



## Magic 0.1

By: Quinn

27 Plays  

Learn the magic of science

**Q** Shoot Passion

**A** Move Left

**D** Move Right

**SPACE** Jump

### Goal:

Transform teachers into guiding angels by shooting them with passion fruits while avoiding obstacles and enemies.

### Strategy:

Collect mandala puzzle pieces, passion, and revelations to overcome obstacles and open the gate to the next level.

# Magic

Ontdek de magie van natuurwetenschappen

Quinn Slagter

## Inhoudsopgave

### Inhoudsopgave

<b>Inhoudsopgave.....</b>	<b>2</b>
<b>Opzet.....</b>	<b>7</b>
<b>Plan van aanpak .....</b>	<b>20</b>
<i>Stappen .....</i>	<i>20</i>
<i>Tijdlijn.....</i>	<i>20</i>
<b>Bijlage.....</b>	<b>22</b>
<i>Logboek.....</i>	<i>22</i>

## Voorwoord/Inleiding

In mijn werkstuk heb ik een educatieve game ontwikkeld met als doel om natuurwetenschappelijk onderwijs meer magie te geven. Door middel van het spelen van deze game worden spelers uitgenodigd om alle elementen die onze natuurlijke verwondering en spirituele ervaring met de natuur stimuleren of tegenwerken te leren zien en te overwinnen. In de game wordt de speler meegenomen op een avontuurlijke reis door drie werelden, gebaseerd op Dante's goddelijke komedie: de hel, het vagevuur en het paradijs met als laagste wereld de school. Ik heb feedback in de game verwerkt, zodat de speler weet wanneer hij iets goed of fout heeft gedaan en zijn leerproces kan verbeteren. Ik heb veel geleerd over programmeren, game design en grafisch ontwerp tijdens het maken van deze game. Ik hoop dat mensen plezier zullen beleven aan het spelen van de game en er misschien ook iets van kunnen leren.





## Concept

Het doel van het spel is om natuurwetenschappelijk onderwijs meer magie te geven door al gamend de speler uit te nodigen om alle elementen die onze natuurlijke verwondering en spirituele ervaring met de natuur stimuleren of tegenwerken te ontdekken en te overwinnen. Het spel bestaat uit vier werelden: de hel, het vagevuur, het paradijs en de school. In de school is er een monster karakter in de vorm van een scheikunde docent die verandert in een engel als je hem met passie bekogelt. Om passie te verzamelen kun je aan de boom der wijsheid schudden om passievruchten te verzamelen. Naast passie kun je ook openbaringen scoren in de vorm van een gloeilamp met een ster eromheen om zo verlichting te krijgen die je helpt in de duisternis van de school.

In de game zijn er echter ook valkuilen bij het leren die worden weergegeven door gesloten boeken in kuilen in de grond van elke level. Een gesloten boek staat voor een gesloten houding, het idee alles al te weten, het geloven in dogma's en het niet benaderen van het leven met een vraagteken. Daarnaast regent het natuuronderwijs vaak met formules, wat letterlijk in het spel gebeurt en eveneens je geestelijke gezondheid aantast (weergegeven door een hersengolf in het menu). Ook verkeerde voeding (in het spel weergegeven door een paddenstoel) kan je leren belemmeren.

De schoonheid van de natuur is in het spel verwerkt op een magische manier. Zo kan de speler door het verzamelen van specifieke items, namelijk de puzzelstukjes van een mandala, de schoonheid van de natuur ervaren op een visueel aantrekkelijke manier. De mandala-puzzelstukjes zijn op verschillende locaties in het spel verborgen en vereisen dat de

speler creatief en nauwkeurig te werk gaat om ze allemaal te vinden. Zodra alle stukjes zijn verzameld, kan de speler ze gebruiken om een prachtige mandala te creëren en daarmee naar een hoger niveau in het spel te gaan. Deze beloning is bedoeld om de speler nog meer te inspireren en te stimuleren om verder te gaan in het spel en de natuur op een dieper niveau te leren kennen.

Het doolhof pad is symbolisch voor de verschillende manieren waarop we kunnen leren en de obstakels die we tegenkomen bij het verkennen van ons brein. Net als in een doolhof moeten spelers soms terug naar het begin om een andere route te verkennen en nieuwe inzichten te krijgen. Het doolhof pad benadrukt ook het belang van volharding en doorzettingsvermogen bij het leren en verkennen.

## Game story

Het concept van mijn game was gebaseerd op de vier werelden die ik voor ogen had: de school, de hel, het vagevuur en de hemel. Elk van deze werelden vertegenwoordigt een aspect van de uitdagingen en valkuilen van het natuuronderwijs en het leren over de natuur.

Door middel van het verzamelen van passie vruchten en het scoren van epiphany elementen kunnen spelers verlichting krijgen en zich een weg banen door de doolhof-achtige paden van het spel. Maar ze moeten ook oppassen voor de valkuilen die worden weergegeven door gesloten boeken in kuilen in de grond van elke level en verkeerde voeding, zoals weergegeven door een paddenstoel in het spel.

De schoonheid van de natuur kan worden verzameld door er een mandala van te maken. Deze puzzelstukjes zijn verstopt in de doolhofachtige paden van het spel en kunnen alleen worden gevonden als de speler alle uitdagingen heeft overwonnen en alle elementen van de natuur heeft ontdekt. Het doolhofachtige pad weerspiegelt de verkenning van ons brein en het leren en ervaren van de natuur.

## Characters/Back story

De scheikunde docent in de game is een belangrijk personage dat de speler constant achtervolgt. Hij wordt afgebeeld met een erlenmeyer waaruit giftige dampen ontsnappen die de geestelijke gezondheid van de speler aantasten. Het personage staat symbool voor het traditionele natuuronderwijs dat vaak op een dogmatische en formulegerichte manier wordt gegeven. Hij staat voor de gesloten houding die sommige docenten kunnen hebben en die het leren kunnen belemmeren.

De speler daarentegen is een leerling die zichzelf wil bevrijden van de giftige invloeden van het traditionele natuuronderwijs en de magie van de natuur opnieuw wil ontdekken. De speler wil leren op een manier die past bij zijn of haar natuurlijke drang om te ontwikkelen en te groeien, net zoals de natuur zichzelf altijd vernieuwt en ontwikkelt. De speler wordt in het spel gestimuleerd om open te staan voor nieuwe ideeën en ervaringen, en om te leren van zijn of haar fouten.

In de game staan de twee karakters symbool voor de twee tegenovergestelde manieren van leren en onderwijzen. De scheikunde docent vertegenwoordigt de gesloten houding die sommige docenten kunnen hebben, terwijl de speler staat voor de natuurlijke drang om te leren en te ontwikkelen op een manier die past bij de eigen persoonlijkheid en interesses. De speler wordt uitgenodigd om de giftige invloeden van het traditionele natuuronderwijs af te schudden en de schoonheid en magie van de natuur opnieuw te ontdekken.

## Game Aesthetics

De game aesthetics zijn van groot belang voor de beleving van de speler en dragen bij aan de betekenis die de game heeft. De esthetiek van de game wordt gekenmerkt door een mix van verschillende stijlen en thema's die samen de ervaring van de speler versterken.

De game is opgebouwd uit vier werelden, elk met een eigen thema en esthetiek. Het allerlaagste level, de school, wordt gekenmerkt door een donkere, grimmige sfeer, waarin de scheikunde docent de speler achtervolgt met een erlenmeyer die de geestelijke gezondheid van de speler aantast. Dit alles wordt weergegeven in een duistere kleurensetting, met donkere schaduwen en scherpe contrasten. Dit creëert een gevoel van dreiging en onzekerheid, wat de speler stimuleert om de juiste keuzes te maken.

De andere drie werelden zijn geïnspireerd op de goddelijke komedie van Dante en worden gekenmerkt door een helderdere, meer kleurrijke esthetiek. De hel is bijvoorbeeld een wereld van vuur en as, waarin de speler wordt geconfronteerd met gesloten boeken in kuilen in de grond, die staan voor een gesloten houding en dogmatische denkpatronen. Het vagevuur is een wereld van licht en schaduw, waarin de speler openbaringen (epiphany) elementen kan scoren om verlichting te krijgen en de wereld beter te begrijpen. Aan de top van het paradijs vindt de speler de mandala-puzzelstukjes die de schoonheid van de natuur vertegenwoordigen en die de speler naar een hoger niveau kunnen brengen.

Al deze elementen dragen samen bij aan de esthetiek van de game, die zich kenmerkt door een duistere, maar ook magische sfeer, die de speler uitdaagt om verder te gaan om de schoonheid van de natuur te ontdekken. De kleuren, de geluiden en de vormgeving van de levels zijn allemaal zorgvuldig afgestemd op dit thema en dragen bij aan de betekenis van de game.



## Game Mechanics in FL

De speler kan door de levels navigeren met de WASDQ-toetsen en de spatiebalk. De speler kan objecten oppakken en verplaatsen door erop te klikken en te slepen. Er zijn verschillende interactieve objecten in het spel, zoals de boom der wijsheid, de paddenstoelen en de gesloten boeken in de kuilen. De speler kan een deel van deze objecten ook oppakken en verplaatsen om bepaalde doelen te bereiken.

Er zijn verschillende obstakels en uitdagingen in het spel, zoals het ontwijken van de erlenmeyer die door de scheikunde docent wordt gedragen, en het vermijden van de paddenstoelen die de geestelijke gezondheid van de speler aantasten. De speler moet ook oppassen voor de gesloten boeken in de kuilen, omdat deze de voortgang in het spel kunnen blokkeren.

Het verzamelen van passievruchten en openbaring-elementen is ook belangrijk voor de voortgang in het spel. De passie kan worden gebruikt om de scheikunde docent te veranderen in een engel die de speler helpt, terwijl de openbaring-elementen de speler verlichting geven en helpen om de schoonheid van de natuur beter te begrijpen.

Het spel heeft ook voortgangsindicatoren in het menu die de geestelijke gezondheid van de speler, de verlichting, de passie en de score weergeven. Ook is er een inventaris voor de verzamelde mandala stukjes en een save-functie waarmee de speler de voortgang kan opslaan per level.

De game mechanics in deze game zijn gevarieerd en complex. Het spel bevat verschillende objecten en elementen die bijdragen aan de spelervaring. Het concept van regendruppels met formules is een belangrijk onderdeel van het spel. Formules zijn afgebeeld als regendruppels die door de zwaartekracht vallen en worden vernietigd door een botsing met een ander object. De regendruppels zijn willekeurig verspreid over de wolken die in een lus regendruppels naar de aarde uitzenden. De timing (eigenlijk de beginhoogte van de regendruppel) van de regenval wordt ook willekeurig bepaald. Wanneer een object in contact komt met water, zal het zich verspreiden.

Een ander belangrijk aspect van de game mechanics is het verzamelen van passievruchten. Deze vruchten worden uitgezonden door de kennisboom en kunnen door de student worden verzameld. Het aantal verzamelde vruchten wordt opgeslagen in een globale variabele en weergegeven in de vorm van een indicator en een nummer in het menu. De passievruchten kunnen ook worden gebruikt om de scheikunde docent te transformeren wanneer deze wordt geraakt.

Een ander object in de game is de aura die om de speler heen hangt wanneer hij zich in het donker bevindt. Dit object heeft geen eigen mechanics en wordt (net als de aanwijshelper) aangebracht door het speler object wanneer dat nodig is.

Daarnaast bevat de game ook een object voor de duisternis die achter andere objecten wordt geplaatst om zo de duisternis te creëren. De hoeveelheid standaardduisternis kan worden aangepast door middel van een globale variabele. Dit maakt het mogelijk om

bijvoorbeeld een schemering te creëren waarbij de speler nog een beetje van de achtergrond kan genieten.



Een ander element is de lichtfeature die in het spel aanwezig is. Het lichtniveau wordt bepaald door de verlichting van de student en de helderheid worden berekend met behulp van een formule die gebruik maakt van globale variabelen en de afstand tussen objecten.

De game bevat ook een elevator object waarbij de speler tijdelijk wordt vergrendeld tijdens de rit. Na de rit wordt de speler weer vrijgelaten.


Het verlies van gezondheid wordt opgeslagen in een globale variabele die wordt weergegeven in de vorm van een indicator en een nummer in het menu. De verlichting wordt berekend door een bundel die controleert of de student zich op een donker blok bevindt en past de aura daarop aan.

De game mechanics worden geïmplementeerd met behulp van verschillende bundels in Flowlab, waaronder beweging van de speler, camera volgen, schadeberekening en meer.



In Flowlab vertaald betekent dit de volgende elementen in alfabetische volgorde;

<b>Afstand</b>	De afstandsbundel berekent de afstand tussen de x,y-positie van het huidige object en de x,y-positie van de speler. De outputeenheid is het aantal pixels.
<b>Aura</b> 	De aura rond de speler wanneer hij in het donker is, heeft zelf geen mechanisme. Het wordt bevestigd door het speler object wanneer het nodig is.
<b>Beweging</b>	In de beweging van de speler, die ook als een aparte bundel is ingesteld, volgt de camera de speler. De positie van de student wordt gecommuniceerd naar andere objecten via een x- en y-globale variabele. De lift was een uitdaging omdat we de speler tijdens de rit van elke beweging moesten vergrendelen. Daarna zou het weer vrij moeten worden gelaten. Dit werd gedaan via drie globale variabelen. De vergrendelingsvariabele van de lift stopt de volgende van de student van de y-positie van de lift die ook in een globale variabele is opgeslagen. De derde globale geeft aan of de lift zich bevindt, wat elke x- of y-beweging van de speler voorkomt door ook schakelaars. Dit geldt ook voor springen. Na een sprong wordt een volgende sprong vertraagd totdat een botsing (met de grond of een ander object) heeft plaatsgevonden.
<b>Docent</b> 	<p>De scheikunde docent heeft de rol van een “monster” in de game, maar ook van het enige score element.</p> <p>Omdat de leraar zich in de duisternis bevindt, moeten we controleren of het onderliggende blok donker is en vervolgens de verlichting-bundel oproepen.</p> <p>Om de animatie van de scheikundeleraar constant te laten bewegen, gebruiken we een oneindige loop. Omdat de leraar naar rechts beweegt,</p>

	<p>betekent de voorwaartse input dat de leraar altijd naar rechts zal bewegen. Door een ray te werpen, kunnen we controleren of de grond ontbreekt wanneer er niets wordt geraakt. Als dat het geval is, draaien we onze beweging om. Hierdoor lijkt de automatisering alsof de leraar zelf draait als hij botst. Hetzelfde geldt voor een botsing met een vast object, maar we reageren alleen als een botsing van links of rechts komt en laten de leraar enige afstand krijgen voordat de botsing opnieuw wordt geactiveerd.</p> <p>Als de leraar botst met een student, verliest de student gezondheid. Als de leraar botst met passievrucht, transformeert hij in een begeleidende engel en krimpt om de student vrijheid te geven en wordt hij vernietigd. De student verdient een score. De mandala opent weliswaar de poort, maar het transformeren van leraren is het werkelijk doel van het spel.</p> <p>De speler heeft de mogelijkheid om verzamelde passievruchten uit te zenden om de docent te transformeren wanneer deze wordt geraakt. Het passie-niveau wordt verlaagd bij elke uitgezonden vrucht.</p>
<b>Duisternis</b>	<p>Het duisterblok wordt achter andere objecten geplaatst om duisternis te creëren. De hoeveelheid standaard duisternis kan worden aangepast door een globale variabele. Dit maakt het mogelijk om een schemer te hebben zodat de speler van de achtergrond kan genieten. Het duisterblok moet worden beïnvloed door de helderheid.</p>
<b>Formule</b>  	<p>De game verspreid de regen willekeurig over de wolk die in een lus regendruppels naar de aarde laat vallen en verspreid de timing van de regenval willekeurig. Water zal zich met elk object verspreiden dus worden vernietigd.</p>
<b>Gezondheid</b>  	<p>De gezondheid van de student kan veranderen na een botsing met een dogma-, formule- of een gif-object. De schade wordt doorgegeven via de globale variabele die de gezondheid opslaat, wat ook wordt weergegeven op een indicator gevolgd door het getal in het menu. Elke schade wordt gevolgd door een geluidseffect. Via een timer neemt de gezondheid geleidelijk toe met de tijd totdat de maximale gezondheid is bereikt. Als de gezondheid nul is, wordt het spel opnieuw gestart.</p>
<b>Kennisboom</b> 	<p>De passievrucht wordt uitgestoten door de kennisboom en vernietigd door de leraar of de student. De pijlen geven aan dat er nieuwe vruchten hangen. Ze zijn bevestigd aan de kennisboom. De pijlhelper voert continu zijn animatie uit en wordt bevestigd door de kennisboom aan zichzelf als de timer aangeeft dat er nieuwe passievruchten zijn. De boom kan nieuwe vruchten laten groeien door een ingestelde timer die wordt bepaald door een globale variabele. Door middel van een randomizer worden in een lus 1 tot 4 vruchten uitgestoten die na een minuut vervallen. De student kan vruchten verzamelen en het aantal wordt opgeslagen in een globale variabele en weergegeven door een indicator en de waarde zelf. Dit geldt ook voor de andere indicatoren trouwens.</p>

<p><b>Lift</b></p> 	<p>Voor de werking van de lift (éénmalig) hebben we een start- en stopblok nodig dat de nabijheid van de lift onderzoekt. Deze blokken stellen de globale variabelen in die de lift bedienen. Bij het begin laten we beide objecten niet zien en niet deelnemen aan het spel op enige manier. Tijdens het spel willen we niet dat de speler de lift ziet stoppen of starten. We simuleren deze automatische werking van de lift.</p> <p>Het liftobject dat niet kan worden verplaatst en de bewegende lift aan het einde vervangt, moet ook worden verborgen en uitgeschakeld. Aan het einde willen we de lift vergrendelen door deze te vervangen door een onbeweeglijke, massieve lift.</p> <p>De beginstatus van de lift is ontgrendeld omdat deze nog niet is gebruikt. Het startliftobject signaleert andere objecten dat de lift begint te bewegen en daarna laten we andere objecten dit verder afhandelen. Wanneer de lift begint te bewegen, kan het startliftblok stoppen met controleren. We willen dat de lift maar één keer werkt, dus duwen we de lift slechts één keer omhoog. We gebruiken een soepele toename van de liftsnelheid door middel van een easy-component. We duwen de lift op de voorkeursmanier (met een push motor) omhoog.</p> <p>We gebruiken de globale variabele die de y-positie van de lift opslaat, wat helpt bij het soepel verplaatsen van objecten die met de lift meebewegen alsof ze door de vloer omhoog worden geduwd. We weten dat we alleen de y-afstand constant moeten houden om mee te bewegen.</p> <p>Omdat het dak van de lift massief is, kan het door de speler ook per ongeluk in de x-richting worden verplaatst. We corrigeren elke x-afwijking door de x-positie op te slaan op het moment dat het dak wordt gecreëerd en deze continu terug te zetten.</p> <p>Als de lift aan het einde is, informeert de eindliftblok eerst andere objecten over het stoppen en geeft daarna pas de opdracht om de lift te vergrendelen en de speler vrij te geven. Deze vertraging zorgt ervoor dat de lift eerst moet stoppen voordat de posities worden vergrendeld. Zonder een timer weet je niet de volgorde van uitvoering. Daarna kunnen we de lift vergrendelen. Wanneer de onbeweeglijke duplicaat het overneemt, wordt de bewegende lift vernietigd.</p>
<p><b>Mandela</b></p> 	<p>De mandala-puzzelstukken worden opgeslagen door de speler.</p> <p>Het spelerobject introduceert zijn eigen randomiseringsbundel omdat de router niet correct randomiseerde. Een van de drie klonen van een mandala-stuk wordt geactiveerd door ze een bericht te sturen. De overgebleven stukken worden vernietigd. Dus 9 van de 27 klonen van mandala-stukken worden bij elke spelstart geactiveerd.</p>

	<p>Wanneer de speler een mandala-stuk verzamelt (botst), wordt hij geïnformeerd door een bericht. De globale variabele die het aantal verzamelde mandala-stukken opslaat, wordt opgehoogd en wanneer deze een filter passeert die controleert of hij 9 heeft verzameld, wordt er een bericht verzonden naar de centrale portaalstukken om doorgang toe te staan en te animeren, zodat duidelijk is dat het portaal open is. De globale booleaanse variabele die de mandala is verzameld, wordt dan ingesteld op waar (1).</p>
<b>Menu</b>	<p>Het menu-object heeft verschillende indicatoren met een getal ernaast die de status van het spel weergeven, zoals de huidige score van de speler, het aantal verzamelde Mandela-stukken en de huidige passie en verlichting van de speler. Deze indicatoren worden geüpdatet door middel van globale variabelen die worden bijgehouden in andere delen van het spel.</p> <p>Het menu-object bevat ook verschillende objecten die de Mandela-stukken weergeven die de speler heeft verzameld. Deze objecten zijn standaard niet zichtbaar, maar worden geactiveerd wanneer de speler een Mandela-stuk verzamelt. Het activeren van een object gebeurt via een bericht dat door de speler wordt verzonden. Deze stukken bleken wel goed te kunnen worden een gestuurd door een en dezelfde parent.</p> <p>Het menu-object is zo ontworpen dat het tijdens het spelen van het spel zichtbaar blijft en de speler altijd op de hoogte houdt van de huidige status van het spel.</p>
<b>Passie</b> 	<p>Het passieniveau, opgeslagen in een globale variabele en weergegeven door een indicator en een getal in het menu, wordt versterkt door een botsing met een passievrucht. De speler kan ook verzamelde vruchten uitstoten om de leraar te transformeren bij het raken ervan. Het passieniveau wordt verminderd bij elke uitstoot.</p>
<b>Portaal</b>	<p>Voor elk mandala-stuk van het portaal wordt dezelfde bundel gebruikt. Er was een ouderobject verbonden, maar vreemd genoeg gaf het niet altijd zijn functionaliteit door aan alle kinderen. Dus de speler stuurt een bericht dat het portaalstuk moet worden getoond nadat het door de speler is verzameld. Dit stuk zal verschijnen in een fade-in op de dekking (alfa-kanaal) maar nog steeds worden gemengd.</p> <p>Wanneer alle negen Mandela-stukken zijn verzameld, activeert de speler de twee middelste portaal-stukken (zodat je echt het portaal in moet gaan) en worden de andere stukken naar een grijze toestand vervaagd om aan te geven dat het portal nu open is. De speler kan dan het portaal binnengaan en naar het volgende niveau gaan.</p>
<b>Speler</b>	<p>Het speler-object heeft de meeste mechanismen geïnstalleerd. Het draagt alle initialisatie en de behandeling van zijn gezondheid, verlichting, passie en de verzamelde mandala-stukken. De initialisatie is verdeeld in acties, variabelen en constanten die aan het begin zijn ingesteld. De standaardacties zijn een waarschuwing die het doel van het spel aangeeft</p>

	<p>en een opgeslagen waarde die het laatste niveau opslaat en het corresponderende niveau laadt.</p>
<p><b>Schijnwerper</b></p>	<p>In de schijnwerper-bundel berekenen en stellen we de RGB-kleuren in die helderheid simuleren. We hebben de afstand (A) nodig (afstandsbundel) en de globale variabelen maximum helderheid (B, een willekeurige waarde op basis van game-ervaring) en de helderheidsniveaus die worden bepaald door de verlichting van de student (C) en de helderheidsstappen (D). Door middel van deze zelfbedachte formule <math>\text{Math.max}(100-(A/32)(B-C*D),0)</math> kunnen we de helderheid van een blok berekenen. Of de helderheid wordt toegepast, wordt beheerd door een schakelaar die de floodlight aan of uit kan zetten. De helderheid van een object wordt gesimuleerd door alle r-, g-, b-kleuren op hetzelfde niveau in te stellen.</p>
<p><b>Verlichting</b></p> 	<p>De verlichtingsbundel begint met het controleren of de student op een donker blok staat en hecht vervolgens het aura-object eraan. De bereikte lichtheid neemt in de tijd af met behulp van een timer, net als bij de gezondheid, maar wordt dan omgekeerd. Bij elke botsing met een epifanie-blok (openbaring), begeleid door een geluidseffect, wordt de verlichting versterkt en dit wordt opnieuw berekend in een helderheidgetal door gebruik te maken van de globale variabelen die de maximale helderheid en de helderheidsstappen opslaan.</p>

## Testing

Het testen van een game is een cruciaal onderdeel van het ontwikkelingsproces. Voor deze game is er op verschillende manieren getest om te zorgen dat de game goed werkt en speelbaar is.

Ten eerste is er veelvuldig gespeeld om te kijken of de verschillende game mechanics goed werken en of de levels goed zijn ontworpen. Hierbij is er vooral gelet op de moeilijkheidsgraad en de balans tussen uitdaging en frustratie.

Daarnaast is er ook visueel getest door te kijken naar de flow van het programma in de behaviour tabs. Hierbij werd er gekeken naar de verschillende waarden die werden doorgegeven in de verschillende bundels van behaviour en of deze waarden de gewenste uitkomsten opleverden.

Tijdens het testen zijn ook verschillende bugs en fouten ontdekt en opgelost. Dit gebeurde door het testen van verschillende situaties en het aanpassen van de code om de fouten te corrigeren. Helaas kent flow lab ook zelf bugs. De ene keer werkt iets niet om bij een herstart ineens wel te werken. Ook andersom, bleek ineens het menu en portaal bij de allerlaatste totale test niet meer te veranderen als alle mandala-stukken zijn verzameld, zonder dat ik iets in de tussentijd aan de game behaviour had aangepast, maar alleen stukken verplaatst.

Al met al is er veel tijd besteed aan het testen van de game om ervoor te zorgen dat het een plezierige en speelbare ervaring biedt voor de spelers. In de reflectie heb ik de testervaringen mede meegenomen. Om het testen te vereenvoudigen heb ik bundels (soms met input en output gemaakt) om het programma in testbare eenheden te verdelen en meer overzicht en daardoor inzicht in de programmastructuur en doorloop te krijgen.

Ik heb aan anderen gevraagd - als een soort acceptatietest - om feedback op de game. Volgens hen moet duidelijk worden gemaakt wat de bedoeling is van het spel en wat goed of slecht is. Het toevoegen van een groene gloed om de goede spullen en een rode gloed om de slechte spullen kan hierbij helpen. Ook moet er gekeken worden naar de snelheid van het lopen en springen, zodat het ontwijken van waterdruppels makkelijker wordt. Tot slot moet er gekeken worden naar het stukje mandala dat mist, wellicht kan er een hint of aanwijzing worden toegevoegd om dit stukje te vinden. Ik zie dit als suggesties voor de volgende update, maar voor nu niet cruciaal om de game niet af te ronden voor nu.

## Reflectie

Ik heb een videogame ontwikkeld en heb veel geleerd tijdens het proces. Het meest tijdrovende en lastige onderdeel van het spel is de lift. Het probleem was echter dat de zwaartekracht de speler naar beneden duwde als de lift stopte en de speler op de grond belandde. Ik bedacht een oplossing waarbij de speler werd losgekoppeld van de lift als deze stopte, en voegde een timer toe om de acties in de juiste volgorde uit te voeren.

Ik gebruikte de motor als object om de lift omhoog te duwen en voegde een Ease element toe voor een mooie liftstijging. Eenmaal boven werd de beweegbare lift vervangen door een niet-beweegbare lift. Ik ontdekte dat als de speler op de lift stond, de gravity de lift naar beneden duwde. Ik heb dit opgelost door een exacte kopie van de lift op de eindpositie te plaatsen en de oorspronkelijke te vernietigen.

Ik wilde dat het spel niet elke keer hetzelfde was, dus ik verstopte de puzzelstukjes op verschillende plekken en plaatste elk puzzelstukje op drie verschillende posities. Zo zijn er 3 tot de macht negen verschillende variaties van het level mogelijk. Echter terwijl bij alle tests de acties die zouden moeten komen als ze allemaal zijn verzameld steeds werkten, bleek bij de allerlaatste test dit niet meer te doen. Omdat deze test heel tijdrovend was, kon ik uit tijdgebrek dit niet meer herhalen om te zien of het een glitch was in FlowLab zelf.

Ik ondervond enkele problemen met de elementen, zoals de router die bijvoorbeeld ad random één van zijn uitgangen kon kiezen, maar de randomnummergenerator bleek niet goed te werken bij de vele routers die ik koos. Ik heb de eigenlijke randomgenerator gebruikt in combinatie met de router en dat werkte beter. Hoewel ik veel dingen in FlowLab probeerde te automatiseren, was er toch veel herhalingswerk nodig. De Parent-functionaliteit leek ook niet geweldig te zijn, dus ik heb alles onder elk element gekopieerd. Het spel bevatte veel fouten, zoals het feit dat je soms langs iets liep en het je tegenhield, terwijl je de tweede keer op dezelfde plek gewoon door kon lopen. Waarschijnlijk heeft dat te maken met updates.

Tijdens mijn ontdekkingen in de praktijk van het bouwen van de game, heb ik besloten om het gebruik van een save-value te vermijden. Ik heb gemerkt dat deze niet handig is, omdat de waarden alleen opgeslagen worden op het apparaat waarop ik de game speel. Als ik de game opnieuw opstart, blijven de waarden in het geheugen staan. Daarom heb ik ervoor gekozen om in plaats daarvan globals te gebruiken om waarden op te slaan en alleen een save-value te gebruiken om de voortgang van het spel bij te houden, zoals het level waar ik ben gebleven.

Tijdens mijn onderzoek naar het toevoegen van licht en donker in de game, heb ik ontdekt dat het introduceren van deze functionaliteit eigenlijk een eenvoudig idee is. Ik kan de afstand berekenen met Pythagoras en vervolgens een beetje uitproberen of objecten helder moeten zijn of niet.

Het lastige was om unieke en authentieke afbeeldingen te genereren. Ik heb hiervoor gebruik gemaakt van kunstmatige intelligentie tools zoals Leonardo.ai en Dalle. Midjourney



bleek de beste tool te zijn, maar deze is helaas niet gratis. Het kiezen van de juiste prompt bleek erg lastig te zijn.

Om het probleem van het niet kunnen communiceren van objecten uit verschillende layers en behavior bundles op te lossen, heb ik zo veel mogelijk naar de gameworld verplaatst en met objectvolgorde gewerkt. Ik heb geleerd dat het eerst afmaken en spelen van de game de beste manier is om fouten op te sporen en bepaalde parameters in te stellen. Hiervoor heb ik een initialize-bundel gemaakt met alle globale constanten, variabelen en acties en heb ik alle instelbare waarden in het studentobject verzameld, omdat deze in elk level aanwezig zijn.

Ik heb veel tijd besteed aan het maken van grafische elementen en het onderzoeken van de gameloga. Dit bleek veel langer te duren dan ik had verwacht. Ik heb ook ontdekt dat er weinig verdiepende informatie beschikbaar is over Flowlab en dat de documentatie tegenviel. Een handige feature die ik later ontdekte, was de hulp- en notitieknop in de editor. Hierdoor kon ik meteen vastleggen waarom ik bepaalde keuzes had gemaakt.

Ik heb de verschillende gamesaves gelabeld met een versienummer, te beginnen met nul om aan te geven dat het spel nog niet klaar is. Dit maakt het makkelijker om de voortgang in kaart te brengen en terug te gaan naar oude versies als dat nodig is. Ik heb alle assets van de game opgeslagen in de map waar ik ook los van de game met allerlei tools afbeeldingen en andere assists heb gemaakt, zodat ik ook een back-up heb.

Ik heb besloten om het verhaal van de game op te delen in de drie werelden van Dante's goddelijke komedie, namelijk de hel, het vagevuur en aan de top de paradijs. Het laatste level is een duistere school, waar ik later als een extra grap op kwam.

Een andere uitdaging die ik tegenkwam bij het maken van de game, was het vinden van de juiste balans tussen uitdaging en frustratie. Ik wilde de speler uitdagen en stimuleren om verder te gaan, maar ik wilde niet dat de game zo moeilijk was dat mensen gefrustreerd zouden raken en opgeven. Daarom heb ik verschillende niveaus van moeilijkheid en uitdaging ingebouwd, zodat de speler op zijn eigen tempo kan spelen en leren.

Een ander aspect dat ik heb opgenomen in de game, is het idee van feedback. Ik wilde ervoor zorgen dat de speler wist wanneer hij iets goed of fout had gedaan, zodat hij zijn leerproces kon verbeteren. Daarom heb ik verschillende soorten feedback in het game menu geïntegreerd, zoals visuele feedback (bijvoorbeeld wanneer een object wordt verplaatst of vernietigd), auditieve feedback (bijvoorbeeld bij het voltooien van een level) en tekstuele feedback (bijvoorbeeld wanneer de speler een fout maakt).

Een ander interessant aspect van het maken van de game was het gebruik van AI-tools om authentieke en unieke afbeeldingen te genereren. Ik heb geleerd hoe ik kunstmatige intelligentie kon gebruiken om afbeeldingen te maken die perfect pasten bij de stijl en het thema van de game. Dit was een geweldige manier om mijn creativiteit te stimuleren en mij te helpen bij het maken van de game. Al met al was het maken van deze game een uitdagend, maar ook een enorm leerzaam proces. Ik zie de game als een eerste start. Later

zal ik meer levels toevoegen, hoewel de eerste level eigenlijk in zich ook al levels heeft zitten. De tweede level speelt als feestje een door AI (AVIA) gegeneerde muziek af.

## Opzet

Het doel is het ontwerpen en ontwikkelen van een 2D-game genaamd "Magic" die spelers de magie van natuurwetenschappen laat ontdekken door het verzamelen van puzzelstukjes om een mandala te maken, waarbij intuïtie en kennis hand in hand gaan.

## Plan van aanpak

### Stappen

- **Conceptontwikkeling:** Maak een storyboard van de game en bedenk de belangrijkste spelelementen, personages en omgevingen.
- **Flowlab Game Creator:** Maak een account aan op Flowlab Game Creator en start een nieuw project. Gebruik de ingebouwde tools om de omgevingen, personages en objecten te ontwerpen. Importeer eventueel eigen afbeeldingen en geluiden.
- **Game mechanics en level design:** Bedenk de acties die de speler kan uitvoeren, welke obstakels ze moeten overwinnen en hoe ze de puzzelstukjes kunnen verzamelen. Maak een schema van de verschillende levels en hoe de speler in elk level de puzzelstukjes kan vinden.
- **Game logic:** Gebruik de visuele programmeertools in Flowlab om de game logica te creëren. Definieer hoe de verschillende elementen van de game met elkaar interacteren, zoals de beweging van de speler en het verzamelen van puzzelstukjes.
- **Testen en verfijnen:** Speel de game meerdere keren door om te zien hoe het voelt en maak aanpassingen waar nodig. Zorg ervoor dat de game uitdagend is, maar niet onmogelijk om te spelen.
- **Toevoegen van audio en visuele effecten:** Voeg audio-effecten en achtergrondmuziek toe aan de game om de spelervaring te verbeteren. Gebruik visuele effecten om de magie van de natuurwetenschappen te laten zien.
- **Testen en debugging:** Test de game uitvoerig en los eventuele bugs en problemen op.
- **Publicatie:** Publiceer de game op de website van Flowlab en/of exporteer de game zodat deze kan worden geüpload naar andere gameplatforms.

### Tijdslijn

- Dag 1 : Conceptontwikkeling en storyboard
- Dag 2: Flowlab Game Creator en omgevingsontwerp
- Dag 3: Game mechanics en level design

- Dag 4: Game logic en testen
- Dag 5: Toevoegen van audio en visuele effecten
- Dag 6: Testen en debugging en publicatie

## Bijlage

### Logboek

Datum	Beschrijving	Tijd(min)
19 maart	Brainstorm, opmaak en plan van aanpak maken	80 min
23 maart	Flowlab tutorials doornemen	150 min
3 April	Brainstorm met J. Koornstra	60 min
6 April	Zelfstandig game bouwen	300 min
7 April	Zelfstandig game bouwen	240 min
8 April	Zelfstandig game bouwen	240 min
9 April	Zelfstandig game bouwen	200 min
10 April	Zelfstandig game bouwen	300 min