Requirements and Analysis Document for: Bravely Segfault

Version: 1.1
Date: 2016-05-29
Authors:

- Philip Tham
- Mikael Ragnhult
- Anton Josefsson
- Stefan Fritzon

1 Introduction

This document describes the development of the game Bravely Segfault.

1.1 Purpose of the Application

The application is an old-school style role playing game (RPG). We will build a basic framework for the functionality of such a game, and create the specific application in that framework. The game is aimed at students of the Computer Science program at GU, in the sense that the story and characters will be based off our experiences from attending said program.

1.2 General Characteristics of the Application

The application will be a computer system, standalone (non-networked), single-player application with a graphical user interface for Windows/Mac/Linux platforms.
The player will be able to control a character in a world by moving around in a movement system on a square-based grid. The player character will be able to interact with other non-playable characters and interactive objects.
The game will be story-based, meaning that it will be divided into multiple chapters. The player will be able to continue through the story by fulfilling certain criteria.
There will be an inventory system where the player can collect so-called items. These items are used to trigger specific interactions that moves the story forward. Specific items can also be equipped by the player to amplify their battle-power.
The game will have a battle system, where the player can fight against enemies in a turn-based manner. The player can make choices as battle actions to interact with and defeat enemies.

1.3 Scope of the Application

This application will mainly be developed to be used on desktop computers.
The main focus is world interaction between the player character and other characters/objects. As such, the battle system will have low priority and will only be implemented if there is time.

1.4 Objectives and Success Criteria of the Project

The project will be considered to be successful when a player is able to play and complete a full chapter of the game story. This includes interaction between the player character and other world objects/characters, as well as item interaction to proceed through the story until the first chapter is completed.

1.5 Definitions, Acronyms, and Abbreviations (wordlist)

RPG	-	Role Playing Game
Inventory	-	A list of game items that a player has access to
World objects	-	Part of the world that you cannot walk on (e.g trees, walls)
NPC	-	Non-playable character
Battleground	-	An area where the character fights enemies
Consumable	-	Items that decrease in amount when used
Equipment	-	Items that can be attached to a player character
Key Item	-	Items that unlocks story progression or other interactions
Framerate	-	The rate at which the application updates

2 Requirements

2.1 Functional Requirements

The player can start a new game.
The player can save a game.
The player can load a game.
The player can move in the grid.
The player can interact/talk with NPCs.
The player can progress the story by interacting with NPCs
The player can fight and defeat enemies.
The player can manage their inventory.

2.2 Non-functional Requirements

2.2.1 Usability

The game should display a guide that instructs the user on button functionality.

There should be an English manual online if the user forgets the button layout.

2.2.2 Reliability

The game should not crash, and should be bug-free.

2.2.3 Performance

The game should run at 60 frames per second at all times on modern hardware. The game should be runnable at interactive framerate on older hardware.

2.2.4 Supportability

The game should not perform differently regardless to the running platform.

2.2.5 Testability

There should be a test suite for non-GUI functionality. These tests should be built from the ground up, starting with the smallest cases (so-called unit tests).

2.2.6 Implementation

As the game will be implemented in Java, the player must have a JRE installed to run it.

2.2.7 Packaging and Installation

The game requires a computer with Java installed.
The game does not require any installation. Instead it will be run as an executable file.

2.2.8 Ethical Concerns

This game will feature characters based on real-life people, as such we will need to seek permission from those people to feature them in the game.

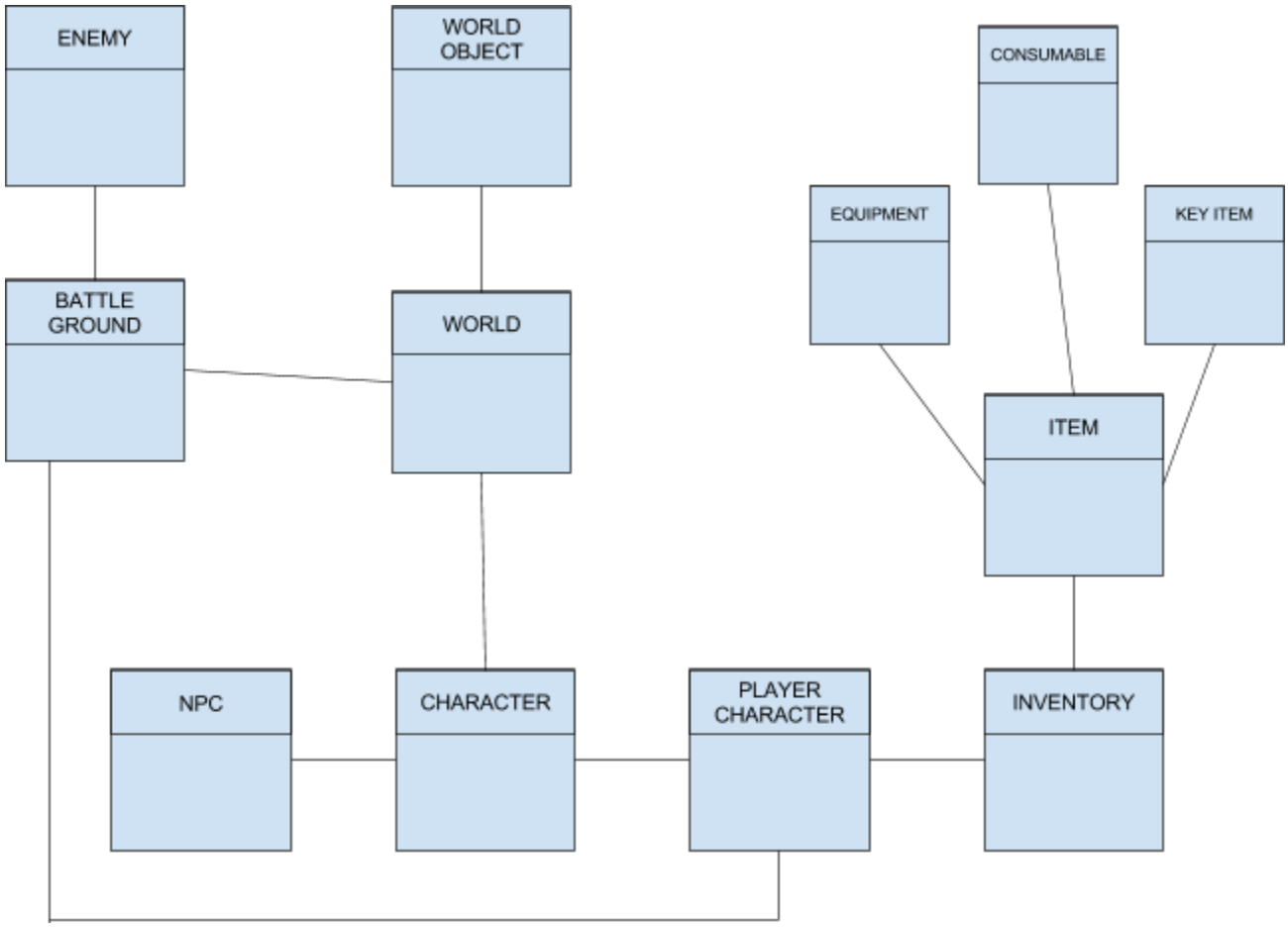
2.3 Application models

Use case model - list of Use case names (text for all in appendix)
Use cases priorities

2.4 References

APPENDIX

Domain model



Use cases

1. Move
   Summary: This is how the player moves their character in the game world
   Priority: High
   Extends: -
   Includes: -
   Participants: The player

   Normal flow (no world object in movement direction)

	Actor	System
1	The player inputs movement direction button	
2		Moves the player character 1 step in the chosen direction
3		Disables movement for a small amount of time

   Alternative flow (world object obstructs movement)

	Actor	System
2.1 World object obstructs movement		Disables movement for a small amount of time

2. New game
   Summary: This is how the player starts a new game
   Priority: High
   Extends: Main Menu
   Includes: -

Normal flow

	Actor	System
1	The player selects the	

	new game alternative from the main menu	
2	The player confirms their choice with the appropriate button	
3		The system starts a new game, loading the game world from the appropriate file and putting the character in the starting position

3. Start interaction

Summary: This is how the player starts an interaction with an NPC
Priority: High
Extends: Move
Includes: Dialogue
Participants: The player

Normal flow (NPC in front of character)

	Actor	System
1	The player moves their character to stand in front of an NPC See Move	
2	The player presses the interact button	
3		The system starts a Dialogue with the NPC in front of the character See Dialogue

4. Dialogue

Summary: This is how the player interact with NPC's dialogue system
Priority: High
Extends: -
Includes: Inventory
Participants: The player

Normal flow (interact with a NPC)

Step	Actor	System
1		A dialogue window starts
2		Gives a list of answers to select
3	Selects an answer with arrows and confirm	
4		Continues the dialogue

Alternative flow

Step	Actor	System
4.1 End of dialogue		Closes the dialogue
4.2 Effect on the game world		Change some of the world objects in some way
4.3.1 Receiving an item		Displays the item being received
4.3.2		The item is added to your inventory
4.3.3		Continues the dialogue (step 4)
4.4.1 Give an item		Dialogue present the option to give a certain item
4.4.2	Selects to give an item	
4.4.3		Remove the item from player inventory and go to 4.2 Effect on the game world

5. Save Game
   Summary: This is how the player saves a game to a file
   Priority: Medium
   Extends: Interacting with NPC/Dialogue?

Includes: -
Participants: The player

Normal flow

	Actor	System
1	The player starts an interaction with the specific NPC that allows you to save the game	
2		The system starts a Dialogue that includes the option to Save Game
3	The player selects the Save Game option in the Dialogue	
4		The system writes a file to a specific location containing information about the current game
5		The system closes the dialogue

6. Load game
Summary: This is how the player starts a new game
Priority: Medium
Extends: Main Menu
Includes: -

Normal flow

	Actor	System
1	The player selects the load game alternative from the main menu	
2		Give a list with load

		files
3	The player selects one load file	
4		The system starts the game, loading the game world from the appropriate file and putting the character in the given position

7. Battle
   Summary: How the player battles in the battleground.
   Priority: Low
   Extends: -
   Includes: Enemy, Player
   Participants: Player

Normal flow

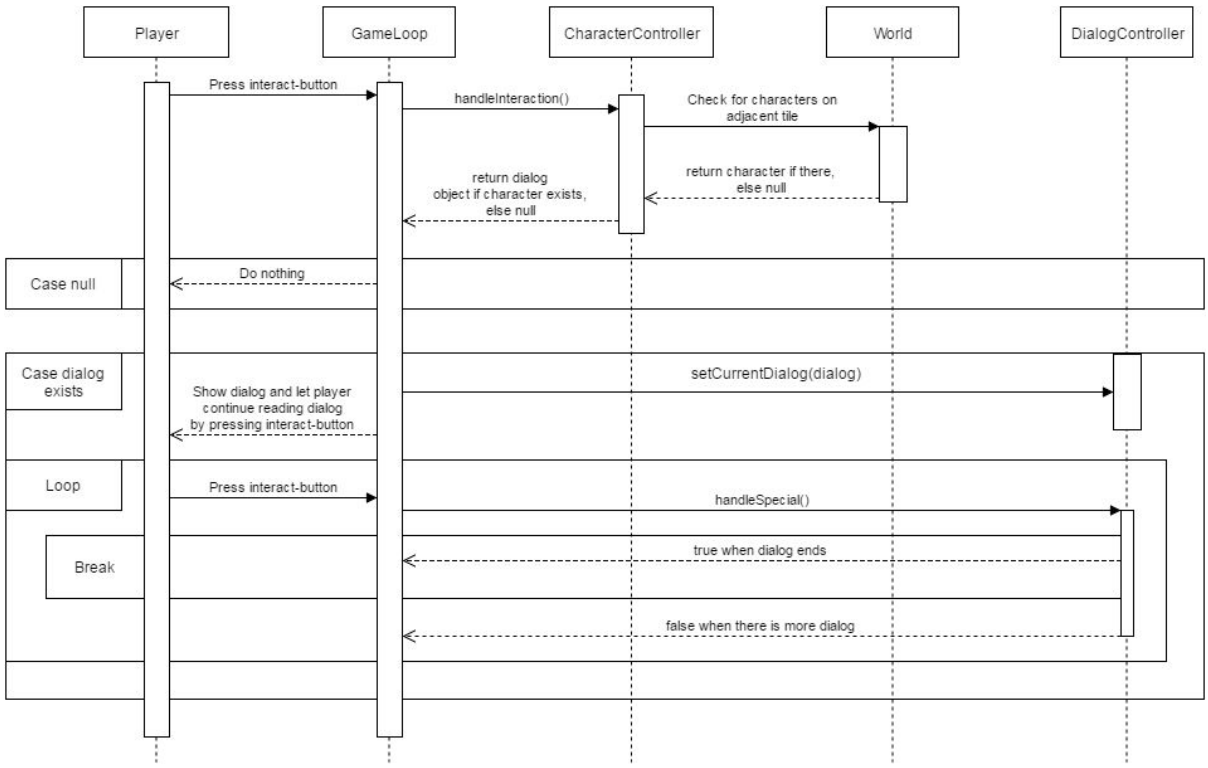
Step	Actor	System
1	Player walks on a grid with battleground effect	
2		The battleground screen opens
3	Player choice a attack	
4		The player misses or hits and recalculate enemy health
5		The enemy misses or hits and recalculate player health
6. Go to step 3		

Alternative flow

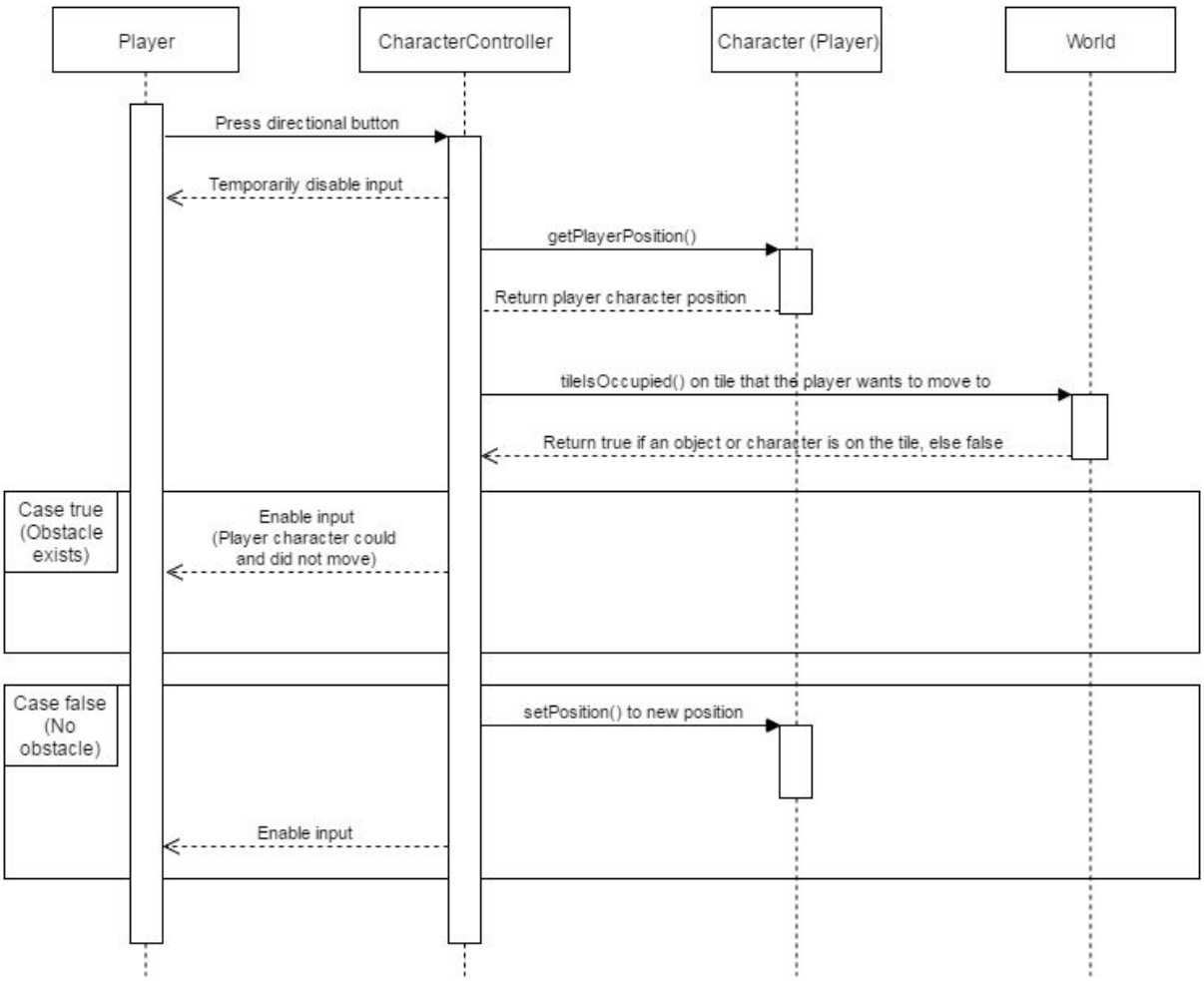
Step	Actor	System
3.1.1	Player choice a item	
3.1.2		Change appropriate values depend on the item.
3.1.3		Enemy hits or misses
6.1 Player health ≤ 0		Dialog screen: “Game over!”
6.2.1 Enemy health ≤ 0		Dialog screen: “Winner”
6.2.2		Experience added
6.2.3		Battleground ends

Sequence Diagrams

Interact Sequence Diagram



Move sequence diagram


Player
CharacterController
Character (Player)
World
Press directional button
Temporarily disable input
getPlayerPosition()
Return player character position
tileIsOccupied() on tile that the player wants to move to
Return true if an object or character is on the tile, else false
Case true (Obstacle exists)
Enable input (Player character could and did not move)
Case false (No obstacle)
setPosition() to new position
Enable input