

Requirements and Analysis Document for HELP ME THIS IS NOT A JOKE THEY HAVE KIDNAPPED ME

Version: 1

2017-03-27

Klas Ludvigsson, Olle Persson, Jesper Persson, Jacob Torrång

1 Introduction

During all of mankind's long history people have always sought out different forms of entertainment. Some people watch or do sports, some watch movies, others play video games. During the somewhat short history of video games we have seen a rapid evolution, from simple 2D graphics and mechanics to full fledged physics based rendering. The purpose of this project to create a first person dungeon crawler game in the same spirit as games like Legends of Grimrock. This will be achieved using an object oriented approach while at the same time using different techniques within games programming.

1.2 Definitions, abbreviations and synonyms:

- Player = the user of the application
- NPC = non-playable character
- Tile = small squares that make up the floor, like tiles in a bathroom
- Dungeon crawler = **from wikipedia** "A dungeon crawl is a type of scenario in fantasy role-playing games in which heroes navigate a labyrinthine environment (a "dungeon"), battling various monsters, and looting any treasure they may find."
- FPS = frames per second
- Friendly = companion to the player
- Hostile = antagonist of the player; wants to attack player
- Item = an item that a player can interact with

1.2 General characteristics:

- Single player game
- First person view
- Being able to save and load game states
- Strict four directions to move
- Can attack enemies
- Can pick up- and use items
- The game ends if the player loses all his hit points or he completes the game objective
- Player can interact with NPCs

- The player has to interact with- and solve puzzles to move through the game

2 Requirements

2.1 Mockups



Mockup of start screen



Mockup of a new game

2.2 Functional Requirements

Player moves forward (1)
Player moves backward (1)
Player moves left (1)
Player moves right (1)

(Player rotates)
Player rotates right (4)
Player rotates left (4)

Player attacks (2)

Player self-destructs (3)
Player saves to file (6)
Player equips weapon (5)
Player uses item (7)
Player quits the game (9)
Player starts new game (8)

Player dies (?)
Player takes damage (?)

Enemy attacks (?)
Enemy moves (?)
Enemy takes damage (2)
Enemy dies (?)

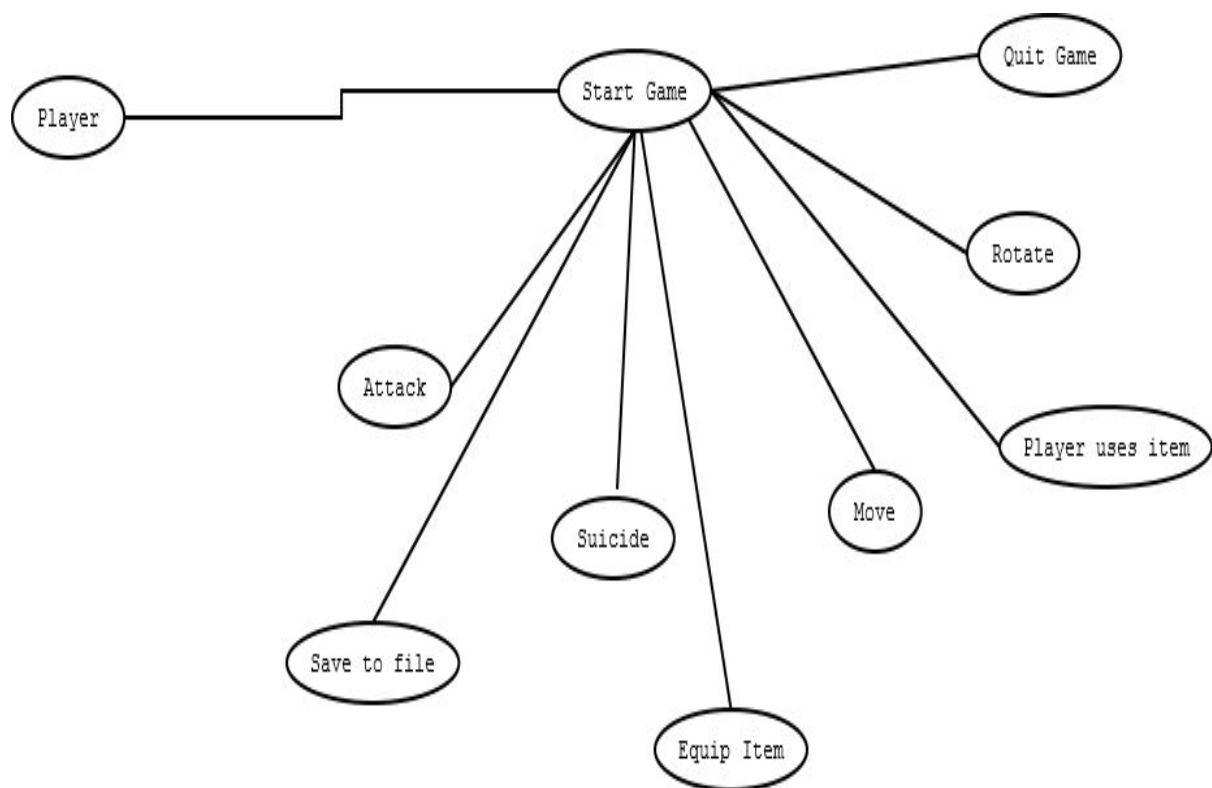
2.3 Non-Functional Requirements

The following requirements should hold:

- Good performance (>60 FPS?)
- Not prone to crashes (Crashes < 1% of average game sessions?)
- Intuitive interface (Given someone who's played a dungeon crawler before, they don't require a tutorial to play the game?)
- Easy to comprehend the game mechanics (Anyone (>90%) should be able to either directly understand the game mechanics or at least that every mechanic is transparent enough that you can test and figure them out?)
- Possible for most people to finish the game (>70% of people should be able to finish the game?)
- Looks decent (>50% of people think it looks decent?)
- Extensible code
- The code should be simple to read and beautiful
- The code should not be stolen from other places and the pictures used will not be taken from copyrighted sources

- The performance should be good with no long loading screens if the hardware holds some minimum requirements (<3sec).
- The application should be supported on Windows and Linux (MAC?)
- And loads of fun!

3 Use cases An UML use case diagram



3.1 Use case listing Use case texts (using the use case template)

UC: 1. Move

Summary: The player presses the arrow key up / down
 Priority: high

Participants: Player

Normal flow of events

The Player tries to move forward and there is an accessible tile in the direction the Player is facing.

| | Actor | System |
|---|--|---|
| 1 | Player presses the arrow key up / down | |
| 2 | | The camera moves from the current tile to the tile in front of / behind the Player, still facing the same direction |
| 3 | | Play walk sound |

Alternate flow

The Player tries to move forward but there is no tile in the direction the Player is facing.

| 2a | Actor | System |
|----|-------|------------------|
| 1 | | Play error sound |

The Player tries to move forward and there is a tile in the direction the Player is facing, but there is a locked door.

| 2b | Actor | System |
|-------|---|---------------------------|
| 1.1.1 | Player doesn't have the key to the door | Play locked sound |
| 1.2.1 | Player has the key to the door | Remove key from inventory |
| 1.2.2 | | Play unlocking sound |

The Player tries to move forward and there is a tile in the direction the Player is facing, but there is an NPC on it.

| | Actor | System |
|---|-------|------------------|
| 1 | | Play block sound |

UC: 2. Attack

Summary: The player attempts to attack

Priority: high

Participants: Player

Normal flow of events

The Player attempts to attack and there is no one on the tile in front of the player

| | Actor | System |
|---|----------------------------------|-----------------------|
| 1 | Player presses the attack button | |
| 2 | | Play attack animation |
| 3 | | Play hollow sound |

Alternate flow

The Player attempts to attack and there is an Enemy on the tile in front of the player

| 3a | Actor | System |
|-------|-------------------------|---|
| 1 | | Calculate damage |
| 2 | | Calculate new hit points of Enemy (current hit points - damage) |
| 3 | | Play hit sound |
| 4 | | Text output “[damage]!” |
| 5 | | Enemy blinks red. |
| 6.1.1 | Enemy's health ≤ 0 | Text output “[Enemy name] died!” |
| 6.1.2 | | Play death rattle sound |
| 6.1.3 | | Enemy fades away from scene |
| 6.2 | Enemy's health > 0 | Text output “[Enemy name]'s health is [Enemy Hitpoints]” |

The Player tries to attack and there is a on the tile in front of the Player.

| 3b | Actor | System |
|-----|---|--|
| 1 | | Calculate damage |
| 2 | | Calculate new hit points of Friendly (current hit points - damage) |
| 3 | | Play hit sound |
| 4 | | Text output “[damage]!” |
| 5 | | Friendly blinks red |
| 6.1 | Friendly turns hostile on Attack | Friendly changes into Enemy |
| 6.2 | Friendly doesn’t turn hostile on Attack | Text output “[Friendly response when attacked]” |

UC: 3. Suicide

Summary: The player inputs “self-destruct” and attempts to kill itself

Priority: low

Participants: Player, System

Normal flow of events

The Player attempts to kill itself.

| | Actor | System |
|---|---|------------------------------|
| 1 | Player presses the self-destruct button | |
| 2 | | “Game Over” dialogue pops up |

UC: 4. Rotate

Summary: Player rotates to the either left or right.

Priority: high

Participants: Player, System

Normal flow of events

The Player tries to rotate to the right.

| | Actor | System |
|---|--|--|
| 1 | Player presses the rotate right / left arrow key | |
| 2 | | The camera rotates 90 degrees clockwise / anti-clockwise |

UC: 5. Equip item

Summary: The player attempts to equip an item

Priority: high

Participants: Player

Normal flow of events

The player attempts to equip an item but already has that item equipped

| | Actor | System |
|---|--|---|
| 1 | Player right clicks an item | |
| 2 | | Menu with "equip item" pops up at the mouse cursor's current position |
| 3 | Player left clicks the option "equip item" on the new menu | |
| 4 | | Play equip sound |

Alternate flow of events

The player attempts to equip an item and that item is not already equipped

| 3a | Actor | System |
|-------|------------------------------------|----------------------------------|
| 1.1.1 | Player has an[other item] equipped | Unequip [other item] |
| 1.1.2 | | [other item] loses its red frame |
| 1.1.3 | | Equip item |
| 1.1.4 | | Item gets a red frame around it |
| 1.1.5 | | Play equip sound |
| 1.2.1 | Player has no other item equipped | Equip item |
| 1.2.2 | | Item gets a red frame around it |
| 1.2.3 | | Play equip sound |

UC: 6 Save to file

Summary: Player saves the game

Priority: mid

Participants: Player

Normal flow of events

| | Actor | System |
|---|----------------------|--------------------------------------|
| 1 | Player save the game | |
| 2 | | Saves the game to a file |
| 3 | | Outputs that the game has been saved |

UC: 7 Player uses item

Summary: Player uses an item

Priority: low

Extends: Non

Includes: Non

Participants: Player

Normal flow of events

| | Actor | System |
|--------------------|-------|--------------------------|
| 1 Player uses item | | |
| 2 | | Apply the effect of item |

UC: 8 Start a game

Summary: The player starts up the game

Priority: high

Participants: Player

Normal flow of events

| | | |
|---|-----------------------|---------------------------------------|
| 1 | Actor | System |
| 1 | Player loads the game | |
| 2 | | Loads the game data |
| 3 | | Starts a game with the loaded data |
| 4 | | Outputs that the game has been loaded |

Alternate flow

| | | |
|----|---------------------------|--------|
| 1a | Actor | System |
| 1 | Player presses "New Game" | |

| | | |
|---|--|--|
| 2 | | The system creates a file on the Player's computer and launches the game |
|---|--|--|

Exceptional flow

| | Actor | System |
|---|-----------------------|--|
| 1 | Player loads the game | |
| 2 | | The system responds that the file is corrupt or the file doesn't exist |

UC: 9. Quit the game

Summary: The player quits the game

Priority: high

Includes: Save game

Participants: Player

Normal flow of events

| | Actor | System |
|---|---------------------|-----------------------|
| 1 | Presses "Exit Game" | |
| 2 | | Exits the application |

Alternate flow

| 2a | Actor | System |
|----|--------------|-----------------------------|
| 1 | | Asks "Save before exiting?" |
| 2 | Presses "No" | |

Alternate flow

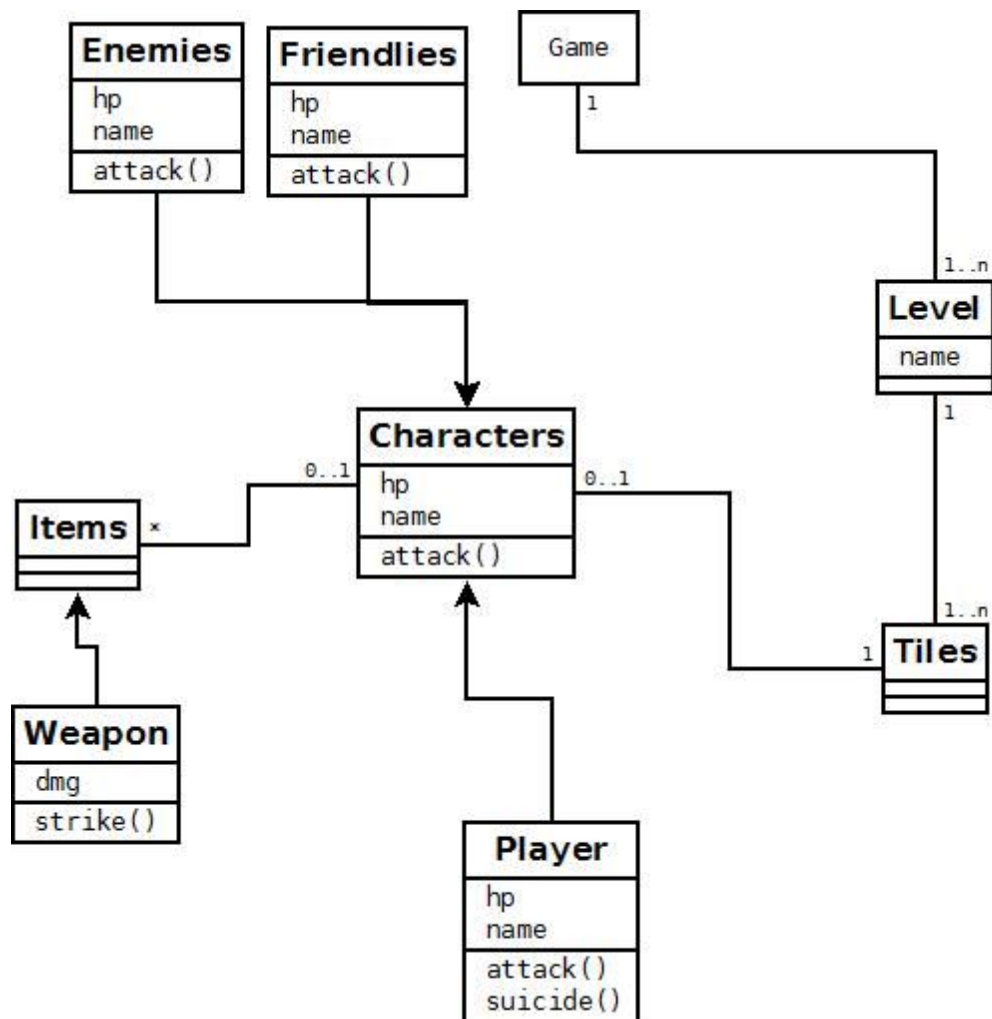
| 2b | Actor | System |
|----|---------------|-----------------------------|
| 1 | | Asks "Save before exiting?" |
| 2 | Presses "Yes" | |

| | | |
|---|--|----------------|
| 3 | | Saves the game |
|---|--|----------------|

Exceptional flow

| | Actor | System |
|---|-------|--------|
| 1 | | |
| 2 | | |

4 Domain model An UML class diagram.



4.1 Class responsibilities Explanation of responsibilities of classes in diagram

The game consists of 1 or more levels which consists of tiles. Every character has a position of exactly one tile and a tile might or might not have a character on it. A character has 0 or more items. An item might be a weapon. A character might be either a player or an enemy. An enemy is an NPC that is hostile towards the player while a friendly NPC is not. The player character is controlled by the human player.

5 References