Coursework Report

Games Engineering

SET09121

40215445

Table of content

1 Introduction

Zombay is a single player shooter game. It focuses mainly on combat. It was inspired from different games from this genre such as Minigore, Dead Nation and Alienation.
The main character controlled by the player is a young girl who is one of the last survivors of a zombie apocalypse. She must survive in this new world as long as she can doing so by killing massive amounts of zombies.

The project repository can be found at:
https://github.com/agrudzinska/zombay.git

2 Changes from the GDD

2.1 Player actions

- A dash function was not implemented due to the lack of time.
- Aiming was changed. The player now can shoot in the direction they are currently facing. Mouse functionality has been removed completely.

2.2 Rules

- Enemy dies upon contact with a bullet.

2.3 Resources

- Weapon upgrades are not included in this version
- Score was not implemented

3 Software design

3.1 Prefabs

The idea of prefabs was borrowed from Unity. Functions spawning all the enemies and menu elements (buttons) are stored in prefabs.cpp to avoid code repetition. It adds more clarity to the code. Enemies are then created in realtime in the Scene update() function.

3.2 Camera

The camera follows the player smoothly. It stops moving before reaching the player so that if they move by only a small distance, it does not follow it.

3.3 AI

Simple AI mechanics were implemented for the enemy entities. All enemies have a Seek() function that makes them chase the player. Different enemy types move at different speeds. Random enemies also have a jump feature implemented, that makes them speed up as they get closer to the player (Fig. 1).

```
if (_jump == false)
{
    auto output = _seek.getSteering();
    move(output.direction * (float)dt);
}
else
{
    auto output = _seek.getSteering();
    if(length(_parent->getPosition() - _player->getPosition()) < 150.0f &&
        length(_parent->getPosition() - _player->getPosition()) > 20.0f)
        move(output.direction * (float)dt * float(2));
    else
    {
        move(output.direction * (float)dt);
    }
}
```

Fig.1

3.4 Physics

The game uses the Box2D physics engine to generate the world, but does not use any more of its functionality.
All the collisions in the world are implemented separately for each of the entities (player, enemies, bullets).
The movement is also implemented individually for each entity.

4 Game implementation

Upon starting the game, the player is presented with a menu (Fig.2). There are buttons for starting the game, viewing game instructions, options (a placeholder button that as of now starts the game, will be used to present the user with different in the future) and quit. Upon clicking the Start button, the player is transported to the game scene (Fig.3). Enemies start spawning after four seconds, giving the player little time to get hold of the controls and locate the spawning points quickly.

Fig.2

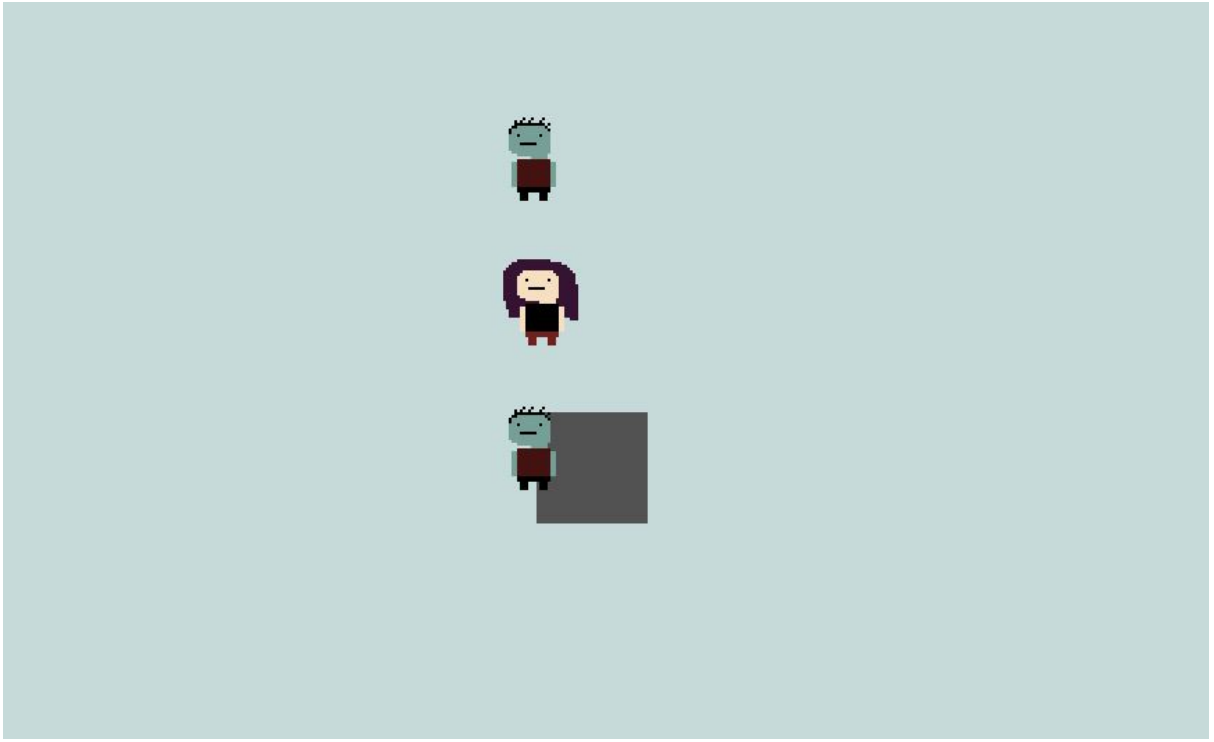

Fig.3

5 Implementation evaluation

The game was not fully implemented as intended. It is missing some of the functionality from the original Game Design Document. It is a basic, playable shooter game that can still be enjoyable for a short period of time. Given more time, more features could be implemented, making it more fun for the player. Possible features to be included in future versions of the product are:

- More advanced AI - implemented different behaviours for different types of enemies would make the game more difficult and add variety to the gameplay.
- Weapon upgrades for the player character
- Adding options - for example remappable controls, graphics options (different resolutions).

6 Playtests

Playtesting was conducted on a group of 5 people who were asked to answer following questions:

- Was the goal of the game clear?
    - 100% answered “Yes”
- Did you have any difficulties understanding the controls?
    - 80% answered “Yes”
    - 20% answered “No”
- How would you rate your overall experience (on a scale of 1 to 10)?
    - Average answer was 7

Additional comments received:

- “Game works just fine, could have more enemy types or implemented map variety.”
- “I like the graphics style. Much cute, such wow.”
- “Game seems a bit unfinished, but is still fun to play. I imagine it will be even more fun when more features are implemented.”

7 Version control

GitHub was used for version control throughout the duration of the project.
https://github.com/agrudzinska/zombay.git

8 Resources

All graphics were made using Pyxel Edit
https://pyxeledit.com/

All sounds designed and created by Finlay McFadzean
https://www.linkedin.com/in/finlay-mcfadzean-096334137/