Requirements and Analysis Document for The game

Version: 1.0

Date: 22/03/18

Author: Farzad Besharati, Therese Sturesson, Philip Nilsson, Linus Wallman

This version overrides all previous versions.

1 Introduction

The current RPG market on PC is lacking in what could be labeled as “Old School” RPG’s. The popular RPG’s of today are often mechanically too easy or linear in design. Our goal is to design and develop a game that caters to fans of games that became popular during the mid to late 1980’s, such as the original Legend of Zelda for the Nintendo Entertainment System.

The application will be a desktop, standalone (non-networked), one-player application with a graphical user interface for the Windows/Mac/Linux platforms.

Some general characteristics:

- It is a one player game.
- The application will be able to save the game, so you will be able to quit the game and come back later.
- It will be an open world game so you will be able to go around freely on the map.
- The GUI will be simple and easy to use. It will have a start menu and when you start the game it will have a bar with information and items and a box where you play.
- There will be enemies (handled by the computer) to defeat, items to pick up and use, points to collect.
- Your character will die and the game will end if your health drops to 0.
- There is no time constraints at all in the game.
- Your points will be saved to a top score list when the game ends.

1.2 Definitions, acronyms and abbreviations

Technical definitions:

- JRE: Java Runtime Environment
- PE: Portable Executable, see references
- Jar: package file format used to aggregate Java class files

Most definitions and terms regarding the core game:

- Player: Representative of the person playing the game
- AI: Representative of entities that aren't the Player
- World: The environment in which the game takes place
- Tile: An even-sided square, the world consists of these
- Character: An entity in the world, could be the AI or the Player
- Player Character (PC): A Character representing the Player
- AI Character: A Character representing the AI

2 Requirements

2.1 Graphical user interface

First off the application will have a fixed size and a fixed GUI that you will not be able to change. The start menu will fill up the whole window. The menu will have four big buttons where you can choose to start the game, Exit the game, or other things.

(Bild)

When you start the game the menu will disappear and the application will show the game. Now the window will have an information and item bar along the lower side of the window. Here you will be able to interact will the items you have, se how much health you have and how many point you have.

(Bild)

The rest of the window will show the game where you can see your character, go around in the world, interact with enemies, pick up items and so on. It should be possible later to add animations.

(Bild)

2.2 Functional requirements

- From the menu the user has the ability to…
    - Start the game
    - Top score list
    - Learn how to play the game
    - Change controller or button/key mapping
    - Quit the application
- While inside the game world the user can…
    - Move about freely in the world
    - Pick up items found in the world

- Transition from one area to another. From the “outside” world into a dungeon, for example
- Attack AI controlled characters and slay these to acquire rewards
- Open an inventory screen from which users can...
    - Manage items in inventory by manipulating them
    - Consume items (e.g bombs)
    - Equip items

Ordering of use cases by priority:

Start Application
Start Game
How to play
Settings
Exit
Attack
Damage
Death
Item activation
Inventory Management
Move
Item pickup

2.3 Non-functional requirements

Any special considerations besides functionality? Usability, reliability, performance, supportability, legal, implementation, … NOTE: Testability mandatory (must have tests)

Usability

The game is designed in such a manner that the user should without any trouble immediately be able to get going in the game with only minor descriptions of how to play the game.

Reliability

There are no critical reliability requirements as it’s not a critical system.

Performance

This is a game after all, which means that there are certain real-time requirements when it comes to rendering and equivalent. This should however not be a problem as this application isn’t as complex, and only uses 2D textures.

The user will be most comfortable playing at a stable 60 fps or higher. Therefore, the application will most likely be limited to 60 fps.

Testability

The underlying game logic should be testable without an accompanying graphical interface.

Technical

The application must be implemented in a manner such that it is easy to scale, further extend its functionalities, and implement comprehensive features such as a client/server-architecture.

Platform independency is achieved using JRE. To run the application, the user will need to have JRE installed.

For usability sakes, it's best to provide a PE (for Windows users) to the user (jar wrapped into PE). This way, we hide information not needed for the user which yields the feeling of opening/executing any other program that they're used to. Same goes for any other data that might confuse the user, such as any library contained in for example dlls.

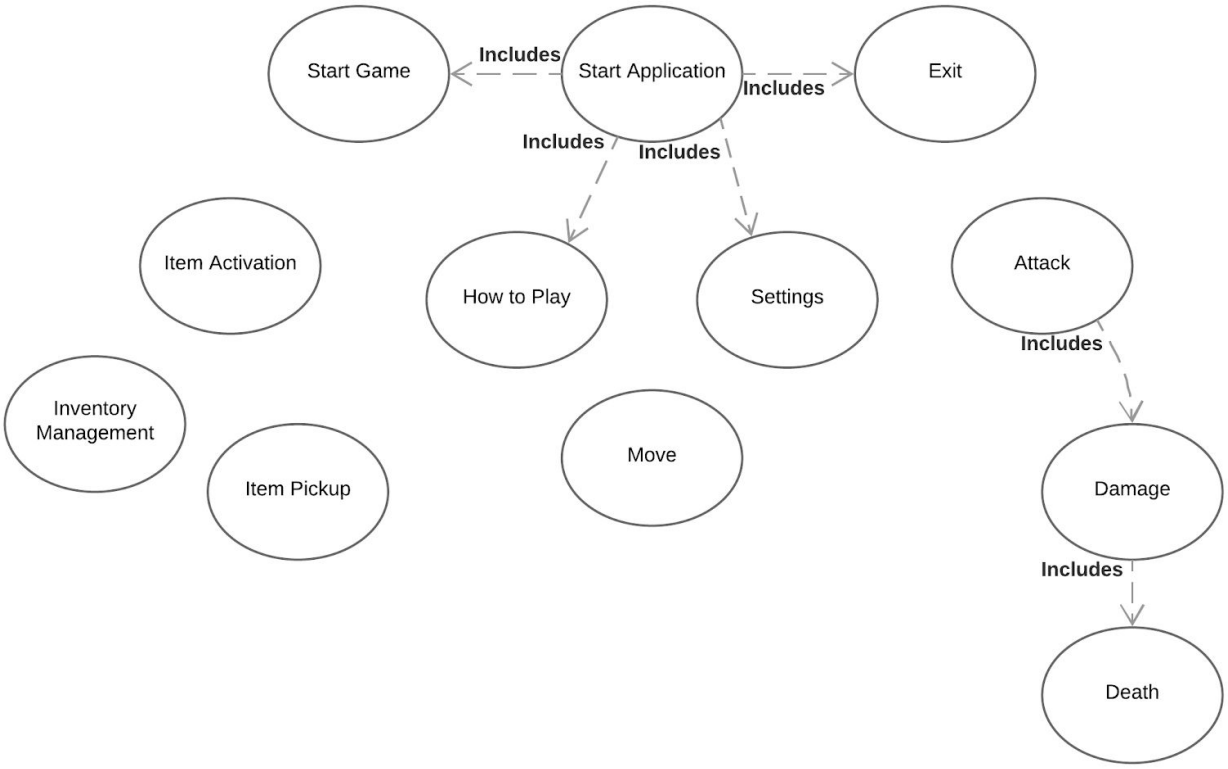
It's to prefer for sake of usability that an installer is provided which unpackages everything and sets it up nice and tidy for the user. However, it will most likely be an app that you get as-is.

The source code will not be included.

List of what's included in the package:

1. A file that contains the application code (jar file(s) wrapped into a PE)
2. All needed resources (assets, map, external libraries, etc.)
3. A README-file containing short documentation and general information regarding how you execute, configure, and install the application

3 Use cases


Start Game
Includes
Start Application
Includes
Exit
Includes
Includes
Item Activation
How to Play
Settings
Attack
Includes
Inventory Management
Item Pickup
Move
Damage
Includes
Death

3.1 Use case listing

UC: 1. Move

Summary: A character moves from one tile in the game world to another
Priority: Mid
Extends: None
Includes: None
Participants: Character

Normal flow

Flow 1: Normal movement, move into an empty tile

	Character	System
1.	Character tries to move into another tile	
2.		Character moves one tile in the chosen direction

Alternate flow

Flow 2: Movement into impassable tile

	Character	System
1.	Character tries to move into an impassable tile	
2.		Position remains unchanged

Flow 3: Movement into tile occupied by an item

	Character	System
1.	Character tries to move into a tile occupied by an item	
2.		Move Character into tile
3.		Ref. UC: 5. Item pickup

Flow 4: Movement into tile occupied by a character

	Character	System
1.	Character tries to move into a square occupied by another character	
2.		Position remains unchanged

UC: 2. Attack

Summary: A Character performs an aggressive action towards another Character
Priority: High
Extends: None
Includes: Damage
Participants: Character

Normal flow

Flow 1: Character attacks a target and hits

	Character	System
1.	Attacks	
2.		Checks if the attack hits the target
3.		Target is hit
4.		Damage is calculated, refer to UC: 3. Damage

Alternate flow

Flow 2: Character attacks but misses or hits nothing

	Character	System
1.	Attacks	
2.		Checks if the attack hits the target
3.		Attack misses

UC: 3. Damage

Summary: A source deals damage to a character, damage dealt is deducted from characters hit points
Priority: High
Extends: None
Includes: Death
Participants: Character

Normal flow

Flow 1: Character takes damage and does not die

	Character	System
1.	Character takes damage from any source	
2.		Character's health decreases by an amount related to the source's attack
3.		Character enters a brief period of invulnerability

Alternate flow

Flow 2: Character takes damage and dies

	Character	System
1	Character takes damage from any source	
2		Character's health decreases by an amount related to the source's attack
3		Health becomes 0, Character dies, move to UC 4. Death

UC: 4. Death

Summary: Character death when damage dealt is greater than or equal to current HP
Priority: High
Extends: None
Includes: None
Participants: Character

Normal flow

Flow 1: AI Character dies

	AI Character	System
1	AI Character has met the requirements for death	
2		Character is removed from World
3		Character drops an item from an item drop table

Alternate flow

Flow 2: Player Character dies

	Player Character	System
1	Player Character has met the requirements for death	
2		Player is presented with Game Over

UC: 5. Item Pickup

Summary: A Character picks up an item in the game world
Priority: Mid
Extends: None
Includes: None
Participants: Player, AI

Normal flow

Flow 1: Player Character picks up consumable item

	Player Character	System
1	Player Character stands on top of a consumable item	
2		Check so that the Player hasn't reached the item limit and that the player has enough inventory space for this item

3		Remove item from the world and increment the amount in the Player Character's inventory

Alternate flow

Flow 2: Player Character picks up consumable item, but does not have enough room

	Player Character	System
1	Player Character stands on top of a consumable item	
2		Check so that the Player hasn't reached the item limit and that the player has enough inventory space for this item
3		Player does not have room in inventory, remove item from the world, inventory remains unchanged.

Flow 3: Player Character picks up direct consumable item

	Player Character	System
1	Player Character stands on top of a direct consumable item	
2		Item's effect is immediately applied to character

Flow 4: Player Character picks up permanent item

	Player Character	System
1	Player Character stands on top of a permanent item	
2		Check so that Item isn't in the inventory already
3		Remove Item from World and place into Player Character's inventory

Flow 5: AI Character tries to pick up an item

	AI	System

1	AI Character stands on top of a any item	
3		Item state remains unchanged

UC: 6. Item Activation

Summary: Player interacts with an item, e.g uses an equipped item.
Priority: High
Extends: None
Includes: None
Participants: Player

Normal flow

Flow 1: Player activates an equipped permanent item

	Player	System
1.	Player presses the item's key	
2.		The item's effect activates

Alternate flow

Flow 2: Player activates an equipped consumable item

	Player	System
1.	Player presses the item's key	
2.		The item's count is decremented and the item's effect activates

UC: 7. Inventory management

Summary: Managing your inventory. This includes things such as equipping items.
Priority: High
Extends: None
Includes: None
Participants: Player

Normal flow

Flow 1: Player opens/closes the inventory

	Player	System
1.	Presses the inventory button	
2.		The inventory hotbar appears/disappears

Alternate flow

Flow 2: Player equips an item in an empty equipment slot

	Player	System
1.	Player presses the directional keys	
2.		The inventory cursor changes position between items
3.	Player presses a key corresponding to an equipment slot	
4.		The item selected by the cursor is put into the corresponding equipment slot

Flow 3: Player equips an item in the same equipment slot as another equipped item

	Player	System
1.	Player presses the directional keys	
2.		The inventory cursor changes position between items
3.	Player presses a key corresponding to an equipment slot	
4.		The item equipped in the slot is replaced with the one selected by the cursor and is put back into the inventory

UC: 8. Start Application

Summary: A start menu where you can choose to start the game, read about it, and quit the game.

Priority: High
Extends: None
Includes: Start Game, How to Play, Settings, Exit
Participants: Player

Normal flow of events

Flow 1: Shows the Start Menu

	Player	System
1.	Starts the application	
2.		Shows the start menu

UC: 9. Start Game

Summary: Starts the main game from the Start Menu
Priority: High
Extends: None
Includes: None
Participants: Player

Normal flow of events

Flow 1: Player starts the game

	Player	System
1.	Clicks on the “Start Game” button	
2.		Switch to the game window and start the main game

UC: 10. How to Play

Summary: Accesses the “How to Play” menu from the Start Menu
Priority: High
Extends: None
Includes: None
Participants: Player

Normal flow of events

Flow 1: Player opens and closes the How to Play menu

	Player	System
1.	Click on the “How to play” button	
2.		Player is presented with the manual for how to play the game
3.	Click on the “Go back” button	
4.		Player is returned to the main menu

UC: 11. Settings

Summary: Accesses the Settings from the Start Menu
Priority: High
Extends: None
Includes: None
Participants: Player

Normal flow of events

Flow 1: Opening and closing the “Settings” menu

	Player	System
1.	Click on the “Settings” button	
2.		Player is presented with the “Settings” screen
3.	Click on the “Go Back” button	
4.		Player is returned to the main menu

UC: 12. Exit

Summary: Close the application
Priority: High
Extends: None
Includes: None
Participants: Player

Normal flow of events

Flow 1: Exit

	Player	System
1.	Click on the “Quit” button	
2.		The game stops and the window disappear

4 Domain model

5 References

[Portable Executable]