CM3070 - Final project

By David Della Rossa

Preliminary Report

Overview

Template

Arcade Game - Game development

I'm going to create a game called **Fightship Arena**.
This is a fast-action manic shooter game, where the player fights in a closed arena against hordes of hostile spaceships, in a time match, managing the weapon arsenal in real-time and building complex tactics against enemies.

The intention is to create a game in the Bullet Hell, fast-action arena-based genre, with some elements of tactics in how to manage weapons and power-ups during the gameplay.

Its main features can be summarised as:

- Fast-action arena-based game
- Fight against hordes of various types of hostile fightships in a time match
- Develop your tactics, managing your weapon arsenal in real-time
- Expand your fightship's arsenal with a wide set of power-ups

In my view of the game, I see three different modalities:

Harvester Mode

Make the highest possible score with one life.
Game goes on until the player dies.
The player is rewarded with time-bonus-score-multipliers at predefined time-checkpoints.

Punisher Mode

Defeat all enemies. The number of enemies is pre-set at the beginning of the level.
They spawn in waves, triggered by the game status.
The number and composition of enemy waves is predefined for each level.
During gameplay, the player can see the number of enemies left to defeat.

Highlander Mode

Stay alive until the time is over. Here, the constant is the time.
The number of enemy waves spawning is variable and depends on the game status.
If score/time is beyond a pre-set threshold, more time can be given. This is considered as normal match time, therefore, if the player dies during this phase, the match is lost.

If score/time is beyond another set threshold, extra time can be given to reach an even higher score. This is considered as extra time, and starts after the normal time is over and the player has won the match. If the player dies during this phase, the match ends, but the player is still the winner.

What is in scope for the project?

Considering the limited time, I will commit to implement only the Harvester Mode.
As a “stretch goal”, if I have time, I will work on the implementation of one of the other two modes.

Who is this game for?

After analysing a number of games in the genre of spaceship shooters and manic shooters, I see that the tactical/strategic aspects of the spaceship shooter games are always left out of the mix.
All examples in these types of games re-interpret the battle in many different ways, varying the speed of action, the number and type of enemies, the graphics, but none of them throws tactics in the mix.
The only one game I’ve found that barely did that, was Gradius (Konami, 1985), but no valid successors after that, wanting to continue exploring the tactical aspects of the fight.
In other types of games, like First person shooters there are several examples of games where the tactical aspects are key to win the battle, like using the right weapon for the right moment, or managing resources efficiently, or controlling key areas of the battleground.
This made me think that a game with these two characteristics, spaceship shooter and tactical aspects, would be a winning combination for a video game.

Target audience

The "Competitor"

Competitor gamers are defined as those gamers that play to best other gamers. The time-limited fights in a close arena, the high number of enemies to defeat, the score system based on score-multiplicators that increase the more the player stays alive, make the experience rewarding and satisfying for this kind of gamers. A high-score table will list the best players' names and their final score, incentivising players to do more and better.

The "Achiever"

Achiever gamers are defined as those gamers that play to reach the highest levels of a tournament or a ladder and are incentivised by the progressive achievements earned during the game. The power-up system will reward their successes with more weapons and add-ons, giving a percentage of completion (in terms of add-ons collected) at the end of each battle, and giving players the chance of replaying the same level in order to achieve better scores.

The "Matusalem"

Matusalem was the longest-lived person in the Bible. He lived for 969 years. Here, by Matusalem gamer, I refer to those older gamers (like me) lucky enough to have the chance

to play the ancestors of this game directly on those big wooden cabinets of the 80's. Hopefully they will feel the same thrill, or just a little bit of it, or just that nice feeling of "Next time I'll do it better!", that made them put another coin in, back in the days.

Literature review

Space invaders

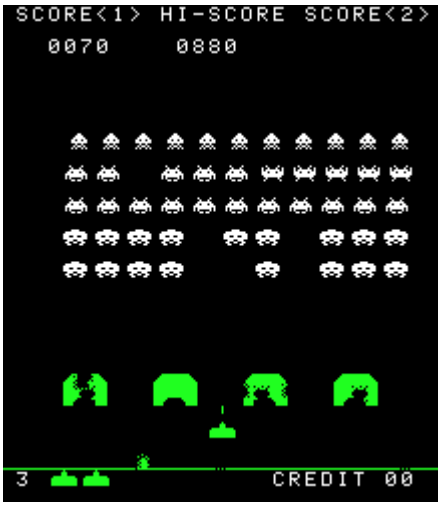

Image 1: Space Invaders

Space invaders (Taito, 1978) was the progenitor of an entire arcade sub-genre, called Spaceship shooters, comprising a vast variety of games which, one after one, evolved the initial idea into more articulated mechanics, with a wide variety of enemies and more and more gorgeous graphics.

Space invaders' clones: Galaxian, Moon Cresta, Galaga

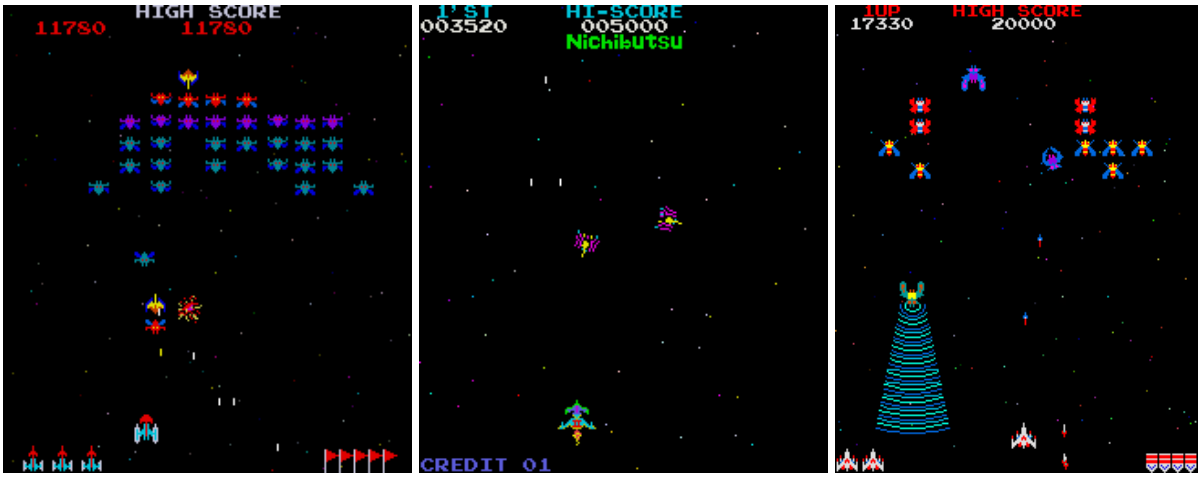

Figure 2: Galaxian, Moon Cresta and Galaga

The first wave of titles after Space Invaders, like Galaxian (Namco, 1979), Moon Cresta (Nichibutsu, 1980), Galaga (Namco 1981) and many more, offered a more various set of enemies, capable of moving in more complex paths, coloured graphics and new elements of game mechanics, like tractor beams, multi-part spaceships, protective shields, etc...

Horizontal and Vertical Scrolling: Scramble and Vanguard

Figure 3: Scramble, Vanguard in horizontal and vertical scrolling

From there, game designers started exploring new ways, new designs, new game mechanics, producing games with horizontal scrolling, like Scramble (Konami, 1981), or Vanguard (Tose, 1981) which starts off as a horizontal scrolling game and turns into a Vertical scrolling game as the level proceeds. Here the player can move in any direction and, like in Vanguard, also shoot in any direction Here appear power-ups as invulnerability or energy, that the player can collect.

Towards new dimensions: Zaxxon and Gyruss

Figure 4: Zaxxon and Gyruss

Designers started also exploring new ways of rendering the game space, like in Zaxxon (Sega, 1982), the first game using an isometric perspective to emulate a 3D environment, Or Gyruss (Konami, 1983) where the spaceship moved in a circle around the centre of the screen like being rolling on the inner surface of a hollow cylinder.

1942 (Capcom, 1984)

Figure 5: 1942

In 1984, Capcom released a new game, 1942, a vertical scrolling game, which shows new distinctive characters: higher density of enemies, and a higher number of bullets on the screen. Players can collect various power-ups which improve the aircraft and its weapons.

More tactics: Gradius (Konami, 1985)

Figure 6: Gradius

Konami released Gradius in 1985, Here the concept of pick-ups is more integrated in the game mechanics, as the user can choose when to use them to their advantage. This opened for new tactics in the gameplay.

The advent of 3D games

But now, an important thing, in the evolution of these games, happened at the beginning of the 90's. The advent of 3D games.

In order to keep their games competitive with the new born FPS genre, the new Spaceship shooter games steered towards emphasizing even more the player engagement, All is moving towards extreme: more enemies, more bullets, higher game speed, higher ability required by the player to proceed in the game. This triggered a shift in the audience, though. Many casual players found this type of game too difficult, too overwhelming, too frenetic. On the other side, this attracted more dedicated and competitive players, thrilled by the more adrenalinic experience these games had to offer.

A new genre is born: Bullet Hell, a.k.a. Manic Shooters

Figure7: Batsugun, DonPachi, Ikaruga, Radiant Silvergun

The first of this new breed, thereby named Bullet Hell, or Manic Shooters, is Batsugun (Toaplan 1993), but many more followed, like DonPachi (Cave, 1995), Radiant Silvergun (Treasure, 1998), Ikaruga (Treasure, 2001), Mushihimesama (Cave, 2005) and many more.

On Steam

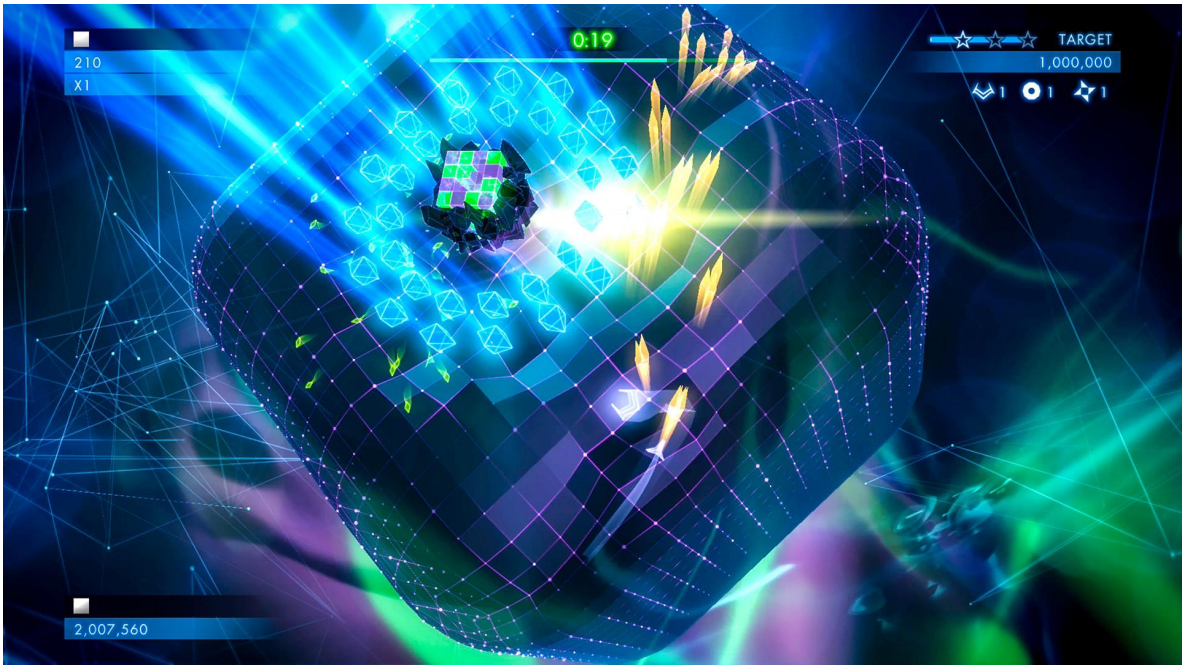

Figure 8: Geometry Wars 3

A few titles in the Bullet Hell genre are available on Steam, like

Geometry Wars (Lucid Games, 2003-2014) and its successors
Crimzon Clover (Yotsubane, 2014)
Gimel Dimension (JOZGamer, 2018),
Netherspace (Konstantinos Katsoris, 2021),
Techbeat Heart (Lost Machine Games, 2021),
Polybius Invaders (Jonni the Dodger, Peter Ward, 2021)

Figure 9: Polybius Invaders, Crimzon Clover

Design

Rules

Single player versus game

Player Controls

The idea is to be able to move the spaceship and quickly change its forward direction at will, so as to have a wider control over the entire arena.
The player controls are the canonical WASD for moving the player's spaceship around.
Three additional buttons, JKL will be used with the following functionalities:

- Button 1 (J) - Fire weapon
- Button 2 (K) - Fire alternative action/weapon
- Button 3 (L) - Enable selection menu

When Button 3 is kept down, the selection menu opens, and time freezes.
Button 3 pressed remaps other buttons like follows:

- W - Change spaceship facing direction to upward
- A - Change spaceship facing direction to left
- S - Change spaceship facing direction to downward
- D - Change spaceship facing direction to right
- Button 1 - Select active weapon for Button 1, among those owned by the player
- Button 2 - Select active action/weapon for Button 2, among those owned.

When Button 3 is released, or after a timeout expires, the selections are applied and time gets flowing again.

NPCs:

All characters will be differentiated by:

- How tough they are to defeat;
- How fast they move;
- Which weapons they are equipped with;
- Which power-ups they leave behind when destroyed;

Enemy: Pawn

Pawns spawn in high numbers and head to the player's fightship, with the intent of colliding with it and destroying it.

They can be dangerous as they move in high numbers towards the player from different directions with the intent of surrounding it.

Enemy: Infantry

Infantry spawn in moderate numbers and stand where they are, floating around and shooting at the player's.

Not too dangerous per-se. They can become insidious when the player is dealing with other enemies, as they keep shooting.

Enemy: Shielder

Shielders spawn in small groups. They are equipped with a protective shield and are able to proceed like a roman testudo formation, protecting other enemies behind them, while moving towards the player.
If the formation is damaged, they rearrange closing ranks.
If alone, they proceed towards the player with kamikaze intent. Other enemies can follow along, protected by their shield.

Enemy: Turret

Equipped with a short range plasma-beam weapon which can damage the player and slow it down in its movements, if it comes too close.
They float around in the arena. If two turrets come in short range, they can form an energy wall where the player cannot pass through and which stops the player's bullets. Destroying one turret would destroy the energy wall.

Weapons

Player starts with only one/two weapons, plus a limited quantity of Serious bomb[1]s and Protective shield, but can acquire more during the gameplay.
Ammunition and energy are left in the arena by some enemies on destruction, but in each arena there is an area where the player can go and have certain weapons and energy recharged.

Multi-cannon

Fires a continuous stream of energy bullets. The shape of the stream and the number of bullets fired in the unit of time changes based on the power-ups collected. Player starts with a double multi-cannon (two streams of bullets).
Ammunition is provided collecting Energy charger power-ups.

Fire-flow

Fires a continuous fire beam. If the fire-flow hits certain objects or certain types of enemies, it keeps flowing all around those objects, hitting also enemies behind them.
Ammunition is provided collecting Fire-flow Ammunition power-ups.

Rocket-launcher

Pressing the fire button, launches one single rocket; releasing the fire button the rocket explodes, damaging all enemies around. This is useful to destroy enemies behind a protective shield.
Of course, rockets also explode when they impact any object.
Ammunition is provided collecting Rocket launcher Ammunition power-ups.

Plasma-wiper

Pressing the fire button, this weapon fires at the closest enemy to the player and from there it spreads to other enemies around. After a short time, all the enemies touched by the

---

[1] Serious bomb is borrowed from Serious Sam game franchise - Croteam 2001-2020

plasma-beam explode in a blast that can damage other enemies around even though they were not touched by the plasma-wiper's beam. During the plasma-beam action, all enemies touched by it are slowed down.
Ammunition is provided collecting Plasma-Beam charger power-ups.

Serious bomb

This weapon is capable of completely clearing up the screen from all enemies.
Occasionally extra Serious bomb ammunition power-ups are spawn in certain levels.

Protective shield

Enables a protective shield all around the player's fightship that stops enemy weapons and pushes enemies away on contact.
There is a maximum amount of seconds per level allowed for the Protective shield. This varies from level to level.

Power-ups

Score multiplier

The initial score multiplier for the player's score is 1.
Each score multiplier powerup increases the total score multiplier, applied to the score gained from each enemy destroyed, by a certain factor.

Energy charger

Increases the energy level by a certain amount.

Fire-flow charger

Increases the amount of Fire-flow ammunition by a certain amount.

Plasma-Beam charger

Increases the amount of Plasma-Beam ammunition by a certain amount.

Rocket Ammunition

Increases the amount of Rocket ammunition by a certain amount.

Serious bomb

Add one Serious bomb to the player’s arsenal

Protective shield charger

Increases the Protective shield by a certain amount.

Tactical use of weapons against enemies

Weapon	Ammo power-up	More Effective on	Less effective on
Multi-cannon	Energy charger	Pawns	Shielders
Fire-flow	Fire-flow charger	Infantry	Pawns
Plasma-wiper	Plasma charger	Shielders	Turrets
Rocket launcher	Rocket	Turrets	Infantry
Serious bomb	Serious bomb ammo	All	None
Protective shield	Protective shield charger	Pawns	Shielders

Technologies in use

Game Engine/Editor: Unity3D

Targeted platforms are:

- Windows 10
- WebGL

Test plan

In the second part of the course, after the Mid-Term deadline, I plan to create a build of the game and ask for a few fellow students, or colleagues to voluntarily offer to test the game.
I plan to produce three versions of the game during the second part of the project.
The testers will do their evaluation purely on a voluntary basis. No form of remuneration will be given.
When they are ready to provide their feedback, a brief questionnaire will be provided, where they can share their thoughts and ideas on improvements, or notify bugs or any type of malfunctioning.
This feedback will then be evaluated and the proper actions will be taken to put tester’s suggestions in the game’s next version.

Feature Prototype

Implemented features:

- Scene management: The game infrastructure manages menus and game levels, using player input to pass from a scene to another.
- Arena: rectangular arena created, where the battles take place
- Score system: takes account of the player's score. Score increases the more enemies the player kills.
- Weapons:
    - Multi-cannon: the weapon can shoot and consume ammo
- Power-ups:
    - Energy charger: released by killed enemies, they boost the player's health level
    - Score multiplier: released by killed enemies, they increase the score multiplier
- Player fightship
    - Can move around using the WASD keys
    - Can shoot enemies
    - Can collect power-ups left over by destroyed enemies
- Enemies:
    - Pawn enemy
        - Pawn enemy instances spawn in pre-defined location in the arena
        - Pawn enemies chase the player
        - When killed, they release randomly selected power-ups
- Unit tests: used a pattern of my creation to separate the game logic from MonoBehaviour classes into a separate Core class. Doing so increases testability of the code, by making it possible to use Moq library.

Evaluation of the prototype

I am very satisfied with the number of features I managed to put into this prototype.
The game so far implemented runs smoothly. The architecture has been designed to be solid, extensible and testable. Some unit-tests have been written, although they still cover a small portion of the code. But the main focus up to this point was to have a running prototype. The unit test code coverage will be improved later.

Architecture

GameManager

The game program starts and instantiates a Main scene containing a GameManager object. This contains all the logic required to stay in memory for the entire duration of the game. This class intercept the player's request to open up menus, pause and resume the game, quit, and other similar actions.
The GameManager class contains a State machine that manages the states Init, Play, Pause and Quit.
When the player selects a new game, this Main scene, staying resident in memory, loads the scene containing the level to start.

LevelManager

This Level scene contains a class LevelManager that instantiates all the necessary components to run that level of the game, such as the PlayerController, the EnemyManager, the ScoreManager and their mutual interaction.
In future improvements, this class too will be provided with a State Machine.

ScoreManager

This class is owned by the Level Manager and takes track of the player's score and the high scores.
The High-Score leaderboard is permanently saved using a ScriptableObject, which persists its status among games.

Health Manager

This class is responsible for taking track of the health status of player and enemies.
Each of these characters contains an instance of this class and is initialized with a health level. During the progression of the game, this level can vary. I.E.: the player can increase it by collecting Energy-charger power-ups. The level can decrease due to collision with enemies or enemy fire.

EnemyManager and enemies

The EnemyManager is responsible for managing the lifecycle of enemies.
It currently spawns enemies in pre-defined spawn-locations at predefined time intervals.
In future improvements, it will manage more complex sequences of enemy spawns during the entire level.
This also detects when an enemy is killed and notifies the ScoreManager (via the LevelManager) so as to increase the Player's score.
All enemies share the same base structure but can be customised at need.
Each enemy is provided with a set of power-ups that can be released when the enemy is killed, based on a probability assigned to each power-up. A random selection, which takes into account this probability value, selects the power-up to spawn.

PlayerController

The playerController receives the input from the Player and takes the proper actions, like move, fire, etc.
This class also detects events like when the player dies.

Weapons and ammo

All weapons rely on a base class structure and behaviour. Each weapon though can customise its behaviour. The same is true for the type of ammo used by the weapon.

Power-ups

Power-ups also share a common base structure, but each can have custom behaviour based on the specific needs.

Initial configuration

All entities that require a specific data configuration, like the player, the enemies, the power-ups, the weapons, etc. are provided with a specific configuration data, implemented as a ScriptableObject.
For example, the Player settings are stored in a ScriptableObject called PlayerSettings, containing values for Initial health value, max speed, etc.
This is a versatile way to efficiently store this data in memory and also to create different configurations for different situations like levels.
For example, I want the player to have max speed = 3 in level 1, but max speed = 5 in level 3. I can create different Settings and assign the wanted settings in each level.

Testing

Who has experience in Unity knows how difficult it can be to implement a proper set of Unit-Tests that cover all important parts of the code.
Although Unity provides a Unit Test framework, that helps testing in static as well as in dynamic situations, the inaccessible constructors in MonoBehaviour-inherited classes is a huge limiting factor, especially for using Mocking techniques.
In my project, I'm working on a workaround that separates the Unity-strictly-related logic and interaction with other Unity entities, like GameObjects and components, from the custom-implemented logic that makes up the game.
In my architecture, each MonoBehaviour-inheriting class contains a corresponding Core class, where the actual game logic is implemented.
The Mono-Behaviour inherited class interacts with the Unity world, acting as a proxy towards the Core class, which is interface based and has a constructor.
This makes the core logic of the game easily testable with the normal tools of a test-coder, like NUnit and Moq, and the test-related patterns applicable too.
For the logic intrinsically Unity-related, the Unity-Test Framework's PlayMode or EditMode are used for testing.
This approach will probably complicate the software architecture a bit and will probably introduce some small delays in the execution (imperceptible, based on my experience), but it will also greatly increase separation of concerns, reduce dependencies, improve testability with all the advantages related, such as more robust code, less prone to regressions.

Future Improvements:

- Player can change its forward direction.
- Player can select weapons from a list.
- Implementation of other types of enemies, Infantry, Shielder, Turret (based on time availability)
- Implementation of other types of powerups: Fire-flow charger, Plasma-beam charger, Protective-shield charger, Rocket ammunition, Serious bomb (based on time availability)
- Other types of weapons: Fire-flow, Rocket-launcher, Plasma-wiper, Serious-bomb, Protective shield (based on time availability)
- Adding sound effects
- Adding animations
- Extending code coverage with unit tests.

Further information

For further information about the game design, please see the github wiki.
For further information about the project management, please see the github project.
For further information about the source code, please see the github repository.
For further information about the game architecture, please see the UML diagram (it requires StarUML).
For the Gantt diagram, please see here.

Note: all these links point to resources which are all Work-In-Progress. The Information therein contained can therefore change at any moment without any type of notice, until the project deadline.

Images

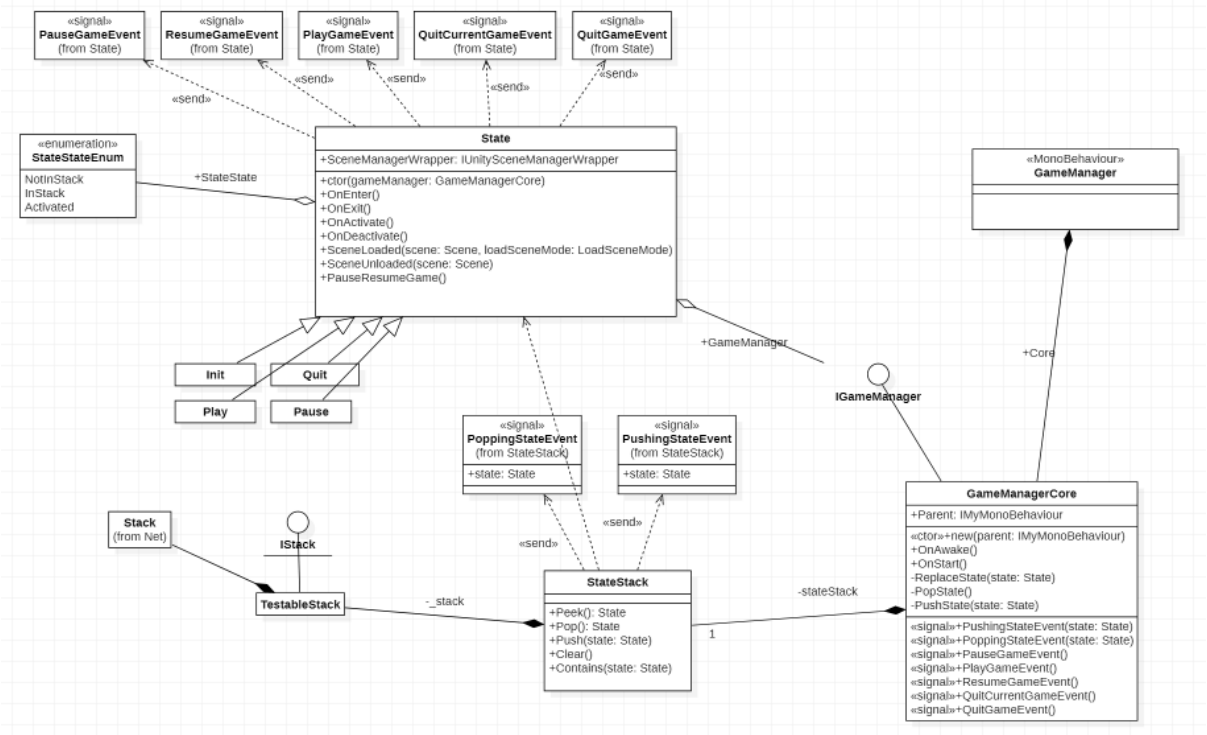

Image 1: Game Manager architecture

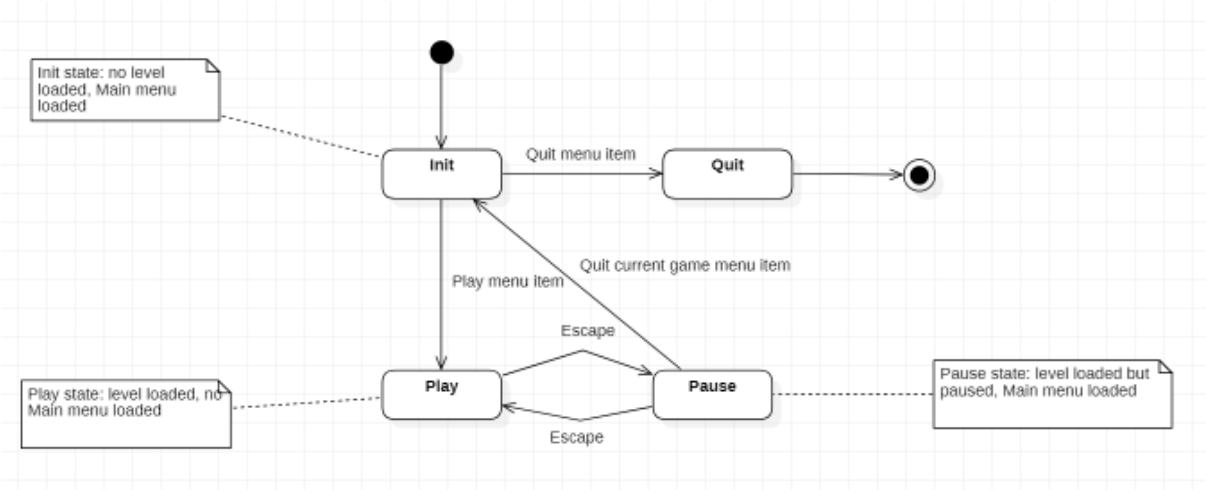

Image 2: Game Manager State Machine

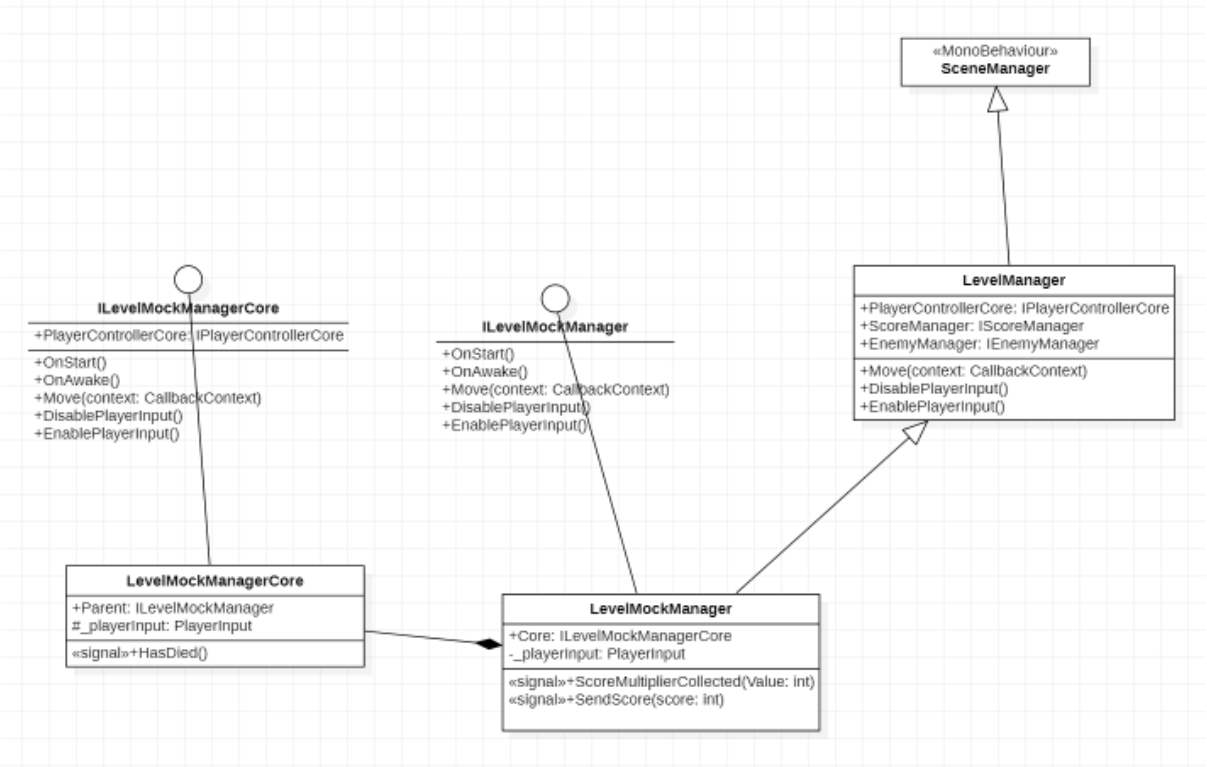

Image 3: Level Manager Architecture

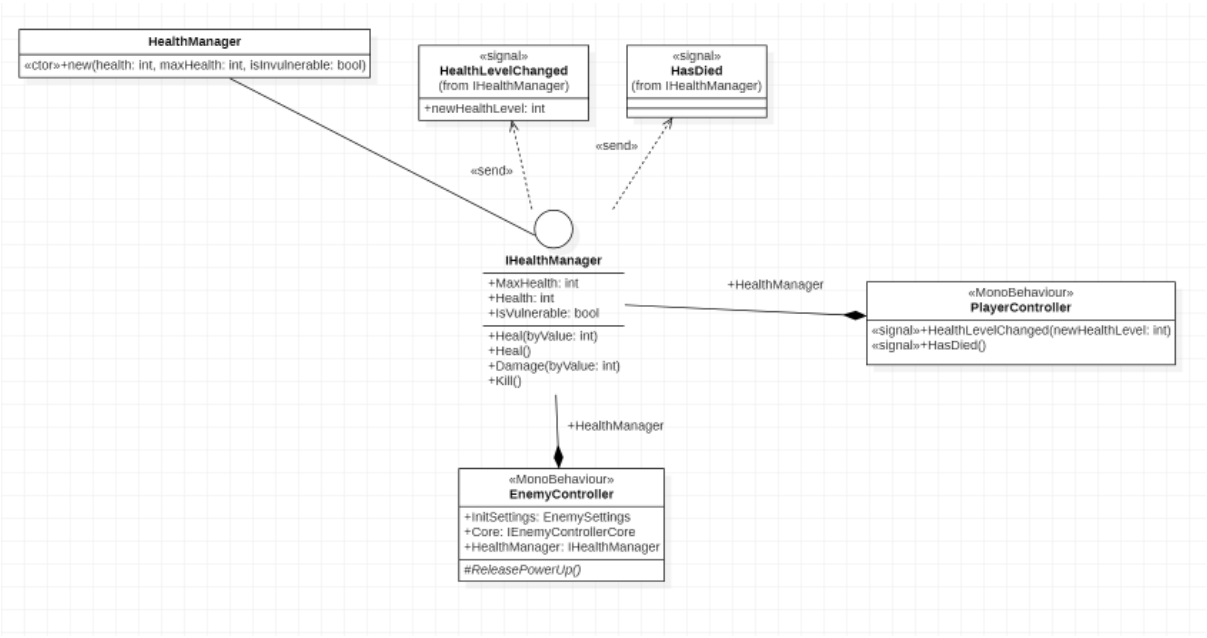

Image 4: Health Manager Architecture

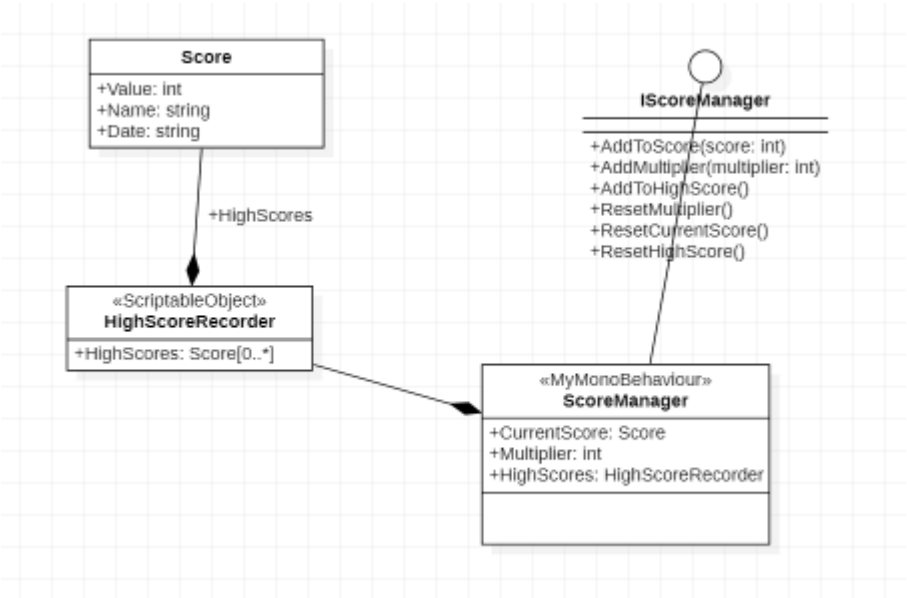

Image 5: Score Manager Architecture

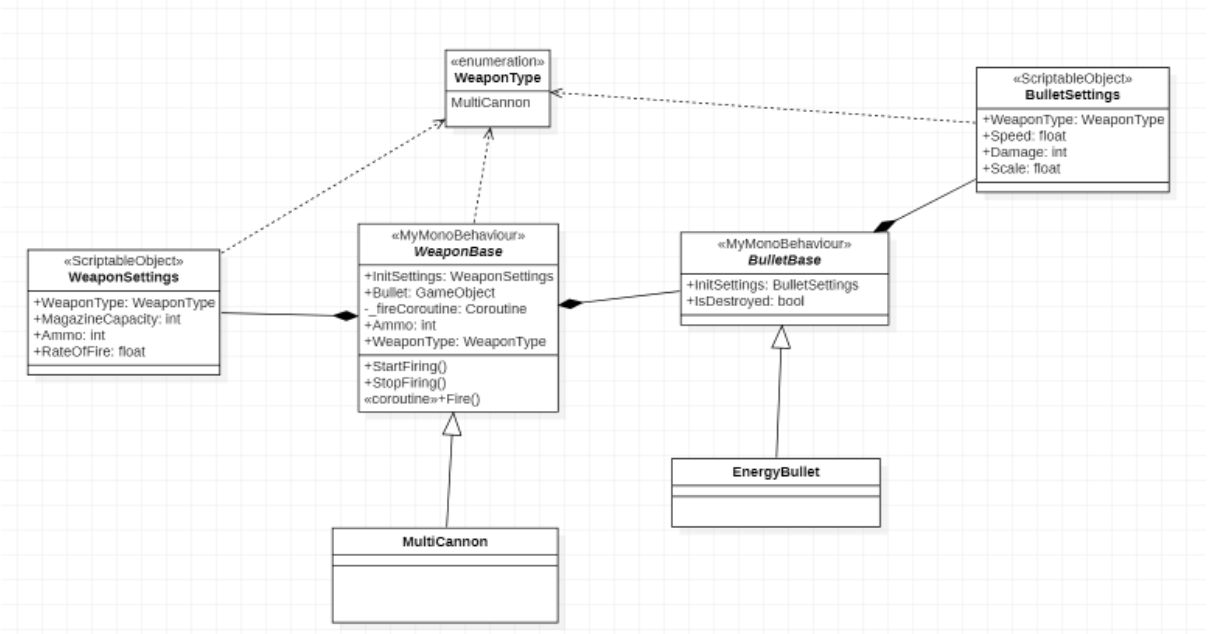

Image 6: Weapons Architecture

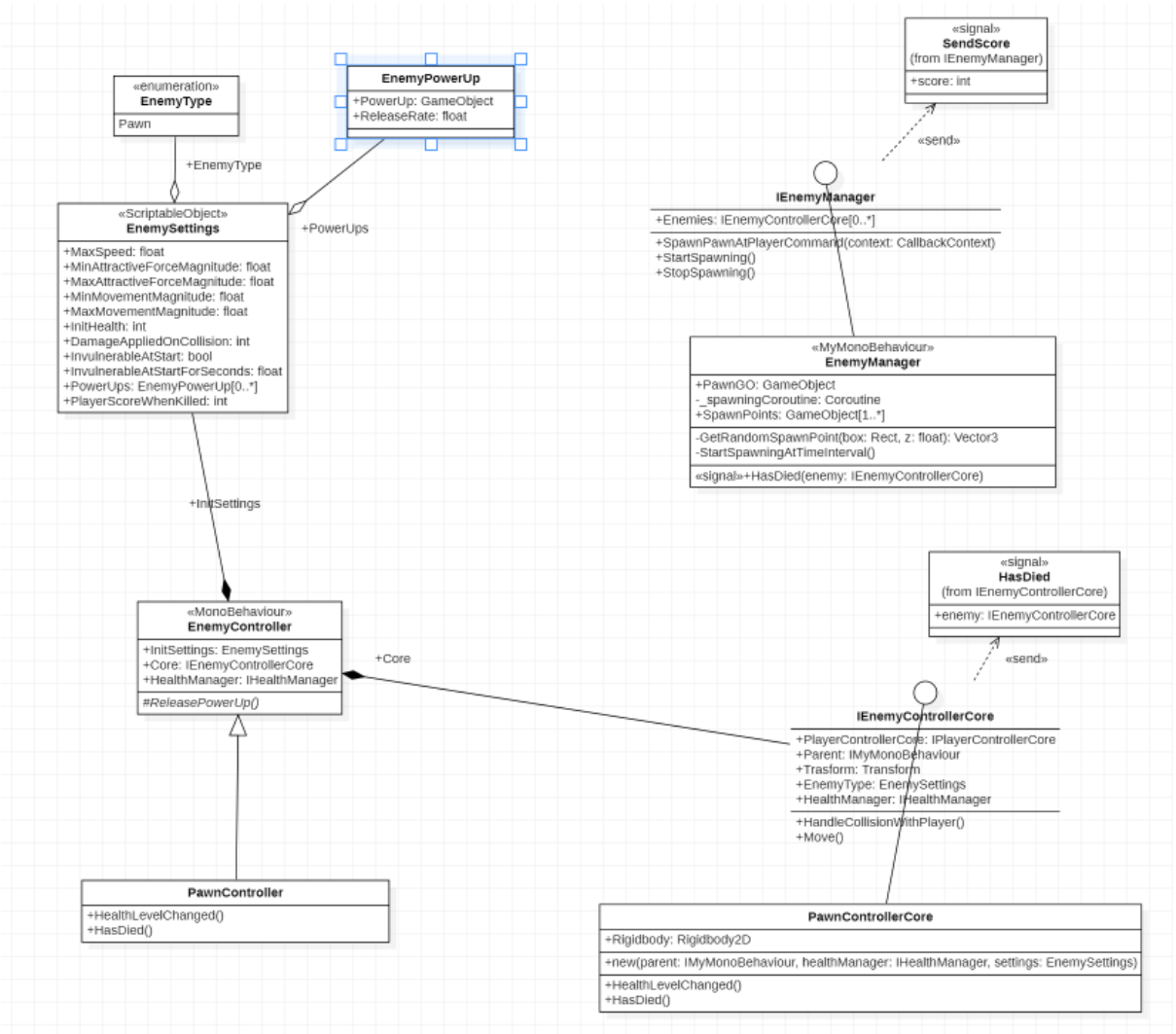

Image 7: Enemies Architecture

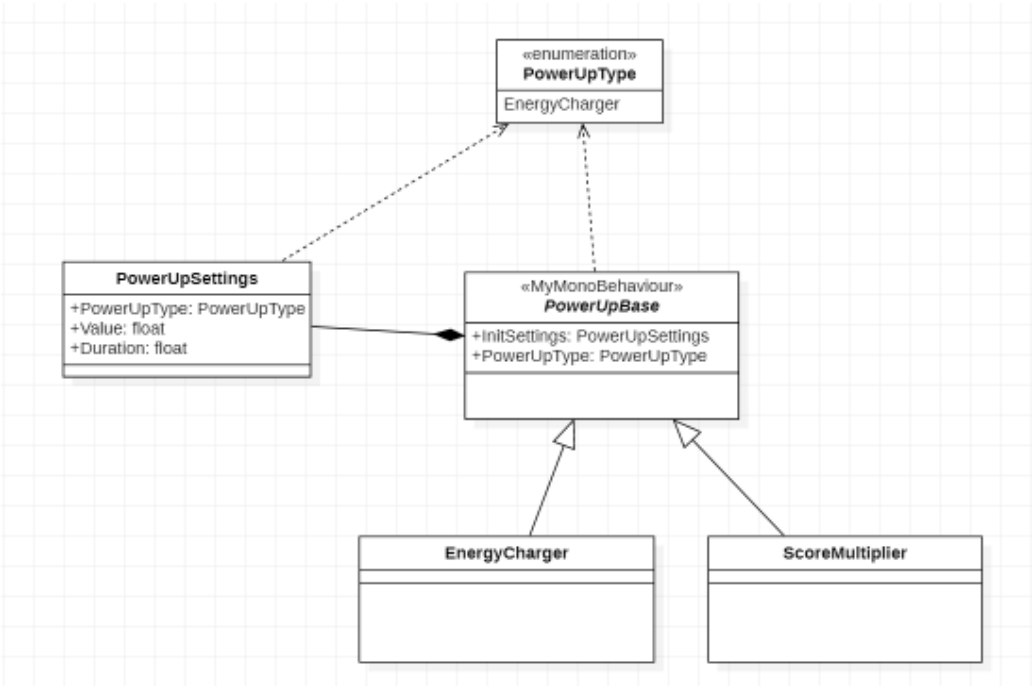

Image 8: Power-ups architecture

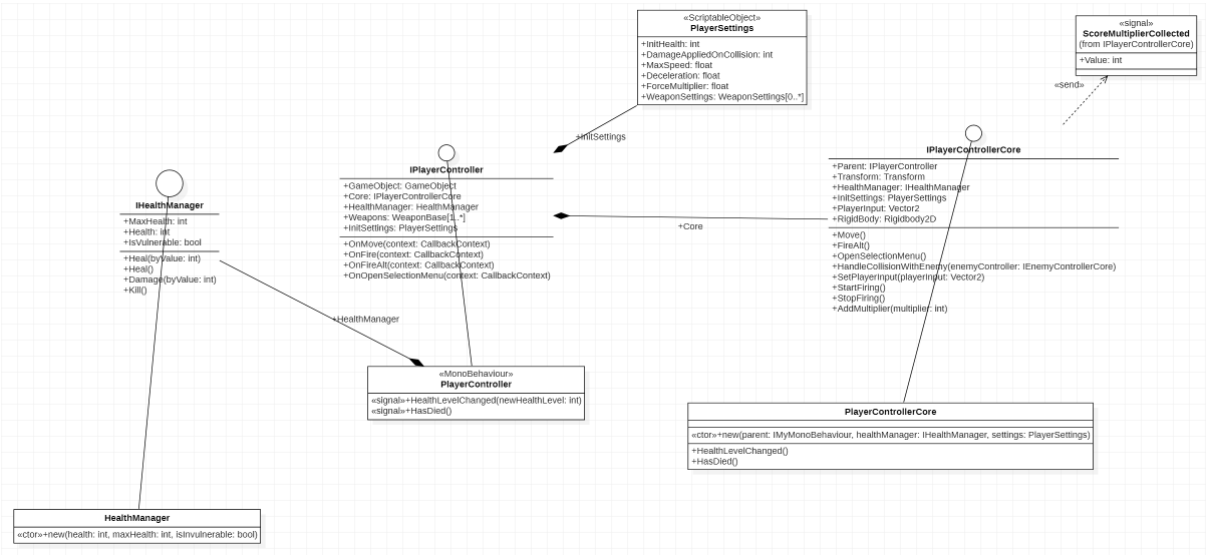

Image 9: Player architecture

Image 10: Power-ups. From left to right: Energy charger, Fire Flow charger, Plasma Beam charger, Protective Shield charger, Rocket ammo, Score multiplier, Serious bomb.
All these images are made by myself with InkScape.

Image 11: Pawn enemy. Made by myself in InkScape.

Image 12: Player fightship (adapted from https://www.cleanpng.com/png-karpman-drama-triangle-color-triangle-equilateral-601066/)

References:

Space Invaders [Online] International Arcade Museum.
Available from: https://www.arcade-museum.com/game_detail.php?game_id=9662
[24/12/2021]

Galaxian [Online] International Arcade Museum.
Available from: https://www.arcade-museum.com/game_detail.php?game_id=7885
[24/12/2021]

Moon Cresta [Online] International Arcade Museum.
Available from: https://www.arcade-museum.com/game_detail.php?game_id=8745
[24/12/2021]

Galaga [Online] International Arcade Museum.
Available from: https://www.arcade-museum.com/game_detail.php?game_id=7881
[24/12/2021]

Scramble [Online] International Arcade Museum.
Available from: https://www.arcade-museum.com/game_detail.php?game_id=9447
[24/12/2021]

Vanguard [Online] International Arcade Museum.
Available from: https://www.arcade-museum.com/game_detail.php?game_id=10293
[24/12/2021]

Zaxxon [Online] International Arcade Museum.
Available from: https://www.arcade-museum.com/game_detail.php?game_id=12757
[24/12/2021]

Gyruss [Online] International Arcade Museum.
Available from: https://www.arcade-museum.com/game_detail.php?game_id=8060
[24/12/2021]

1942 [Online] International Arcade Museum.
Available from: https://www.arcade-museum.com/game_detail.php?game_id=6766
[24/12/2021]

Gradius [Online] International Arcade Museum.
Available from: https://www.arcade-museum.com/game_detail.php?game_id=7987
[24/12/2021]

Batsugun [Online] International Arcade Museum.
Available from: https://www.arcade-museum.com/game_detail.php?game_id=7030
[24/12/2021]

DonPachi [Online] International Arcade Museum.
Available from: https://www.arcade-museum.com/game_detail.php?game_id=13425
[24/12/2021]

Ikaruga [Online] International Arcade Museum.
Available from: https://www.arcade-museum.com/game_detail.php?game_id=10841
[24/12/2021]

Radiant Silvergun [Online] International Arcade Museum.
Available from: https://www.arcade-museum.com/game_detail.php?game_id=9239
[24/12/2021]

Geometry wars [Online] Steam
Available from:
https://store.steampowered.com/app/310790/Geometry_Wars_3_Dimensions_Evolved/
[24/12/2021]

Polybius Invaders [Online] Steam
Available from: https://store.steampowered.com/app/1697050/Polybius_Invaders/
[24/12/2021]

Crimzon Clover [Online] Steam
Available from:
https://store.steampowered.com/app/285440/Crimzon_Clover_WORLD_IGNITION/
[24/12/2021]

Arcade Museum
Available from: https://www.arcade-museum.com/

Arcade video game [Online] Wikipedia
Available from: https://en.wikipedia.org/wiki/Arcade_video_game [24/12/2021]

Golden age of arcade video games [Online] Wikipedia
Available from: https://en.wikipedia.org/wiki/Golden_age_of_arcade_video_games
[24/12/2021]

Shoot 'em up [Online] Wikipedia
Available from: https://en.wikipedia.org/wiki/Shoot_%27em_up [24/12/2021]

History of video games [Online] Wikipedia
Available from: https://en.wikipedia.org/wiki/History_of_video_games [24/12/2021]

Space Invaders [Online] Wikipedia
Available from: https://en.wikipedia.org/wiki/Space_Invaders [24/12/2021]

Galaxian [Online] Wikipedia
Available from: https://en.wikipedia.org/wiki/Galaxian [24/12/2021]

Moon Cresta [Online] Wikipedia
Available from: https://en.wikipedia.org/wiki/Moon_Cresta [24/12/2021]

Galaga [Online] Wikipedia
Available from: https://en.wikipedia.org/wiki/Galaga [24/12/2021]

Phoenix (video game) [Online] Wikipedia
Available from: https://en.wikipedia.org/wiki/Phoenix_(video_game) [24/12/2021]

Scramble (video game) [Online] Wikipedia
Available from: https://en.wikipedia.org/wiki/Scramble_(video_game) [24/12/2021]

Vanguard (video game) [Online] Wikipedia
Available from: https://en.wikipedia.org/wiki/Vanguard_(video_game) [24/12/2021]

Zaxxon [Online] Wikipedia
Available from: https://en.wikipedia.org/wiki/Zaxxon [24/12/2021]

Gyruss [Online] Wikipedia
Available from: https://en.wikipedia.org/wiki/Gyruss [24/12/2021]

1942 (video game) [Online] Wikipedia
Available from: https://en.wikipedia.org/wiki/1942_(video_game) [24/12/2021]

Gradius (video game) [Online] Wikipedia
Available from: https://en.wikipedia.org/wiki/Gradius_(video_game) [24/12/2021]

Batsugun [Online] Wikipedia
Available from: https://en.wikipedia.org/wiki/Batsugun [24/12/2021]

DonPachi [Online] Wikipedia
Available from: https://en.wikipedia.org/wiki/DonPachi [24/12/2021]

Ikaruga [Online] Wikipedia
Available from: https://en.wikipedia.org/wiki/Ikaruga [24/12/2021]

Radiant Silvergun [Online] Wikipedia
Available from: https://en.wikipedia.org/wiki/Radiant_Silvergun [24/12/2021]

Mushihimesama [Online] Wikipedia
Available from: https://en.wikipedia.org/wiki/Mushihimesama [24/12/2021]

Geometry Wars [Online] Wikipedia
Available from: https://en.wikipedia.org/wiki/Geometry_Wars [24/12/2021]

Crimzon Clover [Online] Wikipedia
Available from: https://en.wikipedia.org/wiki/Crimzon_Clover [24/12/2021]

Gimel Dimension [Online] Steam
Available from: https://store.steampowered.com/app/916470/Gimel_Dimension/ [24/12/2021]

Crimzon Clover [Online] Steam
Available from:
https://store.steampowered.com/app/285440/Crimzon_Clover_WORLD_IGNITION/
[24/12/2021]

Netherspace [Online] Steam
Available from: https://store.steampowered.com/app/1749310/Netherspace/ [24/12/2021]

TechBeat Heart [Online] Steam
Available from: https://store.steampowered.com/app/1520330/TechBeat_Heart/ [24/12/2021]

Polybius Invaders [Online] Steam
Available from: https://store.steampowered.com/app/1697050/Polybius_Invaders/
[24/12/2021]

Software used:

Unity editor
Visual Studio 2019 Community Edition
StarUML
Gimp
InkScape
GitHub
GitHub Desktop

Images used

Player fightship image adapted from
https://www.cleanpng.com/png-karpman-drama-triangle-color-triangle-equilateral-601066/