AGP AE1 Assesment

Southampton Solent University | [Company address]

[Document subtitle]

Markus Krugel and foo jia rong

2018

Contents

[Game Description 2](#_Toc528839306)

[User stories 2](#_Toc528839307)

[Diagrams 2](#_Toc528839308)

[Class Diagram 2](#_Toc528839309)

[Flow Diagram 2](#_Toc528839310)

[Gantt Char 3](#_Toc528839311)

[Work Breakdown Structure 3](#_Toc528839312)

[3D Development Techniques 3](#_Toc528839313)

[Point Light 3](#_Toc528839314)

[Anisotropic filtering 3](#_Toc528839315)

[Mip mapping 3](#_Toc528839316)

[Deferred Rendering 3](#_Toc528839317)

[Testing Plans 3](#_Toc528839318)

[Report Markus Krugel 4](#_Toc528839319)

[Report Foo Jia Rong 4](#_Toc528839320)

[References 4](#_Toc528839321)

[Appendices 4](#_Toc528839322)

[Proof for backup methology 4](#_Toc528839323)

[Diagrams 4](#_Toc528839324)

[User Stories 4](#_Toc528839325)

[Class Diagram 4](#_Toc528839326)

[Flow Diagram 4](#_Toc528839327)

[Gantt Chart 4](#_Toc528839328)

[Work Breakdown Structure 4](#_Toc528839329)

# Game Description

*by Markus Krugel*

Our game is a 3D exploration game displayed in a third person view, where the camera is positioned behind the player, which also follows the player. References for this game are games like “Super Mario 64” and “Banjo Kazooie”.

The game starts with a start menu, where the player can select via mouse click if he wants to start the game or exit the game.

After the player selects to start the game, the level with its objects, enemies and the player will spawn. The player can move around the world using the WASD keys and space bar for jumping.

Additionally, the player can pause the game using the enter button. If the game is paused a pause menu will appear where the player can select via mouse click if they want to continue or to go back to the start menu.

Inside the world are also enemies which patrol around in certain areas and chase the player when the player gets in the sight area of the enemy, but they will also lose interest in the player when the player is too far away from him. Once the enemy arrived at the player’s position they will attack the player. The player can also attack the enemy using the “F” Key. If the enemy’s life is under zero they will die and drop an item, which can be collected by the player by simply walking over it. Otherwise if the player dies the screen will get black for a couple of seconds and a game over text will be shown. After that the player will return in the start menu.

There will be static objects, e.g. walls, in the world but also objects which can be moved around by colliding with it. Furthermore, there are objects like coins which can be collected by the player.

An optional goal is to implement NPC characters, which are talking to you when you press “E” in front of them.

# User stories

*by Foo Jia Rong*

fa

# Diagrams

## Class Diagram

*by Markus Krugel*

For every entity which will be displayed on the screen we use the “GameObject” class. Inside this class we store the basic information about the entity like position, model and the collisionBox.

The classes “Character” and ”Object” are derived from this “GameObject” class.

## Flow Diagram

*by Markus Krugel*

The flow diagram displays the general flow we intend to implement to our game.

In the beginning we create our GameManager class, which will handle the logic of the game. Right after that we are creating our Renderer and Camera so that the player can see the game.

Thereafter the StartMenu will be created, where the player can decide between two options via mouseclick. Either they choose to exit the game or they choose to start the game.

When the player decides to play the game, the level and the HUD will be created. The next step is to populate the level with the entities, like the player, enemies and the objects.

Now the update function inside the GameManager class will care about the main loop of the game. Inside this function the current state of the game will update constantly. There are three scenarios to break outside of the normal game loop.

The first option will be when the player dies. Then the screen will turn black for a couple of seconds and a game over message is being displayed. After that we are returning to the StartMenu state.

As a second option to break outside the normal game loop is to pause the game. Then the PauseMenu will be displayed and the update function of the GameManager stops to update the level. Inside the PauseMenu there will be again a decision between two options for the player. At the one hand he can go back to the StartMenu and on the other hand he can continue the game. If they choose to continue the PauseMenu will disappear and the update function will continue to do his work.

The last option is to finish the game. Thereafter the player lands once again in the StartMenu.

## Gantt Char

*by Foo Jia Rong*

## Work Breakdown Structure

*by Foo Jia Rong*

# 3D Development Techniques

## Point Light

## Anisotropic filtering

## Mip mapping

## Deferred Rendering

# Testing Plans

*by Markus Krugel*

# Report Markus Krugel

# Report Foo Jia Rong

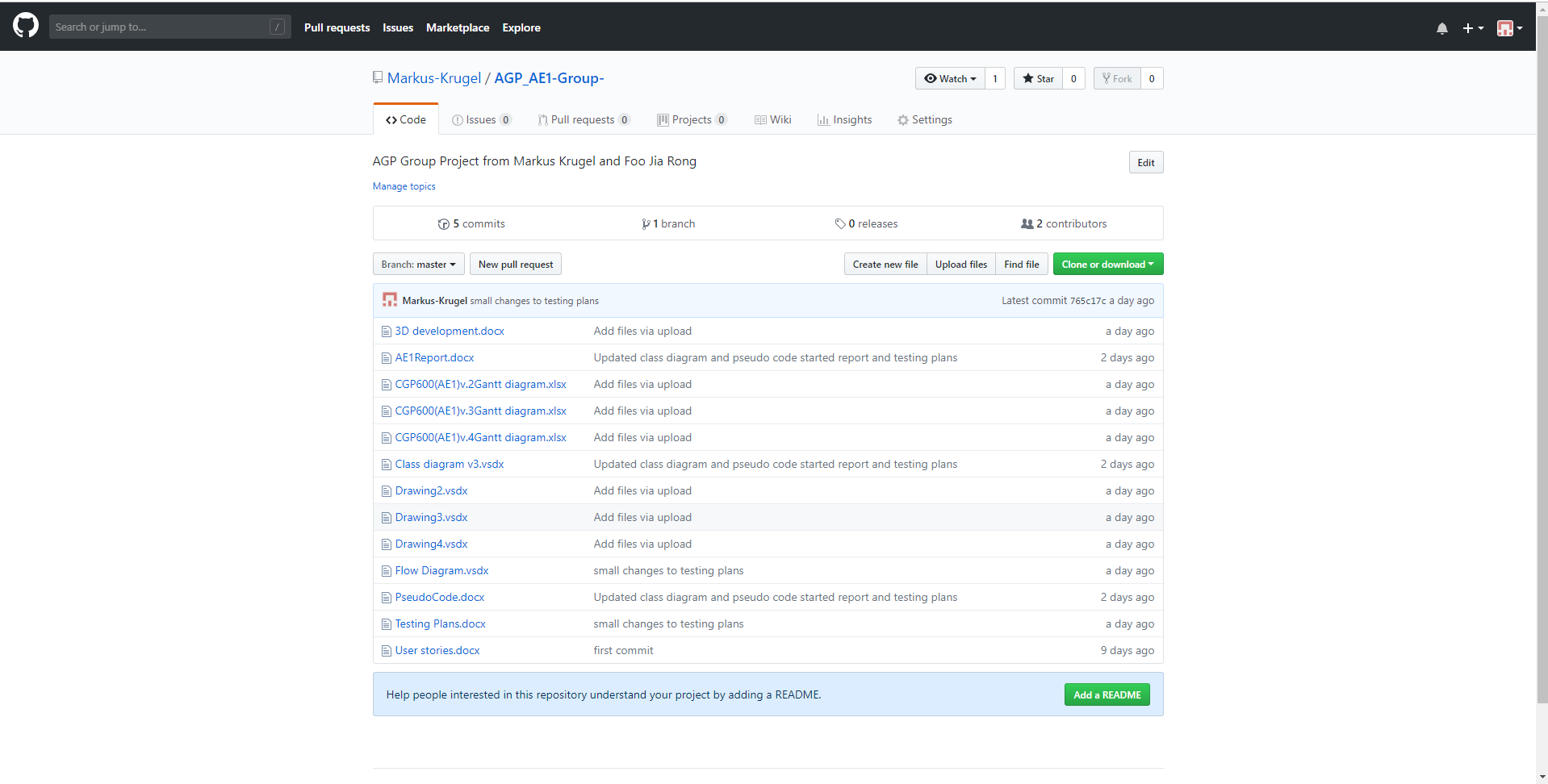
# References

# Appendices

## Proof for backup methology

Link to our Github repository:

<https://github.com/Markus-Krugel/AGP_AE1-Group->



## Diagrams

### User Stories

### Class Diagram

### Flow Diagram

### Gantt Chart

### Work Breakdown Structure