Requirements and Analysis Document for KeyboardChaos (RAD)

Contents

Version: 1.14

Date: 26-05-2015

Author: Alexander Kloutschek, Daniel Wassbjer, Kristoffer Knutsson, Poya Kaboteh

1. Introduction

1.1 Purpose of application

The purpose of the application is to entertain the user, with a local multiplayer game played on a single keyboard.

1.2 General characteristics of application

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

The application will be a round based local multiplayer game with support for 2-4 players. The player who has gathered the most points during a certain amount of rounds will be declared the winner. The default round setting will be time constrained but this can be modified by the users.

1.3 Scope of application

The application will not include an AI (computer controlled) player and will not support solo play (playing alone). The application will not save any interrupted games nor collect any statistics of any form. It will however save the player's controllers used for movement and firing spells.

1.4 Objectives and success criteria of the project

1. 2-4 players should be able to play a full round.

2. The players should be able to customize game controls and the abilities of the chosen character.

1.5 Definitions, acronyms and abbreviations

- GUI, graphical user interface.
- Java, platform independent programming language.
- JRE, the Java Run time Environment.
- AI, artificial intelligence. A computer controlled player.
- Match, a complete set of rounds.
- Round, finite game session where the players gather points.
- Player, the human controlling a character in the game.
- Character, object in game that represents a player and is controlled by a player.
- Spell, the player gets to choose 2 spells to his character. These spells will either be a form of an attack, defense or will be used as utility for the player(like affecting the players position or movement).

- Cooldown, each spell comes with a cooldown. The cooldown is basically a timer in which the spell will be unusable and the player will have to wait for the cooldown to run out before being able to use that spell again. The cooldown will vary between the different spells.
- Fireball, a spell that lets the character shoot out a ball of fire that will damage other players.
- Iceball, a spell that lets the character shoot out a ball of ice that will damage other players.
- Blink, a spell that will teleport the player a short distance.
- Beam,
- Explosion, a spell that will deal damage in a radius around the character that casts the spell.
- Shield, a spell that will shield the character from incoming damage for a short duration.

2. Requirements

2.1 Functional requirements

The players should be able to:

1. Bind their preferred keys to play with.
2. Start a match.
   a) Select spells to start with.
3. Play a round.
   a) Use the two selected spells.
   b) Kill other players' characters to gain points.
   c) Move around their characters on the playing field.
4. Be damaged over time while standing in lava
5. A round should end when one player is remaining.
6. Exit the application. Ends round and match.

2.2 Non-functional requirements

NA

2.2.1 Usability

The game is created for short, high-paced games and therefore it should also be quick and easy for a regular user to go from launch to ready-to-play. The pace of the game also demands a low response time.
Testing with four average computer users should be conducted to assure that the game allows the high pace that is intended.

2.2.2 Reliability

NA

2.2.3 Performance

Actions initiated by, or a result from, a player interaction should not exceed 50 ms delay as worst case.

2.2.4 Supportability

The application should be ready for extensions in the form of maps and additional spells. The estimated time of adding a map or spell which (code wise) is ready should not exceed one work day.

2.2.5 Implementation

The application will use the Java environment as a way to be runnable regardless of which platform it's being used on. Users need to have JRE installed and configured, and the application downloaded and installed.

2.2.6 Packaging and installation

The application comes as a zip-file containing;

- A jar-file containing the code for the application
- A resource bundle
- Installation guide in the form of a README-file.

2.2.7 Legal

There are no known legal issues regarding concept of game or name.

2.3 Application models

2.3.1 Use case model

See Appendix for diagram and descriptions.

2.3.2 Use cases priority

1. Exit
2. Play
3. Move
4. UseSpell
5. StartMatch
6. SetPlayerControl
7. SetPlayerSettings

8. SetSpells
9. SetPlayerName
10. SetGameSettings
11. SetMap
12. SetGameMode
13. GoToMenu
14. GoBackToGameSettings

2.3.3 Analysis model

See Appendix for UML diagram.

2.3.4 User interface

See Appendix for User Interface.

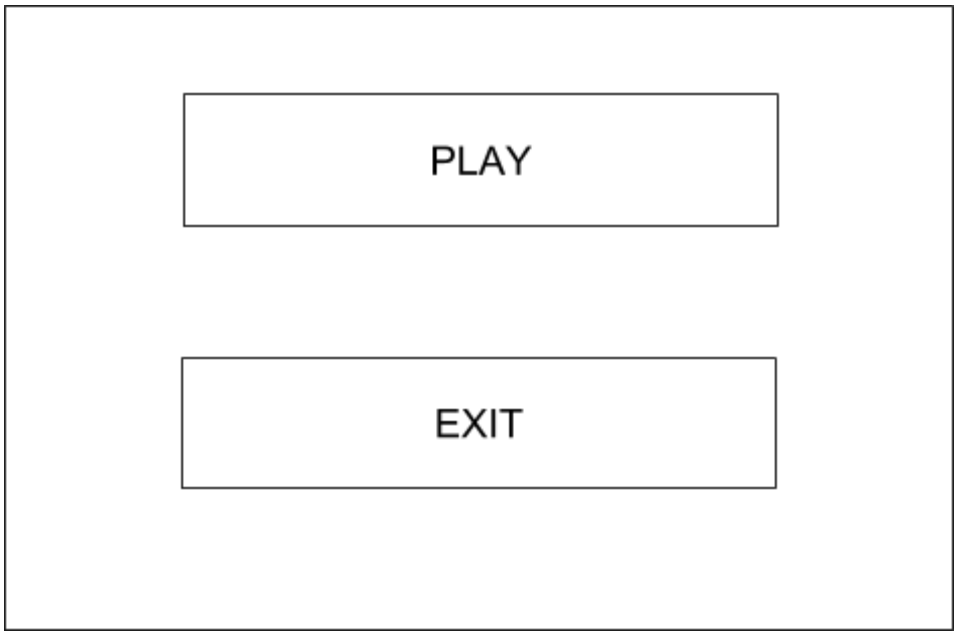
2.4 References

Appendix

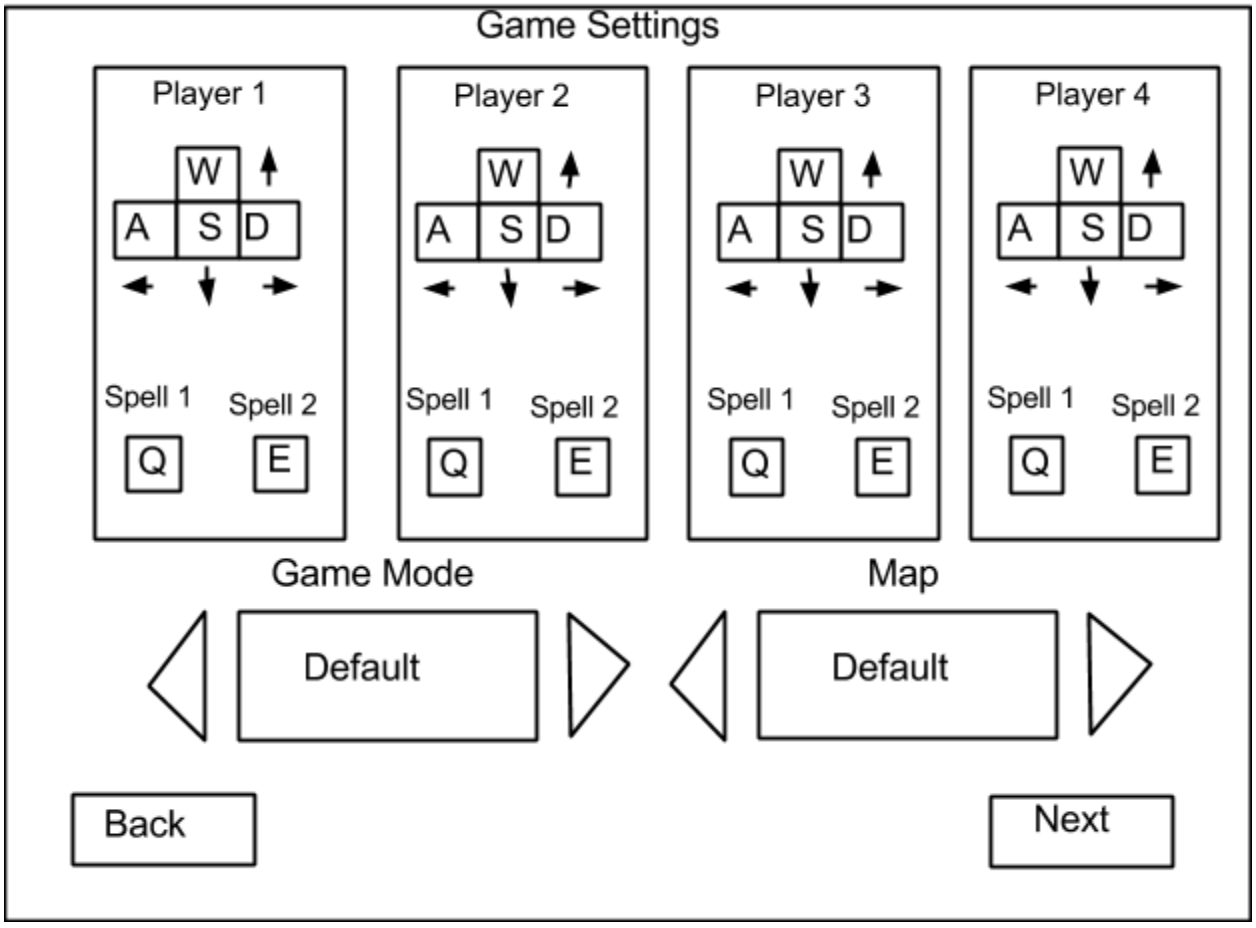
GUI

Preliminary GUI.

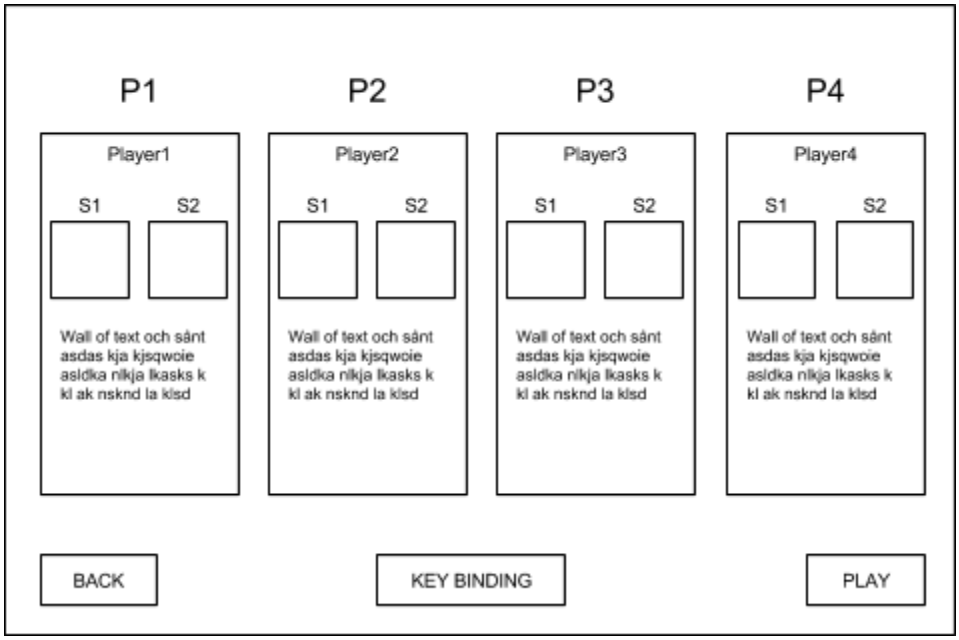
Main menu:



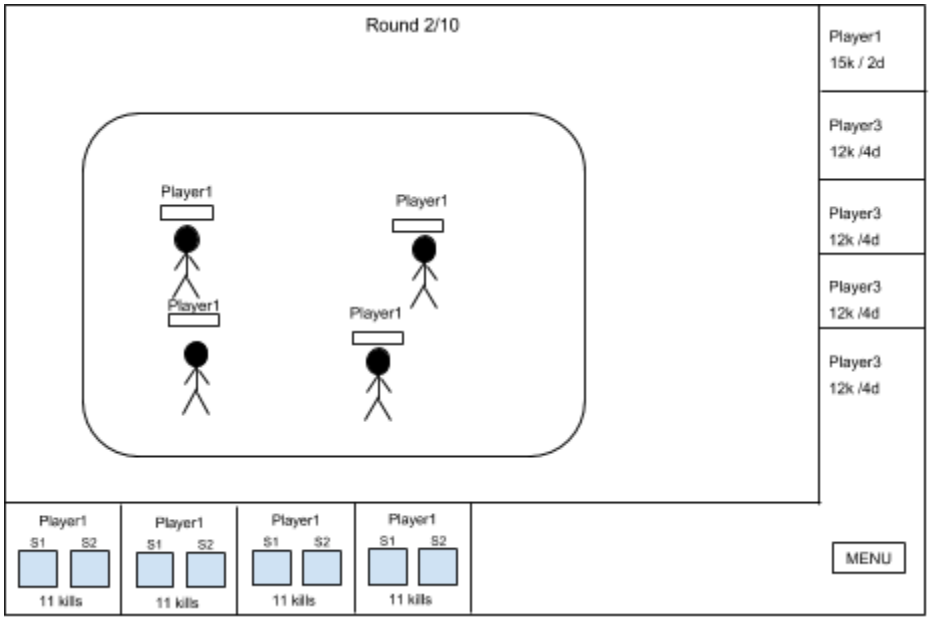
Game settings:



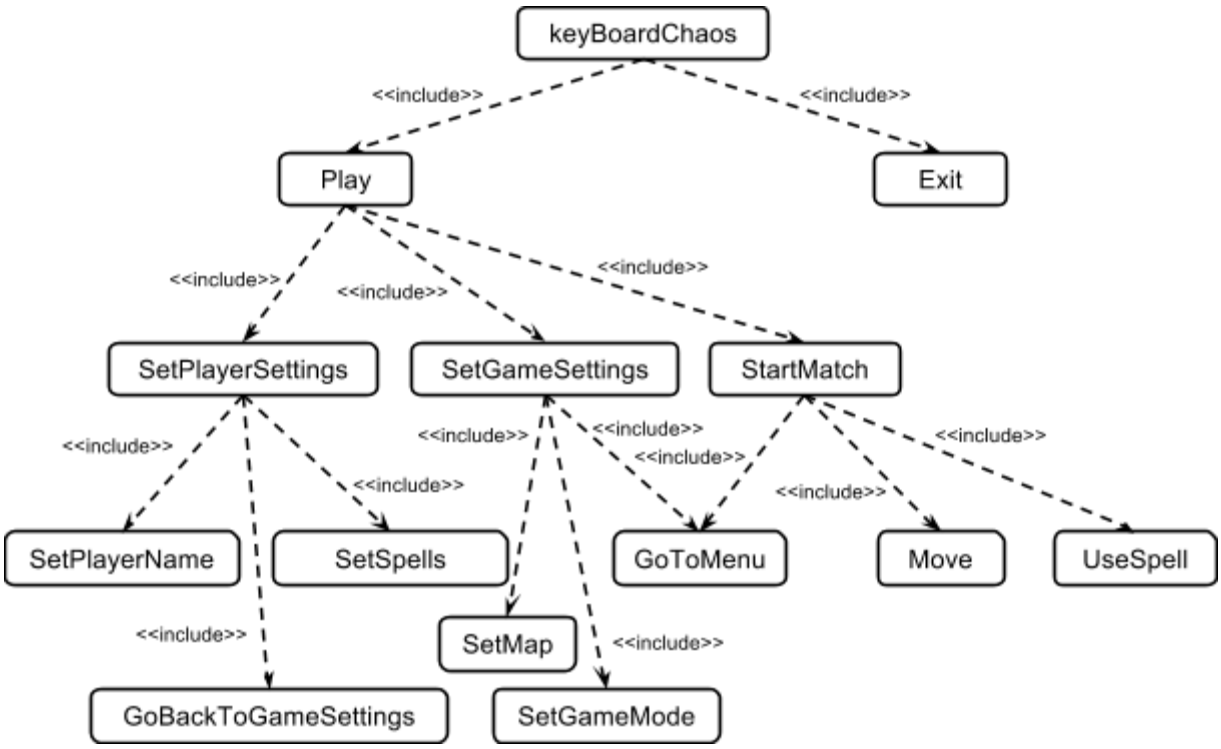
Player settings:



In-game:



Use cases (overview)



Use Case: Exit

Summary: Exits the application
Priority: High

Extends: -
Includes: -
Participators: One of the players, the application
----Pre UC: Start the application

Normal flow of events

	Actor	System
1	User presses the exit button	
2		System closes the application

Use Case: Move

Summary: Moves the player either left, right, up, down or diagonally by pressing one or more of the keys controlling the player's movement.
Priority: High
Extends: -
Includes: -
Participators: One of the players
----Pre UC: Play, StartMatch

Normal flow of events

	Actor	System
1	The player presses one or more of his/her directional keys.	
2		Applies a force to a player's in-game character in it's direction.
3	The player releases all his/her directional keys.	
4		The player's in-game character slowly stops.

Alternative flow

Flow 2.1.1 - Player is moving in lava

	Actor	System
2.1.1		A force is applied to a players character, a contact with lava as been detected.
2.1.2		Player's character loses a certain amount of health at an even time interval.

Flow 4.1.1 - Player stops in lava

	Actor	System
4.1.1		Refer to *4*.
4.1.2		Refer to *2.1.2*.

Flow 2.1.2.1 - Player dies in lava

	Actor	System
2.1.2.1		The player's character dies and disappears from the screen.

Flow 3.1.1 - Player collides with a solid object (such as a wall or a player)

	Actor	System
3.1.1	Player presses one or more keys towards a solid object, such as a player or a boundary "wall".	

Flow 3.2 - Player releases one or more keys, still keeping one or more keys pressed

	Actor	System
3.2.1	The player releases one or more of the directional keys, still keeping one or more key pressed.	
4.1.1		The player's in-game character keeps

		moving, but the force is applied in the direction which the player is facing.

Use Case: UseSpell

Summary: The character uses an equipped spell.
Priority: High
Extends:
Includes:
Participators: One of the players that presses either of his assigned spell buttons
----Pre UC: Play, StartMatch

Normal flow of events

	Actor	System
1	The player presses his assigned spell button	
2		The system displays that the player's character fires away a fire ball.
3		The Fireball hits another player and deals damage.

Alternative flow

Flow 3.1 - The player kills an enemy with his spell

	Actor	System
3.1.1		The spell that the player has casted kills an enemy. The system will display that the player that casted the spell has gained a point.

3.1.2		The killed player's character is removed from the screen.

Flow 2.1 - The player misses his shot

	Actor	System
2.1.1		The system displays that the player's character fires away a fire ball. The Fireball doesn't hit anything and is removed after a specified time.

Flow 2.2 - The player has chosen blink as his spell

	Actor	System
2.2.1		The system moves the player's character a certain amount of tiles in the direction that the character is facing.

Flow 2.2.1 - The player has chosen blink as his spell and teleports onto a physical object

	Actor	System
2.2.1.1		The system tries to instantly move the player's character's position to an accepted position.

Flow 2.4 - The player has chosen beam as his spell

	Actor	System

2.4.1		The system tries to instantly move the player's character's position to a tile closer to the player until either a valid tile is chosen or the player will simply not teleport.

Flow 2.5 - The player has chosen explosion as his spell

	Actor	System
2.5.1		The system displays an explosion effect around the player's character with a fixed radius, dealing damage to the other players within the explosion radius. May invoke *3.1*.

Flow 2.6 - The player has chosen shield as his spell

	Actor	System
2.6.1		The system displays a shield around the player and a timer that indicates for how long the shield will be up. During this time the player will be invulnerable and can't take damage. Eventually the shield will time out and disappear.

Flow 2.7 - The player has chosen ice ball as his spell

	Actor	System
2.7.1		The system displays that the player's character fires away a ice ball.

2.7.2		The ice ball hits another player and deals damage. May invoke *3.1*.

Flow 2.7.1 - The ice ball misses

	Actor	System
2.7.1.1		The system displays that the player's character fires away a ice ball. The ice ball doesn't hit anything and is removed after a specified time.

Use Case: Play

Summary: One of the players presses the play button.
Priority: High
Extends: -
Includes: SetPlayerSettings, SetGameSettings, StartMatch
Participators: One of the players, the application
----Pre UC: Start the application

Normal flow of events

	Actor	System
1	User presses the play button.	
2		The System displays a screen where the users can change their settings such as key bindings.

Analysis model

Preliminary analysis model.