Bedrijfspunt: Game met save systeem

Naam: Sten de Boer

Studentnummer: 500773940

Datum: 08-07-2021

Docent: Marianne Bossema

Inhoudsopgave

1. Reflectie: Save systeem

Hier wordt besproken met welke technieken er data kan worden opgeslagen in Unity, welke techniek er is gebuikt om binnen de game data op te slaan en waarom er voor deze techniek is gekozen. Als laatst wordt er gereflecteerd op het proces voor het maken van het save systeem.

1.1 Save systeem technieken en gekozen techniek

Voor het opslaan van data in Unity zijn er drie technieken. Deze drie technieken zijn PlayerPrefs, JSON en de binary formatter. Hieronder wordt er besproken wat welke techniek inhoud.

1.1.1 Player prefs

Player prefs is een systeem dat al ingebouwd is in Unity. Dit maakt het makkelijk en snel om te gebruiken binnen een game, gezien het binnen een paar regels code allemaal werkt. Binnen de player prefs kunnen er alleen maar een paar basis data types worden opgeslagen en niet de meer geavanceerde data types die Unity gebruikt. Dit houdt in dat de enige data die kan worden opgeslagen nummers (int/ float) en letters (string) zijn. Verder kunnen er geen booleans (true/ false data) worden opgeslagen en er kunnen ook geen lijsten worden opgeslagen. Dit houdt dus in dat voor elk deel dat moet worden opgeslagen een apart variabele moet worden aangemaakt. Ook is deze data heel makkelijk aan te passen door spelers en zouden ze dus bijvoorbeeld extra levens in de game kunnen aanmaken.

1.1.2 JSON

De data kan ook worden opgeslagen in een apart JSON bestand. In dit bestand kunnen wel alle soorten basis datatypen worden opgeslagen. Dit houdt dus in dat er int/ float, string een booleans kunnen worden opgeslagen. Er kan ook een lijst van deze datatypen worden opgeslagen in een JSON bestand. Deze techniek is wel moeilijker om te implementeren, maar er kan wel veel meer data worden opgeslagen met deze techniek. Het moeilijke met deze techniek is alleen dat de data nog steeds makkelijk kan worden aangepast door spelers. Om dit moeilijker te maken zou de data door een encryptie systeem moeten worden gehaald om het moeilijker te maken om de data aan te passen.

1.1.3 Binary formatter

De data kan ook met een binary formatter worden opgeslagen. Net zoals het JSON systeem kan dit systeem alle basis data soorten en ook hele lijsten van deze soorten data opslaan. In plaats van deze data op te slaan in makkelijk te lezen code wordt deze data opgeslagen in binaire taal. Dit maakt het moeilijk om de data aan te passen. Voor deze techniek hoeft er dus geen extra encryptie te worden gebruikt om de data veilig te maken van de speler.

1.1.4 Gekozen techniek

Ik heb ervoor gekozen om de binary formatter techniek te gebruiken binnen de game. Player prefs heeft niet genoeg mogelijkheden om data goed op te slaan en is alleen voor de meest simpele data. Data met een JSON opslaan maakt de data te makkelijk bereikbaar en

aanpasbaar voor de spelers. Een heel encryptie systeem hiervoor maken zou veel extra werk zijn, als het ook makkelijker kan door alle data in binaire taal te zetten wat de meeste spelers het moeilijk maakt om de data aan te passen. Met een goed encryptie systeem zou data opslaan in een JSON bestand waarschijnlijk nog veiliger zijn, maar dat is meer werk dan dat ik tijd had om deze game te maken.

1.2 Reflectie op maken save systeem

Doordat alle technieken om data op te slaan alleen basis type data kunnen opslaan betekent dit dat als ik een nieuw script aanmaak kan ik deze niet in zijn geheel opslaan. Dus om data goed op te slaan moet er rekening mee worden gehouden dat de script dat moet worden opgeslagen makkelijk op te slaan is met basis data types. Bijvoorbeeld als een speler items heeft in de game, dat alle items in de game een vaste index hebben. Hierdoor kunnen die items die de speler heeft in een lijst met ints worden opgeslagen en hoeft item data niet apart per script opgeslagen te worden. Verder is het ook handig om zo weinig mogelijk data te hoeven op slaan, gezien alles wat er wordt opgeslagen ook weer correct moet worden ingeladen. Hoe meer data hiervoor moeten worden opgeslagen, des te groter de kans dat er bugs komen terwijl er data wordt ingeladen. Ook moet er voor elk stuk data dat er wordt opgeslagen meer code worden geschreven voor opslaan en inladen wat kan leiden tot onoverzichtelijkheid en onnodige code.

2. Reflectie: Duidelijkheid game

Hier wordt besproken hoe de playtest verliep en wat er onduidelijk was voor de spelers tijdens de playtest. Als laatst zal er worden besproken waar ik de volgende keer op zal letten tijdens het designen van een game.

2.1 Verlopen playtest

Het doel van de playtest was om de game uit te spelen zonder dat ik de playtesters hielp met het ontdekken van hoe de game werkt. De spelers hadden een half uur tot drie kwartier om de game uit te spelen. Alle mechanics van de game zelf waren duidelijk voor de playtesters. Iedereen heeft de game kunnen uitspelen in de bovengenoemde tijdsperiode. Tijdens de playtest kwamen nog wel een paar problemen naar voren binnen de game. Deze waren dat de spelers sneller naar links liepen dan dat ze naar rechts gingen. Doordat ze te snel naar links accelereerde konden de spelers niet goed objecten aan de linkerkant van hun character neerzetten, wat soms belangrijk is binnen de game gezien objecten soms op specifieke posities moeten staan. Verder kunnen er ook objecten binnen muren geplaatst worden, wat verder niet voor problemen zorgt, gezien de objecten ook weer uit de muren gehaald kunnen worden. Dit ziet er alleen niet heel net uit. Ook volgt de sleutel de speler soms van een te verre afstand wat voor verwarring kan zorgen en de speler kan laten denken dat deze de sleutel niet goed heeft. Hierdoor zou de speler dan niet goed de deur kunnen openen. Als laatst is het mogelijk om in sommige levels vast te raken in bepaalde gebieden, waardoor de speler het hele progress van het level moet opgeven en het level moet herstarten. Dit geeft een vervelende ervaring voor de spelers en dus zou er een bepaald block moeten komen dat er voor zorgt dat alleen de speler doodgaat en terug naar de startpositie gaat. Deze kunnen dan op alle locaties worden geplaatst waar het mogelijk is om vast te komen zitten en op die manier kan de speler altijd veilig terugkomen bij de start, zonder dat er veel acties herhaald hoeven te worden.

2.2 Reflectie design game

Doordat de sleutel de speler volgt was dachten sommige spelers dat de sleutel de deur moest aanraken, terwijl dit niet het geval is en de deur altijd open gaat, zodra de speler de deur heeft opgepakt. Misschien is het dus beter om de sleutel, zodra deze door de speler wordt opgepakt naar de deur te laten vliegen en de deur een open deur sprite te geven om aan te geven dat de deur open is in plaats van de sleutel de speler te laten volgen. Ook is het waarschijnlijk handig om voor uitleg van controls binnen de game plaatjes te gebruiken in plaats van alleen maar tekst. Het probleem was nu dat het soms niet duidelijk was dat de speler objecten kon vasthouden door X ingedrukt te blijven houden. Hierdoor konden ze latere puzzels niet oplossen en moest ik dit ze verder uitleggen. Spelers hebben een grotere kans dat ze wel naar de plaatjes kijken in plaats van een heel blok tekst te lezen en dus zouden ze dan alles begrijpen. Verder zou het ook mogelijk zijn om in een van de intro levels de speler te forceren om een object rond te laten dragen, zodat ze alle belangrijke mechanics leren van de game. Ook zou de tekst in overwereld kunnen worden vervangen met plaatjes, voor dezelfde redenen als hierboven benoemd zijn. Daarnaast mist de game nog wat visuals voor acties, zoals de game opslaan, de opgeslagen game verwijderen en het afsluiten van de game/ een level. Deze laatste twee zouden ook nog een pop-up menu

moeten hebben die vragen aan de speler of ze zeker weten dat ze een van deze acties willen voltooien, gezien de speler nu met een misclick van een knop heel wat progressie kan verliezen, zonder dat ze dat willen. Ook is het design van de level soms te dichtbij elkaar, waardoor het bewegen in de game niet helemaal lekker loopt, gezien de speler vast loopt tegen objecten of objecten niet goed kwijt kan neerleggen, zonder er tegenaan te lopen en ze per ongeluk te verplaatsen. Dit kan een vervelende ervaring geven voor de speler, doordat ze niet soepel kunnen doen wat ze willen doen en bepaalde acties moeten herhalen die eigenlijk maar een keer gedaan zouden moeten worden. Als laatst missen de sloten in de game die uit gaan met de knoppen een sprite voor als ze uit staan. Hierdoor kan het onduidelijk zijn waar ze terug zouden komen als er geen object op het slot staat en dit geeft extra moeilijkheid aan de speler, waar dit niet nodig zou zijn.

3. Urenverantwoording

Datum	Uren	Actie
24-06-2021	6 uur	Art assets gevonden, speler beweging, camerabeweging, zwaartekracht wissel en animaties tijdens zwaartekracht wissel
25-06-2021	4 uur	Alle speler animaties toegevoegd, cooldown UI van zwaartekracht wissel, bugfix voor niet vallen, meer art gevonden voor wat niet in asset pack zat, cannon creation (verwijderde feature)
28-06-2021	2 uur	Grapple cannon gemaakt (Naar speler verplaatst), eerste level gemaakt
30-06-2021	6 uur	Overworld gemaakt, grapple aan speler gemaakt en cannons verwijderd, fixed prefabs, fixed player animations, tweede level toegevoegd
01-07-2021	6 uur	4 extra levels toegevoegd (intro/ crossroad/ first up path levels), UFO progression toegevoegd, button lock openers added
02-07-2021	6 uur	Alle levels voor intro/ crossroad en first up path afgemaakt, Levels voor second up path gemaakt
04-07-2021	4 uur	Bomb mechanic gemaakt, slimes gemaakt, right path1 gemaakt, begin right path 2 gemaakt.
05-07-2021	6 uur	Right path 2 afgemaakt, weight mechanic gemaakt, left path 1 gemaakt en begin left path 2 gemaakt.
06-07-2021	8 uur	Eind left path 2 gemaakt, alle uitleg van mechanics toegevoegd en playtest uitgevoerd, begin save systeem onderzoeken/ maken
07-07-2021	8 uur	Save systeem onderzoek/ afgemaakt, links loop bug gefixt, eind scherm gemaakt, start verslag
08-07-2021	4 uur	Verslag afgemaakt

4. Bibliografie

Brackeys, director. *SAVE & LOAD SYSTEM in Unity*. 2018. *Youtube*, https://www.youtube.com/watch?v=XOjd_qU2Ido.

Comp-3 Interactive, director. *Saving Data using JSON Serialization [Unity Tutorial]*. 2020. *Youtube*, https://www.youtube.com/watch?v=aV2OA4f5ru8.

Talbert, Lance. “Saving Game Data with Unity.” *Redgate*, 3 februarie 2020, https://www.red-gate.com/simple-talk/dotnet/c-programming/saving-game-data-with-unity/.