

Edinburgh Napier University, School of Computing

Students:

Przemyslaw Szumlanski (40212392 )

Rafal Fryderyk Ozog ( 40217610 )

# Final Report

---

Module: Games Engineering ( SET09121 )

Module Leader: Kevin Chalmers

Project: 2D Platform Shooter Game – 'Shoot2Kill'

## 1. Introduction

As a coursework for Games Engineering module, we were assigned to create a game in Visual Studio, using SFML. Throughout this module we gained valuable knowledge of how to make a game in C++, what the advantages and limitations of SFML library are, and how to work in a team to accomplish common software development project. The result of this teamwork is a 2D shooter game, called Shoot2Kill.

Shoot2Kill is a platform shooter that follows a story of Simon, an army veteran who goes on a quest to rescue his kidnapped daughter. He must defeat his enemies and face the Boss before he can accomplish his mission. The goal of this game is to progress to next levels, and eventually, to rescue the protagonist's daughter. The player has to shoot incoming enemies and avoid their shots. Every bullet that hits the player will hurt him, and the game is lost when the Health Points drop to 0.

The game was inspired by classic platform shooters, like Contra or Duke Nukem. We chose this kind of game because of our fond memories, related to playing platform shooters.

## 2. Changes

The game differs in some ways from our original vision for the game, as stated in the game design document.

The story required a few changes to make the plot more complete. Our main protagonist has been working with the police to reduce the influence of the Mafia on his town. For this reason, the Mafia kidnaps the protagonist's daughter. It's a matter of honour for him, and he wants to deal with them alone. This is different to the story in the game design document, where the local police are corrupt, and the Mafia has much control over the police and the city.

The game has different features as well. We decided to leave out the First Aid Kits, which were supposed to replenish player's health. Instead, the protagonist is healed when he progresses to the next level. The player has to finish the level without dying. If the player dies, he/she can restart from the same level.

We changed the weapon set as well. Shotgun is now replaced by a bazooka, which has ammo capacity of just 1, and shoots a single big projectile. We decided to change this because we wanted to add an exciting weapon that would give the player a sense of power, even just against one opponent. The bazooka projectile does much greater damage; however it does not explode yet. We are planning to add explosions in future versions.

We have also limited the number of levels. The game features 6 levels instead of 8, as described in the game design document. We decided to cut down on the number of levels to reduce repetitiveness and present more of the game during demonstration. Additionally, we

decided against the possibility to go back to previous levels, as we didn't see a good reason to revisit them. Since all the enemies are killed, there is nothing for the player to experience in these previous levels. We will keep this option in mind however, and possibly implement it in the next versions.

Finally, we decided to add new elements to the game. The final boss fight now features moving turrets, which shoot big projectiles that the player has to avoid.

### 3. Software design

To create this game, we strictly followed the object oriented programming approach and the Entity Component Model, based on it. Thanks to this, we were able to make our own components that we could reuse at different times throughout the game. We also followed general structure and recommendations from the workbook, which helped us make the game better, and the code more understandable. Besides these approaches, the game also features implementation of AI to control the actions of enemies.

#### 3.1. Characters based on Entity Component Model

During the planning stage we have decided to create our characters ( and also majority of other entities actually ) following the Entity Component Model. This approach allowed us to create our own functionality pieces, which could be easily added to different characters and

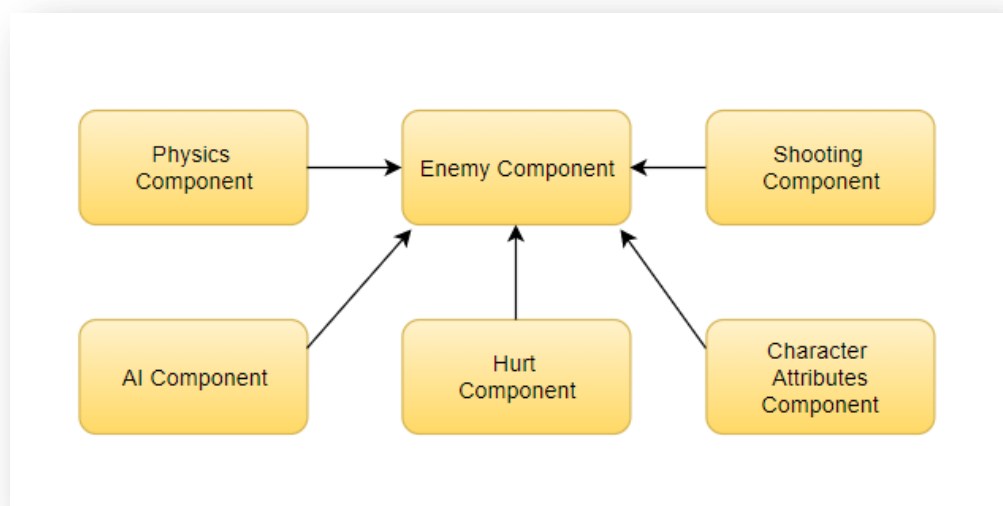


Figure 1 Enemy Component

managed in readable way. For example each enemy is an Entity object, containing one general 'Entity' component. This one in turn – is related to other ones, which add features like shooting, being hurt (when shot), physics, AI, character attributes (like HP, ammo, current gun).

### 3.2. Enemy AI

The game contains implemented Artificial Intelligence logic, which is responsible for enemy's behaviour. Actions undertaken by bot depend on situation, distance from a player and his current movement direction. During the ordinary area patrol, the intelligent enemy will be walking (changing direction sometimes), jumping, and generally waiting for our player with his gun. When the distance to player is close enough, bot starts the attack. To make the game more playable bots make short breaks during attacks (giving more power to player). Features requiring time countdown have been implemented using 'freezer variables'. These are simple numbers counting down with each 'update' method call. When the counter

reaches 0 – the proper action occurs and the variable is reset.

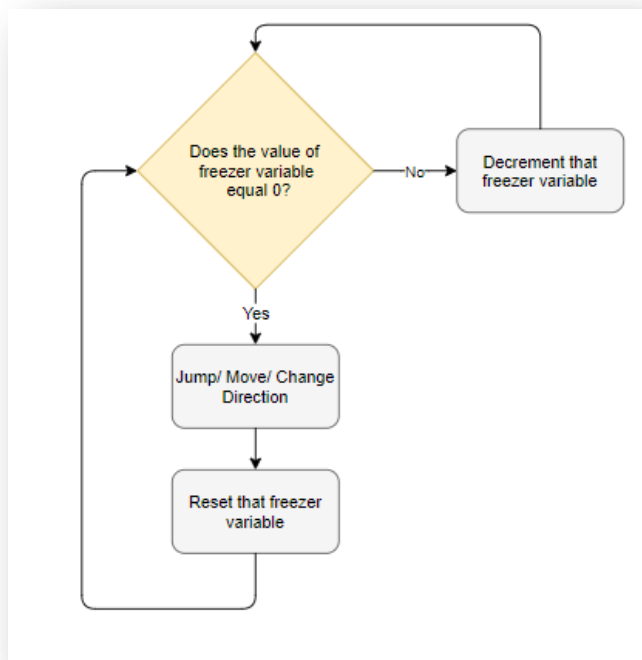


Figure 2 Freezer Variable Working Scheme

### 3.3. Optimisation

The last stage of our programming phase was related to code optimisation. We have proceeded our own investigation to figure out which elements are the heaviest load for the game. This approach led our project to following changes:

- trying to use weak pointers more often, rather than shared ones – especially in situations, when we only need pointer to get to know if the object still exists or not,
- using list and its proper number to access entities within the 'ents' ( Entity Manager ) – instead of much slower 'find' function ( using tags ),
- 

## 4. Game description

Shoot2Kill is a 2D Platform shooter. The objective of the game is to defeat the enemies and progress in levels. The player is introduced to the story via a short introduction. The protagonist, Simon, is on a rampage to rescue his kidnapped daughter Camilla. Camilla was kidnapped by the Mafia, as a revenge on Simon for helping the police in stopping the Mafia's

terrible influence over the city. The player needs to defeat mafia recruits and get to the Boss in order to rescue his only child.



Figure 3 Main Menu

During the game, the player must go through levels with different platforms and kill the enemies he/she encounters. The game features different weapons, like Colts, AR15 and Bazooka, each deadlier than the previous one. The enemies are not dummies however – they will do everything in their power to stop the player from getting to their boss.

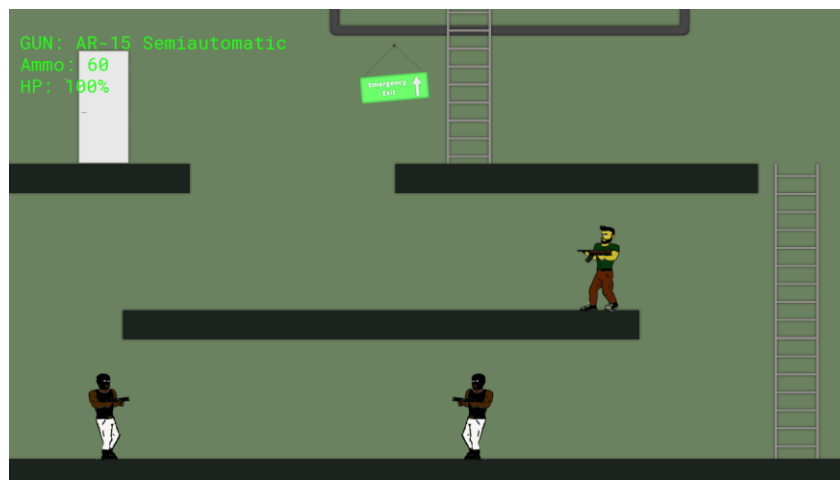


Figure 4 Screenshot 01

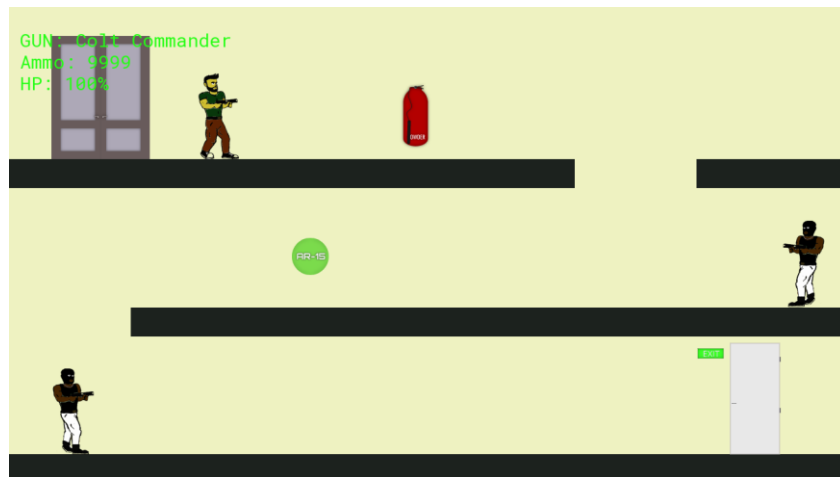


Figure 5 Screenshot 02

## 5. Evaluation

This coursework gave us an excellent opportunity to use our creativity in games development. As this is only the prototype, we are happy with the outcome. The assignment proved to be challenging, but this only fuelled our determination and made this whole process more fun. Based on the final product, as well as the work we put into this project, we consider the results to be more than satisfactory. However, in order to thoroughly evaluate our game, we need to compare the game to our original concept, as well as other games in this genre.

### 5.1. Comparison against the original concept

The game does not venture too far from our initial idea for the game. We set out to create a simple and entertaining game with nice graphics and fun story. Based on this, we consider this prototype to be a success. However, the original concept differs from the final implementation in some ways. The final product includes new ideas, like different weapons and objects. The original concept specified more variety in enemies, but we steered away from this idea because of time limitations. There were also some ideas we decided to omit in order to reduce unnecessary clutter. Overall, the game is different from the original in some ways, but we are very happy with the outcome.

### 5.2. Comparison against other games in the genre

The market for a 2D platform shooter is filled with great classics, and some new releases, which add new and interesting ideas to this genre. We tried to create a game which went beyond very simple graphics, and which was entertaining to play. To best reflect on the game, we will compare it to two big titles in 2D platform shooters.

Contra was a great success among shoot-'em-ups, and created a benchmark for future titles. We attempted to imitate the entertaining side of Contra, and we are happy with the results

so far. Our prototype features way fewer objects and enemies than Contra, but we included a funny and engaging storyline, which Contra lacked. Another great feature of the classic was the multiplayer mode. We considered this option, but decided to focus on the single player first, to make it as good as we could. However, we will add this option in future versions.

While imitating the fun and engaging world of Contra, we also wanted to make something a bit more graphically pleasing. Based on this, we consider our artwork to be more than satisfactory. However, to compete in the current market, we need to compare our game to the biggest titles. The newest and hottest release in this genre is Cuphead. With its beautifully drawn graphics, there are very few titles that can compete with this game. We also went with drawing our sprites, and this process proved to be more complicated than we previously thought. To have a chance at competing with Cuphead, we would need a lot more drawings and more fluidity. However, based on the amount of time we had, we believe our prototype does the job well.

Overall, Shoot2Kill compares well with many 2D platform shooters. Our game differs from the titles listed above, mainly with having multiple platforms. We put a lot of effort into this game, and thanks to SFML and Box2D libraries, this task was made much easier. We are happy with the outcome we got, especially considering the size of the development team, as well as the timeframe.

### **5.3. Quality of the game**

Shoot2Kill is in its prototype form and should be considered a work-in-progress. We are happy with the quality of the game so far, and will strive to improve it. As for any project that's in its pre-release form, there are problems and issues which need to be resolved. However, we are still satisfied with our progress and we feel that the game helped us immensely with understanding how games development works.

### **5.4. Possible improvements**

Shoot2Kill offers fun gameplay and easy-on-the-eye graphics; however there are improvements that need to be implemented before the game could compete in today's gaming market. The first improvement to the game is much more variety in game objects, like enemies, turrets, and physical objects that the player can hide behind. Another improvement is with fluidity in graphics. As the levels and characters are all drawn by hand, making fluid animations takes a lot of time. However, due to the size of our development team, as well as the timeframe, we consider our game to be more than satisfactory.

## 6. Resources and references

- SFML library
- ECM library, Engine library – Kevin Chalmers (taken from the 'Platform' project)
- Character textures, including Simon, Enemies, Boss and Camilla – Szymon Duleba (Graphic Designer)
- Bullets textures – Szymon Duleba
- Promotional Graphics – Szymon Duleba
- Sound effects ( gun sounds ) – soundbible.com