

Dr. One Escape

Een game maken in Unity is niet onmogelijk.

Introductie

Het plan was om een eigen game te maken. Bij het kiezen van dit onderwerp werd tegen ons gezegd dat heel veel leerlingen het hadden geprobeerd maar het nog nooit iemand was gelukt. Wij zijn de uitdaging aangegaan. Deze escape room game is gemaakt in Unity en verder volledig van scratch opgebouwd. Het bestaat uit bijna 20 models en meer dan 550 regels code.

Schrijven is schrappen

Excluding merges, **2 authors** have pushed **13 commits** to main and **13 commits** to all branches. On main, **139 files** have changed and there have been **605,911 additions** and **477,802 deletions**.

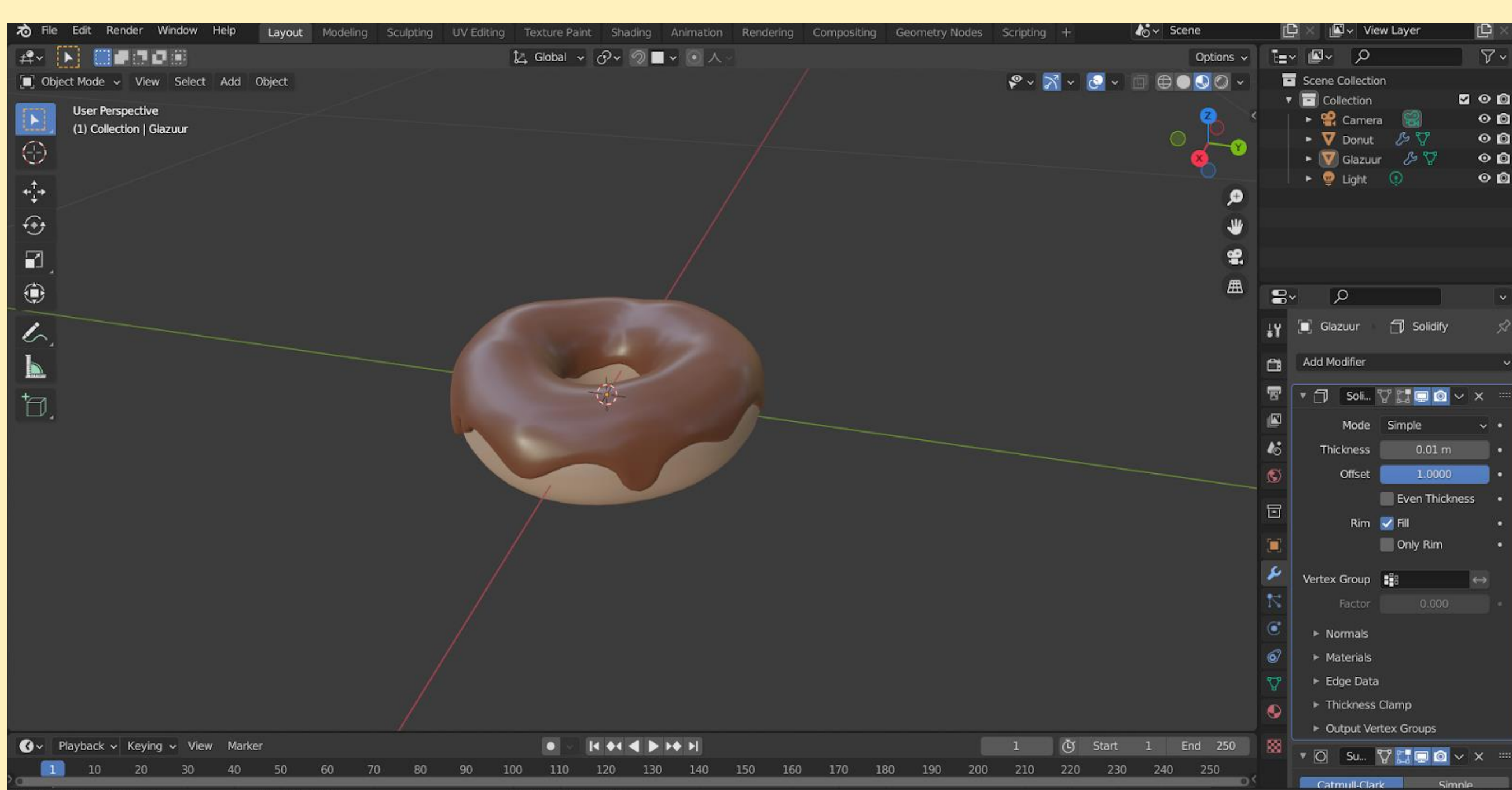


Bron: Github insights

Modelleren

Ieder object in de game heeft een herkenbaar uiterlijk nodig. Om de game een persoonlijke uitstraling te geven zijn alle objecten voor de game gebouwd en op elkaar afgestemd. Dit is gedaan in het modelleerprogramma *Blender*. Hieronder is een 3D-model van een donut weergegeven, die uiteraard in het spel is verwerkt. Eerst moet de basis van de vorm worden gemaakt. Bij het ene object gaat dit soepeler dan bij het andere, afhankelijk van hetgene dat je na probeert te maken. Bij gecompliceerde vormen is meer precisie nodig. Zodra je een basis hebt om op te bouwen kun je je op de details gaan focussen, in dit geval het aanbrengen van het glazuur. Om het object zo realistisch mogelijk te laten ogen is het hier van belang de vormen niet al te voorspelbaar te plaatsen. De laatste stap is het ontwerpen van het material waar het object uit bestaat. Hierbij let je op de kleur, maar ook op factoren zoals de ruwheid en de weerkaatsing van het licht. Stel je dit goed af op wat het uiteindelijk moet worden, en je krijgt een herkenbaar resultaat.

Een zelfgemaakte donut



Bron: donut.blend (Blender)

Uitproberen?

Dat kan! De eerste vijf levels van Dr. One Escape zijn nu af. De game is als .exe bestand te downloaden door deze QR-code te scannen. Het is speelbaar op de PC.



Code

Dr. One Escape is geschreven in C#. Bovenstaand script zorgt ervoor dat iedere tick de camera beweegt als de speler op J of L drukt. De Update() functie voert zichzelf iedere frame uit. Input.GetKey() is true als de toets wordt ingedrukt. Het stuk code daarna wordt dus uitgevoerd wanneer de J in dit geval wordt ingedrukt. De transform.Rotate() functie laat het object waar de transform component bij hoort draaien, in dit geval de camera. Vector3.down is een Vector3 met de waarde (0, -1, 0). Dit wordt gebruikt als basis om mee te rekenen en staat voor een richting. Om te voorkomen dat het beeld schokkerig wordt, wordt Time.deltaTime (de tijd tussen het huidige en vorige frame) gebruikt om de draaiing te berekenen. De rotation_speed kan in de editor makkelijk aangepast worden om de optimale draaisnelheid te krijgen. Deze code zorgt, samen met veel andere codes, ervoor dat de escaperoom werkt.

Source code

```
// Update is called once per frame
Unity Message | 0 references
void Update()
{
    if (Input.GetKey(KeyCode.J))
    {
        transform.Rotate(Vector3.down * Time.deltaTime * rotation_speed);
    }
    if (Input.GetKey(KeyCode.L))
    {
        transform.Rotate(Vector3.up * Time.deltaTime * rotation_speed);
    }
}
```

Bron: Dr. One Escape source code

Gameplay

In Dr. One Escape speel je als drone. Je bevindt je in een vierkante kamer, en je doel is de deur te openen en eruit te ontsnappen. Hiervoor kun je objecten gebruiken die in de kamer te vinden zijn. Met sommige objecten kun je interacties uitvoeren, en sommige zijn op te pakken en te verplaatsen. Het is aan de speler zoveel mogelijk manieren te vinden om de kamer te verlaten.

Enkele beelden uit Dr. One Escape



Bron: Dr. One Escape

Besturing

[W][A][S][D]: Beweeg de speler
[SHIFT]: Omlaag vliegen
[SPACE]: Onhoog vliegen
[1]...[9]: TV zender wisselen
[E]: Object loslaten
[Q]: Boek lezen
[J][L]: Camera draaien
[ESC]: Pauzemenü openen
Voortgang bekijken

Pauzemenü



Bron: Dr. One Escape

Onderwerp

Het plan was om een eigen game te maken. Bij het kiezen van dit onderwerp werd tegen ons gezegd dat heel veel leerlingen het hadden geprobeerd maar het nog nooit iemand was gelukt. Wij zijn de uitdaging aangegaan.

Unity of Unreal

Bij het kiezen tussen twee game engines hebben we advies gekregen op [discord](#). Unreal gebruikt C++ en Unity gebruikt C#. Omdat ik al ervaring had met C++ dacht ik dat Unreal de beste keuze was. Maar op discord werd gezegd dat Unity voor beginners makkelijker aan te leren is. Bovendien is een profielwerkstuk 80 uur, wat erg weinig is voor de meeste games. Wanneer het leren van de game engine al 80 uur is, blijft er weinig tijd over voor het maken van de game. C++ en C# lijken ook veel op elkaar, dus zou het kunnen programmeren in C++ juist een voordeel zijn. Daarom hebben we uiteindelijk voor Unity gekozen.

<https://learn.unity.com/course/create-with-code>

Rigidbody.velocity en transform.Translate()

- Bewegingen gaan nu via Rigidbody.velocity in plaats van transform.Translate(). Hierdoor wordt er rekening gehouden met physics en hoeft er geen outofbounds code meer te zijn. De speler kan niet meer door voorwerpen heen bewegen. (logboek, 3-8-2021)

In de cursus werden bewegingen door middel van transform.Translate() uitgelegd. Het probleem daarmee was dat Translate() physics negeert bij hogere snelheden. Daardoor kon de speler door de muur heen vliegen. Om dat op te lossen had ik een outofbounds check gemaakt, maar daardoor kon de speler niet meer door de deuropening vliegen. Door in het #unity-dev kanaal van GameDevLeague te zoeken, heb ik de oplossing gevonden: Rigidbody.velocity. De bewegingen daarvan zien er hetzelfde uit als transform.Translate() maar hierbij worden physics wel gebruikt. Een andere optie was Rigidbody.AddForce(), maar daarbij werd de speler met een soort gooibeweging bestuurd. Dat zou vooral bij het vliegen niet handig zijn.

Data serialization

Voor de escaperoom moest de voortgang van de speler opgeslagen worden. Dit kan via playerprefs of via data serialization. Playerprefs is makkelijker te programmeren, maar data serialization is veiliger. Na uitgezocht te hebben hoe data serialization werkt, heb ik dit gebruikt voor het opslaan van de voortgang.

That's not a bug, that's a feature

We waren eerst van plan om nog een geheime kamer te maken waar je in kan komen door met de muis op de knop te drukken. Daar bleken we te weinig tijd voor te hebben, maar de deuropening en de ideeën zijn er nog. Aangezien de volledige escaperoom met uitzondering van het pauzemen gespeeld kan worden zonder muis vonden we het beter om de geheime kamer te openen zonder de muis. Tijdens het schrijven van de HeldObject code, was er een bug. Als je vanaf de zijkant tegen het object bewoog, ging het object onder je hangen. Dat gebeurde ook als je op de

grond bewoog. Als je daarna niet omhoog vloog en het object zou loslaten, viel het door de vloer heen. De enige manier om daarna het blokje terug te krijgen is door het level te resetten. Om dit op te lossen, had ik een code geschreven waarmee het object alleen als je via de bovenkant erop vloog kon worden opgepakt. Bij het testen van de escaperoom was het door de belichting moeilijk in te schatten of je boven het object vloog. Daarna kwamen we op het idee om het door de grond laten vallen van een object de trigger te maken voor de geheime kamer.

Feedback verwerken

Hey Qun, Sarah en Frans,

Ik ben momenteel jullie project aan het bekijken en in de eerste plaats ben ik onder de indruk. Zeker als je nog nooit iets met Unity hebt gedaan is het dapper om gelijk met 3D te beginnen ipv 2D, maar dat heeft jullie niet afgeschrokken.

Er staat een solide fundering voor een escaperoom-achtig spel waar zeker op doorgebouwd kan worden.

Het gebruik van placeholders voor meerdere objecten is slim, zo kan je je spel door ontwikkelen zonder opgehouden te worden door artists.

De kast is wel erg mooi, is die zelf gemaakt? Zo ja, goed werk! Zo nee, slim om te kijken wat er online allemaal beschikbaar is.

Ook goed gebruik gemaakt van overerving in de items die je vast kan houden, zo voorkom je dubbele code te schrijven.

Het draaien van de camera is erg cool en ook mooi dat de controls van de speler aangepast worden aan de hoek waarmee je kijkt, dat maakt het besturen een stuk makkelijker. Wel had ik graag ergens willen zien wat de controls waren, nu moest ik dat uit de code lezen.

Het checken van de juiste morse code is ook slim gedaan, maar kijk uit met het indrukken van de knop. Als als beide een speler en een object de knop indrukken, en dan verlaat één van de twee de knop weer wordt dat gezien als het loslaten van de knop, terwijl er nog iets het aan het indrukken is.

Het checken van de afstandsbediening is iets minder. Zoals het nu is kan je niet 2 keer hetzelfde cijfer achter elkaar invoeren. Ik zou misschien `Input.GetKeyDown` gebruiken (dat kan je slim doen zodat je niet 10 keer dubbele code hoeft te schrijven) of de muis gebruiken om op de cijfers te klikken (want niet elk toetsenbord heeft een numpad).

Ik vind het wel jammer dat er weinig feedback is naar de speler, ik weet niet echt of ik iets goed of fout doe, hoeveel van een code ik al heb ingevuld, of ik een level heb gehaald, etcetera.

Een camera die de speler volgt zou helemaal cool zijn (maar is wel lastig), anders misschien een wijdere hoek zodat er meer tegelijk in beeld is.

Verder vind ik de mappenstructuur een beetje rommelig. Een goede verdeling van waar welke bestanden komen te staan is erg belangrijk om fijn aan een project te werken. Probeer daar afspraken over te maken en je daar aan te houden.

Geen bestanden los in je Assets map, onderverdeling als er te veel bestanden in een map komen, verzamel alle materials in een aparte map, dat soort dingen.

Veel programmeurs gebruiken ook dat soort afspraken in de namen van je variabelen. Hoofdletters voor public, kleine letters voor lokaal, beginnen met een underscore voor private, etcetera. Dat kan je helpen je code leesbaarder te maken.

Ik heb ook nog een foutje gevonden: Als je de deur opent in het pauzemenu kan je de speler weer besturen, maar sluit het menu niet.

Voor de verlichting (en de schaduwen) zou ik in ieder geval een directional light recht naar beneden willen hebben wijzen, want dan kan ik altijd zien wat het exacte punt is waar ik overheen vlieg voor als ik iets wil laten vallen of recht naar beneden wil gaan. Nu is het lastig om te zien waar ik precies ben. Schuine schaduwen zijn natuurlijk wel mooier op de rest van de objecten om die meer diepte te geven, dus misschien kan je daar een combinatie tussen vinden.

Laat je niet misleiden door mijn lijst van op- en aanmerkingen, over het algemeen is het een redelijk solide project wat voor beginners erg knap is.

Ik vind het wel lastig om te beoordelen wat jullie wel en niet zou moeten kunnen, dus ik heb gewoon zoveel mogelijk opgeschreven wat er in me op kwam.

Veel succes verder met het project, ik zou het graag nog een keer zien als het af is!

Groetjes,
Midas

Op basis van deze feedback heb ik een aantal aanpassingen gemaakt. (Logboek, 15-2-2022, 15:30-16:11)

TV besturing

Ik had uiteindelijk gekozen voor de numpad omdat het invullen van een code makkelijker is voor de gebruiker op die manier. Het alternatief was door middel van de pijltjestoetsen omhoog en naar beneden. Daarna heb ik, op basis van de feedback, de input veranderd in de nummers bovenaan het toetsenbord omdat niet ieder toetsenbord een numpad heeft.

Afronding

Hoewel ik graag nog een UI had willen toevoegen met de besturing, geluiden bij het uitvoeren van bepaalde acties en response bij het halen van een level was daar te weinig tijd voor. Op de laatste dag voor de deadline heb ik de models van Sarah in de escaperoom gezet en aanpassingen gemaakt zodat alles ook met de models nog werkte. Hier heb ik een executable file van gemaakt met een setup wizard en deze op github gezet. Nu kan de game gespeeld worden. Op de poster zal te zien zijn hoe de besturing werkt. Er zal ook een qr-code op staan om de game te downloaden.

Logboek

15-2-2022

15:30-15:47

Door een combinatie van directional light en het oude licht is het makkelijker om diepte in te schatten tijdens het spelen.

15:47-16:05

De afstandsbediening werkt nu met normale nummers. Als de savefile corrupted is wordt deze nu automatisch gereset.

16:05-16:11

Hierarchy opgeruimd.

16:11-17:03

Geprobeerd om de TV Guide cube te veranderen in een boek, maar bij het veranderen van de visuals viel het boek door de grond heen. Het is opgelost door een box collider te gebruiken in plaats van een mesh collider.

10-1-2022

15:45-15:50

Book is nu een child class van HeldObject.

21-12-2021

15:45-16:10

Bug fix:

Data serialization werkt nu. De progress class kan nu de completion value van de 5 levels opslaan.

16:20-16:40

De UI is aan het begin van de game gesloten en kan geopend worden via de Escape toets. Wanneer de UI wordt geopend is de huidige voortgang te zien. De voortgang wordt altijd opgeslagen, zelfs als de game tussendoor uit wordt gezet.

16:40-17:00

UI buttons werken door het toevoegen van een event system.

17:00-17:12

Bewegingen van de speler worden geblokkeerd als de game op pauze staat. De bewegingen zijn weer mogelijk als de speler op "Open Door" drukt.

17:12-17:22

Level 4 toegevoegd: Door het plafond heen weggaan.

17:22-17:26

Het invoeren van een TV-zender geoptimaliseerd.

Bugs:

- Soms wordt de gamedata file leeg gemaakt, de oorzaak is nog niet bekend.

Nog te doen:

- Code netter maken
- Primitives vervangen
- Textures toevoegen
- De morsecode in een boek zetten
- De cijfercode in de video's zetten

18-12-2021

17:45-19:55

De game kan nu opgeslagen worden. Er is een pauzemenü waarin je je voortgang gaat kunnen zien en een knop kan vinden voor level 5.

Bugs:

- Knoppen in het pauzemenü kunnen niet worden ingedrukt
- Serialization werkt niet, dit kan komen doordat er geen eerste save is

Nog te doen:

- KeyBinds voor het openen van de pauze UI (esc)
- Controls bevriezen tenzij de level 5 knop is ingedrukt
- Visuals verbeteren

3-12-2021

14:55-17:45

Level 3 gemaakt:

De tv kan nu video's afspelen. De afstandsbediening kan opgepakt en gebruikt worden via de keypad.

2-11-2021

9:38-10:20

Gewerkt aan de afstandsbediening van de tv.

Plan:

- Kanaal omhoog/naar beneden met arrowUp en arrowDown
- Bijhouden in welke volgorde de kanalen worden gewisseld voor een code

1-10-2021

8:43-8:45

Het boek kan niet meer knippen.

8:46-8:48

Tijdens het lezen kan de speler niet bewegen.

Bug:

Als je tijdens het bewegen op Q drukt, ga je door met bewegen. (opgelost om 8:53)

8:48-9:50

De tv kan een video afspelen

24-9-2021

12:20-14:25

Gewerkt aan interactie met boeken: Een boek kan nu openen en sluiten

Bug:

Als je Q te lang ingedrukt houdt, sluit het boek ook meteen, waardoor het boek gaat knippen

19-9-2021

14:02-14:16

Als je E indrukt, kan je een object pakken en loslaten.

Bugs:

- Het item volgt de speler heel langzaam
- Het item schokt veel als je stilstaat

14:43-15:05

Je hoeft niet op E te drukken om het item te pakken, maar je pakt het automatisch door ertegenaan te vliegen.

Bug fix:

- De schokken zijn weg

- De snelheid waarop het item de speler volgt is hetzelfde als de beweegsnelheid van de speler

Bugs:

- Wanneer je via de zijkant tegen een blokje vliegt dat op de grond ligt, komt het blokje onder de grond terecht. Als je het daarna loslaat, valt het door de grond heen onder de escaperoom.

15:05-15:42

Het blokje wordt alleen opgepakt als je er via de bovenkant op vliegt. Hierdoor is de eerdere bug opgelost. De knop code is aangepast waardoor de deur ook open gaat als je er een object op gooit.

16:16-16:23

Als je uit de escaperoom gaat, wordt het level gereset. In de GameManager wordt bijgehouden welk level de speler uitspeelt. De camera wordt nu bestuurd met J en L in plaats van 4 en 6.

16:23-16:50

De deur kan geopend worden door een morsecode in te voeren met de knop.

10-9-2021

12:20-14:25

Ik heb de camera een focal point gegeven. Dit wordt bestuurd met 4 en 6 op de numpad. De verlichting is veranderd van directional light naar een lamp die binnen de escaperoom schijnt. Er zijn muren toegevoegd achter het scherm. De escaperoom is nu een volledige kamer met meer primitives.

Bug:

- Het werkt alleen op de 4 en 6 boven het toetsenbord

8-9-2021

13:10-15:35

We hebben samen op school levels bedacht in de tijd die daarvoor gegeven was en de exacte indeling van de escaperoom. We hebben ook het herhalingsysteem bedacht.

4-8-2021

12:47-12:54

De quiz over unit 4 gemaakt.

12:54-13:33, 5.1

Geleerd:

- Hoe de OnMouseDown() functie werkt
- Hoe je Torque (draaiing) kan toevoegen aan objecten met een Rigidbody
- Het verschil tussen perspective en isometric view en hoe je dat in de camera kan instellen
- Waar je een GameManager voor gebruikt

13:33-14:02, 5.2

Ik kan nu:

- Een TextUI toevoegen
- De TextUI via code aanpassen
- TMPPro gebruiken en de anchor points in TMPPro

14:01-14:34, 5.3

- Communiceren via code met een Button in de UI
- De scene opnieuw laden waardoor de inhoud gereset wordt naar hoe het was voor je op play drukt

14:34-15:00, 5.4

Ik heb een title screen gemaakt waarin je de difficulty kan aanpassen. Ik kan via de functie AddListener() via code listeners toevoegen in plaats van via de unity editor. Daarna heb ik met empty game objects de objecten in de scene geordend.

3-8-2021

12:30-12:47

Ik heb de code van de escaperoom netjes gemaakt.

12:47-12:55

Door middel van de quiz uit de cursus heb ik bevestigd dat ik unit 3 snap.

12:55-13:25, 4.1

Ik heb geleerd hoe je de camera op een focal point richt (camera instellen als child object van de focal point) en uitleg gekregen over global en local directions. Hierdoor weet ik hoe je orbit view kan programmeren in het camerascript.

13:25-13:57, 4.2

Ik kan nu:

- Bouncy physics gebruiken via een physicsmaterial
- Berekeningen doen met Vector3 objecten
- Vector3.normalized gebruiken

13:57-14:27, 4.3

Ik kan nu:

- Powerups toevoegen
- IEnumerator en de WaitForSeconds() functie gebruiken om de code te laten "wachten"
- Coroutines gebruiken
- Gameobjects zichtbaar en onzichtbaar maken met SetActive()

14:35-14:53, 4.4

Ik kan nu:

- FindObjectsOfType() gebruiken
- Array.Length gebruiken

14:53-15:15

Challenge 4 gedaan

16:13-17:15, escaperoom

Basic gameplay mechanics toegevoegd:

- De deur gaat open als je op de knop drukt
- Je moet de knop ingedrukt houden, anders sluit de deur
- Bewegingen gaan nu via Rigidbody.velocity in plaats van transform.Translate(). Hierdoor wordt er rekening gehouden met physics en hoeft er geen outofbounds code meer te zijn. De speler kan niet meer door voorwerpen heen bewegen.
- De speler wordt gedetecteerd als hij de kamer uit gaat

- De camera kan draaien en is gericht op een focal point in het midden van de kamer

2-8-2021

10:41-11:30, unit 3.1

Ik ben aan unit 3 begonnen van de cursus. In unit 3.1 heb ik geleerd hoe Rigidbody constraints werken, hoe je physics kan gebruiken, hoe AddForce werkt en hoe je de Rigidbody door middel van code kan aanpassen.

11:35-12:03, unit 3.2

Ik heb geleerd hoe je een herhalende achtergrond kan maken waardoor het lijkt alsof je achtergrond eindeloos doorgaat, hoe je via CompareTag() gameobjects in de code kan identificeren en hoe je dezelfde data in verschillende scripts kan gebruiken.

12:38-13:05, unit 3.3

In deze unit heb ik geleerd hoe animaties werken en hoe je deze animaties kan besturen en aanpassen via een code.

13:11-13:35, unit 3.4

Ik heb geleerd hoe je special effects kan toevoegen zoals:

- Particles
- Sound effects
- Achtergrondmuziek

13:45-14:06

Ik heb de challenge van unit 3 gedaan.

14:12-15:18

In de escaperoom heb ik de PlayerController script gemaakt. De speler beweegt met wasd of de pijltjestoetsen. Met shift en spatie kan de speler omhoog en naar beneden bewegen. Het klikken van de muis wordt gedetecteerd. Er wordt gecontroleerd of de speler te ver uit de escaperoom beweegt en dan wordt de beweging ongedaan gemaakt. Bewegingen gebeuren via transform.Translate().

30-7-2021

14:26 tot 15:56

Ik heb de top down game challenge van de unity cursus die te vinden is op <https://learn.unity.com/course/create-with-code>. Daarbij heb ik gebruik gemaakt van een spawn manager, prefabs en user input.

16:33-16:44

Ik heb de globale vormen van items in de escaperoom toegevoegd (de muren, de speler, blokjes als de boekenkast, de tv en boeken etc.). Daarna heb ik de basic materials toegevoegd voor kleuren om items uit elkaar te kunnen houden. De camera is op de juiste plek ingesteld en de layout van de escaperoom is gemaakt. Daarna heb ik geleerd hoe je een backup van het project maakt.

16:44-17:00

Ik heb de quiz gemaakt en daarmee bevestigd dat ik weet:

- Hoe collisions en interactie met collisions werkt
- Wat prefabs zijn en hoe ze gebruikt worden
- Hoe je een spawnmanager maakt
- Wat Debug.Log() is

Unit 1-2

Iedere unit kostte me ongeveer 4-5 uur. Bij de eerste 3 units heb ik geen logboek bijgehouden, maar mijn snelheid zal ongeveer hetzelfde zijn geweest.

Bronnen

GameDevLeague: <https://discord.gg/0TYNJfCU4De7YIk8>

Unity cursus: <https://learn.unity.com/course/create-with-code>

Unity documentation: <https://docs.unity3d.com/Manual/index.html>

Data saving: <https://www.red-gate.com/simple-talk/development/dotnet-development/saving-game-data-with-unity/>

Github gebruiken met Unity: <https://www.youtube.com/watch?v=qpXcvS-g3g>