

Conceptvorming

De keuze om een Android en een Windows build te maken leek me een makkelijke uitweg. Ik heb daar nog even over getwijfeld, maar ik heb toch besloten om deze twee platformen te kiezen zodat ik niet aan het einde van het blok in de stress zou raken omdat dingen onverwachts niet werken. Sowieso zijn Android en Windows voor mij het makkelijkst te testen, omdat ik een android telefoon en een windows laptop heb.

Ik heb in Unity gewerkt omdat ik hier nog steeds veel in kan en wil groeien. Elk project dat ik er in kan maken is mooi meegenomen.

Met die keuzes in stand, ben ik gaan nadenken over wat ik wilde maken. Ik neigde eerst naar een platformer. Hier kreeg ik geen vast idee voor waar ik echt iets mee wilde doen, tot ik ORB bedacht. In ORB bestuur je niet een character, maar de wereld om hem heen. Door de omgeving te draaien begeleid je ORB naar een doel in het level. Onderweg kom je obstakels tegen waar je langs moet zien te komen. De bedoeling is om het doel zo snel mogelijk te bereiken.

Iteraties

Ik was eerst van plan om hier een 2D spel van te maken. Bij het bouwen van de wereld en de controls merkte ik dat het heel moeilijk was om het punt van zwaartekracht in een 2D game van plek te veranderen. Ik dacht er over om de wereld dan te verplaatsen, met een apart gameobject als pivot point wat de speler volgt. Ook dit kreeg ik niet functionerend. Na even verder spitten kwam ik er achter dat het heel makkelijk is om het zwaartekrachtpunt te verplaatsen in 3D physics. Daarom heb ik het spel opnieuw opgebouwd in 3D physics. Wanneer de zwaartekracht wordt verplaatst wordt ook de camera gedraaid, zodat de speler altijd het gevoel heeft dat het zwaartepunt beneden zit. Zo creëer ik alsnog de illusie dat de hele wereld wordt gedraaid.

Dit spel had ook gebruik kunnen maken van de tilt functie van de meer recentere mobiele telefoons. Omdat mijn telefoon dit echter niet heeft, zou dit heel lastig zijn geweest om te testen. Daarom heb ik voor een andere smartphone specifieke manier van controls gekozen: touch. Aaron heeft mij geholpen met de code waarmee ik de angle tussen twee posities kan berekenen. Hiermee heb ik een script kunnen maken waarmee je heel smooth de camera met twee vingers kan draaien.

Een laptop of pc heeft over het algemeen geen tilt. Daarom moest het spel hierop met andere controls gespeeld worden. Ik heb gekozen voor de pijltjestoetsen omdat dit natuurlijk aanvoelt en omdat ik dacht dat de code hiervan heel anders zou zijn dan die van de touch input. Het bleek dat dit zo ongeveer in één regel code gedaan kon worden – terwijl er bij de touch meerdere extra functies nodig waren. Wat dat betreft had ik gelijk dat het erg anders zou zijn.

Design Patterns

Een logische stap was om gebruik te maken van het Abstract Factory design pattern. Deze stuurt de platform specifieke code door naar de game code. Elke platform specifieke class heeft een functie die een float waarde returnt, welke direct in een variabele wordt opgeslagen. Met deze variabele wordt er gekeken hoe ver de wereld gedraaid moet worden.