaartoe en krijgt een introductie over hoe hij het circuit moet fixen.

Tijdens dit licht er een metalen pijp op met 'click me' erop.

"Hey! I can use that for that alien queen thing!"

De ontwikkeling:

Het beeld slaat weer over naar het normale overall view scherm en deze keer lichten drie panelen op.

"Only three more to go"

De speler loopt naar ze toe en repareert alle onderdelen. Tijdens deze reparaties bevinden zich andere metalen pijpen, die als de speler er op klikt opgepakt worden.

De oplossing:

De deur wordt opengeduwed door de aliens, dit zijn er een beetje meer dan ervoor.

"I just can't catch a break can I?"

"Let's find the big one and put an end to this!"

De speler vecht de aliens in de control room.

Er komt een gele pijl op het scherm die de weg naar de alien koningin laat zien.

De speler vindt de alien queen door de pijl te volgen en ondertussen aliens te vechten.

De ontknoping:

De deur van de ruimte waarin de alien koningin zich bevindt gaat open en de speler loopt naar binnen.

"That thing is way bigger in real life..."

"Certain death here I come?"

Er komen kleine instructies op het scherm over hoe de speler de grote alien kan verslaan door de grote gevonden 'wapens' te vinden.

De speler vecht de eindbaas.

Mogelijkheid einde 1:

De speler had niet genoeg grote wapens om de eindbaas te verslaan en er komt een scherm van game over nadat de eindbaas de astronaut heeft geraakt omdat hij geen wapens (meer) had.

Mogelijkheid einde 2:

De speler had genoeg wapens om de eindbaas te verslaan, maar niet genoeg om dit snel te doen, de eindbaas gaat pas nadat ze het schip ernstig heeft beschadigd.

De laatste scène is van de astronaut die erachter komt dat hij gewonnen heeft, maar geen contact meer kan opnemen met de aarde, hij zit vast op het station... alleen.

"At least astronaut Stan isn't here I guess..."

"Hell what am I saying, I'd even take that french bastard right now..."

"Anything but this"

Mogelijkheid einde 3:

De speler had alle wapens gevonden en verslaat zonder enige moeilijkheden de eindbaas.

"Suck on that alien trash! Whoo!"

Het scherm gaat naar een grote afbeelding van de aarde met een telefoon in het midden met drie steeds opkomende stipjes.

Er komen twee poppetjes aan beide kanten van het scherm, een astronaut en een andere astronaut in een andere kleur.

Astronaut: "Anyone there? This is space station Delta 7 do you read?"

Persoon van Aarde: "This is ALA 4, we read you loud and clear Delta 7"

Astronaut: "Somethings happened here, I'm gonna need an extraction pod."

Persoon van Aarde: "What happened?"

Astronaut: "Trust me, you don't wanna know"

Persoon van Aarde: "..."

Persoon van Aarde: "ETA four hours, hang tight in there"

Het visuele aspect, de koppeling van de levels en het narratieve aspect zijn nu af. De game moet een passend visuele uitstraling hebben. Zo komt het gevoel van het onderwerp het beste over. De koppeling van levels moet ook goed gebeuren, anders lijkt de game warrig en is het moeilijk om goed te volgen. Door goede koppelingen te maken en na te denken over wat er wanneer moet gebeuren kan er goede structuur in de game worden gebracht.

Het script is misschien wel een van de belangrijkste dingen die gemaakt moet worden. Door deze van te maken is er een goede én complete opbouw gemaakt van de game. Alles om de game te moeten maken is voorbereid.

Hoe ziet de technische toekomst van videogames eruit?

De toekomst van technologische ontwikkelingen heeft een grote invloed op videogames. Sommige worden speciaal ontworpen voor videogames en sommige zijn enkel met kijk op de technologie zelf, maar kunnen goed gebruikt worden voor games.

Virtual Reality is een van de ontwikkelingen die specifiek voor gameontwikkeling is gebouwd. Het zorgt ervoor dat het voelt alsof je daadwerkelijk in het spel zit. Dankzij een VR bril en twee controllers voor recht en links handige acties kun je spellen zoals Phasmophobia doen. Het spel Phasmophobia is een horror spel waarin je met vrienden online samen bovennatuurlijke wezens moet jagen en vastleggen. Dit is dus een spannend spel om te doen. De VR bril verbetert de ervaring hiervan. Doordat je als het ware echt in de wereld zit, (omdat je zicht meebeweegt met hoe je hoofd draait) wordt het spel nog spannender. Ook heb je Beat Saber, een VR ritmespel, totaal tegenovergesteld van Phasmophobia. Dankzij de VR functie is het spel interactiever, je moet daadwerkelijk de 'sabers' bewegen om de goede blokjes te raken aan de goede kant. VR wordt steeds beter, de eerste uitvinding echt als virtual reality te beschouwen was de Sensorama, uitgevonden door Morton Helig (zie bijlage 6). De Sensorama was een uitvinding uit 1962, en simuleerde een fietstocht door de stad Brooklyn. De machine gebruikt zicht, geluid, geur en beweging om een realistische middag te simuleren. Verder ging deze uitvinding toen echter niet omdat Helig niet genoeg financiering had voor dit project.

Als we kijken naar het verschil tussen toen en nu, is er te verwachten dat VR al snel zich nog meer zal ontwikkelen. Nu kunnen we spellen zoals Phasmophobia spelen, kleine wereld spellen met gelimiteerde beweging opties. Over 20 jaar kunnen we waarschijnlijk in grote wereld games met meerdere bewegings opties kunnen spelen. Het eerder genoemde spel, Baldur's Gate

3, is een open wereld spel met veel acties, van beweging tot spreken. De mogelijkheden van VR zijn nog niet zo groot, wel wordt hier constant aan gewerkt. In de toekomst kunnen spellen zoals Baldur's Gate 3 dus misschien wel in VR zijn.

Een andere ontwikkeling die steeds meer in videogames wordt gebruikt is AI. Dit wordt voornamelijk gebruikt om beslissingen te maken. Algoritmes waarmee de computer kan besluiten welke bewegingen gemaakt moeten worden door bijvoorbeeld vijanden in een game. Als een karakter dat geen lange afstand wapens heeft wilt aanvallen, dan moet de computer beslissen dat dit karakter dichterbij komt en bij een bepaalde afstand zijn aanval doet. Een ander voorbeeld zijn de algoritmes die deze bewegingen omzetten in daadwerkelijke animaties. De beslissingen die gemaakt worden gaan gepaard met de algoritmes waarmee de objecten bewegen. Als een karakter dichtbij staat en een aanval doet dan zal dit (in sommige gevallen) eerder een stealth attack¹⁴ zijn, Als je als speler een vijand van een iets grotere afstand aanvalt, kan het zijn dat je hiermee wel karakters in de game mee waarschuwt en dus je plek verradt. Dit moet allemaal in een paar seconden worden beslist door de computer, de AI.

¹⁴ Stealth Attack: Een aanval die op een sluipende manier wordt gemaakt, het blijft onopgemerkt waardoor geen andere vijanden in de buurt door hebben dat de speler er is.

AI en VR zijn dan ook heel goed te combineren. Op de open dag van Breda University of Applied Sciences, was er een demonstratie van de combinatie van AI en de gameontwikkeling. Er liep iemand rond in een pak waar bewegingssensoren op zaten. Erachter bevond zich een heel groot televisie scherm met daarop een groot fantasy wezen in een lege ruimte. Na goed te kijken tussen de persoon en het scherm was te zien dat elke beweging van de persoon over werd genomen naar het karakter op het scherm.

De AI die geprogrammeerd was nam de taak op van het wezen op het scherm te laten bewegen en hierbij de goede verandering van het wezen aan te geven. Zette de persoon een stap naar voren, dan zette het wezen op het scherm dat ook. Niet alleen de plek van het wezen was veranderd, maar ook het 3D object wat het beeld van het karakter laat zien veranderde om een stap na te bootsen. Dit gebeurde allemaal in één keer, onder de controle van AI.

De voorspelling bij VR is dus te bereiken met AI. In de toekomst is er een grote kans dat het mogelijk is om kleine sensoren op je kleding te plakken op plekken zoals je handen, ellebogen, knieën en voeten, een nabootsing van het pak. Samen met een controller voor beide handen en een VR bril zou je in een open wereld geplaatst kunnen worden en kunnen spelen alsof je daadwerkelijk een karakter was in dit spel, alsof dit jouw nieuwe realiteit is. Een heel goed voorbeeld hiervan is Sword Art Online, dit is een sci-fi serie waarin de verhalen van mensen in een fantasiewereld worden afgespeeld. De mensen hierin bevinden zich in hun 'echte leven' in hun kamer thuis. Het is sci-fi, maar het heeft een grote kans om waarheid te worden.

Conclusie

Na dit onderzoek naar deze engines is er geconcludeerd dat Godot de beste keuze is, het is uitdagend maar niet onhaalbaar om mee te kunnen werken. GameMaker is te simpel en Unity en Unreal zijn voor dit project geen slimme keuzes vanwege de kennis die je al moet hebben om deze te gebruiken. Voor een echte beginner in de game ontwikkeling, is GameMaker een perfecte start om je vaardigheden te ontwikkelen, kan je geen code dan kun je eerst leren hoe de samenhang werkt met GML visual, heb je dit onder de knie dan kun je beginnen met de kennis die je hebt en hierop code toepassen. Je kunt met GameMaker eerst de basics leren voordat je verder gaat met gevorderde dingen.

Een game is interessant te maken door er een verhaal aan toe te voegen. Verhalen zijn volgens meerdere gamers een belangrijk deel. Voor sommigen is het niet belangrijk en vinden games zoals shooters fijner, wat de meeste in gemeen hebben is het online co-op¹⁵ genre. Verbinding vinden is iets wat veel gamers doen, dit komt naar voren uit onderzoek aangezien een groot deel van de gevraagde (bijlage 3) als favoriete game WoW¹⁶. Dit spel heeft zowel meerdere verhaallijnen als ondersteuning om online met vrienden het spel te spelen. Ook komt eruit, dat zelf je eigen verhaallijn van de game te kunnen beïnvloeden, een grote bijdrage kan geven aan hoe interessant de game is. Meerdere mogelijkheden zorgen ervoor dat er meer te doen is. Het geeft je het gevoel dat je in de wereld van de game zit en dat je eigen keuzes invloed hebben. Snelle games zorgen ervoor dat men de aandacht erbij moet houden en dat het niet saai wordt. Games zoals deze worden sneller populair dan de meeste.

Met een scene tree is de game overzichtelijk te maken. Het tweede model kan worden geïmplementeerd bij het maken van de

¹⁵ Online co-op: Online Cooperation, online kunnen spelen met meerdere mensen.

¹⁶ WoW: World of Warcraft, een video game.

game, de sprites, de geluiden en de gameplay is wat voor de details zorgen. Het is belangrijk om te weten hoe een verhaal in een game in elkaar zit. Door de opbouw op te schrijven en het model van Bizzocchi te gebruiken is er eenvoudig een verhaal te creëren. Dankzij een scene tree is het gemaakte verhaal overzichtelijk te implementeren in de game en kan er één voor één aan de onderdelen worden gewerkt.

De game moet natuurlijk ook een passend visuele uitstraling hebben. Zo komt het gevoel van het onderwerp het beste over. De koppeling van levels moet ook goed gebeuren, anders lijkt de game warrig en is het moeilijk om goed te volgen. Door goede koppelingen te maken en na te denken over wat er wanneer moet gebeuren kan er goede structuur in de game worden gebracht.

Het script is misschien wel een van de belangrijkste dingen die gemaakt moet worden. Door deze van te maken is er een goede én complete opbouw gemaakt van de game. Alles om de game te moeten maken is voorbereid.

Hoeverre de ontwikkeling van AI in deze industrie komt is nog maar een gok. Wel kan er een goed beredeneerde gok gemaakt worden als we kijken naar wat de mens tot nu toe al bereikt heeft en waar er aan gewerkt wordt. In de toekomst is er een grote kans dat het mogelijk is om kleine sensoren op je kleding te plakken op plekken zoals je handen, ellebogen, knieën en voeten, een nabootsing van het pak. Samen met een controller voor beide handen en een VR bril zou je in een open wereld geplaatst kunnen worden en kunnen spelen alsof je daadwerkelijk een karakter was in dit spel, alsof dit jouw nieuwe realiteit is. Een heel goed voorbeeld hiervan is Sword Art Online, dit is een sci-fi serie waarin de verhalen van mensen in een fantasiewereld worden afgespeeld. De mensen hierin bevinden zich in hun 'echte leven' in hun kamer thuis. Het is sci-fi, maar het heeft een grote kans om waarheid te worden.

Literatuur

- Engine, G. (z.d.). *Godot Engine - free and open source 2D and 3D game engine*. Godot Engine. <https://godotengine.org/>
(Geraadpleegd op 2 oktober)
- YoYo Games. (z.d.). *Make 2D games with GameMaker | free video game maker*. GameMaker. <https://gamemaker.io/en>
(Geraadpleegd op 2 oktober)
- *Unity Real-Time Development Platform | 3D, 2D, VR & AR Engine*. (z.d.). <https://unity.com/>
(Geraadpleegd op 2 oktober)
- *Unreal Engine | The most powerful real-time 3D creation tool*. (z.d.). Unreal Engine. <https://www.unrealengine.com/en-US>
(Geraadpleegd op 2 oktober)
- RetroAnalyst. (2020, 1 oktober). Asteroids: By the Numbers. *RETROGAME DECONSTRUCTION ZONE*.
<https://www.retrogamedeconstructionzone.com/2019/10/asteroids-by-numbers.html>
(Geraadpleegd op 13 november 2023)
- Bizzocchi, J. (2006). Games and Narrative: An Analytical Framework. *the Journal of the Canadian Games Studies Association*, 1(1).
<http://www.sfu.ca/~bizzocch/documents>Loading-citation.pdf>
(Geraadpleegd op 14 november)

- Carlquist, J. (2003). *Playing the story: Computer games as a narrative genre*. Carlquist | Human IT: Journal for Information Technology Studies as a Human Science.
<https://humanit.hb.se/article/view/144/148>
(Geraadpleegd op 14 november)
- Filipović, A. (2023). THE ROLE OF ARTIFICIAL INTELLIGENCE IN VIDEO GAME DEVELOPMENT. *Kultura Polisa*, 20(3), 50–67.
<https://doi.org/10.51738/kpolisa2023.20.3r.50f>
(Geraadpleegd op 16 november)
- GameMaker. (2022, 3 november). *How to make a classic arcade game in GameMaker* [Video]. YouTube.
<https://www.youtube.com/watch?v=nwlvT-L9vFg>
(Geraadpleegd op 16 november)
- Kaan Alpar. (2023, 17 maart). *How to make asteroids in Godot 4 (Complete tutorial)* [Video]. YouTube.
<https://www.youtube.com/watch?v=FmIo8iBV1W8>
(Geraadpleegd op 16 november)
- Lin, B. A. (2007). *Narrative Interface design: The use of interface elements to enhance the narrative experience in videogames*.
https://summit.sfu.ca/_flysystem/fedora/sfu_migrate/8286/etd3174.pdf
(Geraadpleegd op 4 december)

- *History.* (z.d.).
<https://cs.stanford.edu/people/eroberts/cs181/projects/2010-11/VirtualWorlds/styled/page1.html>
 (Geraadpleegd op 21 december)
- *De geschiedenis van virtual reality.* (z.d.). Vrendle.
<https://vrendle.nl/blog/de-geschiedenis-van-virtual-reality>
 (Geraadpleegd op 21 december)
- Millington, I. (2019). *AI for Games* (3de editie). Taylor & Francis Group.
file:///home/chronos/u-c7d2325520afb129861f065f6b7c68e2cca46df3/MyFiles/Downloads/9781351053303_previewpdf.pdf
 (Geraadpleegd op 21 december)
- Kaan Alpar. (2023, 13 mei). *2D platformer Spritesheet Animations (Godot 4)* [Video]. YouTube.
<https://www.youtube.com/watch?v=sVJEaYNOUNw>
 (Geraadpleegd op 11 januari)
- Chris' Tutorials. (2022, 6 juli). *Godot 4 Crash Course for Beginners - GameDev 2D Top Down Tutorial* [Video]. YouTube.
<https://www.youtube.com/watch?v=Luf2Kr5s3BM>
 (Geraadpleegd op 11 januari)
- Coco Code. (2023, 18 oktober). *Start Your Game Creation Journey Today! (Godot beginner tutorial)* [Video]. YouTube.
<https://www.youtube.com/watch?v=5V9f3MT86M8>
 (Geraadpleegd op 19 januari)

- DevWorm. (2023, 3 april). *How to Create an RPG in Godot 4 (step by step)* [Video]. YouTube.
<https://www.youtube.com/watch?v=pBoXqW4RyE>
(Geraadpleegd op 19 januari)

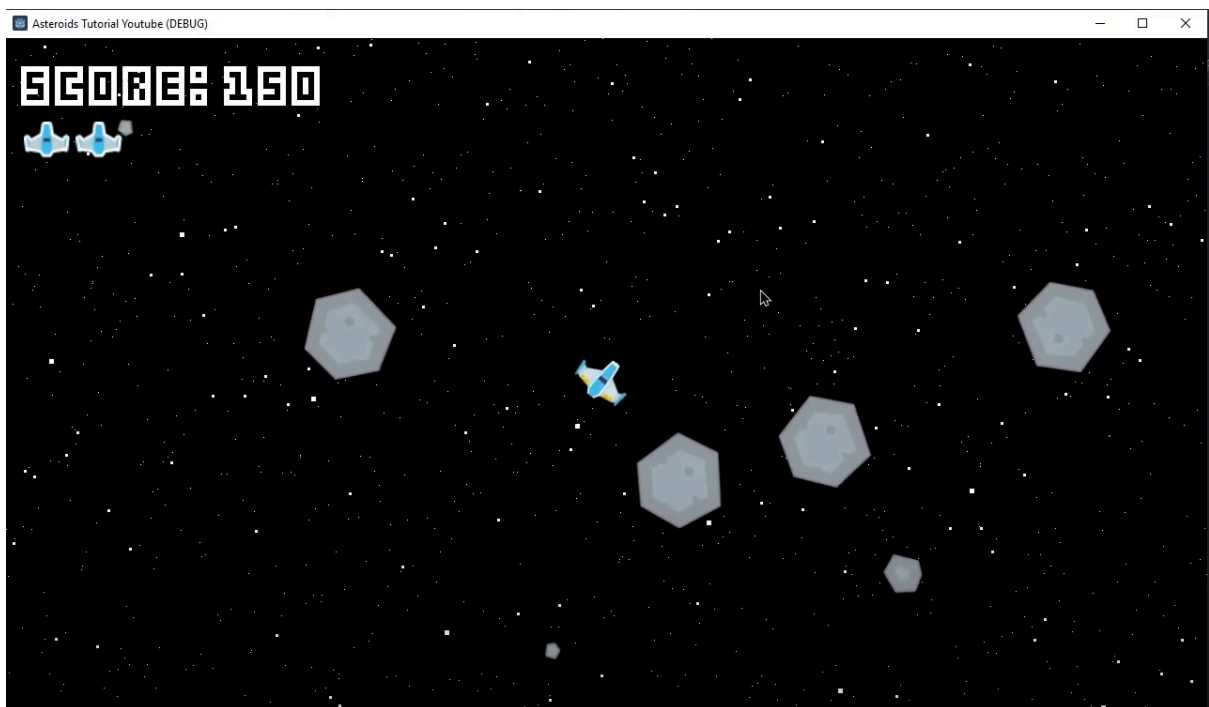
Bijlage

Bijlage 1

Schermafbeeldingen van het proces van de game *Asteroids* in Godot en Gamemaker.

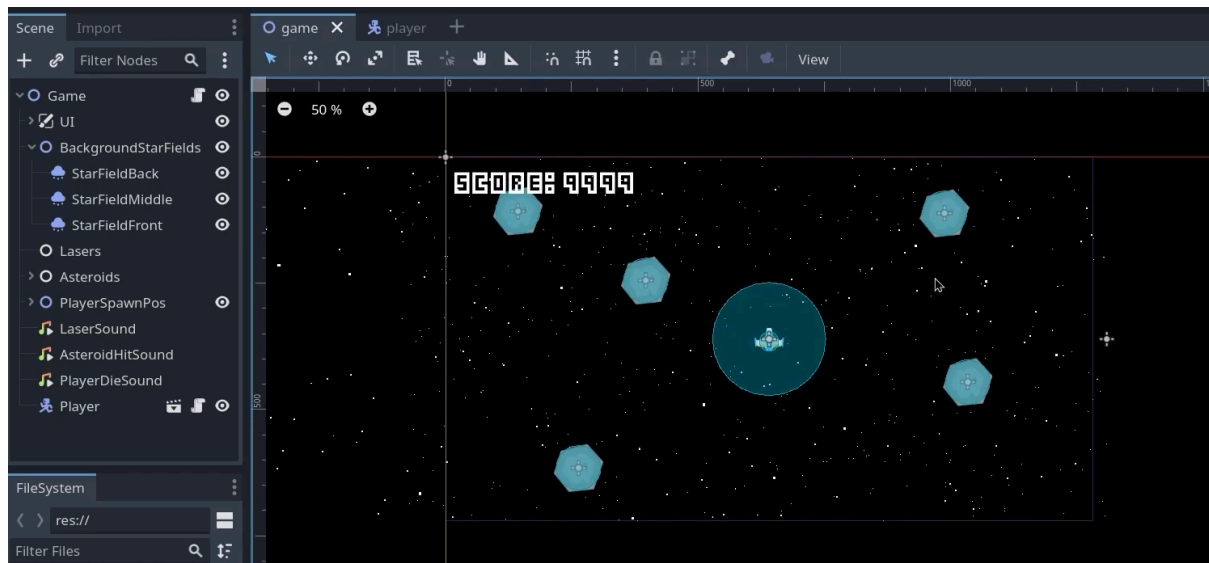
Godot:

Onderstaand is een in-game screenshot van het eindproduct van de test in Godot.



Onderstaand is een screenshot waarin de objecten zich bevinden. Links is te zien hoe alle objecten met elkaar verbonden zijn. Deze zijn hier onderling gekoppeld en kunnen

zo meegenomen worden in de scripts.



Asteroid.gd:

Regel 1, de class name extends, koppelt de code aan het aangemaakte Area 2D van de asteroïde.

Erna worden verschillende variabelen aangemaakt. Een var is een normale variabele, een enum is een variabele waar in dit geval drie mogelijke waarden aan zijn gegeven. Een @export voor de variabele zorgt ervoor dat de variabele uit het script wordt gehaald en naast de code te zien is ook als de code is weggeklikt. Dit zorgt ervoor dat de variabele snel met één klik verwijderd kan worden, bij de ene variabele kan dit gebruikt worden om de grootte van de variabele te testen. Een @onready zorgt ervoor dat de scene niet over load, dit keyword zorgt ervoor dat de variabele pas laad als het nodig is.

Bij de punt variabele wordt de code eronder geraadpleegd om te kijken hoeveel punten gegeven moeten worden bij de grootte van een asteroïde.

```

1  class_name Asteroid extends Area2D
2
3  signal exploded(pos, size, points)
4
5  var movement_vector := Vector2(0, -1)
6
7  enum AsteroidSize{LARGE, MEDIUM, SMALL}
8  @export var size := AsteroidSize.LARGE
9
10 var speed := 200
11
12 @onready var sprite = $Sprite2D
13 @onready var cshape = $CollisionShape2D
14
15 ▼ var points: int:
16 ▼ | get:
17 ▼ |   | match size:
18 ▼ |   |   | AsteroidSize.LARGE:
19 |   |   |   | return 100
20 ▼ |   |   | AsteroidSize.MEDIUM:
21 |   |   |   | return 50
22 ▼ |   |   | AsteroidSize.SMALL:
23 |   |   |   | return 25
24 ▼ |   |   | _:
25 |   |   |   | return 0

```

Hier worden met de enum variabele verschillende vormen en plaatjes gegeven aan het object asteroïde, dit is wat ervoor zorgt dat het van vorm verandert.

```

▼ func _ready():
  | rotation = randf_range(0, 2*PI)
  |
  ▼ | match size:
  ▼ | | AsteroidSize.LARGE:
  | | | speed = randf_range(50,100)
  | | | sprite.texture = preload("res://assets/textures/meteorGrey_big2.png")
  | | | cshape.shape= preload("res://resources/asteroid_cshape_large.tres")
  ▼ | | AsteroidSize.MEDIUM:
  | | | speed = randf_range(100,150)
  | | | sprite.texture = preload("res://assets/textures/meteorGrey_med1.png")
  | | | cshape.shape= preload("res://resources/asteroid_cshape_medium.tres")
  ▼ | | AsteroidSize.SMALL:
  | | | speed = randf_range(100,200)
  | | | sprite.texture = preload("res://assets/textures/meteorGrey_tiny1.png")
  | | | cshape.shape= preload("res://resources/asteroid_cshape_small.tres")
  | # print(speed) om te checken of de speed ook echt 'random' veranderd

```

Hieronder wordt de radius en de plaats van de vorm geraadpleegd. Zo kan de computer bepalen wanneer de asteroïde het scherm verlaat, zal het weer verschijnen aan de andere kant.

Bij de functie 'explode', explodeert de asteroïde en wordt de grootte gecheckt en punten gegeven. Hieronder wordt aangegeven dat als de player in de body van een asteroïde komt dat de speler dan 'doodgaat'.

```
44 func _physics_process(delta):
45     global_position += movement_vector.rotated(rotation) * speed * delta
46
47     var radius = cshape.shape.radius
48     var screen_size = get_viewport_rect().size
49     if (global_position.y+radius) < 0:
50         global_position.y = (screen_size.y+radius)
51     elif (global_position.y-radius) > screen_size.y:
52         global_position.y = -radius
53     if (global_position.x+radius) < 0:
54         global_position.x = (screen_size.x+radius)
55     elif (global_position.x-radius) > screen_size.x:
56         global_position.x = -radius
57
58 func explode():
59     emit_signal("exploded", global_position, size, points)
60     queue_free()
61
62 func _on_body_entered(body):
63     if body is Player:
64         var player = body
65         player.die()
66
```

Game.gd:

```
1 extends Node2D
2
3 @onready var lasers = $Lasers
4 @onready var player = $Player
5 @onready var asteroids = $Asteroids
6 @onready var hud = $UI/HUD
7 @onready var game_over_screen = $UI/GameOverScreen
8 @onready var player_spawn_pos = $PlayerSpawnPos
9 @onready var player_spawn_area = $PlayerSpawnPos/PlayerSpawnArea
```

Hier wordt de score en de levens die de player heeft bijgehouden. Bij de volgende functie wordt de game opgestart, alle variabelen worden gereset en de game kan (opnieuw) beginnen.

```
13 ▼ var score := 0:
14 ▼ » set(value):
15 » » score = value
16 » » hud.score = score
17
18 ▼ var lives: int:
19 ▼ » set(value):
20 » » lives = value
21 » » hud.init_lives(lives)
22
23 ▼ func _ready():
24 » game_over_screen.visible = false
25 » score = 0
26 » lives = 3
27 » player.connect("laser_shot", _on_player_laser_shot)
28 » player.connect("died", _on_player_died)
29 »
30 ▼ » for asteroid in asteroids.get_children():
31 » » asteroid.connect("exploded", _on_asteroid_exploded)
32
33 ▼ func _process(delta):
34 ▼ » if Input.is_action_just_pressed("reset"):
35 » » get_tree().reload_current_scene()
```

De `.add_child` functie zorgt voor een extra object gekoppeld aan het object laser. Als deze laser bij de volgende functie de asteroïde raakt, verandert de asteroïde van vorm, als het de SMALL asteroïde is zal deze verdwijnen, 'pass'.

Bij elke geëxplodeerde asteroïde zal er een extra asteroïde deelnemen op het scherm komen, zo gaat het spel door.

```

37  ▾ func _on_player_laser_shot(laser):
38      >| $LaserSound.play()
39      >| lasers.add_child(laser)
40
41  ▾ func _on_asteroid_exploded(pos, size, points):
42      >| $AsteroidHitSound.play()
43      >| score += points
44  ▾ >| for i in range(2):
45  ▾ >|     match size:
46  ▾ >|     >|     Asteroid.AsteroidSize.LARGE:
47  ▾ >|     >|     >| spawn_asteroid(pos, Asteroid.AsteroidSize.MEDIUM)
48  ▾ >|     >|     Asteroid.AsteroidSize.MEDIUM:
49  ▾ >|     >|     >| spawn_asteroid(pos, Asteroid.AsteroidSize.SMALL)
50  ▾ >|     >|     Asteroid.AsteroidSize.SMALL:
51  ▾ >|     >|     >| pass
52
53  ▾ func spawn_asteroid(pos, size):
54      >| var a = asteroid_scene.instantiate()
55      >| a.global_position = pos
56      >| a.size = size
57      >| a.connect("exploded", _on_asteroid_exploded)
58      >| asteroids.call_deferred("add_child", a)

```

Als de speler dood gaat zal er een leven afgaan, als er geen levens meer zijn dan zal er een game over screen komen, als er wel levens over zijn komt er een kleine time out voordat er weer een schip in beeld komt en je verder kan.

```

60  ▾ func _on_player_died():
61      >| $PlayerDieSound.play()
62      >| lives -= 1
63      >| player.global_position = player_spawn_pos.global_position
64  ▾ >| if lives <= 0:
65      >|     >| await get_tree().create_timer(2).timeout
66      >|     >| game_over_screen.visible = true
67  ▾ >| else:
68      >|     >| await get_tree().create_timer(1).timeout
69  ▾ >|     >| while !player_spawn_area.is_empty:
70      >|     >|     >| await get_tree().create_timer(0.1).timeout
71      >|     >|     >| player.respawn(player_spawn_pos.global_position)
72

```

Game_over_screen.gd:

Bij de restart button herlaad de game.gd scene en kan het spel opnieuw gespeeld worden.

```

1  extends Control
2
3  ▾ func _on_restart_button_pressed():
4      >| get_tree().reload_current_scene()
5

```

Hud.gd:

Dit geeft de waarden van de variabelen die veranderen. De levens en de score, het zorgt ervoor dat de score en de levens zichtbaar zijn op het scherm.

(Zowel hier als in de volgende foto geeft hij een rode lijn bij het laden van de scène, echter werkt dit wel, het geeft aan dat de scène niet gevonden wordt maar de game runt wel en laat de scène zien.)

```

1  | extends Control
2
3  ▾ @onready var score = $Score:
4  ▾ >| set(value):
5      >| >| score.text = "SCORE: " + str(value)
6
7  var uilife_scene = preload("res://scenes/ui_life.tscn")
8
9  @onready var lives = $Lives
10
11 ▾ func init_lives(amount):
12 ▾ >| for ul in lives.get_children():
13     >| >| ul.queue_free()
14 ▾ >| for i in amount:
15     >| >| var ul = uilife_scene.instantiate()
16     >| >| lives.add_child(ul)
17

```

Laser.gd:

Deze code geeft eigenlijk alleen de richting waarin de laser wordt geschoten. Door de movement vector `.rotation(rotation)` krijgt de laser de richting van de kant waar het schip naar wijst. Zo heb je controle over waar het schip de lasers schiet.

Ook wordt er hier aangegeven dat als de laser een asteroïde raakt, deze explodeert en de laser verdwijnt (queue_free).

```
1 extends Area2D
2
3 @export var speed := 500.0
4
5 var movement_vector := Vector2(0, -1)
6
7 func _physics_process(delta):
8     global_position += movement_vector.rotated(rotation) * speed * delta
9
10 func _on_visible_on_screen_notifier_2d_screen_exited():
11     queue_free()
12
13 func _on_area_entered(area):
14     if area is Asteroid:
15         var asteroid = area
16         asteroid.explode()
17         queue_free()
18
```

Player.gd:

Hieronder wordt de rate_of_fire aangegeven. Zonder de onderstaande code zouden de lasers op milliseconde naast elkaar afschieten, wat zorgt voor een witte lijn. Om bursts van lasers te krijgen wordt er een kleine time-out toegevoegd dat 0.15 seconde duurt.

```

1  class_name Player extends CharacterBody2D
2
3  signal laser_shot(laser)
4  signal died
5
6  @export var acceleration := 10.0
7  @export var max_speed := 350.0
8  @export var rotation_speed := 250.0
9
10 @onready var muzzle = $Muzzle
11 @onready var sprite = $Sprite2D
12 @onready var cshape = $CollisionShape2D
13
14 var laser_scene = preload("res://scenes/laser.tscn")
15
16 var shoot_cd = false
17 var rate_of_fire = 0.15
18
19 var alive := true
20
21 func _process(delta):
22     if !alive: return
23
24     if Input.is_action_pressed("shoot"):
25         if !shoot_cd:
26             shoot_cd = true
27             shoot_laser()
28             await get_tree().create_timer(rate_of_fire).timeout
29             shoot_cd = false

```

Deze onderstaande code zorgt voor de rotatie en beweging van het schip. Als er op W,A,S,D, wordt gedrukt dan gaat het schip de bijbehorende richting in. Dit gebeurt met rotate_right/left en move_backward/forward. Deze hebben in de eigenschappen van de game engine een gekoppelde toets (W,A,S,D)

Daaronder is net als bij de asteroïde code geschreven dat ervoor zorgt dat het schip op het scherm blijft, als het schip het scherm verlaat zal deze aan de andere kant weer verschijnen.


```

31 func _physics_process(delta):
32     if !alive: return
33
34     var input_vector := Vector2(0, Input.get_axis("move_forward", "move_backward"))
35
36     velocity += input_vector.rotated(rotation) * acceleration
37     velocity = velocity.limit_length(max_speed)
38
39     if Input.is_action_pressed("rotate_right"):
40         rotate(deg_to_rad(rotation_speed*delta))
41     if Input.is_action_pressed("rotate_left"):
42         rotate(deg_to_rad(-rotation_speed*delta))
43
44     if input_vector.y == 0:
45         velocity = velocity.move_toward(Vector2.ZERO, 3)
46
47     move_and_slide()
48
49     var screen_size = get_viewport_rect().size
50     if global_position.y < 0:
51         global_position.y = screen_size.y
52     elif global_position.y > screen_size.y:
53         global_position.y = 0
54     if global_position.x < 0:
55         global_position.x = screen_size.x
56     elif global_position.x > screen_size.x:
57         global_position.x = 0
58

```

Hieronder bevindt zich eerst de code voor de actie van het schieten zelf. Hier geeft het de rotatie aan waar in het script van de laser verder mee gewerkt wordt zoals uitgelegd. Eronder test het programma of de variabele True of False is met ==. Door de twee waarden te vergelijken kan het zien wanneer de actie van 'die' nodig is. Als het schip geraakt is en dood is gegaan zal de sprite ervan onzichtbaar worden en zal het signaal aangeven om een reset of een extra leven te geven.

Bij de functie respawn gebeurt het tegenovergestelde, het vergelijkt de variabele weer, als het 'alive' is dan zal de sprite weer zichtbaar worden. Ook neemt hij de globale positie weer aan waar het schip zich eerst in bevond. Natuurlijk start je opnieuw en wordt de vector van de snelheid dus naar 0 gezet.

```

59  ▾ func shoot_laser():
60    »  var l = laser_scene.instantiate()
61    »  l.global_position = muzzle.global_position
62    »  l.rotation = rotation
63    »  emit_signal("laser_shot", l)
64
65  ▾ func die():
66  ▾ »  if alive==true:
67    »  »  alive = false
68    »  »  sprite.visible = false
69    »  »  cshape.set_deferred("disabled", true)
70    »  »  emit_signal("died")
71    »  »
72
73  ▾ func respawn(pos):
74  ▾ »  if alive==false:
75    »  »  alive = true
76    »  »  global_position = pos
77    »  »  velocity = Vector2.ZERO
78    »  »  sprite.visible = true
79    »  »  cshape.set_deferred("disabled", false)
80

```

Player_spawn_area.gd:

Dit spreekt net als de game_over.gd voor zich. Hier spawned de asteroïde op de plek waar hij origineel vandaan kwam.

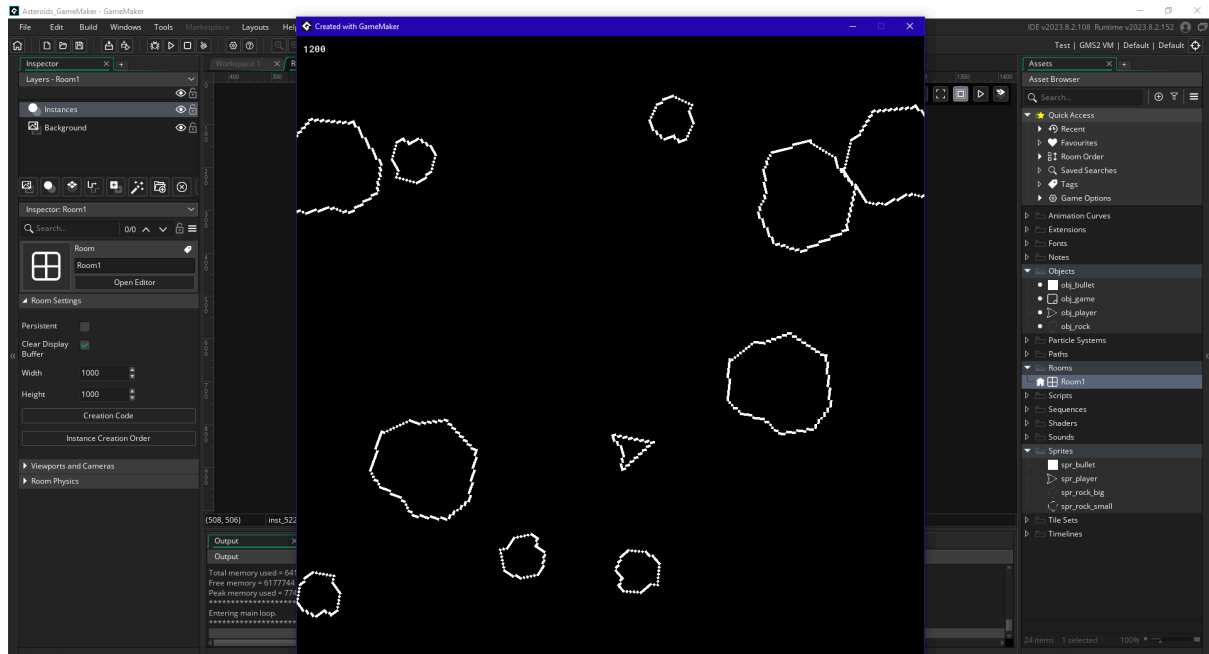
```

1  extends Area2D
2
3  ▾ var is_empty: bool:
4  ▾ »  get:
5    »  »  return (!has_overlapping_areas() && !has_overlapping_bodies())
6

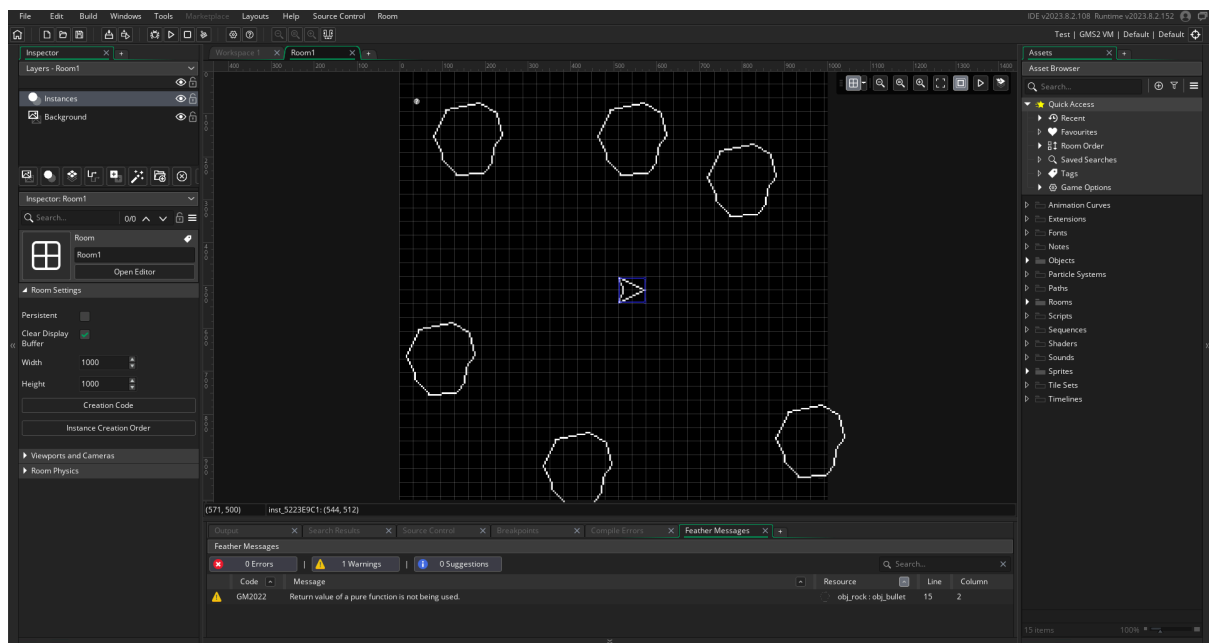
```

Gamemaker:

Onderstaand is een in-game screenshot van het eindproduct van de test in Gamemaker.

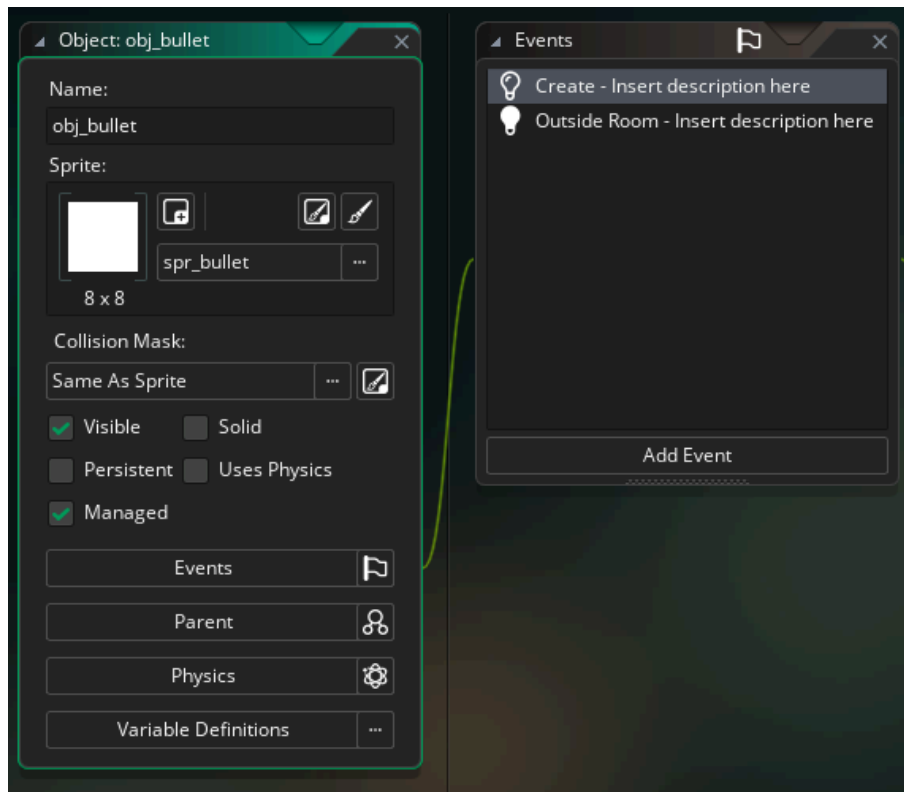


Onderstaand is een screenshot van de 'room'¹⁷ waarin de objecten zich bevinden.

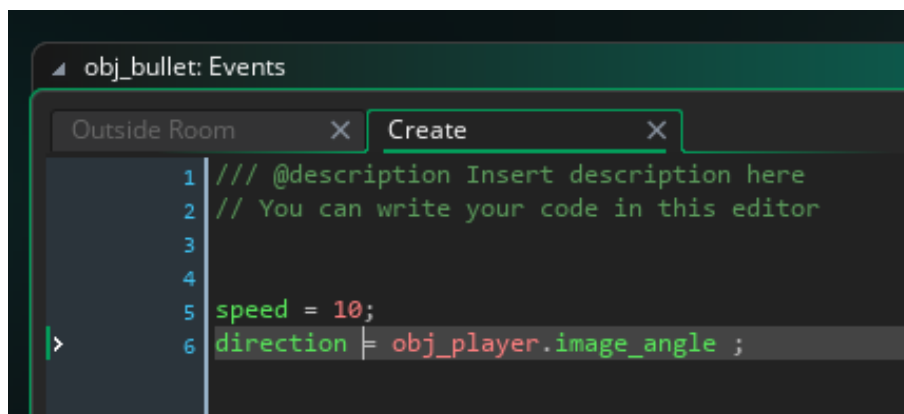


¹⁷ Room: Een 'kamer' waarin objecten geplaatst kunnen worden, alles in de room komt na het starten van de game op het scherm te staan, mits het een sprite heeft.

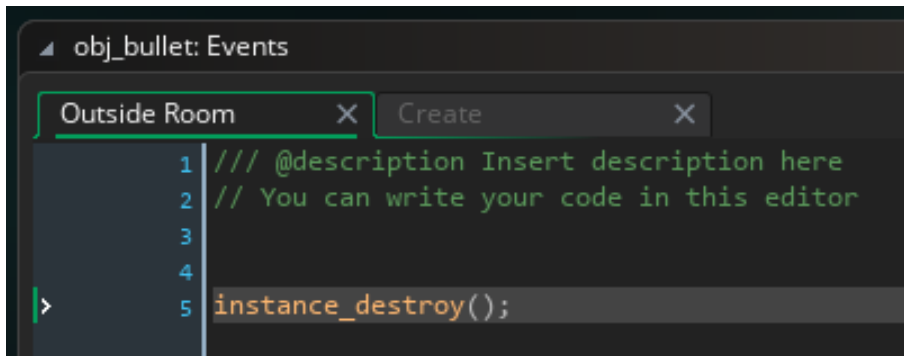
Hier zijn alle acties en eigenschappen verbonden aan het object 'bullet'



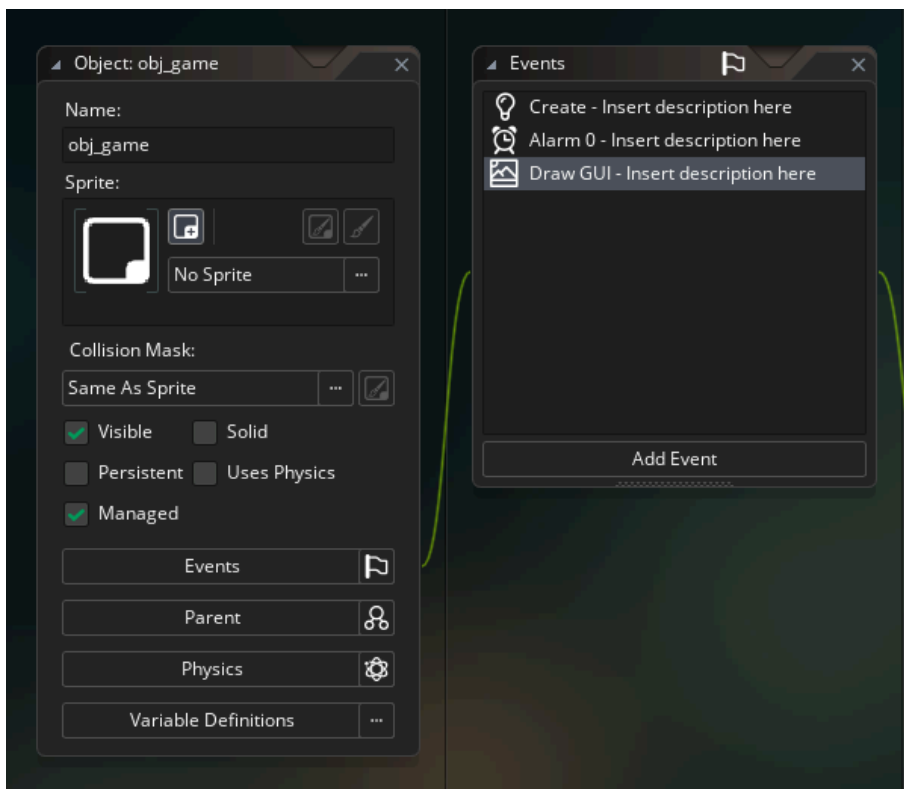
Speed geeft de snelheid van de 'bullet' aan, de image_angle geeft de directie van waaruit de bullet schiet (de image angle van het object 'player').



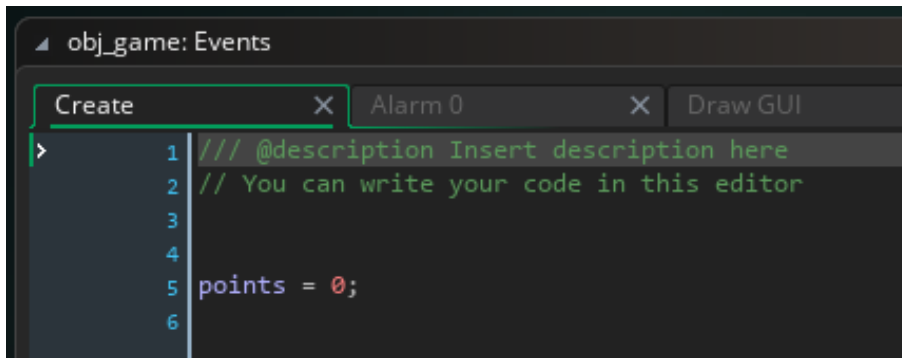
Dit zorgt ervoor dat de bullet stopt met bestaan als het uit het scherm gaat.



Hier zijn alle acties en eigenschappen verbonden aan het object 'game', deze heeft geen sprite omdat het puur voor room functies is.

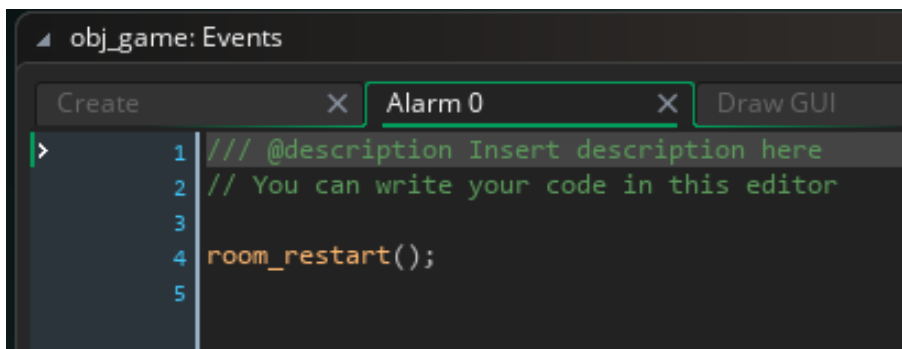


Hierbij worden bij de start van de game de punten op 0 gezet.



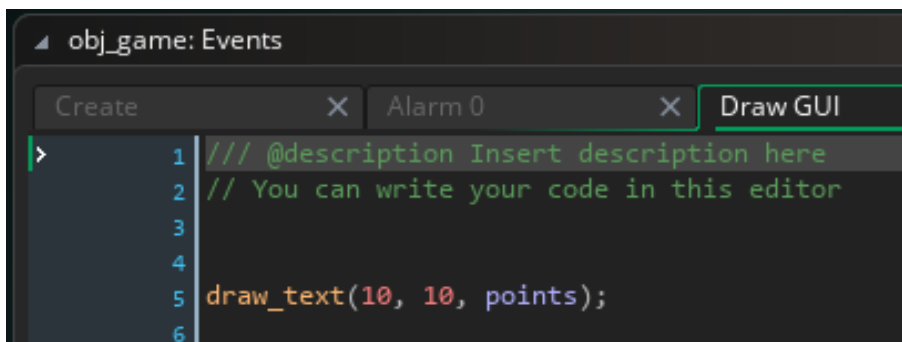
```
obj_game: Events
Create X Alarm 0 X Draw GUI
> 1 /// @description Insert description here
  2 // You can write your code in this editor
  3
  4
  5 points = 0;
  6
```

Dit spreekt voor zich, het start de room opnieuw, het reset.



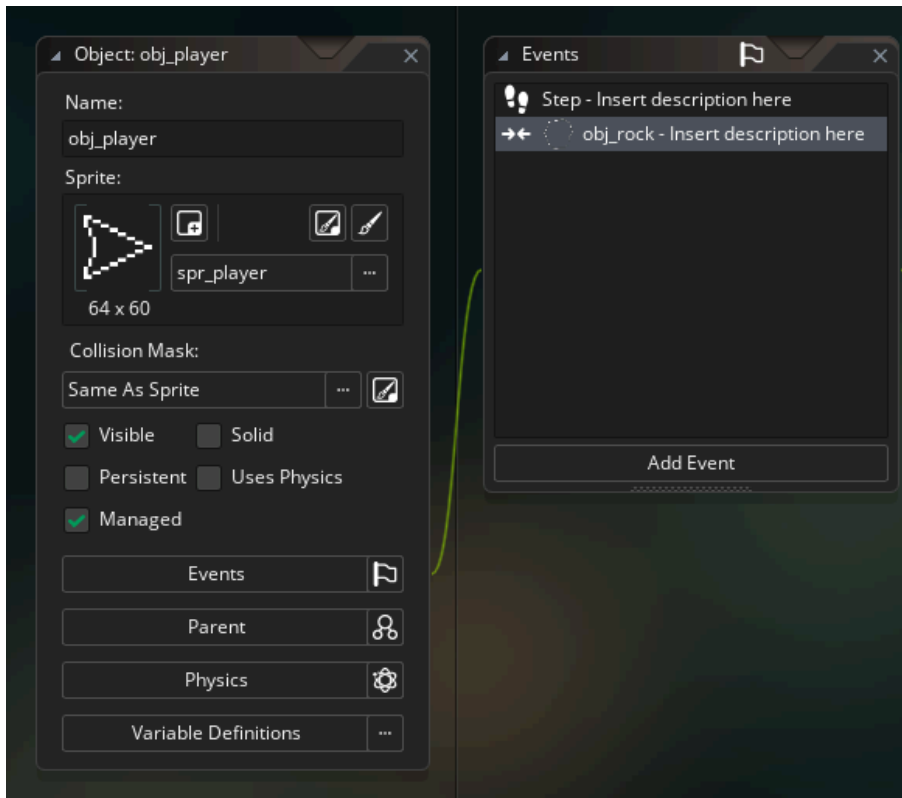
```
obj_game: Events
Create X Alarm 0 X Draw GUI
> 1 /// @description Insert description here
  2 // You can write your code in this editor
  3
  4 room_restart();
  5
```

Hierbij zorg je ervoor dat de score van de punten boven in de hoek komt (10 pixels van de rand af).



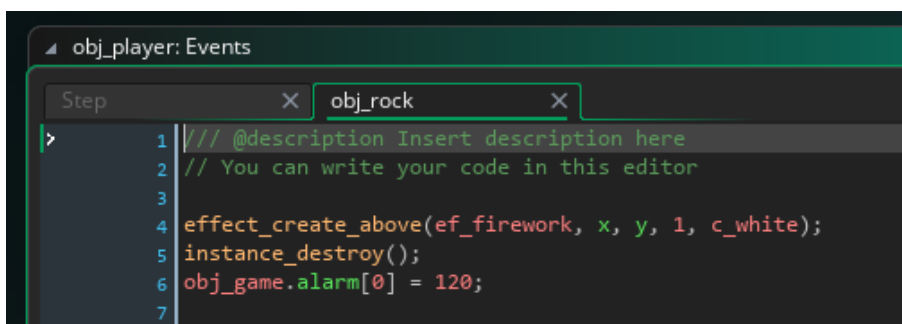
```
obj_game: Events
Create X Alarm 0 X Draw GUI
> 1 /// @description Insert description here
  2 // You can write your code in this editor
  3
  4
  5 draw_text(10, 10, points);
  6
```

Hier zijn alle acties en eigenschappen verbonden aan het object 'bullet'



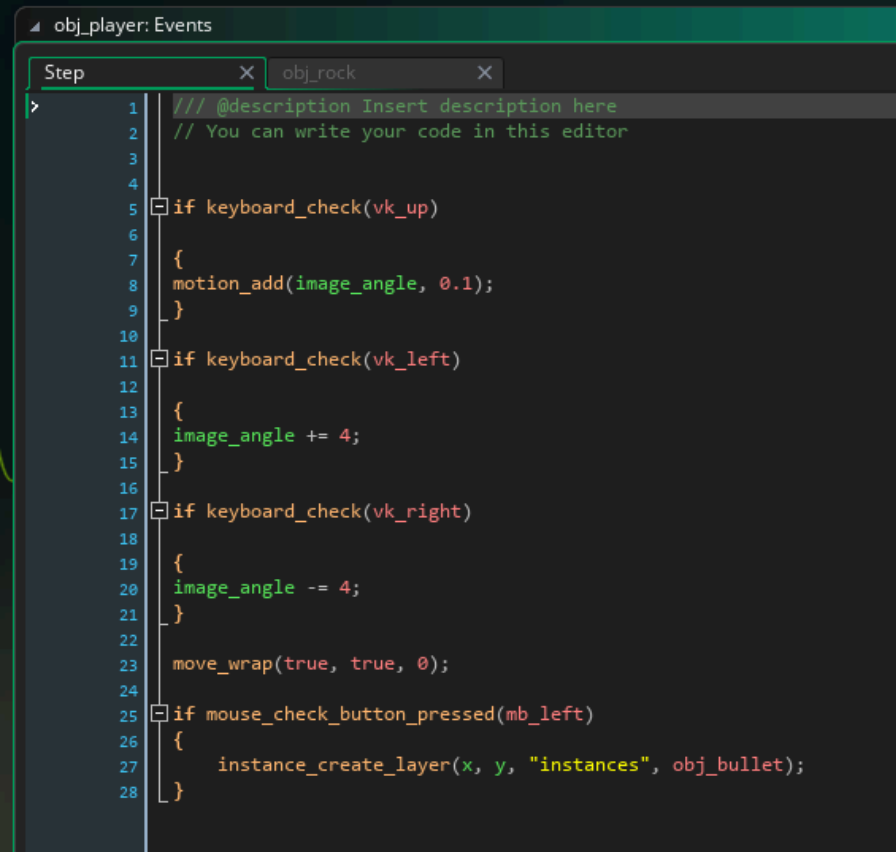
Zoals boven te zien is dit een collision actie, dit gebeurt alleen als de 'player' met een 'rock' botst.

Eerst wordt er een effect gecreëerd in de kleur wit : `c_white`. Daarna wordt het `obj_player` vernietigd en stopt het spel > `obj_game.alarm[0]`, dit verwijst naar alarm 0 van object 'game' en dan restart de room.



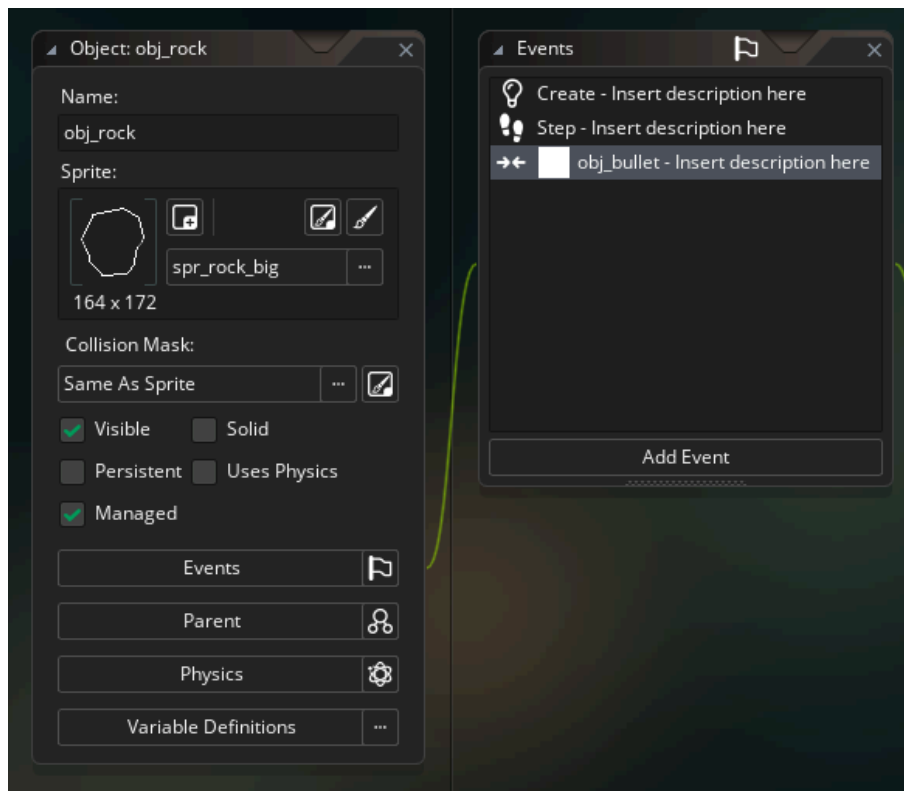
Met deze code laat je de 'player' vooruit en links en rechts bewegen (`vk_up`, `vk_left`, `vk_left`). Image angle zorgt ervoor dat het object draait en dus van richting verandert.

Bij een muisklik (mouse_check_button_pressed) schiet je een 'bullet'

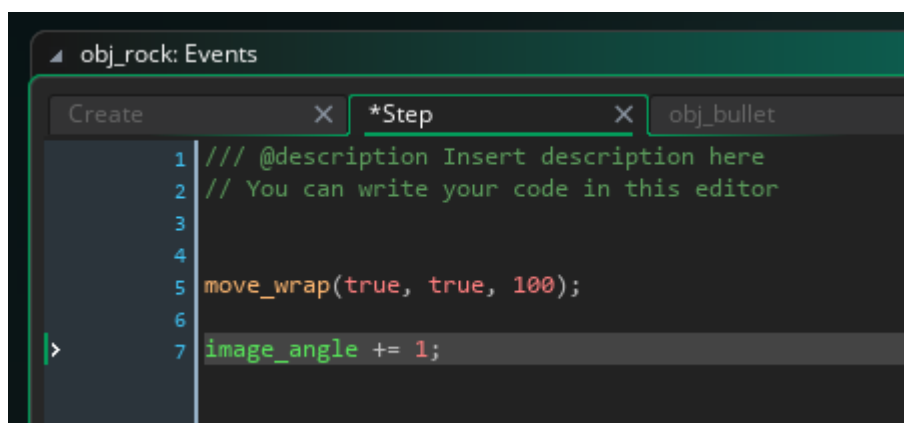
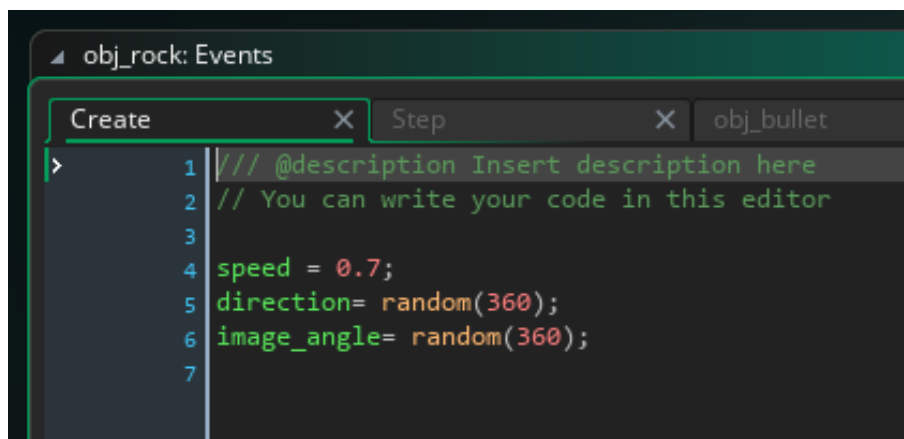


```
obj_player: Events
Step x obj_rock x
> 1 /// @description Insert description here
2 // You can write your code in this editor
3
4
5 if keyboard_check(vk_up)
6 {
7     motion_add(image_angle, 0.1);
8 }
9
10
11 if keyboard_check(vk_left)
12 {
13     image_angle += 4;
14 }
15
16
17 if keyboard_check(vk_right)
18 {
19     image_angle -= 4;
20 }
21
22 move_wrap(true, true, 0);
23
24
25 if mouse_check_button_pressed(mb_left)
26 {
27     instance_create_layer(x, y, "instances", obj_bullet);
28 }
```

Hier zijn alle acties en eigenschappen verbonden aan het object 'rock'



Dit geeft de snelheid en de directie van de 'rock's, de directie is random en kan alle kanten op (360) net als de draaiing van de afbeelding gekoppeld aan het object .



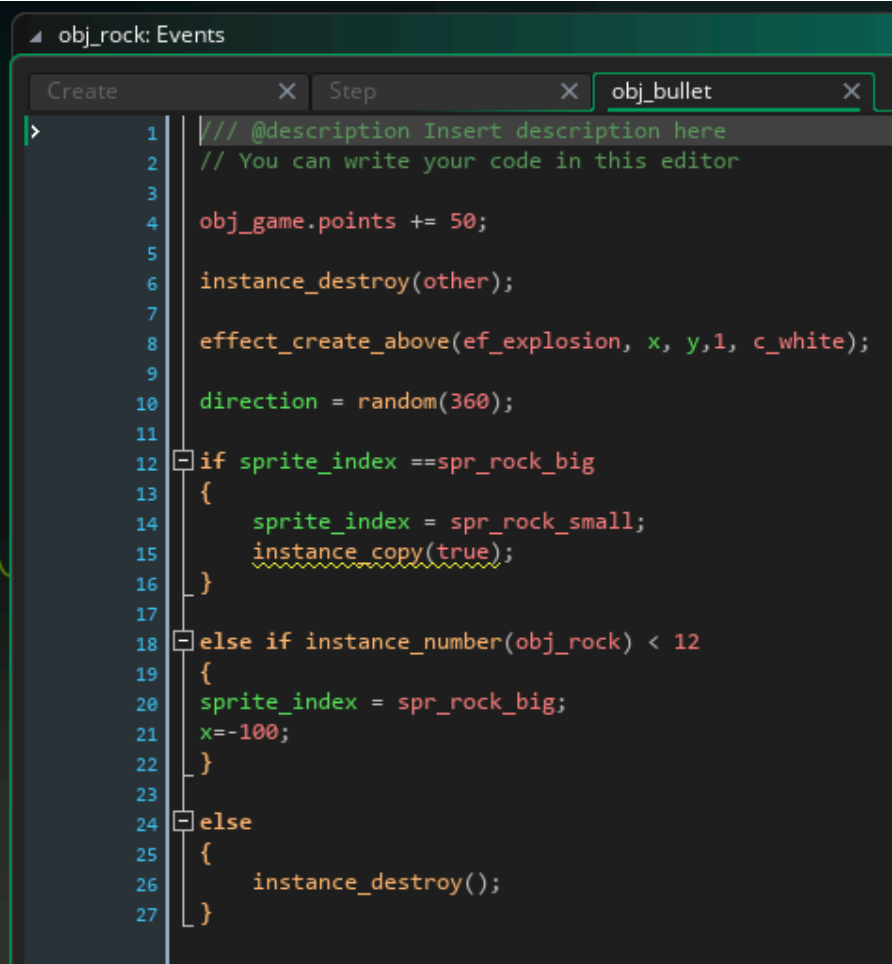
Bij een collision met een bullet komen er 50 punten bij de score van de aangemaakte points in het game object.

Daarna wordt de bullet vernietigd > (other)

Hierbij wordt een explosie gecreëerd net als eerder bij de collision van de player en een rock.

Erna wordt voor gezorgd dat als het een grote rock is, er een kleinere sprite van de rock als afbeelding komt en deze wordt gedupliceerd: de rock split als t ware in tweeën. Als er meer dan 12 rocks zijn wordt de rock geteleporteerd naar 100 pixels weg. Als het geen sprite_index = spr_rock_big heeft, dan is het een kleine rots, als je een kleine rots schiet gaat deze kapot

(instance_destroy())



```
obj_rock: Events
Create X Step X obj_bullet X
>
1  /// @description Insert description here
2  // You can write your code in this editor
3
4  obj_game.points += 50;
5
6  instance_destroy(other);
7
8  effect_create_above(ef_explosion, x, y, 1, c_white);
9
10 direction = random(360);
11
12 if sprite_index == spr_rock_big
13 {
14     sprite_index = spr_rock_small;
15     instance_copy(true);
16 }
17
18 else if instance_number(obj_rock) < 12
19 {
20     sprite_index = spr_rock_big;
21     x = -100;
22 }
23
24 else
25 {
26     instance_destroy();
27 }
```

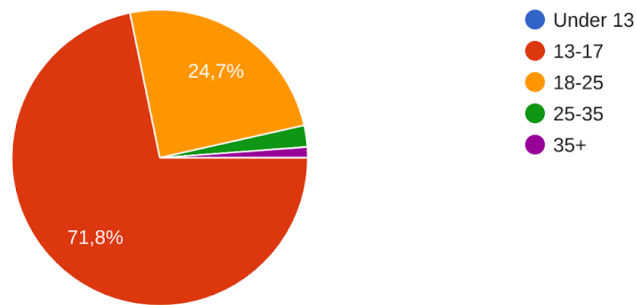
Bijlage 2

Enquête

Vraag 1: Hoe oud ben je?

1. How old are you?

85 antwoorden

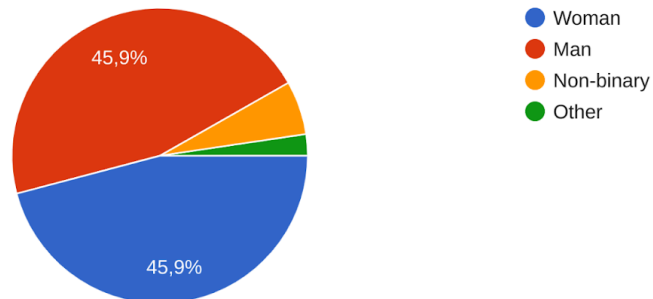


- De 71,8% van de mensen die de enquête hebben ingevuld zijn 13 tot 17 jaar oud. Het overige kwart is bijna volledig uit 18-25 jarige, dit was namelijk 24,7%. Als we dit vergelijken met de Nederlandse bevolking is dit geen representatieve groep, dit komt omdat vooral mensen tussen de 13-17 jaar oud de enquête hebben ingevuld. De bevolking bestaat niet zoals in de enquête, uit tieners en jongvolwassenen maar ook voor een groot deel uit mensen boven de 30. Wel is er voor de enquête de goede doelgroep bereikt, mensen die videogames spelen zijn namelijk vooral jongeren.

Vraag 2: Met welk gender identificeer je jezelf?

2. What gender do you identify as?

85 antwoorden

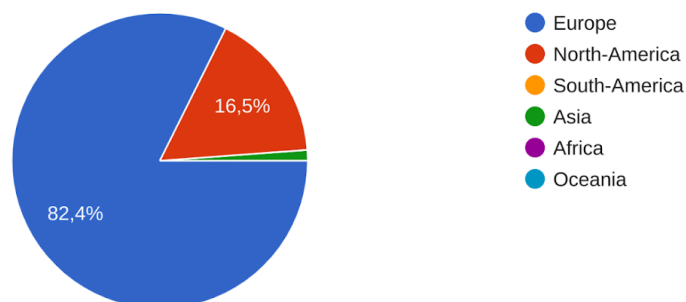


- Grappig genoeg, was het percentage mannen en het percentage vrouwen dat onze enquête ingevuld was gelijk. Hiernaast identificeren 5 van de 85 mensen zich als non-binair en hebben 2 van de 85 mensen voor anders gekozen. Er is dus geen meerderheid als het aankomt op gender in ons onderzoek.

Vraag 3: Uit welk continent kom je vandaan?

3. What continent are you from?

85 antwoorden

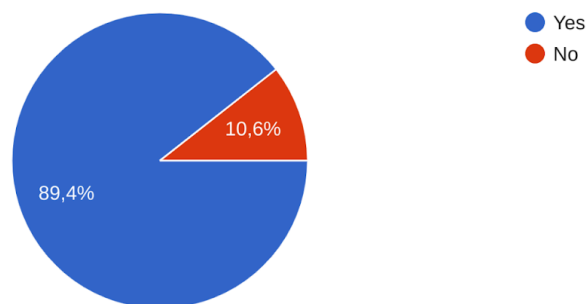


- 82.4% van de mensen komt uit Europa, wat logisch is aangezien wij zelf ons ook in Europa bevinden. Wel hebben we dankzij een social media platform Discord antwoorden van 14 mensen uit Amerika en van 1 iemand uit Azië kunnen krijgen. We hebben deze mensen de enquête laten invullen door de vragen Engels te maken. Hierdoor hadden we een groter bereik. Dit was belangrijk want videogames hebben ook een groot bereik en worden niet alleen in Nederland gespeeld. Bijna iedereen die de enquête heeft ingevuld komt uit Europa (namelijk 82,4%), we kunnen het dus niet over een wereldwijde situatie hebben met de enquête maar eerder de situatie in Europa.

Vraag 4: Speel je video games?

4. Do you play videogames? (If not you can skip question 5 through 9 and go on to question number 10.)

85 antwoorden



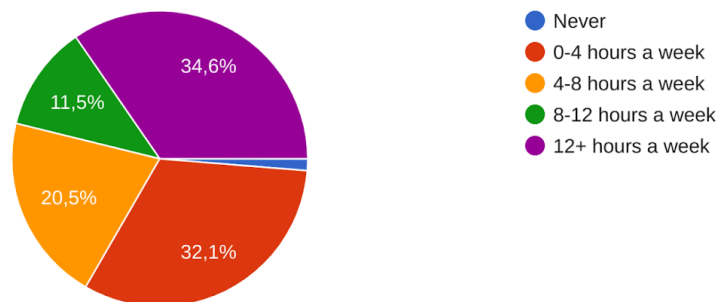
- Natuurlijk was dit belangrijk om te weten. We hebben de enquête goed verspreid aangezien maar 9 mensen van de 85 geen video games spelen. Dit is goed omdat we dus duidelijk de doelgroep hebben bereikt die we nodig hadden voor deze enquête: mensen die videogames spelen. Wel is het merkwaardig dat het dan 76 antwoorden hadden moeten

zijn bij de andere vragen die over videogames spelen gaan. Dit is niet zo, het zijn er namelijk 78. Dit was dus een resultaat waar we veel mee kunnen omdat bijna iedereen games speelt. De 10,6% komt zeer waarschijnlijk van familieleden en vrienden die geen games spelen maar voor ons de enquête hebben ingevuld.

Vraag 5: Zo ja, hoe vaak?

5. If yes, how often?

78 antwoorden



- 32,1% van de mensen speelt 0-4 uur per week. Daarnaast speelt 20,5% van de mensen 4-8 uur per week. Deze tijden zijn hoogstwaarschijnlijk onze medeleerlingen en vrienden die de enquête hebben ingevuld. Naast school is er namelijk niet veel tijd over. Er is 11,5% die 8-12 uur per week gamen. Er spelen 34,6% van de mensen maar liefst 12+ uur per week.

In totaal spelen 43,6% van de mensen 8 uur lang of meer games per week. Dat is aardig wat als je kijkt naar de tijd die school of een baan op kunnen nemen. Voor veel mensen is gamen echter een ontspannings bron. Dat zou kunnen verklaren waarom deze mensen zoveel video games spelen. De 43,6% mensen die 8 uur of meer gamen, kunnen

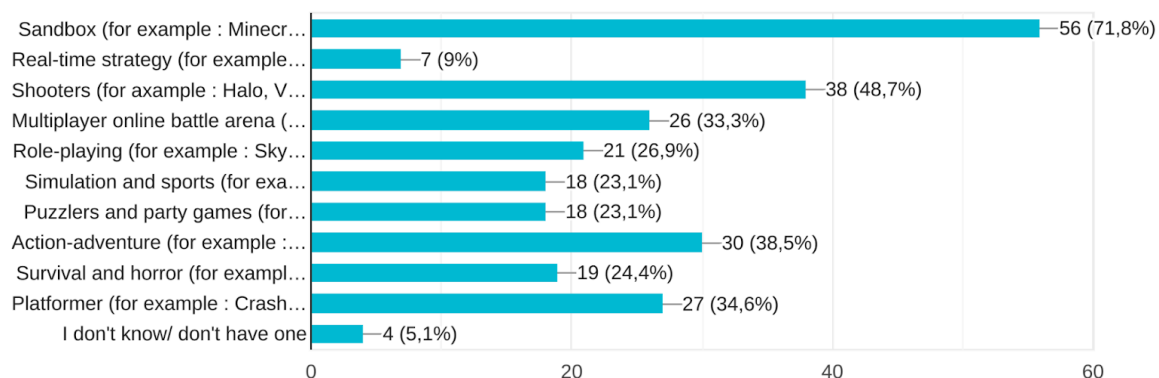
wellicht een verslaving hebben. Dit kunnen wij niet zeker zeggen want we hebben geen goed beeld over de situatie, de kans dat er mensen zijn die verslaafd zijn aan gamen en de enquête hebben ingevuld is echter wel aanwezig.

Vraag 6: Als je videogames speelt, welk genre speel je dan het liefst?

Bij deze vraag was het mogelijk om meerdere antwoorden aan te vinken, hierdoor zal de som van de percentages hoger zijn dan 100%.

6. When you play videogames, what kind of genre do you prefer to play?

78 antwoorden



- 71,8% van de mensen vinden met het genre sandbox leuk, Minecraft staat hierbij als voorbeeld en is ook al tijden een heel geliefd spel dat door veel mensen wereldwijd gespeeld wordt.
- Maar 9% vindt Real-time strategy leuk, spellen zoals World of Warcraft vallen hieronder. Dit is de op een na minst populaire categorie.
- Shooter games zijn door 48,7% van de mensen aangekruist als een favoriet genre. Je ziet deze redelijk vaak terugkomen in de media omtrent video games, bijvoorbeeld Halo of Valorant.
- De multiplayer online battle arena is ook redelijk geliefd, er hebben namelijk 33,3% van de mensen dit als een van de beste genres gekozen. Dit zijn

spellen zoals Fortnite, het zou best kunnen dat mensen toen Fortnite een rage was een zeer populair was, een meer gekozen genre zou zijn. Hoewel de trend nu voorbij is en je kan zien dat sommige mensen dit soort spellen gewoon nog spelen.

- Role-play spellen worden door net geen 26,9% van de mensen als geliefd genre gekozen. Spellen zoals de Witcher vallen hieronder, in deze spellen doe je quests en speel je echt als dat personage. Uit de cijfers is te zien dat niet veel mensen dit het liefst spelen. Vaak wordt er gezegd dat echte nerds dit spelen, we hebben dus ook echte nerds bereikt met deze enquête.
- Simulation en sport games zoals Forza Motorsport zijn niet heel populair en zijn door maar 23,1% van de mensen gekozen. Je hoort ook niet veel over deze spellen dus dit was te verwachten
- Puzzlers en party games zijn net zoveel gekozen als simulation en sports games (ook 23,1%). Je hoort toch wel vaker over spellen zoals Jackbox of the Talos Principle dan dat je mensen over Simulation games hoort praten. Dit kan betekenen dat de meeste mensen die de enquête hebben ingevuld niet graag samen een spel spelen, waarbij je bij elkaar moet zijn. Maar juist liever samen in hun eigen plekje spelen (of helemaal niet samen spelen).
- Games zoals Assassin's Creed zijn zeer populair tegenwoordig. 38,5% van de mensen hebben deze gekozen, hieruit blijkt dat Assassin's Creed en

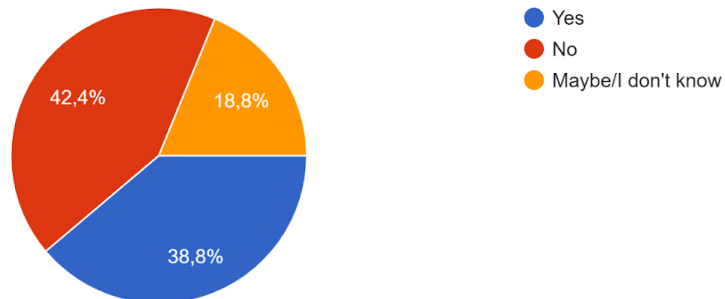
soortgelijke spellen nog steeds goed gaan in populariteit.

- Survival en horror games zijn heel spannend, maar toch vinden 19 mensen van de 76 dit een goed genre om te spelen. Laatst was Resident Evil ook een zeer populair spel geworden, dit kan er dus invloed op hebben.
- Platformspellen zoals Crash Bandicoot en heel veel anderen zijn nog steeds redelijk populair, iedereen die games speelt heeft wel eens een platform gespeeld. Het is echter geen zeer populaire favoriet als je kijkt naar de enquête. 34,6% van de mensen hebben een platformer gekozen. Dat is een redelijke hoeveelheid maar net iets lager dan je zou verwachten als je denkt aan de hoeveelheid platformer games die er zijn.
- Natuurlijk zijn er ook een paar mensen geweest die het niet wisten of er geen hadden. Dit waren er 4.

Vraag 11: Stimuleren advertenties je om videogames te spelen

11. Do advertisements stimulate you to play a game?

85 antwoorden

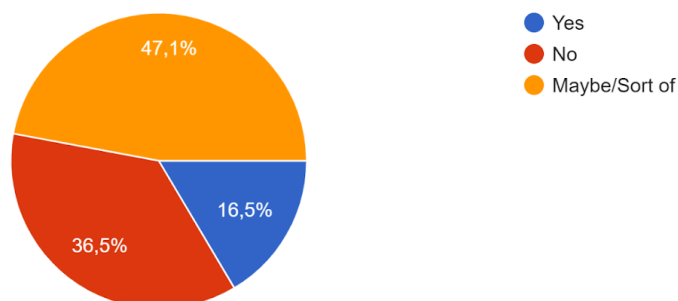


- Het verschil tussen 'ja' en 'nee' is bij deze vraag niet erg groot. Echter wint 'Nee' met een kleine meerderheid (42,4%). Hieruit valt te concluderen dat het voor de meerderheid van de mensen niet zal werken, maar dat advertenties zeker wel effect hebben op meer dan 38,8% van de mensen. 18,8% weet het niet.

Vraag 12: Weet je hoe videogames werken [de code etc.]

12. Do you know how games work internally? (the code, the design and the art)

85 antwoorden



- Iets meer dan 16,5% van de mensen zegt 'goed' te weten hoe videogames werken. 47,1% van de mensen

weet iets over de interne werking, maar niet compleet. Dit zou kunnen betekenen dat ze weten hoe ze gemaakt worden, maar dat ze het niet zelf zouden kunnen omdat ze hier niet genoeg kennis voor hebben. 36,5% zegt eerlijk dat ze geen idee hebben hoe games in elkaar zitten. Dit was op zich te verwachten, veel mensen spelen namelijk games maar weten niet hoe ze in elkaar zitten. Hier ben je ook niet echt mee bezig als je een game speelt.

Bijlage 3

Vragenlijst

De vragenlijst is in het Engels om mensen zo goed mogelijk te kunnen laten beantwoorden en om meerdere mensen te kunnen bereiken.

How old are you?

- 38 (3x)
- 29 (2x)
- 15
- 33
- 32
- 43
- 28
- 39

Which country are you from?

- United Kingdom (5x)
- The Netherlands (2x)
- Norway
- Sweden

How often do you play games?

- Daily/Every day (11x)
- Whenever I have free time

Why do you play games?

- Relaxation and/or fun (3x)
- Escape daily life
- Because I enjoy the different challenges and worlds you can experience
- Escape from the stress of work. Destress and relax. and Fun, of course!
- I play games as a form of enjoyment and a way to escape the real world.

- It's relaxing and I love the cooperation that's required for some games. As well as good stories.
- To relax, kill time, for entertainment, socialising with friends through teamwork and for the in-game accomplishments.
- Because it's fun and engaging. It also provides as a break from the otherwise "stressful" days of work and worries. And it in addition improves your language and communication skills, and your coordination among other things.
- Always enjoyed playing games of any nature, since being a kid. Grew up with a PS1 and so-on, along with a computer. So, very memorable for me and I love that games enable you to escape into another world. Games are a positive distraction for me from the stresses of work etc. It's a sense of accomplishment, getting through a level, achievement, piece of loot or even more, completing a game (rare for me!). They are wonderfully immersive and I especially love games set in a fantasy environment, though in the past I've dabbled in all sorts of genres.

What makes a game interesting for you?

- Depth story and gameplay
- Social interaction, customisation
- a lot of content and online multiplayer
- Story, and the freedom to solve problems in creative ways.
- Plot and good worldbuilding (dont create a new world with lore if its not needed)

- The difficulty, longevity, open world exploration, creativity (puzzles, problem solving, building).
- The aesthetics and story as well as the gameplay is what draws me to a game. I also really enjoy multiplayer-based games.
- I tend to lean more towards simulation games and horror style games, so jump scares definitely make games interesting for me. I do also enjoy puzzle games as these get my brain working.
- Easy to pick up, difficult to master. Games need some learning curve in order to be interesting to me. I like games with replayability where you can slowly master the game. A good UI goes a long way, controls should be intuitive.
- This is a tough question to answer as, in my honest opinion, it entirely depends on the type of game. I play games across many genres. For this, and for context, I will base this off of the MMO "genre". You may think it is what is just within the game content itself that makes them interesting. However with MMOs, it's also the social element to it. Being able to play the content with friends / strangers within. Solo content within the MMO environment isn't always fun and can just "drag on" and get quite lonely / boring. Having content built around the social aspect and the requirements for it makes it insanely more interesting. Of course, you still need to have the solo elements and ensure you cater to that part of the fanbase that do want solo.
- It needs to grab my attention. Easy to grasp the concept but not a walk in the park - good level of difficulty or ability to change difficulty as suits. Attractive graphics are a must for me - I am a fan of cartoon-y style graphics (e.g. WoW, Spyro, Crash Bandicoot,

Simulation games) but also of real-esque games (e.g. GTA, Gran Turismo Racing, Assassin's Creed). Having a story is important albeit I don't pay attention always to the text! I love games that aren't just linear; where you can influence the direction your characters goes in and do things in the order you want. A linear story is boring for me, especially if you can't make decisions - I love interactions with NPCs where you can choose the outcome from options and this then changes things in future! Social aspect is a part of it for me, albeit WoW is the only game (that's not a board game) that I play with others. I very much tend to be happy doing my own thing, as I can find playing with others brings you back into reality a bit too much at times.

Is a story in a game important for you?

- No
- Yes (2x)
- Extremely
- Yes but not always.
- If it's an RPG definitely. Other genres not so much.
- Overall, yes. In certain fast-paced shooters, not so much.
- Yes for the most part, but sometimes its nice to play a "mindless" grinding game if you only have for example 20 minutes to spare.
- A story isn't the most important thing in a game for me. Whilst It's nice to have in certain games, these aren't always needed and can be played perfectly fine without one (or skipping it!).
- 50/50 for yes and no. Dependant on the genre of game, a story can either ruin it. Break the immersion. It can also be the soul of the game and the game is nothing without one. All dependant on the style of game you're playing. A single person, immersive linear game - a story would make that game.
- Yes, it's important but it does depend on the game. I like playing Simulators such as Zoo Tycoon, Sim City, Cities Skyline, Planet Zoo and they don't really have a story. But if it's a roleplaying game (e.g. Fable, Witcher, Skyrim!), it definitely needs to be a good story to pull me in. I don't find games such as shooters appealing for this reason - I don't like the genre and

can't imagine they have any story at all. They're also often social games and I wouldn't enjoy that (especially if it's pvp style).

Which game is your absolute favourite?

- World of Warcraft (3 x)
- Due to how long I've played it (over 15 years), it has to be WoW and all the expansions. Played it since I was 15, had times where I stopped playing, but since I started working ~22, there's not been many times since that I've given up playing it - it's a great relax from work.
- The game I've definitely spent most time with is World of Warcraft, so because of that I assume that would be my answer. Heroes of Might and Magic III is also fairly high up there.
- World of Warcraft hands down
- Grounded
- Cities Skylines
- Zelda: A link to the past
- My overall favourite game would have to be 'The Sims 4'
- This is hard but I think I'm going to go with Final Fantasy VII, not the remake.

Why is this game so great?

- (I think World of Warcraft is great because) Social aspect, teamwork for raiding and dungeons.
- (I think World of Warcraft is great because) Been 20 years of my life Met some great people in it and I'm addicted

- (I think World of Warcraft is great because) MMO open world game that can be played with a lot of friends at the same time. It is constantly updated and expanded upon.
- (I think World of Warcraft is great because) The game has it's ups and massive downs, but overall just 80% about it is amazing to me. The gameplay, playstyle, storyline & graphics are "top notch" for the type of game it is and what it is aiming for.
- (I think World of Warcraft is great because) It's constantly being updated. I have a LOT of memories about this game and made some great friends. It's very familiar to me.. it feels like coming home when I play it, especially if I visit old zones or areas in the game. There's always something to do in the game and keep me busy, but at the same time, if I just want to dabble in the game with something easy and watch a tv show, I can do both. That's not something I can do in most other games I play, where I feel I need to be 100% immersed in it. Therefore I get to multitask when I play WoW. I very much like to have 'goals' in a game and achievements were introduced into the game many years ago. They've continuously been added to and I feel a sense of accomplishment in earning more achievements and seeing where I stand on a guild, server and wider basis. I'm competitive in that nature and as I've such a history in the game, I feel advantaged against others and don't want to let that go by giving up on the game! Back in the day, I was always so sad when a game finished and there'd be no more content, so I love my favourite game continually releasing new content.
- (I think World of Warcraft is great because) The social aspect of it is great and I (for the most part) really enjoy the lore.

- (I think Grounded is great because) Its open world and Great
- (I think City Skylines is great because) It's easy to get started on your city but you soon realise there's 100 variables involved that you didn't consider at first. Or that choices you make early on affect the rest of your build. Each city has an RPG element to it as well, you need to be invested (or even attached) to the world / city. I love the creative freedom it gives you while still providing general structure.
- (I think Zelda: A link to the past is great because) very good story and worldbuilding. And very fun gameplay!
- (I think The Sims 4 is great because) I think it's probably my favourite game because I do enjoy the fact that the game is almost life-like, in the sense that you can make it as much like your own life as possible, or you could go in the complete opposite direction. I like the fact that there is no 'win or lose' and you can go at your own speed. This game gives me the opportunity to be creative. It's the game I have spent most hours playing and is the one that I always end up coming back to, even if I haven't played for a while.
- (I think Final Fantasy VII is great because) Part cause of nostalgia from my childhood part the story and part that I had never played anything like it before.

Which game do you dislike most?

- raid shadow legends
- Minecraft
- Payday 3
- Any sports game.

- My most disliked game was 'Zoo Tycoon'
- Lord of the Rings Online.
- That's a really difficult question, because I tend to know in advance whether I'll probably like a game or not, so don't tend to even try and play it. I'm very picky in deciding to play a new game, it's very rare that I buy/play a new game as I need to be certain I'll enjoy it. As mentioned above, shooters are probably my least favourite, or football games, along with anything horror themed. My partner plays a game on occasion called Deathwing which when I see it, looks like a horror-shooter...
- League of legends.
- A matter of engagement I'd say. I don't dislike anything if it's fun. Mobile games?
- Hard to say as I would have stopped playing within hours and forgotten all about them

Why is this game so bad?

- (I think Raid shadow legends is bad because) its very pay to win which makes it an unfair system
- (I think Minecraft is bad because) The graphics of it is horrible. It's basically a game where you play with legos, but still somehow manages to look worse than if you were to create stuff out of legos.
- (I think PayDay 3 is bad because)The game launched with a lot of bugs and a bad multiplayer experience due to server connectivity issues. The prequel was around for 10+ years and had a lot of content added to it through DLC. Payday 3 felt barebones and empty in comparison.
- (I think Sports games are bad because) I just don't like sports games.
- (I think Zoo tycoon is bad because) I didn't enjoy this game due to the fact that the actual game play itself wasn't self explanatory and you seem to need to read through a lot of the instructions just to get an idea of how things work, when games like these shouldn't really need to be explained. I also didn't enjoy the graphics of the game itself and there are similar games out there that are of better quality. From what I remember, I also found a few bugs after a short amount of game-play too.
- (I think Lord of the Rings Online is bad because) This game used to be great, quite literally great. Then the rights got sold, it moved to a new developer and they drove it into the ground. It went free to play, content died down, development went away. No more needs to be said other than it was once a great and now it is not and I will not touch it again ever.
- (I think Games like Deathwing are bad because) To me, the game looks stressful and terrifying. I play a game to

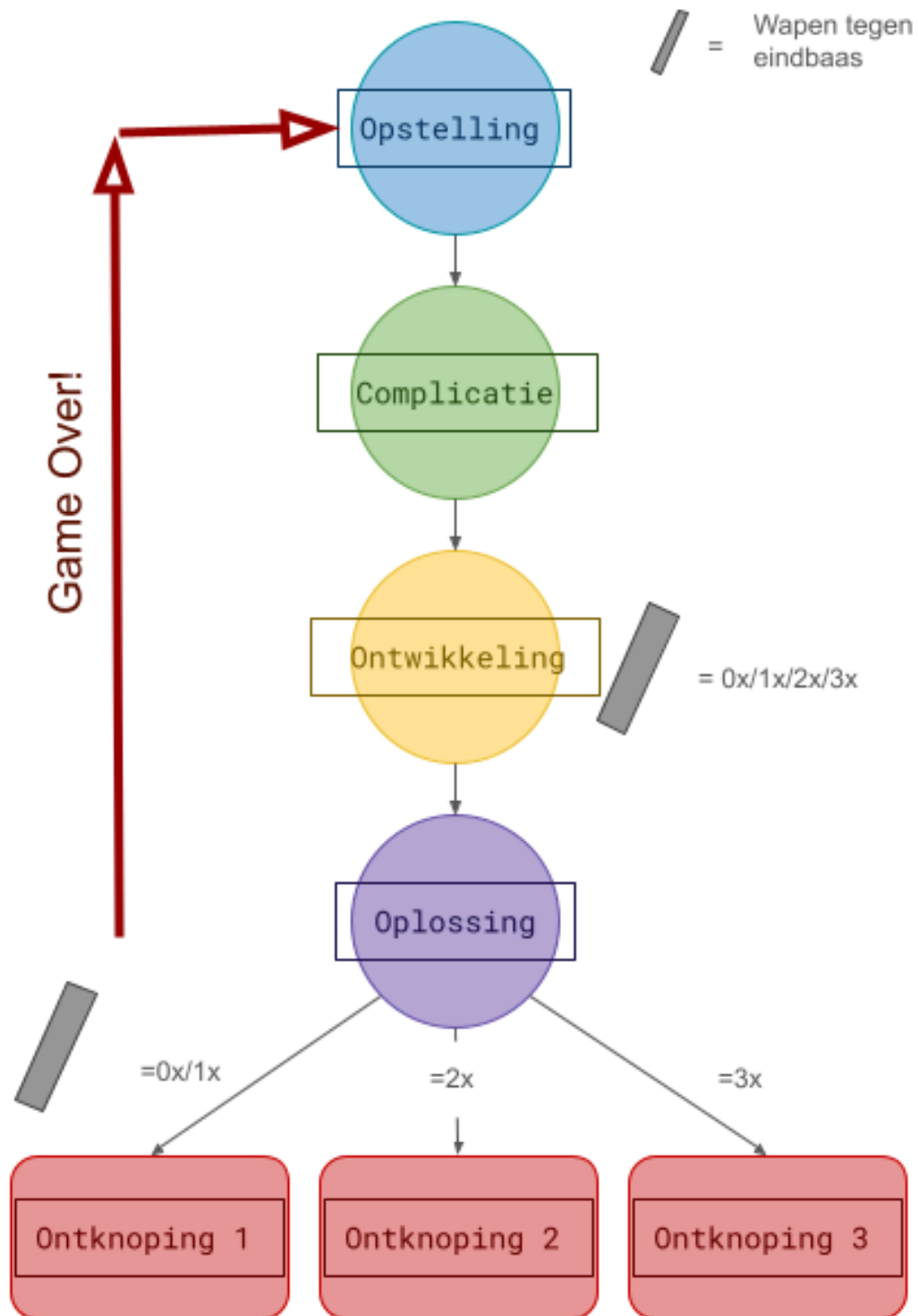
have fun, relax and it looks the opposite of those to me! I also like to be able to see my character which is the antithesis of first person shooters.

- (I think League of Legends is bad because) Toxic
- (I think mobile games are bad because) Cause it's not genuine and just want your money?
- Generally I find games that offer fake choices very bad. This happens in wow: bfa¹⁸ with the whole sylvanas quest line. You're essentially given the choice whether you support an important characters leadership of her faction but in the end it matters very little (just a few different lines if dialogue, zero ingame consequences). Same is the infamous Mass Effect ending that essentially was a 'pick a color' simulator where whichever choice you made your character died. It felt pointless.

¹⁸ BFA: Battle For Azeroth, een uitbreidingspakket van het spel World of Warcraft.

Bijlage 4

Scene tree



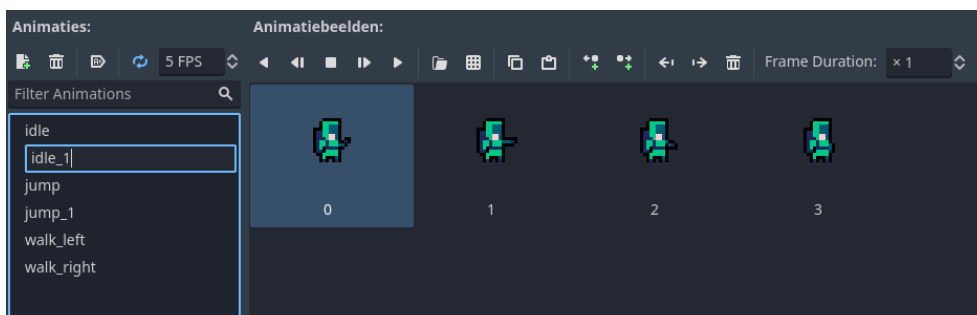
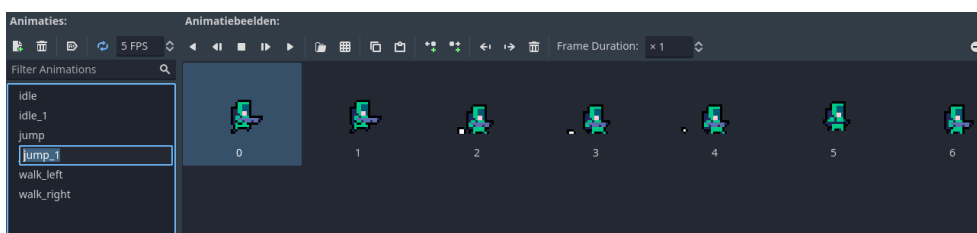
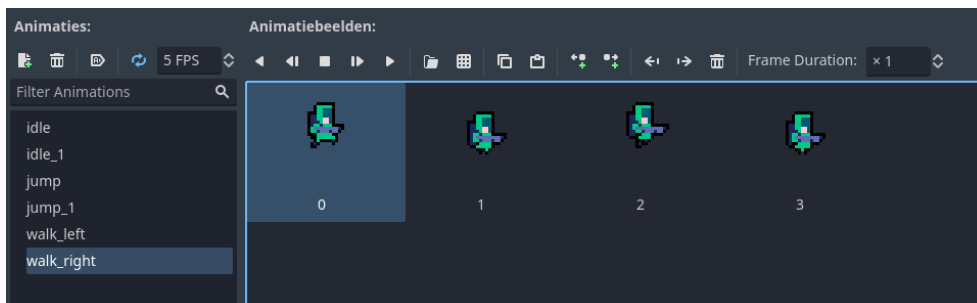
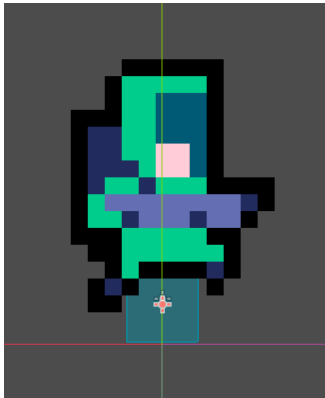
Bijlage 5

De gebruikte objecten zijn afkomstig van itch.io¹⁹

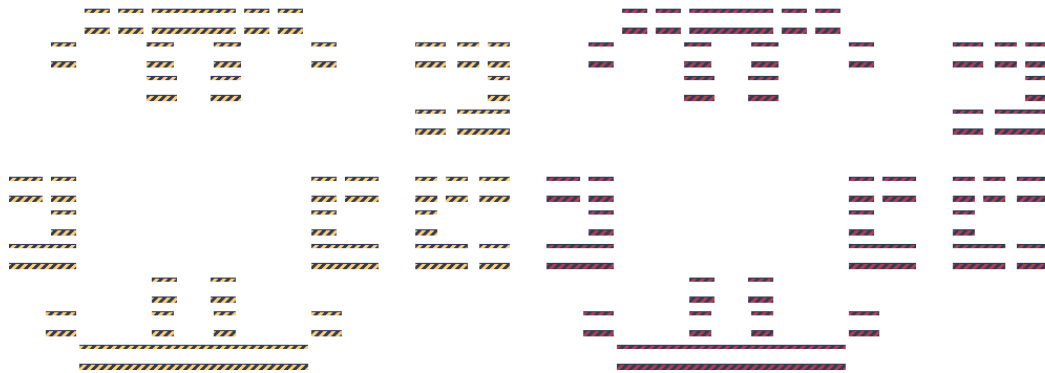
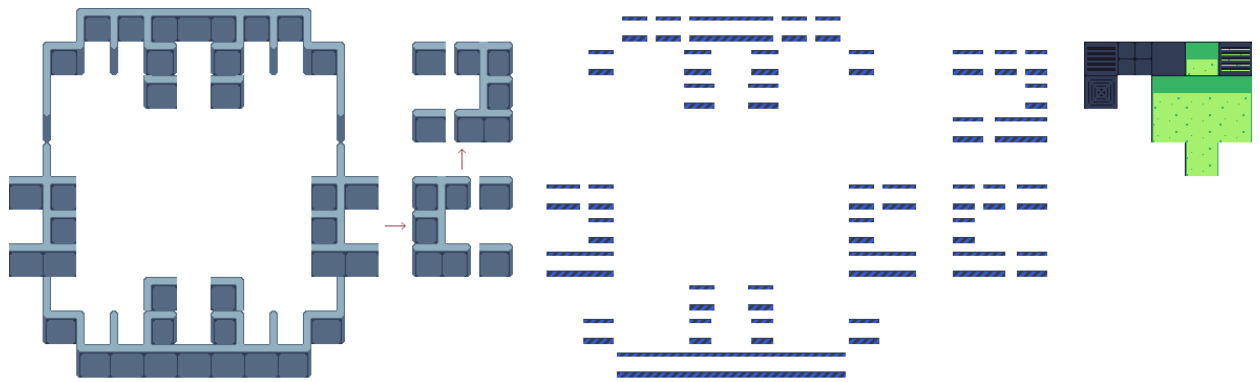
De animatie en sprites:

Hieronder zijn de sprites te vinden en hoe de animaties gemaakt zijn. De assets komen van de volgende webpagina :

<https://trevor-pupkin.itch.io/outer-buddies>



¹⁹ Itch.io: Een website waarop vaak gratis games, sprites, geluidsfragmenten en andere objecten voor game design op gedeeld worden.



Bijlage 6

De Sensorama uit 1962 uitgevonden door Morton Helig



Bijlage 7

Het logboek

Totale tijd aan besteed:

Totaal =	95,28333333
----------	-------------

Dag	Tijd (minuten)	Werk gedaan
25-3-2022	50	Aan enquête gewerkt
31-3-2022	50	Aan enquête gewerkt
1-4-2022	50	Aan enquête gewerkt
4-4-2022	50	Klaar met de enquête en opgestuurd
13-5-2022	50	Gewerkt aan het analyseren van de antwoorden
16-5-2022	40	Gewerkt aan het analyseren van de antwoorden
30-5-2022	50	Gewerkt aan het analyseren van de antwoorden gelijk klaar met het analyseren ervan.

Dag	Tijd (minuten)	Werk gedaan
22 mei	50	Besloten over het onderwerp videogames, dit is gedaan door gewoon even te brainstormen.
25 mei	100	Hoofd- en deelvragen bedacht en ingeschreven voor ideeën op het forum.
14 juni	100	Idee verder uitgewerkt en het plot en werking van de game, gedacht aan verhaal enzovoorts.
27 juni	120	Gezocht naar sprites voor de game zelf om te gebruiken in de gameplay, dit is gedaan op itch.io een website waarin naast andere dingen, je gratis sprites kan vinden om mee te werken, de gekozen sprites mogen gebruikt worden bij projecten die gedistribueerd worden.
5 juli	40	Gezocht naar sprites voor de game zelf om te gebruiken in de gameplay, zelfde als de dag ervoor.
26 juli	150	Begonnen met het leren van C++/C# (uit het boek 'Beginning C++ Through Game Programming (4th Edition) by Michael Dawson (2006) ')
5 aug	60	Verder gegaan met het leren van C++/C# en opnieuw gekeken naar mogelijke game engines (onderzoek voor deelvraag 1)
10 aug	30	Verder gegaan met het leren van C++/C# (uit het boek 'Beginning C++ Through Game Programming (4th Edition) by Michael Dawson (2006) ')
4 sept	20	Besloten het PWS alleen te maken vanwege complicaties in de samenwerking en hoofd en deelvragen een beetje aangepast
11 sept	100	Nieuw idee besloten, nog steeds zelfde onderwerp, dit is weer bereikt door te gaan zitten en te brainstormen.
22 sept	70	Verder gekeken naar idee van de game, hoe dit er visueel een beetje uit gaat zien, door een beetje rond te kijken op itch.io is dit een beetje tot stand gekomen

23 sept	40	Naar geluidselementen gezocht die bij de game zouden kunnen passen. Fragmenten zoals muziek, tekst en sound effects.
28 sept	20	Nog een paar sprites gezocht, ook dit was weer op itch.io , weer hebben alle gebruikte bronnens op deze website toestemming gegeven voor het gebruiken.
2 okt	115	Onderzoek gedaan naar verschillende game engines, omdat ik al eerder geïnteresseerd was in de ontwikkeling van games had ik alle engines behalve Godot al op mijn computer staan, hierdoor wist ik er al een groot deel over. Verdere onderzoek is voornamelijk gedaan op de websites van de engines zelf.
3 okt	50	Verder gegaan met het leren van C++/C# (uit het boek 'Beginning C++ Through Game Programming (4th Edition) by Michael Dawson (2006) ')
14 okt	40	Aan de inleiding en eerste deelvraag gewerkt, en de indeling van het verslag al in het document gezet zodat ik niks vergeet toe te voegen.
23 okt	100	Aan de inleiding en eerste deelvraag gewerkt, vertelt over waarom ik informatie heb gekozen en waarom het onderwerp een game is.
24 okt	30	Verbeteringen gemaakt aan het document na de reacties van begeleider, deze reacties ook meegenomen voor ide de toekomst, bijvoorbeeld de bijlagen, niet vragenlijsten enzv. in het verslag maar in de bijlage en er naar verwijzen.
27 okt	100	Onderzoek gedaan naar verschillende game engines en hiermee de eerste deelvraag verder uitgewerkt
31 okt	20	Verbeteringen gemaakt aan het document na de reacties van begeleider
3 nov	50	Opgezochte sprite packs doorzocht en een verzameling aan sprites gedownload
5 nov	20	Wat extra toegevoegd aan deelvraag 1, vooral spelling verbeteren, hier en daar een zin toevoegen en de conclusie beter verwoord en uitgebreider gemaakt.
13 nov	80	Projecten aangemaakt in zowel Godot en GameMaker, deelvraag 1 verder uitgewerkt
14 nov	60	Onderzoek gedaan naar waarom games een verhaal hebben
15 nov	120	Deelvraag 2 verder uitgewerkt en onderzoek gedaan naar verhalen. Dit onderzoek heb ik vastgelegd, het gaat hier om het deel van de geschiedenis en de veelzijdige van verhalen en hoe deze in games worden verwerkt Vragenlijst gemaakt over wat een game nou precies interessant maakt, om te kijken of het uitgevoerde onderzoek ook daadwerkelijk waar was.
16 nov	30	Vragenlijst verspreid naar mensen waarvan ik weet die videogames spelen om zo accuraat mogelijke antwoorden te krijgen. Onderzoek gedaan naar de toekomst van games, dit duurde niet heel lang aangezien ik op deze dag niet veel tijd had, wel vond ik het belangrijk om even alvast naar mogelijke artikelen te kijken, ik heb er echter geen gevonden deze dag.
17 nov	50	Gelijk na de eerste antwoorden een groot deel van de geanalyseerd en deelvraag 2 verder uitgewerkt met deze nieuwe informatie.
21 nov	150	Vragenlijst overgenomen en in de bijlage 2 gezet en geanalyseerd. Dit had ik een beetje onhandig gedaan waardoor ik de vragen per ongeluk door elkaar had gezet, ook werkte google docs een beetje vervelend dus dat zorgde ervoor dat ik een hoop dingen wat langamer deed.

22 nov	70	Tutorial van Gamemaker gevolgd: dit was redelijk snel en simpel en binnen 40 minuten was de hele asteroids game gemaakt Een brede kijk gedaan naar Godot om niet blind in de tutorial te beginnen. Een beetje rondgespeeld met functies van bijv Nodes, Textures en Collision Shapes plaatsen.
23 nov	170	Meer aan het verslag gewerkt en deelvraag 1 bijna afgerond na het volgen van nog een deel van de Godot tutorial: Informatie van het proces en de gebruiksvriendelijkheid van Godot uitgelegd/vastgelegd.
30 nov	85	Meer aan de Godot tutorial gewerkt en screenshot van process in bijlage toegevoegd: het schip heeft een vorm, collision shape en een sprite en kan terug in het scherm verschijnen als het er buiten gaat
4 dec	70	Uitleg bij de code geplaatst van GameMaker test, hierbij per ongeluk een stuk code vergeten toe te voegen, dit moet er later nog in. Toepassingen van het bizzocchi model vastgelegd, en een kleine analyse hiervan, hierbij heb ik ook gelijk het model gebruikt om een deel van de lay-out van de game alvast vast te leggen in het verslag.
8 dec	40	Puntjes op de i gezet bij opdracht 2 en een kleine tussentijdse conclusie over welke dingen belangrijk zijn in een game, deelvraag 2 is nu af.
14 dec	130	Godot tutorial gevolgd: beweging controls van het schip zijn nu af, zowel als het schieten. Het volgen hiervan ging redelijk goed, op sommige momenten wel even langzaam omdat de tutorial een beetje snel ging en ik niet helemaal begreep waar de code voor was, dit moest ik dus nog even zelf analyseren.
15 dec	75	Verder gewerkt aan deelvraag 1 door verder te werken aan de Godot, asteroides in de game kunnen nu van vorm veranderen en bewegen. Gewerkt aan het tweede model van Bizzocchi en de toepassing daarvan in deelvraag 3, voorbeeld van Detroit become human toegevoegd om zo goed mogelijk het tweede model uit te leggen.
19 dec	140	Gewerkt aan het tweede model van bizzocchi in deelvraag 3, hierbij nog een voorbeeld van een game waarin dit model wordt toegepast genoemd. Ook de analyse van het gemaakte werk vastgelegd en beredeneerd dat het huidige werk veranderd moet worden, dit laten zien en uitgewerkt hoe dit veranderd moet worden om het onderzoek dat gedaan is zo goed mogelijk te incorporeren in de game. Deelvraag 4 is bijna af. Gekeken naar de eisen en het logboek uitgebreider gemaakt om deze te laten doen aan de eisen die gesteld zijn, logboek was eerst zeer beknopt. Dit in de toekomst meenemen door uitgebreid het werk dat gedaan is te omschrijven.
9 jan	190	Verder gegaan met het deel van de tutorial maar bij het exploderen van de asteroïde ben ik vast komen te zitten, class hides a global script, kwam meerdere keren op het scherm en de asteroides wilden niet veranderen van grootte. Aan gewerkt maar nog geen oplossing gevonden. Idee is om de volgende keer een deel van de nieuwe code te verwijderen hopen dat het ergens een volgorde van input was.
10 jan	160	Code verwijderd en het deel van de tutorial opnieuw begonnen, de fout is weg maar nu is er een andere fout. Heel lang gezocht en gekeken naar websites zoals de hulp website van Godot zelf en stackoverflow. Geen oplossing gevonden.
11 jan	90	Aangezien ik vast liep met de voor onderzoek maar toch al goed verstand had van Godot ben ik verder gegaan met het main

		praktische deel van de game. Game levels en physics toegepast, en een goede grootte voor het scherm.
12 jan	150	Geprobeerd animaties toe te voegen, dit ging fout en heb meerdere tutorials gevonden en deze gecombineerd om een goede animatie van de speler te maken. De speler kan nu bewegen over het scherm. Een heeft een goede grootte op het scherm.
14 jan	40	Opnieuw gekeken naar de fout van het vooronderzoek van Asteroids, niks kunnen vinden en hulp gevraagd aan docent.
16 jan	20	Mailtje gestuurd naar iemand die de docent had aangeraden om vragen aan te stellen.
19 jan	70	Kleine dingen toegevoegd en research gedaan naar het main praktische onderdeel, wat meer passende sprites gezocht voor de game
20 jan	30	Reactie gehad op mijn mailtje, gewerkt aan het advies en heb de layers van de asteroïde veranderd, maar het werkte niet, opnieuw een mailtje gestuurd.
22 jan	90	Begonnen met het toevoegen van de gekozen spritesheets in de achtergrond.
26 jan	170	Gewerkt aan het commentaar dat op 10 januari door de docent is gegeven. Heb de samenvatting verbeterd en een kleine draft van de eindconclusie gemaakt. Ook heb ik gewerkt aan de laatste deelvraag die nog niet afgerond was. Ik heb een deel van de game een beetje opgebouwd maar loop vast met het koppelen van de acties van de speler en de reacties die de achtergrond moet hebben daarop. Heb een mailtje gestuurd naar de persoon aangeraden door de docent voor advies.
27 jan	350	Op het advies van de persoon aangeraden door de docent heb ik van de eerdere correspondentie over de asteroids game, het hele object van de asteroïde opnieuw aangemaakt, dit was veel werk maar het werkt nu wel! Gelijk ben ik verder gegaan met de rest, de score van het schieten van de asteroïde
3 feb	130	Heb de proef game Asteroids afgemaakt en de foto's van deze code in de bijlage 1 gezet.
7 feb	90	Heb de uitleg van de code van de asteroids game in Godot uitgelegd in de bijlage. De bijlage van deelvraag 1 is nu helemaal af.
8 feb	360	Heb door de dag heen veel gewerkt aan de game en veel bronnen geraadpleegd met aan het werken van de game praktisch. Ik heb er veel moeite mee en heb nog geen reactie gehad op mijn mailtjes. Ik krijg bij de gecodeerde acties van de speler veel errors en weet niet waarom. Ik ben opgegeven moment net als bij de proef game asteroids in godot, opnieuw begonnen, ik had al de kennis al opgedaan en dit ging redelijk vlot. Er is nu echter een ander probleem, ik heb eerder in het proces al een error en krijg de achtergrond niet goed. Alle sprites overlappen en willen niet meer werken.
12 feb	80	Ik heb opnieuw geprobeerd hier aan te werken maar de deadline wordt krap en ben dus nu begonnen na te denken hoe ik het pws compleet kan maken als de game niet werkt. In eerdere mailtjes en gesprekken heeft de docent gemeld dat het eindwerk van de game zelf niet het project kan zijn en dat het gaat om het proces en het verslag.
16 feb	20	Na goed nadenken heb ik besloten de game zelf maar los te laten, geen reactie op mijn mailtjes en te veel errors in de game waardoor de game niet eens opstart zijn in de korte dagen niet meer op te lossen. Ik ga nu focussen op het proces schriftelijk vast te leggen. Ik heb hiervoor dan ook een mailtje naar de docent gestuurd.

18 feb	120	Ben begonnen en heb gelijk een goede start gemaakt met het maken van de laatste deelvraag. Ik heb een kladversie van een samenvatting gemaakt van wat ik gedaan had en wilde doen voor de game. Gelukkig had ik schriftelijk een heel idee uitgewerkt over wat en wanneer er moet gebeuren in de game, dit ben ik nu aan het overnemen en dat is bijna gebeurd.
19 feb	100	Ben verder gegaan aan alles en heb het onder verdeeld in kopjes om het overzichtelijk te maken. De kladversie is nu overgenomen en ga ik verder uitwerken om het op niveau te krijgen
20 feb	320	Heb een grote read through gedaan van het hele verslag en het taalgebruik verbeterd. De conclusie is nu af behalve het deel van de laatste deelvraag. Ook heb ik gewerkt aan de laatste deelvraag en het script toegevoegd.
21 feb	100	Heb het hele script afgemaakt en daarmee ook gelijk de eindconclusie. Het verslag is nu af en uitgeprint en klaar om ingevuld te worden.