RAD

1 Introduction

Purpose of application
The project aims to create a game of the genre "Roguelike". It is simply meant to entertain the user.

General characteristics of application
A "Roguelike" is a game based on a very old game called "Rogue". Its characteristics are somewhat close to a roleplaying game. Our idea is that you progress in a 2-dimensional environment with a single character, going through different rooms in some kind of dungeon. On the way through the dungeon there will be enemies trying to stop you with possible platforming involved. At the end of the dungeon there will be some kind of boss creature.

The game will be pretty short but very difficult, and feature "permadeath" meaning that if your character dies, you lose all progress and have to start over. To make things more interesting the map and gameplay is not the same every time you play (typical characteristics of a "Roguelike"). The map will be randomly generated or at least put different rooms in different orders so that you don't know what challenge is next. Enemies are also placed randomly.

There will be a great number of items changing the dynamics of combat and every time you find one it will be randomly generated.

Scope of application
A single player only game.

An approximation of minimum content to be added:
- One main character.
- 10 different enemies.
- 30 items.
- One boss.
- 20 rooms.

Objectives and success criteria of the project
The player should be able to move the main character around by walking and jumping. Move from room to room and face enemies with varying AI functionality. Killing enemies award experience points to increase your character's strength. In some rooms chests will be placed with loot, loot will include a big set of items that alters combat. Getting killed ends the game. Killing the boss at the end wins you the game.

Definitions, acronyms and abbreviations

- Roguelike
    - A game with permanent death and a randomly generated game world.
- Hero
    - The player controlled character.
- Enemy
    - A character that tries to kill or in other way hinder the Hero. Controlled by the computer.
- Boss
    - Special enemy category. Boss enemies are unique enemies with special abilities and looks. In order to progress through the game, the player has to beat bosses.
- Key
    - A key opens locked doors and chests in the game. Can only be found.
- Weapon
    - A tool for killing enemies. Weapons can be found or bought. The player can only keep a limited number of weapons at a time.

2 Requirements

Functional requirements

1. Start a new game.
2. Play the game
    a. Move the player character left and right
    b. Make the player character jump
    c. Attack enemies
    d. Pick up items
    e. Exit rooms and enter new ones
    f. Challenge and defeat a boss
3. Restart the game
4. Exit the game

Non-functional requirements

Usability

The game will have intuitive controls for players who are used to this kind of game.

Actions taken by the player should affect the game as soon as possible. Starting a new game or restarting the current game should be available with one or two key presses. The player should be challenged by the difficulty of the game, but never by the controls themselves.

Reliability
-

Performance
Response is critical in a real time action game. Almost instant response is very important to give the game a good feel. The game should be able to run at full speed even on older computers.

Supportability
The application is going to be available for PC/Mac/Linux.

Implementation
The game will be written using Java and the LibGDX framework.

Packaging and installation
-

Legal
All audio and visual material used are either self made or taken with the creator's consent.

Application models

User case model
See appendix

Use cases priority

1. Start game
2. Hero Walk
3. Hero Jump
4. Hero Attack
5. Shoot/Throw
6. Open chest
7. Equip item
8. Attribute menu
9. Spend attribute points
10. Pause game
11. Resume game
12. Exit game

Analysis model
See appendix.

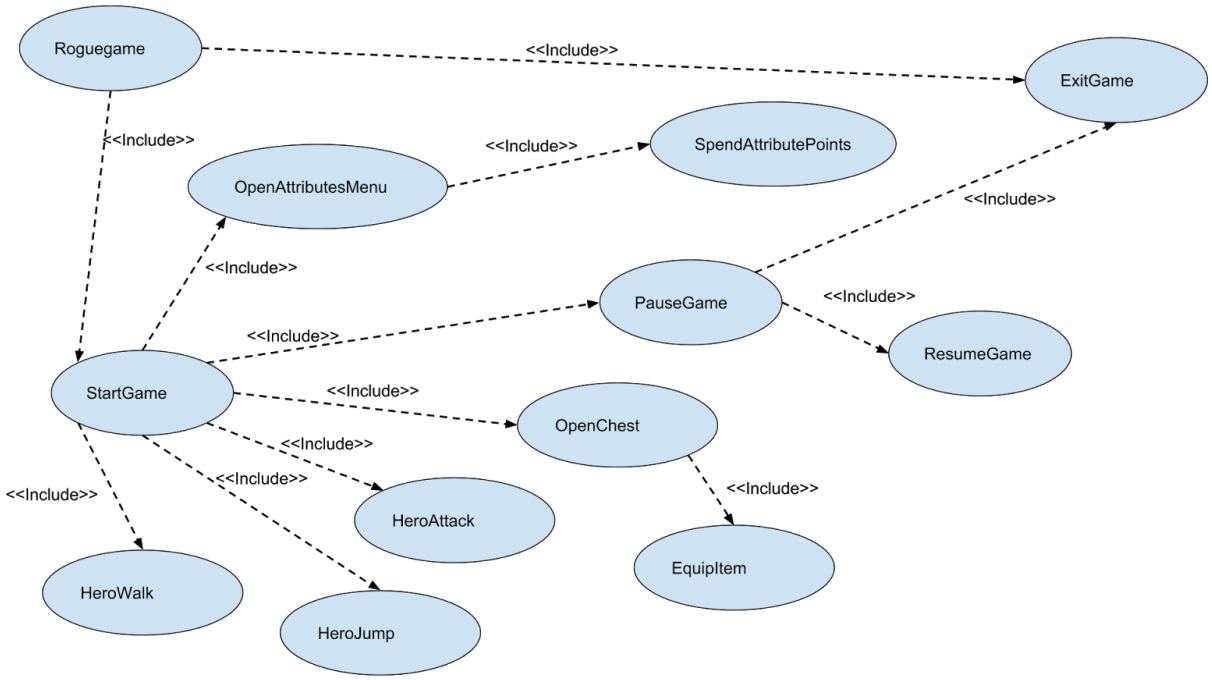
User interface
At startup or after a game over, the user will be located in the “Main Menu”, a simple GUI with some kind of background displaying the name of the game. There will also be a few clear buttons like “Start Game”, “Exit Game” and possibly “Settings”. When playing the game there will be some kind of interface showing the player what items he/she has equipped. Health will be displayed either by a “bar of health” or with other visual feedback (eg. the character getting bloody or the entire screen flashing red).

On the top of the screen is a map display showing in what room the hero is located, and what rooms have already been visited.

References

Appendix

User case model



User cases

Use Case: HeroWalk

Summary: This is how the player character is made to walk left/right.

Priority: High

Extends: -

Includes: -

Participators: Player

Normal flow of events

	Actor	System
1	Presses left/right key	
2		Moves the player character in the left/right direction. Scrolls the screen if the current room is big enough.

Alternate flows

Flow 2.1 Player character faces an adjacent wall.

	Actor	System
1		The player character remains still or is stopped if moving.

Flow 2.2 Player character faces an adjacent enemy character.

	Actor	System
1		The player character remains still or is stopped if moving. Player takes damage.

Flow 2.3 Player character is dead.

	Actor	System
1		The player character remains still.

Analysis model