1. Introduction

1.1. Purpose of the system

The sytem is used for entertainment of the user. It is supposed to be a challenging experience which for ces the player to evolve his skills and navigate
carefully through the environment of game.

1.2. Scope of the system

The System is quite small as it is only used for educational purposes and to get used to the processes of developing a video game.

1.3. Objectives and success criteria of the project

The objective of the project is to create a small video game with at least a playable level and a completly planned out and structured development.

1.4. Definitions, acronyms, and abbreviations

The playable character is named "L41k4".
The research facility is named "Heidelberg Dynamics"

1.5. References

1.6. Overview

2. Proposed system

2.1. Overview

2.2. Functional requirements

The player is required to be able to move the playable character throughout the levels. WASD for up, do wn, left, right.
The player character should be able to aim using the mouse, e.g. the character aims in the cursor directi on at all times.
The player character should move throughout a level, interacting with the environment. e.g. bump into a wall, walk down a corridor, etc. (Collision).
The playable character and the enemies have a certain number of hitpoints. When they run out the char acter is eliminated.
The game is required to have multiple useable weapons which the player can pick up and between whic h they can swap.
The game features a weapon - named RBG - that shoots up to two small projectiles per second.
The game features a weapon - named Superperforator - that shoots a laser beam that damaged all ene mies in its area of effect.
There also have to be different enemies which can attack and damage the player.
There is an enemy type - named Scientist - that approaches the player for melee attacks.
There is an enemy type - named Security Turret - that is stationary and shoots small projectiles into the players direction.
The player must also be able to defeat these enemies with a certain amount of attacks.
To complete a level the player must be able to collect key cards which unlock the next level.
The player is healed to full health after each level completion.
After the last level is completed the player is shown an end screen that shows their successful escape.
When the players health reaches 0 the player is shown an end screen that shows their failed escape trial
.

2.3. Nonfunctional requirements

2.3.1. User interface and human factors

The game features a health bar that is visable to the player at all times.
The game contains a overview of the weapons the player has currently and which weapon is equipped at the moment. This overview is visable at all times.
The game cotains a indicator that shows the player if they already collected the key card of the current l evel.
A new game can be started by the user within one click.
The volume of the game is changable within four clicks.
The programm can be closed within two click.
The controls of the game can be learned by the within 3 minutes.

2.3.2. Documentation

There should be a documentation of at least the most important diagramms that help to understand the concept of the game in detail and as a whole. (see 2.5)
There should be a short documentation of all SCRUM meetings.
THere should be a documentation of changes made in the development process.

2.3.3. Hardware considerations

A PC with at least Windows 10 or a contemporary GNU/Linux variant,
At least 2 GB RAM
At least a GTX 750 or equivalent GPU
At least 2 GB of SSD Storage
At least a Ryzen 3 1200/Core i5-5200U or equivalent.

2.3.4. Performance characteristics

The game should run with at least 30 FPS.
A reaction to input should happen within 15 ms.

2.3.5. Error handling and extreme conditions

Errors and exeptions should be logged and reported to the developers.

2.3.6. Quality issues

All the code is written in Chromium coding style.
We use line breaks after 80 characters.
We always use Test-Driven Development

2.3.7. System modifications

There is just a log used for error handeling.

2.3.8. Physical environment

There is no internet connection required to play the game.
There is a mouse and keyboard needed to play the game.
There are no accessibility settings due to the games short development time.

2.3.9. Security issues

Due to the fact that the game uses no internet connection and no data from the player so there should b e no security issues.

2.3.10. Resource issues

2.3.11. Audio

The game features its own soundtrack that is running in the background during the main menu and the playthrough.
There are sounds for object collision.
There are sounds for shooting and eliminating enemies.
There is a sound effect when the player gets hit.
There is a sound effect for weapon swaps.
There is a sound effect for level completion and victory.
There is a sound for picking up items.
There is a sound effect when the players health reaches 0.

2.4. Pseudo requirements

The game should feel smooth and responsive at all times.
The game is designed in pixelart.
All levels and the overall design of the characters should match the setting of the research facility.

2.5. System models

2.5.1. Scenarios

2.5.2. Use case model

2.5.3. Object model

2.5.3.1. Data dictionary

2.5.3.2. Class diagrams

2.5.4. Dynamic models

2.5.5. User-interface -- navigational paths and screen mock-ups

3. Glossary