Zelfevaluatie GDC7 Sound

Naam: Frank Severijns   
StudentNr: 2383764   
Datum: 2-12-2015

Voor mijn GDC7 Sound project heb ik een Space Invaders spel gemaakt waarin een aantal game mechanics beinvloed worden door muziek.

Toen ik aan het project begon had ik voor ogen om de beat uit de muziek te lezen en de bewegingen van de speler, aliens en lasers daarop te synchroniseren. Dit bleek achteraf veel moeilijk dan ik had gedacht en had uiteindelijk geen werkende implementatie kunnen maken.  
Voor bronnen heb ik voornamelijk op het internet gezocht naar “Onset detection” en “Beat detection”. Hierdoor vond ik een wordpress blog (<http://www.badlogicgames.com/wordpress/?cat=18>) waar onset detection uitgebreid beschreven wordt. Deze tutorial was helaas gebaseerd op Java, ik vond het daardoor lastig om dit op Unity toe te passen en ben daarom verder gaan zoeken naar een Unity-gerelateerde bron. Daardoor stuitte ik op een github project (<https://github.com/allanpichardo/Unity-Beat-Detection>) en een pastebin met een Unity script (<http://pastebin.com/myXiu97R>, bijna exact hetzelfde als de github code). Door deze in mijn project te implementeren hoopte ik een idee te krijgen van hoe ik de beat detection wou laten werken.   
De beat detection die uit deze code kwam werkte goed (genoeg), en ik besloot daarom om de code zelf over te nemen om zo meer inzicht te krijgen in hoe het werkt (en de onnodige code eruit te krijgen). Ik snapte het uiteindelijk alleen op een basis niveau en niet genoeg om zelf een werkende implementatie te krijgen (mijn eigen versie, gebaseerd op dat script, las door onbekende reden geen beats uit.)   
Wel kreeg ik een zeer ruwe versie werkend waarbij ik de output data van de Audiosource uitlas en daaruit een gemiddelde berekende, pieken die boven dat gemiddelde uitkwamen werden aangezien als een beat. Dit was echter verre van precies genoeg.

Toen ik bij de oplevering een reparatie moest doen koos ik, gebaseerd op feedback van de docent, ervoor om beat detection te laten voor wat het is. Ik had het gevoel dat ik op dat gebied niet veel verder zou komen. In plaats daarvan zocht ik naar een manier om de intensiteit (decibel waarde) van de muziek uit te lezen en daarmee de aliens te beinvloeden. Hierbij kreeg ik hulp van Sander Dings, die mij een code sample uit zijn eigen project gaf. Door gebrek aan documentatie kon ik er echter weinig mee, omdat ik niet goed begreep wat de code nou eigenlijk deed. Later vond ik toevallig een bron (<http://answers.unity3d.com/questions/157940/getoutputdata-and-getspectrumdata-they-represent-t.html>) waarop Sander blijkbaar zijn code had gebaseerd. Dankzij de uitleg die erbij gegeven wordt en de comments in de code kreeg ik al meer inzicht in wat de code doet.  
Uiteindelijk heb ik deze code toegepast om de aliens in het spel van gedrag te laten veranderen op dasis van de decibel waarde uit de code (bij hoge dB (boven 0) gaan de aliens sneller en schieten ze vaker, bij lage dB (onder 0) bewegen ze langzamer en schieten ze minder.  
Toen ik het uitlezen van de dB werkend had heb ik met Tom Vaessen besproken wat ik nog meer zou kunnen doen, omdat ik zelf vond dat alleen de dB waarde berekenen waarschijnlijk niet genoeg zou zijn. Tom leverde het idee om de frequency uit het spectrum te lezen en het schieten van de aliens te baseren op de pitch, in plaats van de dB waarde. Dezelfde bron gaf had hier ook een implementatie voor, waar ik mijn eigen op gebaseerd heb. Ook hiervoor heb ik mijn best gedaan om de code zo veel mogelijk te begrijpen.  
In mijn uiteindelijke versie schieten de aliens vaker bij hoge pitch waardes.