

The background of the top half of the page is a pixel art illustration of a space scene. It features a dark blue space with various shades of blue and purple nebulae. There are several stars, some of which are bright white with four-pointed flares. A small, crescent-shaped planet is visible in the upper left. A larger, more detailed planet with a blue and white striped pattern is on the left side. A white, jagged, zigzag line representing a rocket's trail curves from the bottom left towards the center. At the end of this trail is a small, pixelated rocket ship with a red and white body and a blue nose cone.

# Technisch verslag

## Totally

## original

## space shooter

**Gemaakt door:**

Jari Fuijkschot

1760874

Stijn Oosting

1766982

Tim Stolker

1761781

Alexander de Graaff

1761310

**Vak:**

Thema Game

**Docent:**

Gerald Ovink

**Klas:**

V2D

**Opdracht:**

Game - Thema Game

**Team:**

B

<b>Inleiding</b>	<b>3</b>
<b>Algemene Informatie:</b>	<b>3</b>
<b>Classes</b>	<b>4</b>
State Machine:	4
SplashClass:	4
MainMenuClass:	4
LeaderBoardClass:	4
CharacterMenu:	5
GameControl:	5
GameOver:	5
Enemy:	5
Enemy*	5
Boss1:	5
Upgrade	5
Action:	5
PlayerStats:	5
Rectangle:	5
Picture:	5
DisplayTexts:	5
<b>Diagrammen</b>	<b>5</b>

# Inleiding

In het volgende hoofdstukken vind je informatie over de code van het spel “**Totally Original Space Shooter**”. Dit spel is gemaakt door studenten van HBO ICT, Technische Informatica op de Hogeschool Utrecht.

## Algemene Informatie:

**Code:** C++ 14

**Libraries:**

<SFML/Graphics.hpp>

<SFML/Audio.hpp>

<iostream>

<functional>

<fstream>

<vector>

**Bestanden:**

File naam:	Extensie
Main	.cpp
SplashScreen	.hpp
Statemachine	.hpp
MainMenu	.hpp
LeaderBoard	.hpp
CharacterMenu	.hpp
GameControl	.hpp
Enemy	.hpp
Objects	.hpp
Upgrade	.hpp
GameOver	.hpp
leaderboard	.txt
images	folder
sounds	folder
Makefile	-

# Classes

## State Machine:

In de main.cpp wordt de state machine aangeroepen, dit is een class die alle states regelt van de game. Deze states bepalen in welke fase van het spel de speler zich bevindt, er zijn 6 states:

SPLASH, MENU, LEADERBOARD, CHARACTER, GAME en GAMEOVER.

Elke state zal dan de bijbehorende class loop aanroepen die alles regelt wat er in de bepaalde state moet gaan gebeuren. Een aantal van deze class loops returned een string waar in staat naar welke state hij hierna moet gaan, dit kan bijvoorbeeld bij de GAME state die "GAMEOVER" en "MENU" kan returned die naar de bijbehorende state leidt.

## SplashClass:

De SplashClass regelt de Splash screen die afspeelt aan het begin van het opstarten. Deze class regelt voornamelijk de animatie die afspeelt en doet verder niet veel. Deze class returned niks.

## MainMenuClass:

De MainMenuClass regelt alles wat er in de MainMenuLoop gebeurt. Hier gaat het programma door een loop die telkens de functies ProcessInput() en Render() aanroepen. In bijna alle klassen komen deze functies voor en handelen een groot deel van wat de speler kan zien op het scherm.

### Render():

Render regelt, zoals de naam al zegt, het renderen. Dit doet het door alle lijsten van objecten te lopen en ze steeds individueel te drawen op de window.

### ProcessInput():

Deze functie registreert wat en waar de speler geclicked heeft en of de speler iets op het toetsenbord heeft ingedrukt. Dit wordt doorgegeven aan de action class die vervolgens die input verwerkt.

Er zijn in de klasse op 3 knoppen te clicken, 2 van deze knoppen returnen de bijbehorende string naar de state machine. Dit zijn "GAME", "LEADERBOARD" en de 3de knop sluit het spel.

## LeaderBoardClass:

De LeaderBoardClass laat op het scherm de top 10 hoogste scores met de namen zien. Deze top 10 scores worden opgeslagen in een text document. Hierbij laat het ook een return knop zien. Als de gebruiker hierop klikt zal hij terug gaan naar het MainMenu.

## CharacterMenu:

De CharacterMenu klasse regelt alles wat te maken heeft met het Character selection screen.

Hierin kan de speler kiezen tussen de 3 characters. Als de gebruiker heeft geklikt op 1 van de 3 characters zal het spel starten met stats die gebaseerd zijn op de gekozen character. Indien de speler op de return knop drukt zal de speler weer terug gaan de Main Menu

Deze klasse heeft 4 verschillende return strings, 3 voor de verschillende characters en 1 voor het returned naar main menu. De 3 verschillende character returns veranderen de waarden van de 'PlayersStats' class die vervolgens meegegeven wordt aan de GameController class.

## GameControl:

In de GameController class wordt alles geregeld wat te maken heeft met het spelen van het spel. Hierbij gaat de loop telkens door de functies ProcessInput(), Update() en Render(). Deze functies handelen een groot aantal van de verschillende elementen van het spel waaronder het kunnen bewegen, schieten, het spawnen van enemies, het spawnen van de trader, etc. Hierbij maakt de klasse, naast zijn eigen functies, gebruik van een verschillend aantal andere classes zoals Upgrade en Enemies. Een van de belangrijkste functies van GameController is Update().

Update();

GameControl is niet de enige klasse die gebruikt maakt van de Update() functie maar is zeker degene die er het meest gebruik van maakt. Deze functie wordt om de 20 ms aangeroepen en kijkt dan of er iets is veranderd in het spel. Deze functie regelt bijvoorbeeld het bewegen van de kogels, het bewegen van de enemies en regelt ook de trader die om de 3 waves langkomt. De trader biedt de speler 3 upgrades aan in ruil voor credits.

Zoals al eerder is genoemd is, heeft GameController 2 return mogelijkheden. Dit zijn "GAMEOVER" en "MENU". "MENU" wordt gereturned zodra de speler op esc drukt om het spel te pauzeren en vervolgens op de main menu knop drukt. "GAMEOVER" wordt gereturned zodra de speler zijn levens 0 zijn en roept de GameOver class aan.

## GameOver:

De Gameover class kijkt na het doodgaan van de speler of de speler een score heeft die hoog genoeg is om in de top 10 te komen. Zo niet, dan returned de class "MENU" en wordt de speler gelijk gestuurd naar Main Menu. Zo wel, dan kan de speler een naam invoeren en vervolgens op enter drukken om de score+naam op de leaderboard te zetten. De class

returned vervolgens "LEADERBOARD" waardoor de speler naar leaderboard wordt gestuurd.

## Enemy:

In de Enemy.hpp file staan verschillende soorten enemies. Deze inherited allemaal van de klasse Enemy.

## Enemy

Enemy is een superklasse en heeft dus een aantal virtuele functies die de children van deze klasse kunnen gebruiken. Deze klasse wordt gebruikt als de basis waar alle enemies van het spel op gebaseerd zijn. Deze klasse heeft meerdere functies waaronder move() en shoot().

Er zijn verschillende soorten enemy classes:

- enemy0
- enemy1
- enemy2
- enemy3
- enemy5
- enemy6
- enemy10
- enemy15
- enemy20
- enemy30
- boss1

Enemy0 is iets anders dan de andere enemies, deze enemy krijgt de speler nooit te zien en heeft verder geen input op de game. Deze enemy zorgt er alleen voor dat er altijd een enemy in het spel aanwezig is om segmentation faults en andere error's te voorkomen.

De andere enemies zijn zo goed als hetzelfde, het enige wat hun uniek maakt is hun shoot() en move() en een aantal variabelen. De move() en shoot() functies worden aangeroepen vanuit update() in GameControl.

## Upgrade:

De upgrade class is een superklasse met meerdere subklassen. Elke klasse staat voor een upgrade. Als de upgrade wordt opgepakt door de speler worden de waardes van de speler aangepast en de bijbehorende credits van de speler afgehaald.

De subklassen zijn:

- upg\_damage2
- upg\_tripleFire
- upg\_reload50
- upg\_damage1
- upg\_shield

- upg\_health1
- upg\_reload30
- upg\_shipspeed1
- upg\_reload20

De upgrades verschijnen elke 3 rondes. Er komt dan een “trade ship” het beeld in met 3 verschillende upgrades. Elke upgrade heeft een rarity. Zo heb je “Common” met een kans van 50%. Dit zijn de slechtste upgrades. Daarna komt “Uncommon” met een kans van 30%. Deze zijn net iets beter dan de common upgrades. Na uncommon komt “Rare” met een kans van 15%. Bij deze rarity zit ook de shield upgrade, dit is een speciale upgrade die 2 schilden voor het schip zet om een kogel op te vangen, een schild verdwijnt zodra het geraakt wordt door een kogel. Als laatste heeft de crate een kans om een “Legendary” upgrade te zijn met een 5% kans. Deze bevat de “Triple Fire” upgrade. Door deze upgrade schiet de speler met 3 kogels tegelijk.

Elke upgrade heeft verschillende kosten. Zo kost Triple Fire 250 credits en Ship Speed 1 maar 15. Bij het doden van enemies krijgt de speler tussen de 1 en 3 credits.

### Action:

De action klasse zorgt voor het afhandelen van gebruikers input. Zo gebruikt GameController een actionlist die functies gebruikt voor het klikken van de pijltjestoetsen, spatiebalk en escape key. Bij de pijltjestoetsen wordt het schip bewogen. Bij spatie wordt er een kogel afgevuurd en bij de escape key wordt het spel op pauze gezet.

### PlayerStats:

In de klasse PlayerStats staan, zoals de naam al zegt, alle stats van de speler. De speler heeft de volgende stats:

- reloadTime
- shipSpeed
- bulletCount
- waveCount
- enemyKilled
- playerScore
- bulletAlignment
- size
- health
- bulletDamage
- credits
- tripleFire
- spawnShields

Elke stat heeft een standaard waarde per schip. Zo heeft het kleine schip een snellere reload tijd dan het standaard schip. Ook heeft het kleine schip een kleinere hit box en is iets sneller met minder levenspunten. Daarnaast doet het kleine schip minder damage om te compenseren voor de snelheid. Het grote schip heeft meer levenspunten, is langzamer maar doet meer damage dan het standaard schip. Deze stats worden ook links en rechts van het beeld getoond.

## Rectangle:

Rectangle klasse maakt gebruik van een sfml rectangle object. De rectangle klasse heeft een position, color, size en shape. Deze klasse wordt gebruikt voor de muren in het spel. Verder wordt het gebruikt bij het leaderboard, bij het character selection screen en death screen. De rechthoeken worden als achtergrond gebruikt bij deze states. Verder heeft de rectangle klasse een aantal bruikbare functies. Zo kan de shape worden opgevraagd met `getShape()`, de rectangle gemoved worden door `move()` en gechecked worden of de muis zich op dezelfde plek bevindt als het object met `clicked()`.

## Picture:

De Picture klasse maakt gebruik van een sfml picture object. Picture wordt gebruikt om plaatjes in te laden en deze op een plek op het scherm neerzet. Deze klasse wordt gebruikt voor de backgrounds, schepen, upgrades, knoppen en visuals zoals de speler en enemy bullets. De picture klasse heeft verschillende functies. Zo kan er gechecked worden of een picture iets raakt met de `collision()` functie. Ook kan er gechecked worden of een picture zich op dezelfde plek als de muis bevindt met de `clicked()` functie. Verder kunnen verschillende eigenschappen van de picture worden opgevraagd zoals: `getType()`, `getPosition()` en `getShape()`. Ook kunnen plaatjes worden geroteerd, jumped, moved en scaled. Dit biedt veel bruikbaarheid in verschillende andere klassen.

## DisplayTexts:

De displayText klasse zorgt ervoor dat een aantal playerstats op het scherm getoont worden. Links staan de gegevens Reload Time, Ship Speed, Bullet Damage en Bullet Count. Rechts staan de gegevens Wave, Enemies Killed, Score, Player Health en Credits. Hierdoor kan de speler makkelijk zijn score, credits en andere gegevens zien. Alle stats worden ook geüpdatet als de speler bijvoorbeeld schiet, een enemy killed of een upgrade pakt.



# Diagrammen

