Requirement and Analysis Document for 4Students

Version: 1.6

Date: 2017-05-25

Authors: Niklas Baerveldt, Erik Lundin, Lucas Ruud

1 Introduction

We are creating a single player 2D-sidescroller. This application is intended as a fun game with a new experience every time you play. The application should be easily started from a computer when you have a break or a small amount of time to spend.

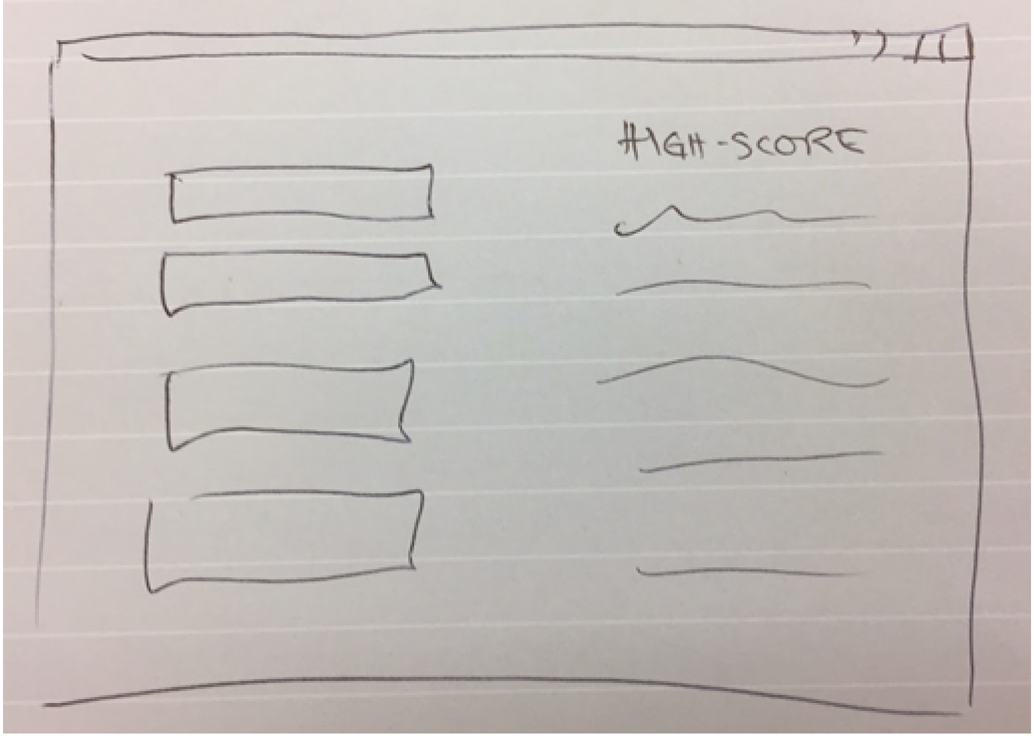
In the game, your character will need to survive and keep moving along the level for as long as possible. The level behind the player will disappear so the player needs to keep moving or he will fall of the map and die. The player should be able to move horizontally, jump vertically and attack his enemies. The enemies will be AI controlled and attempt to hinder the player from making progress. The levels will be generated by the computer and consist of platforms and obstacles. It will be possible to affect the generation of the level by collecting powerups.
The player will have a score which will be acquired by interaction with the game world.

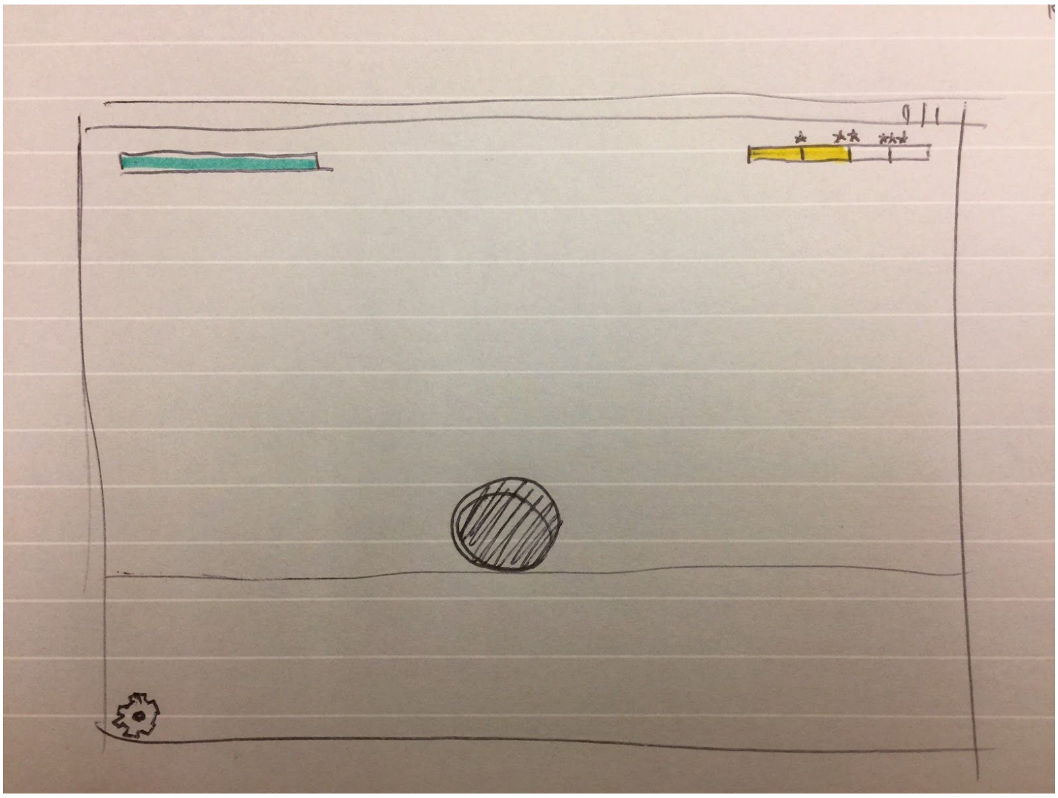
1.2 Definitions, acronyms and abbreviations

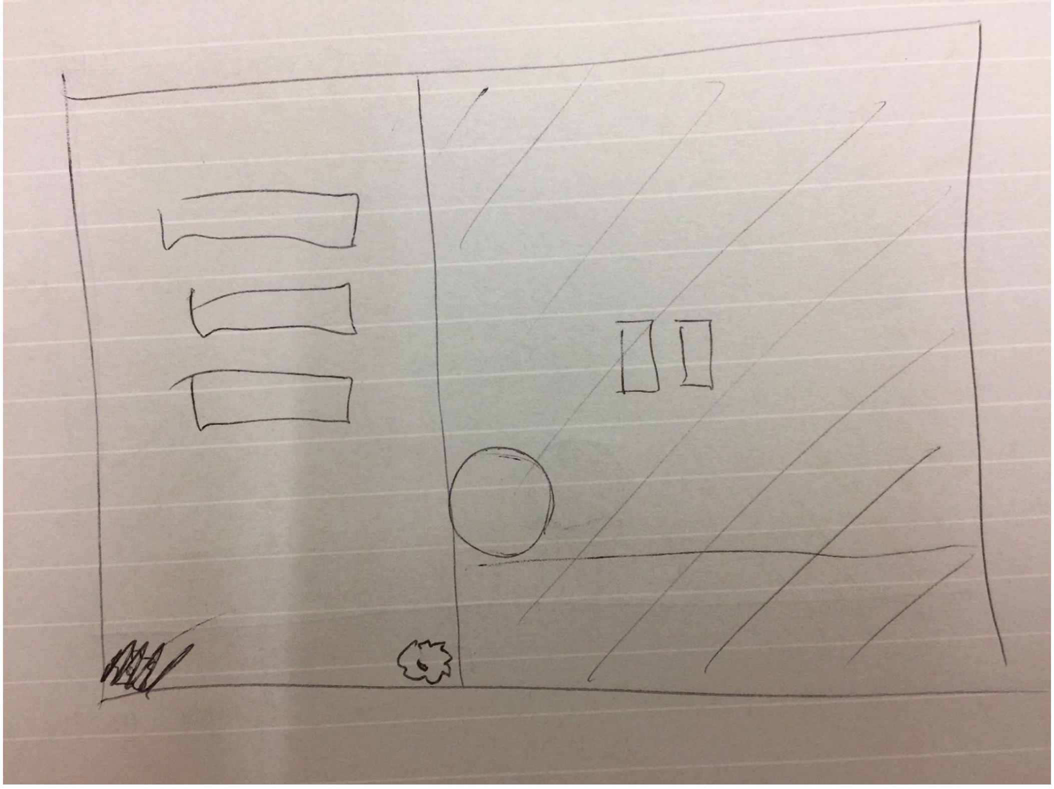
Player: The entity controlled by the user
Enemy: An AI-controlled entity within the game
Power-up: A game changing effect.
Obstacle: An object interfering with the players ability to continue.
Health: A measurement of the player's life, if depleted the player dies
Projectile: A representation of something the player shoots, deals damage to enemies.
User: The person using the application.

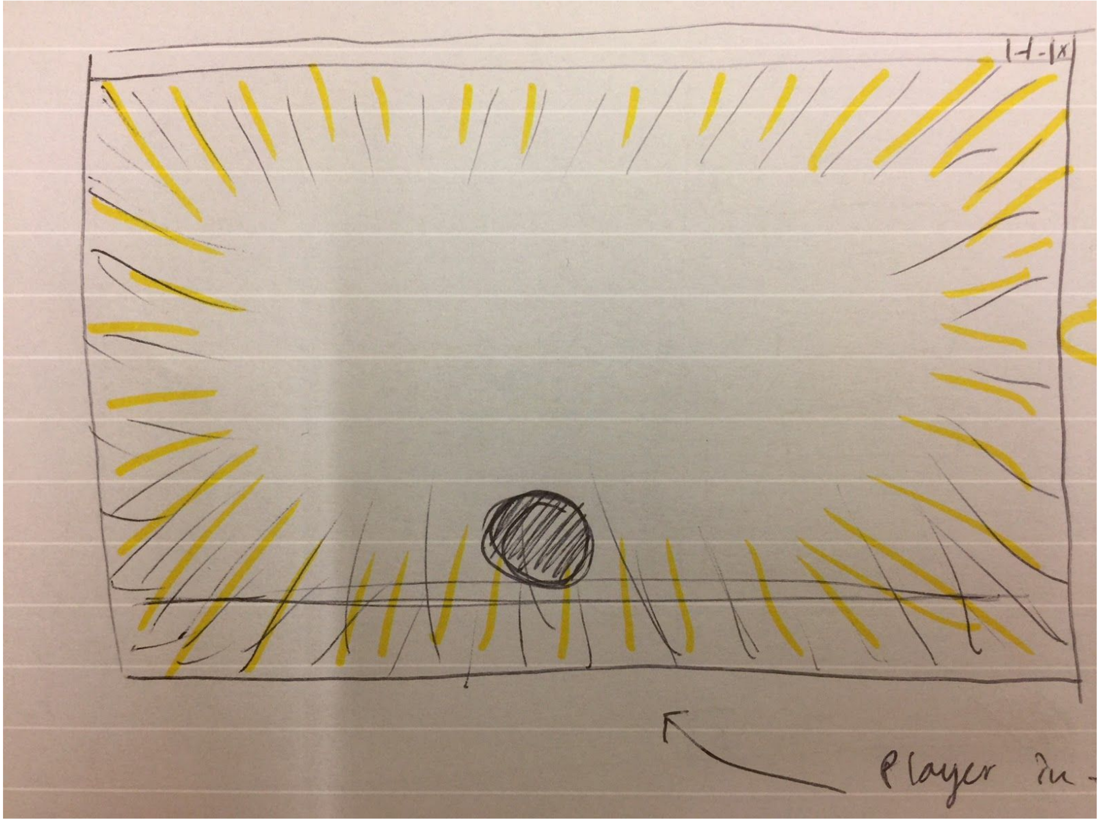
2 Requirements

2.1 User interface



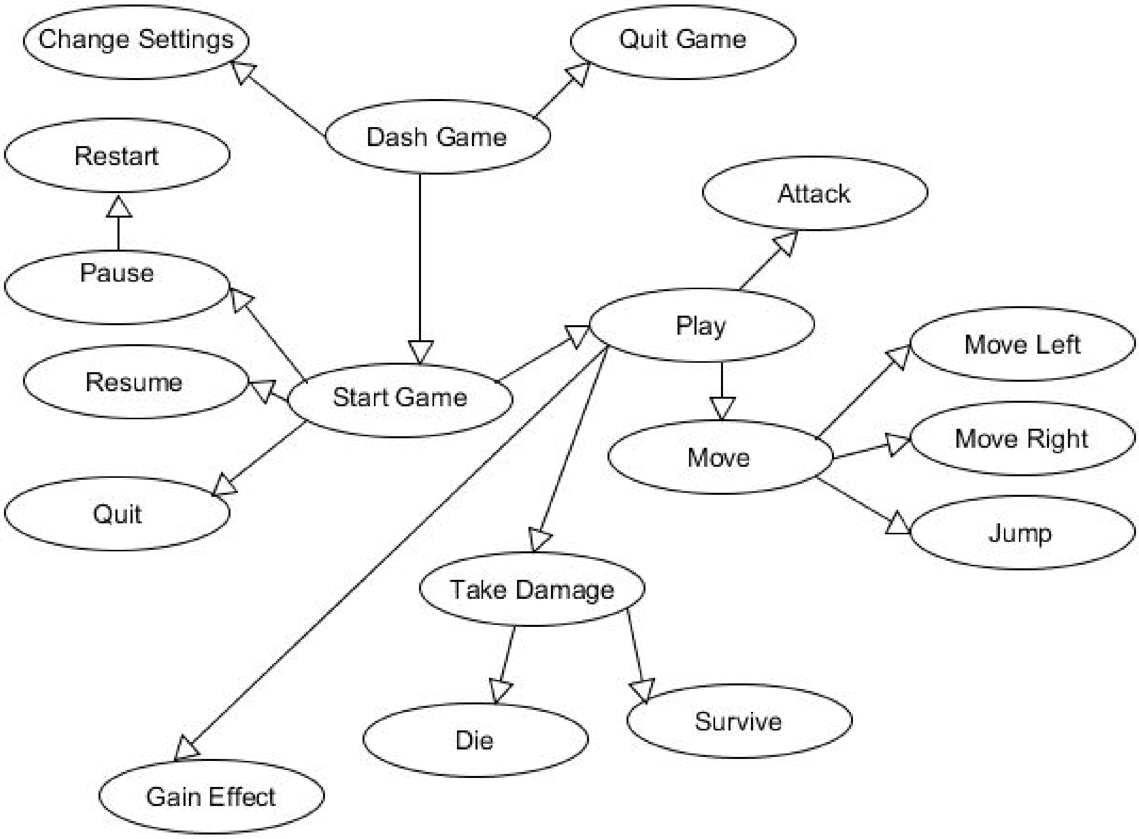






|-|-|x|
Player in -

2.2 Functional requirements



The user will be able to:

- Start game
- Quit the game
- Edit settings
- Move
    - Horizontally
    - Jump vertically
- Attack
- Acquire powerups
- Acquire points
- Pause game
- Take damage

2.3 Non-functional requirements

- The program should be able to perform with a framerate of at least 25 fps.
- The program should be testable.
- The level will be randomly generated
- The player should be able to move to the right infinitely

3 Use cases

3.1 Use case listing

Use case: rangedAttack
Summary: The user presses a button in order to perform an attack
Priority: Medium
Extends: Attack
Includes:
Participators: Player

Normal Flow of Events:

	Actor	System
1	Presses the button for ranged attacks	
2.1 Player misses		Miss-ranged-animation plays
2.2 Player hits an enemy who dies		Hit-animation plays Enemy-die animation plays Enemy disappear from the level
2.3 Player hits an enemy who lives on		Hit-animation plays Enemy-hit animation plays

Use case: GainEffect

Summary: The player gains an effect, the map gets modified

Priority: Medium

Extends: Move

Includes:

Participants: player

Normal Flow of Events:

the player moves over an EffectIcon

	Actor	System
1	the player moves over an EffectIcon	
2		the EffectIcon disappears
3		The map Changes
		An EffectAnimation is displayed on the screen

Use case: Pause

Summary: The presses the pause button

Priority: Medium

Extends: Play

Includes: QuitToMenu, Resume, Restart (Only represented here)

Participants: player

Normal Flow of Events:

The player presses the “pause”-button and chooses to resume.

	Actor	System
1	The player presses “pause”-button	
2		Pause menu is drawn with the buttons “Resume”, “Restart” and “Quit to Menu”
3	Player presses the “Resume”-button	
4		The pause menu disappears, The game starts again.

Alternate Flow:

The player presses the “pause”-button and chooses to restart.

	Actor	System
1	The player presses “pause”-button	
2		Pause menu is drawn with the buttons “Resume”, “Restart” and “Quit to Menu”
3	Player presses the “Restart”-button	

4		The pause menu disappears, The map is drawn to starting point. The enemies respawns

Alternate Flow:

The player presses the “pause”-button and chooses to quit to menu.

	Actor	System
1	The player presses “pause”-button	
2		Pause menu is drawn with the buttons “Resume”, “Restart” and “Quit to Menu”
3	Player presses the “Quit to Menu”-button	
4		The pause menu disappears, The main menu is drawn

Use case: goToSettings

Summary: The user wants to change the settings of the game

Priority: Low

Extends: Game

Includes: goToSettings

Participators: Player

Normal Flow of Events:

	Actor	System
1	Presses button for settings	
2		Changes menu to settings-menu

Use case: changeASetting

Summary: The user wants to change the settings of the game, i.e difficulty

Priority: Low

Extends: goToSettings

Includes:

Participators: Player

Normal Flow of Events:

	Actor	System
1	Presses the button of the setting they want to change	
2.1 The user changes the difficulty		Changes the text of the button to represent the new choice Changes the game's' settings to reflect the new choice
2.2 The user changes a keybinding		The system shows a new dialog that prompts the user to choose a new key for the action chosen
2.2.1	The user presses the desired key	
2.2.1.1 The user pressed a "legal" key		Closes the dialog Changes the text of the button to represent the new choice Changes the game's' settings to reflect the new choice
2.2.1.2 The user pressed a forbidden key		Shows a warning text informing the user to pick a different key The user is back at 2.2
2.2.1.3 The user pressed ESC		Takes the user back to the settings menu

Use case: TakeDamage

Summary: The user takes damage

Priority: High

Extends: Play

Includes: Die, Survive

Participants: The player

Normal Flow of Events:

the player takes damage and survives

	Actor	System
1	The player takes damage	
2		DamageAnimation is played
3		Players stamina is reduced
4		StaminaReducedAnimation is played

Alternate Flow:

The player takes damage and dies

	Actor	System
1	The player takes fatal damage	
2		DeathAnimation is played
3		go to Die

Use case: Move

Summary: The user moves by pressing a “move” button

Priority: High

Extends: Play

Includes: Jump, Move right, Move left

Participators: Player

Normal Flow of Events:

The user moves with no restrictions

	Actor	System
1	The user presses on of the “movement-keys”	
2		Move the player in the designated direction

Alternate Flow:

The user moves to the right

	Actor	System
1	The user presses the “move right”-key	
2		Move the player to the right a set amount of pixels

Alternate Flow:
The user moves to the left

	Actor	System
1	The user presses the “move left”-key	
2		Move the player to the left a set amount of pixels

Alternate Flow: The user jumps

	Actor	System
1	The user presses the “jump”-key	
2		Move the player up a set amount of pixels. Move the player down the same amount of pixels.

Alternate Flow:
The user moves and takes damage

	Actor	System
1	The user presses one of the “movement-keys”	
2		Move the player a set amount of pixels in the designated direction. The player receives damage.
3	See Take damage	See Take damage

Alternate Flow:

The user moves and crosses the finish-line

	Actor	System
1	The user presses one of the “movement-keys”	
2		Move the player a set amount of pixels in the designated direction. The player crosses the finish-line.
3	See Win	See Win

Use case: Quit Game

Summary: The user quits the game from the title screen

Priority: Medium

Extends: Game

Includes:

Participators: Player

Normal Flow of Events:

The user presses the “quit game” button, the game shuts down.

	Actor	System
1	The user presses the “quit game”-button	
2		Save progress. Shut down the game.

Use case: Start Game

Summary: The user starts the game from the title screen

Priority: High

Extends: Game

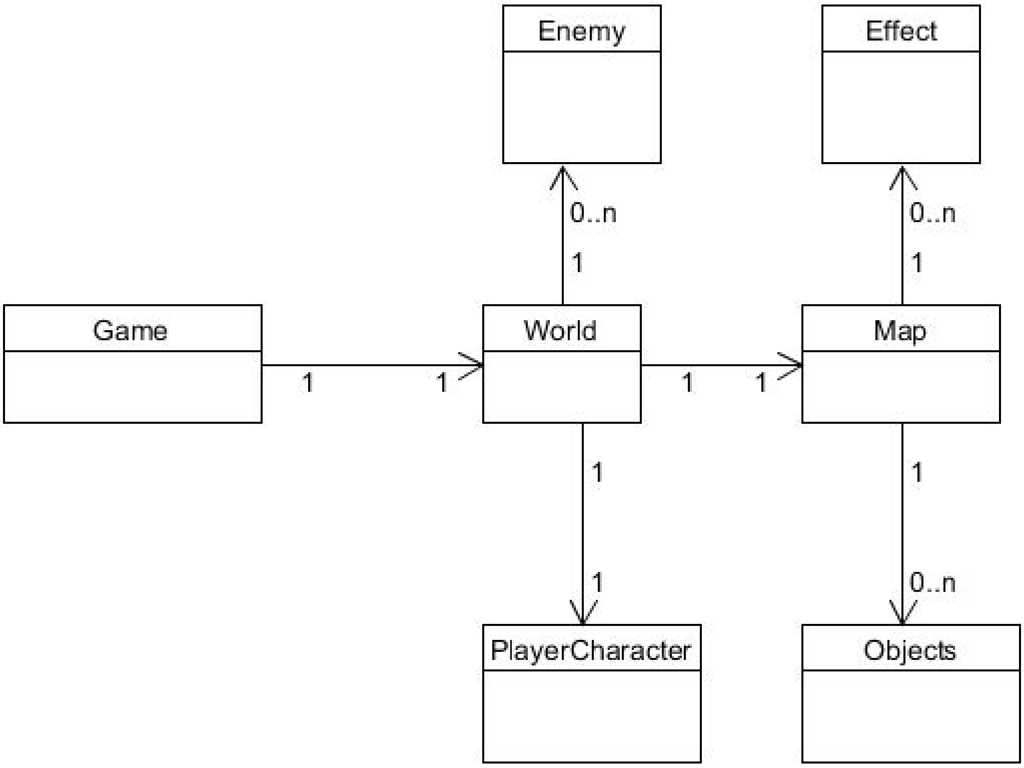
Includes:

Participators: Player

Normal Flow of Events: The user presses the start game button and the game starts.

	Actor	System
1	The user presses the “start game”-button	
2		Display the level select screen.
3	See choose level	See choose level

4 Domain Model



4.1 Class responsibilities

Game: The overall representation of the game.

World: Holds information about the player, enemy and location of powerups

Map: Holds information about the obstacles and what powerup is in effect

Player: Holds information regarding the player character, such as health, position, velocity etc.

Enemy: Enemies placed on the level.

Effect: Something the player can touch to change how the map is generated.