

Requirements and Analysis Document for Group 28

Version: 2

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
- Alignment = hostile and friendly are alignments
- 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:

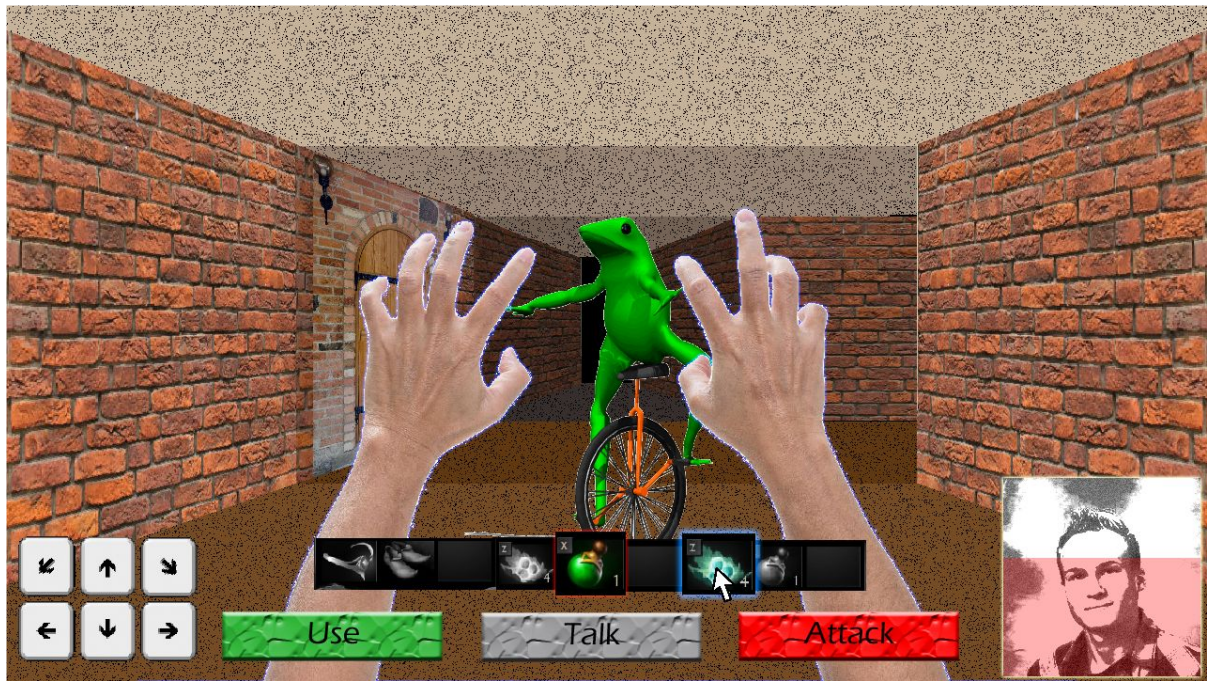
- Single player game
- First person view
- Being able to save and load game states
- Strict four directions to move
- Can attack NPCs regardless of alignment
- Can pick up- and use items
- Each item has its own effect when used
- The game ends if the player loses all his hit points or he completes the 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 game session



Mockup of a game session with an NPC response

2.2 Functional Requirements

Ids below are on the form "FR number", where number is the corresponding list number.

1. Player should be able to move forward
2. Player should be able to move backward
3. Player should be able to strafe right
4. Player should be able to strafe left

5. Player should only be able to walk to accessible tiles
6. Player should be able to unlock doors
7. Player should be able walk through doors
8. Player should be able to rotate right
9. Player should be able to rotate left
10. Player should be able to talk to friendly NPC
11. Player should be able to attack NPC
12. Player should be able to attack nothing
13. Player should be able to take damage from NPC
14. Player should be able to save the game state
15. Player should be able to load the previous game state (if there is one)
16. Player should be able to equip items
17. Player should be able to pick up items
18. Player should only be able to have one equipped item at a time
19. Player should be able to attack with the equipped item
20. Player should be able to use equipped item on self
21. Player should be able to change equipped item
22. Player should be able to quit the game
23. Player should be able to start a new game state
24. The System's NPCs should be able to attack the player
25. The System's NPCs should be able to move
26. The System's NPCs should be able to take damage

2.3 Non-Functional Requirements

References to “minimum requirements” refers to:

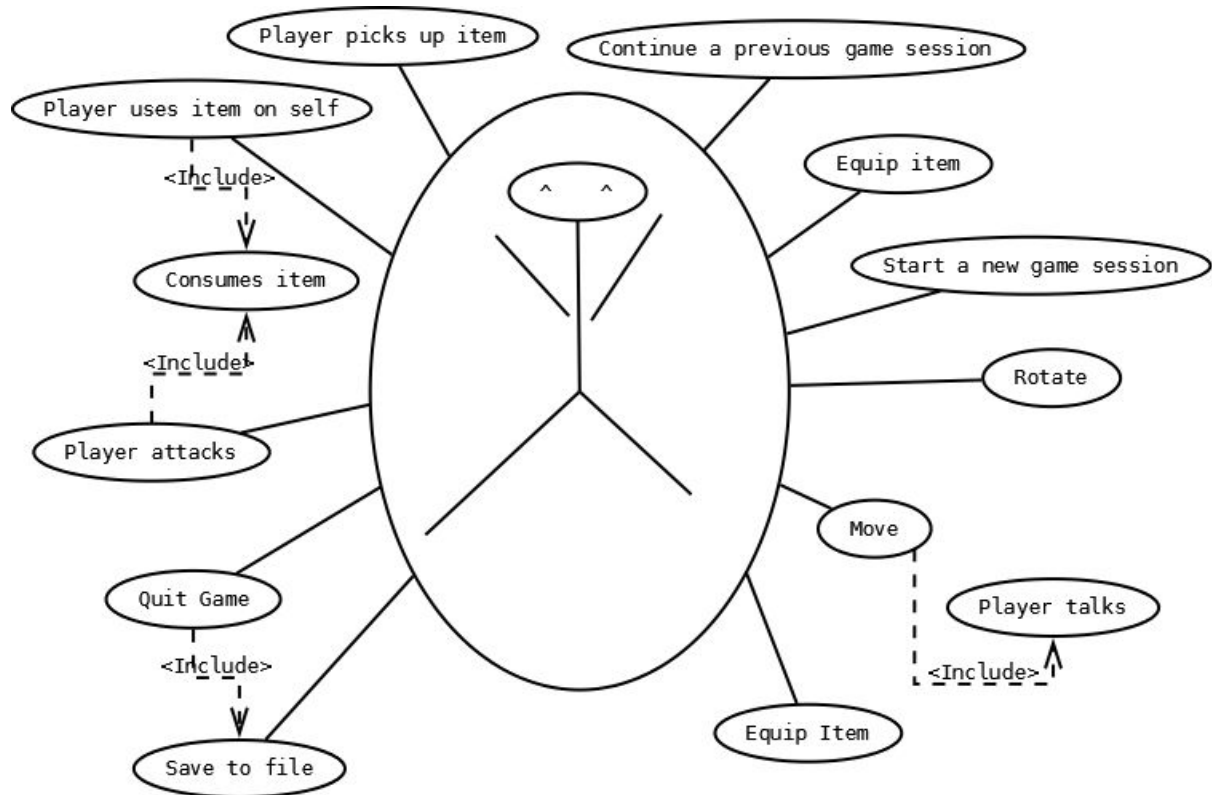
PC with at least; a processor from the Sandy Bridge generation, a GTX 760 (or corresponding AMD card) and 8 GB of RAM

Ids below are on the form “NFR number”, where number is the corresponding list number.

1. The game should have an average of 60 FPS at runtime at minimum requirements.
2. The game should have an average of running 99% of the time in a game session
3. The game should support Windows and Linux
4. Players should be able to start the game from the menu without help, at least 90% of new players
5. Players should be able to figure out the controls for moving and rotating the character without outside help, at least 90% of new players in period of 10 minutes
6. Players should be able to complete the game, at least 70% of players should be able to finish the game within 50 hours of active gameplay
7. The game should be fun!

3 Use cases

3.1 Use cases UML



3.1 Use cases

UC 1. Move

Summary: The player presses the arrow key up / down

Priority: high

Participants: Player

Includes: UC 11 (player talks)

Normal flow of events

The Player tries to move to a tile

	Actor	System
1	Player presses the arrow key up / down / right / left	
2		The player moves to the tile in front / behind / to the right / to the left of the player, facing the same direction as before
3		Play walk sound

Alternate flow

There is a wall blocking the way

2-3a	Actor	System
1		Play error sound (no movement)

There is a locked door blocking the way and the player doesn't have a key

2-3b	Actor	System
1	Player doesn't have the key to the door	
2		Play locked sound

There is a locked door blocking the way and the player has the appropriate key

2-3c	Actor	System
1	Player has the key to the door	
2		Remove key from inventory
3		Play unlocking sound

There is an NPC on the tile that the player attempts to move to

2-3e	Actor	System
1		Play block sound

UC 2. 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 3. 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 clicks an item	
2		Equip item
3		Item gets a red frame around it
4		Play equip sound

Alternate flow of events

There is already an item equipped

2a	Actor	System
1	Player has an[other item] equipped	
2		Unequip [other item]
3		[other item] loses its red frame

UC 4. 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

Exceptional flow

2-3a	Actor	System
1		System tries to save to a file but fails
2		Displays an error message

UC 5. Player attacks

Summary: Player attacks

Priority: high

Participants: Player

Include: UC 7 (Consume item)

Normal flow of events

Player attacks NPC on tile in front of player with an item that does damage and has a special effect

	Actor	System
1	Player presses the attack button	
2		Check the effect of equipped item
3		Calculate damage
4		Apply damage to NPC
5		Apply item effect to NPC

Alternate flow

Player has no item equipped

2a	Actor	System
1		Choose the effect of player's default item (fist)

Player is facing tile with no NPC on it

3-4a	Actor	System
1		Apply damage on empty tile
2		Apply effect on the empty tile

UC 6. Player uses item on self

Summary: Player uses an item on self

Priority: high

Participants: Player

Includes: UC 7 (Consumes item)

Normal flow of events

Player uses an equipped item on self

	Actor	System
1	Player presses button to use item on self	
2		Check the effect of equipped item
3		Calculate damage
4		Apply damage to self
5		Apply item effect to self

Player has no item equipped

2a	Actor	System
1		Choose the effect of player's default item (fist)

UC 7: Consumes item

Summary: Player uses on self / attacks with consumable item

Priority: high

Participants: Player

Normal flow of events

Player uses on self / attacks with consumable item, and there is only 1 stack of the item

	Actor	System
1		Item is removed from inventory

Alternate flow

There is more than 1 stack of the item

	Actor	System
1		Decrease the item counter by 1

UC 8. Continue a previous game session

Summary: The player continues a previous game session

Priority: high

Participants: Player

Normal flow of events

	Actor	System
1	Player presses the Continue button	
2		Load previous game state from file
3		Start game with game state
4		Output that the game has been loaded

Exceptional flow

Tries to load, but there is no previous data

2-4a	Actor	System
1		There is no previous data to load
2		Exception: No previous save to load!

UC 9. Start a new game session

Summary: The player starts a new game session

Priority: high

Participants: Player

Normal flow of events

1-a	Actor	System
1	Player presses the New Game button	
2		The system saves the starting game state to a file
3		Start game with the starting game state

Exceptional flow

2-3a	Actor	System
1	System doesn't have the privileges to create a file	
2		Exception: Cannot create save!

UC 10. 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		Asks "Save before exiting?"
3	Presses "Yes"	
4		Saves the game
5		Exits the application

Alternate flow

2-4a	Actor	System
1		Asks "Save before exiting?"
2	Presses "No"	

Exceptional flow

2b	Actor	System
1		The system can not save
2		Outputs "Save failed!"
3		Outputs "Try to save again?"

UC 11. Player talks

Summary: The player talks to an NPC in front of the player

Priority: high

Participants: Player

Normal flow of events

The player walks on a tile and there is an item on that tile

	Actor	System
1	Player presses "Talk"	
2		Dialogue popup: [NPC response]
3		Play dialogue popup sound
4	Player presses any button, or clicks with the mouse anywhere	
5		Dialogue disappears

UC 12. Player picks up item

Summary: The player picks up an item

Priority: high

Participants: Player

Normal flow of events

The player walks onto a tile, there is an item on the tile and there's a slot in the player inventory

	Actor	System
1		Put item in available slot in inventory

Alternate flow

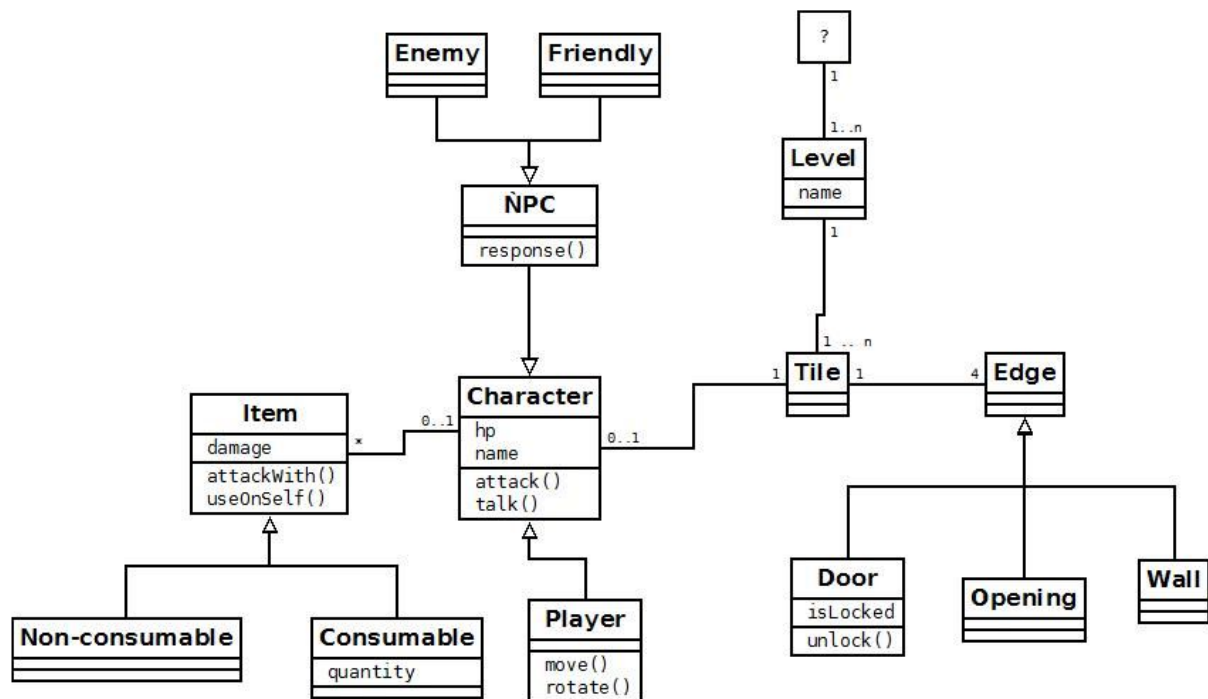
It's a consumable item and there is already one instance of that item in the inventory

1a	Actor	System
1		Increment item counter by 1

There is no available item slot

1b	Actor	System
1		Play full sound

4 Domain model



4.1 Class responsibilities Explanation of responsibilities of classes in diagram

The game consists of 1 or more levels which consists of tiles. Every tile has 4 edges. An edge is either a door, opening, or a wall. 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 has a damage and can either be attacked with or used on self. Items are consumable or nonconsumable. A character has a name, hp and can attack and talk. A character is either an NPC or a player controlled character. An NPC is either an enemy or a friendly. A player can move and rotate.