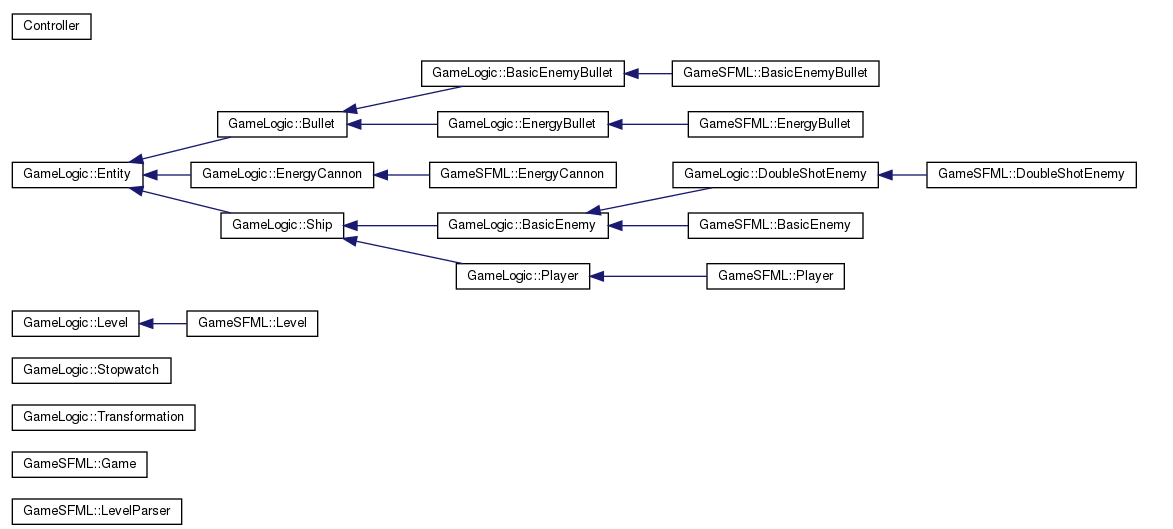
Ian Dierckx  
20141756

**Rapport Project Gevorderd Programmeren:**

**Space Invaders**

**Algemene Structuur:**

Voor de datastructuur heb ik me voor een deel gebaseerd op het voorbeeld gegeven in de opgave. Zo heb ik een klasse Entity waar het grootste deel van de andere klassen zijn van afgeleid. De klasse Entity heeft zelf 3 subklassen. Zo is er de klasse Bullet, uit deze worden zowel de kogels geschoten door de speler als de kogels geschoten door de aliens afgeleid. Ook is er de klasse EnergyCannon, meer uitleg over deze klasse volgt later. Tenslotte is er de klasse Ship, hieruit wordt worden zowel de Player klasse als de BasicEnemy klasse afgeleid.



Elke klasse die ook visueel zichtbaar moet zijn heeft op zijn beurt een subklasse in de GameSFML namespace die ervoor zorgt dat deze correct wordt weergegeven op het scherm. De afgeleide klassen van Entity met een SFML subklassen zijn de volgende: BasicEnemyBullet, EnergyBullet, EnergyCannon, BasicEnemy, DoubleShotEnemy en Player. Voor elk van deze klasse volgt later meer uitleg.

Buiten de Entity klasse zijn er nog een aantal klassen die hier niet van worden afgeleid. Zo is er de Controller klasse, de Level klasse, de Stopwatch klasse, de Transformation klasse, de Game klasse en de LevelParser klasse. Ook van deze klassen volgt er meer uitleg later.

**Bullet en zijn subklassen:**

Deze klassen representeren alle kogels die worden afgeschoten in het spel. Zo hebben we de BasicEnemyBullet, deze wordt afgeschoten door alle alien. Deze klasse weet enkel wat hij moet doen als er een collision is met de Player, waarbij deze de Player health met 1 verlaagt en als deze op 0 komt te staan de Player zal vernietigen wat neerkomt op een game over. Alle andere collisions worden wel geregistreerd door de collision check, maar worden genegeerd.

De tweede klasse die wordt afgeleid van Bullet is de EnergyBullet klasse. Deze representeert de kogels afgeschoten door de EnergyCannons en worden gebruikt door de speler om de aliens neer te schieten. Deze zijn collision handler zal enkel collisions met enemies afhandelen en de rest negeren. Bij een collision met een enemy zal het de enemy’s health met 1 verlagen en als deze op 0 komt te staan vernietigen.

Zowel BasicEnemyBullet als EnergyBullet hebben een afgeleide SFML-versie die ervoor zorgt dat deze correct op het scherm wordt afgebeeld.

**EnergyCannon:**

In plaats van de originele versie van Space Invaders te implementeren heb ik gekozen om er zelf een variant op te verzinnen. Bij deze variant kan onder andere de speler zelf niet schieten en moet deze de EnergyCannons die verspreid staan over de level gebruiken. Zodra de Player onder een EnergyCannon staat kan hij deze gebruiken om te schieten, als er nog genoeg kogels in zitten. Zodra een kanon voor een bepaalde tijd niet wordt gebruikt zal het automatisch herladen tot zijn maximum aantal kogels, wat normaal 8 is. Ik heb ervoor geopteerd om geen collision te voorzien met de EnergyCannon klasse. Een mogelijke uitbreiding die ik dus nog zou kunnen hebben doen was om ervoor te zorgen dat een kanon vernietigd kon worden door de aliens. Ook EnergyCannon heeft een SFML subklasse die ervoor zorgt dat deze correct op het scherm wordt weergeven.

**Ship en zijn subklassen:**

Ship zelf heeft 2 subklassen, de Player klasse en de BasicEnemy klasse. De Player klasse, zoals de naam doet vermoeden, stelt de speler voor. Hoewel de speler visueel niet een schip voorstelt leidt deze toch af van Ship. De reden hiervoor is dat de keuze om de speler geen schip te maken later werd gemaakt dan dat de Player en Ship klassen werden gemaakt en ik ben vergeten om de Ship klasse te hernoemen naar iets dat beter zou passen. Qua interne werking is de Player niet veranderd na de keuze om deze visueel geen schip te maken.

De tweede klasse die afleidt van Ship is BasicEnemy. Deze representeert de doorsnee enemy en wordt ook gebruikt als parent klasse van de DoubleShotEnemy klasse, een meer geavanceerde alien die 2 levens heeft en dus 2 keer moet geraakt worden door de speler.

Zowel Player, BasicEnemy als DoubleShotEnemy heeft een SFML subklasse die ervoor zorgt dat deze correct op het scherm verschijnen.

**MVC pattern:**

In de opgave stond dat we het Model View Controller (MVC) pattern moesten gebruiken en dit heb ik dan ook gedaan. Het Model wordt gedaan door de alles in de GameLogic namespace. De View is gerepresenteerd door de GameSFML namespace welke alle SFML subklasse bevat van alles in de GameLogic namespace (waar nodig natuurlijk). Het Controller deel wordt geregeld door de Controller klasse. Deze neemt de input gekregen door de View en regelt wat het Model er mee moet doen. De Controller volgt een singelton pattern aangezien er steeds maar 1 controller mag zijn.

**Stopwatch en Transformation klassen:**

De Stopwatch klasse zorgt ervoor dat de game op alle PCs even snel loopt en de Transformation klasse zorgt voor de vertaling van de coördinaten gebruikt in de Game Logic naar de coördinaten gebruikt door SFML. Beide klassen volgen het singleton pattern zoals gevraagd in de opgave.

**Level klasse:**

Ik heb geopteerd om een Level klasse te implementeren. Deze houdt alle informatie bij die van toepassing is op een enkel level in het spel. Dit is onder andere bijvoorbeeld alle aliens die nog tot dit level behoren of de kogels die momenteel afgeschoten zijn. Door hier een klasse van te maken kon ik mijn LevelParser gemakkelijk van een json file een object van deze klasse laten maken per level.

Level heeft ook een SFML subklasse welke ervoor zorgt dat alles dat tot het level behoort op het scherm wordt getekend door de draw functie van alle objecten op te roepen.

**Game klasse:**

De main game loop bevindt zich in de Game klasse. De klasse onder andere creëert de window waar alles in wordt getekend.

**Geen observer pattern:**

Zoals te merken is heb ik geen observer pattern gebruikt, ook al werd dit wel gevraagd in de opgave. De reden hiervoor is dat het een tijdje duurde voor ik goed begreep waarvoor de observer pattern diende en eens ik het begreep zag ik niet echt een meerwaarde om hetgeen dat al was gemaakt van het project om te bouwen zodat ik er een observer aan kon toevoegen.