**Input/ Output documentatie**

Pelle Meijer & Nanne van Zanten

**Project concept:**

Voor ons project hebben we gekozen om onderzoek te doen naar audio input. Dat doen wij door een 2d game te bouwen waarin je de game bestuurd met je stem. In de game speel je als een piloot die obstakels moet ontwijken, hoe luider en hoger je stem is de hoger dat de piloot vliegt.

Daarin moeten wij onderzoek gaan doen naar: Audio input, frequenties en voor de obstakels dachten wij aan een terrein gegenereerd met perlin noise.

**Dag 1:**

Vandaag hebben wij gekeken naar talen en engines waarin wij de game kunnen maken. Wij dachten na over C++, Javascript, GameMaker en P5JS. Uiteindelijk zijn wij voor P5JS gegaan omdat het een hele fijne liberary is voor prototypen van 2d games. P5JS heeft ook al een ingebouwd microfoon object en het is crossplatform dus dat maakt het nog gunstiger ook zijn wij erg ervaren in javascript.

**Dag 2:**

Vandaag zijn wij onderzoek gaan doen naar het p5 Object p5.AudioIn(). Dit object creëert een microfoon op de pagina en geeft ons toegang aan de data. Hiermee hebben wij een rondje op en neer laten gaan met de volume van je stem.

**Dag 3:**

Vandaag hebben wij de terrain generation gemaakt. Hiervoor hebben wij de noise functie van p5 gebruikt. We hebben daarmee perlin noise gegenereerd en aan de bovenkant en onderkant daarmee rotsen geplaatst.

**Dag 4:**

Vandaag hebben wij ervoor gezorgd dat de rotsen met een bepaalde naar links gaan en dat als je de rotsen aanraakt dat je dan opnieuw begint. Dat hebben wij gedaan door te kijken hoe hoog de rots is op de x positie van de speler en dan te kijken of de speler daar onder zit zo ja dan is de xpos van de wereld weer 0. Toen we dit eerst implementeerden was er een enorme frame drop, Omdat wij in de for loop die de rots op elke x rij tekende keken of de x positie die we die loop aan het tekenen waren gelijk was aan de player x pos, zo ja? Kijk dan of de player y boven de hoogte is van de y van die pixelrij. Alleen dat is natuurlijk veel de zwaar als je dat 200x elke frame wilt doen. Dus hebben wij dat een keer per frame gedaan, maar toen moesten wij de noise wel weer opnieuw bereken.

**Dag 5:**

Vandaag hebben wij de player sprite gemaakt en geïmplementeerd. En we hebben ook geprobeerd het project uit de p5 online editor te halen en local te kunnen bewerken. Alleen jammer genoeg als wij het zelf in chrome wouden openen wou chrome ons niet toegang geven om de mic te gebruiken. Daarmee hebben wij die dag veel me gepield maar is het ons uiteindelijk niet gelukt.

**Dag 6:**

Het is ons vandaag eindelijk gelukt om de chrome microfoon aan de praat te krijgen. Het probreem was namelijk dat wij de mic meteen op de eerste frame wouden gebruiken, maar chrome wilt eerst player input voordat het een microfoon toestaat. Dus hebben wij uiteindelijk in de update gekeken of we al toegang hebben tot de mic, zo ja? Dan word de game geladen. Anders wacht de computer op een click op het scherm van de player.

**Dag 7:**

Vandaag hebben wij besloten om de game local multiplayer te maken. Wij hebben gekozen om Node.js te gebruiken want dat is een hele handige socket server library voor javascript. Het is ons vandaag gelukt om op een socket te comuniceren.

**Dag 8:**

We hebben vandaag de game omgebouwd zodat beiden clients y pos meegeeft en inneemt en dat het die waarden displayed als een vijand.

**Dag 9:**

Vandaag hebben wij ervoor gezorgd dat de vliegtuigen elkaar kunnen inhalen.