RAD

Blast In The Past

Version: 0.1

Date: 2015-03-26

Authors: Mattias Johansson, Bjarne Gelotte, Jonas Graul Sagdahl, Erik Cervin-Ellqvist

This version overrides all previous versions.

1 Introduction

"Blast in the Past" - A present day soldier gets sent back in time to the Middle ages to fight knights. Top-down shooter with 2D-graphics and enemies with simple AI, power-ups, levels, etc.

1.1 Purpose of application

A game solely made to be fun to play. It should also be a challenge so the user will be proud of him-/herself if they beat it.

1.2 General characteristics of application

The game will be a top-down, single-player 2D-shooter with retro graphics and lots of action. It will take place in the Middle ages, where the user plays as a character that has been sent back in time. There will be different levels and the character will be able to pick up weapons and defeat enemies (medieval warriors). The user will also need to solve simple puzzles in order to progress (activating switches, picking up keys to unlock doors and chests, etc.). The character will have some sort of health-bar and an inventory which will allow the user to switch between weapons or equipment.

1.3 Scope of application

The game should be 10 levels long. Each level should adhere to an overarching medieval theme but also be distinguishable from each other, both graphically and in terms of how the level plays out (difficulty, which enemies appear and the like). The levels should be between 2 and 5 minutes long. Every even numbered level is a boss stage, meaning that the level ends with the user facing a particularly strong enemy or sets of enemies. There should be an inventory system which can be used to manage items and equipment.The game should have 3 difficulty settings: easy, medium and hard. Harder difficulties will increase the amount of health enemies have and decrease the amount of damage the PC can take before the user loses. On harder difficulty settings the enemy AI will be more aggressive towards the character. The game should include som puzzels where the user must find or activate certain objects (levers, buttons) to progress. The game should include power-ups like bullet time and double damage. There should be 5 different normal enemy types and 5 different boss monsters. The levels should have some basic obstacles like rocks and trees. Music and

sound effects should be included in the game. The sound effects should accompany appropriate actions like firing weapons, walking, picking up power-ups and so on.

1.4 Objectives and success criteria of the project

The game should consist of two maps (two levels), on level two there should be a boss encounter. The user should be able to move the character freely in the room where there is no obstacle. Each map/level should have its own weapon, hidden somewhere within the map, which the user should be able to loot. The user should then be able to choose which weapon to equip through an inventory system. When a weapon is equipped, the user should be able to fire it, assuming he/she has ammo.

1.5 Definitions, acronyms and abbreviations

HP - Health Points, signifies the amount of damage a character can absorb before dying.

2D - two dimensions

AI - Artificial Intelligence

Ammo - Ammunition

Boss - A particularly tough and dangerous enemy

Power-up - An item which gives the player certain positive abilities like increased damage, faster movement for example. Expires after a short time (less than a minute).

NPC - Non-Player Character, a character that is not being controlled by the user.

PC - Player Character

GUI - Graphical User Interface

2 Requirements

In this section we specify all requirements.

2.1 Functional requirements

A user should be able to:

1. Move in eight different directions (N, NE, E, SE, S, SW, W, NW)
2. Fire a weapon
3. Die when life-bar is depleted

4. Pick up weapons
5. Switch between weapons
6. Automatically come to the next level once he/she has cleared a level
7. Turn volume on/off
8. Change keybindings
9. Pause the game to open a menu
10. Restart level via in-game menu
11. Exit to main menu via in-game menu
12. Save via in-game menu
13. Close the application
14. Pick up and use power-ups
15. Aim the weapons with a mouse and get a clear indication of where the PC is aiming

2.2 Non-functional requirements

2.2.1 Usability

The application should be easy to use.

2.2.2 Reliability

The application should not crash. Saved data should not be lost.

2.2.3 Performance

The game should run smoothly and be able to be tabbed down and up easily. The application should start fairly fast when executed.

2.2.4 Supportability

Help should always be accessible in the menu. It should be described in a README file how the program can be executed.

2.2.5 Implementation

The code should be well written and well structured.

2.2.6 Packaging and installation

The game will be packaged to a jar-file using Maven and no installation will be needed. A README-file will also be included.

2.2.7 Legal

N/A

2.3 Application models

2.3.1 Use case model

Our detailed use cases include:

1. Fire weapon
2. Loot from chest
3. Pick up power-up
4. Save game
5. Change keybindings

See appendix for UML diagram and textual descriptions.

2.3.2 Use cases priority

1. Fire weapon
2. Loot from chest
3. Pick up power-up
4. Save game
5. Change keybindings

2.3.3 Domain model

See appendix.

2.3.4 User interface

See appendix for sketches.

The in-game UI consists of:

* Heart icons, to symbolize the characters health
* Top-down view of the map with the PC in the center of the screen
* A inventory bar at the bottom of the screen
* An indicator of which weapon is currently equipped and how much ammo you have, in the bottom left corner of the screen

You will aim with the mouse, which will look like a crosshair. The inventory will hide itself when not active, to keep the UI clean.

The main menu will consist of the following items:

* Continue
* New Game
* Load Game
* Options
* Credits
* Exit

The in-game menu will consist of the following items:

* Continue
* Save Game
* Load Game
* Options
* Exit to Main Menu
* Exit to Desktop

The options menu will have two tabs, Sound and Keybindings.

Appendix

GUI

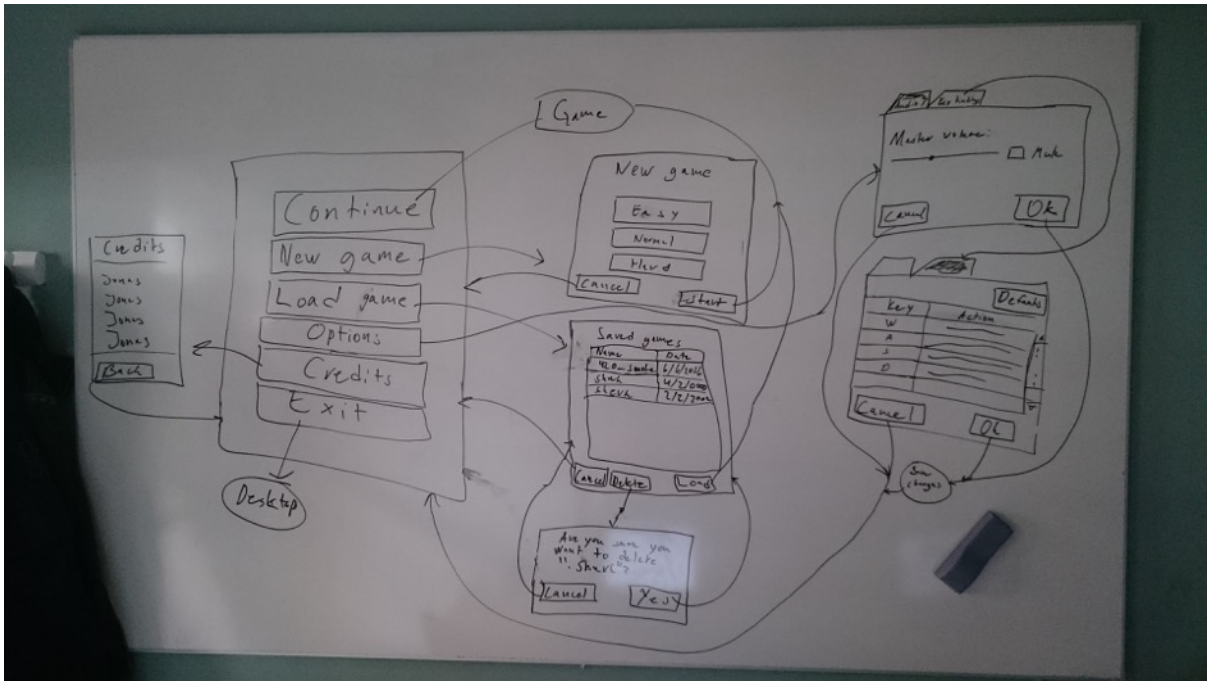

Figure 1. Main menu

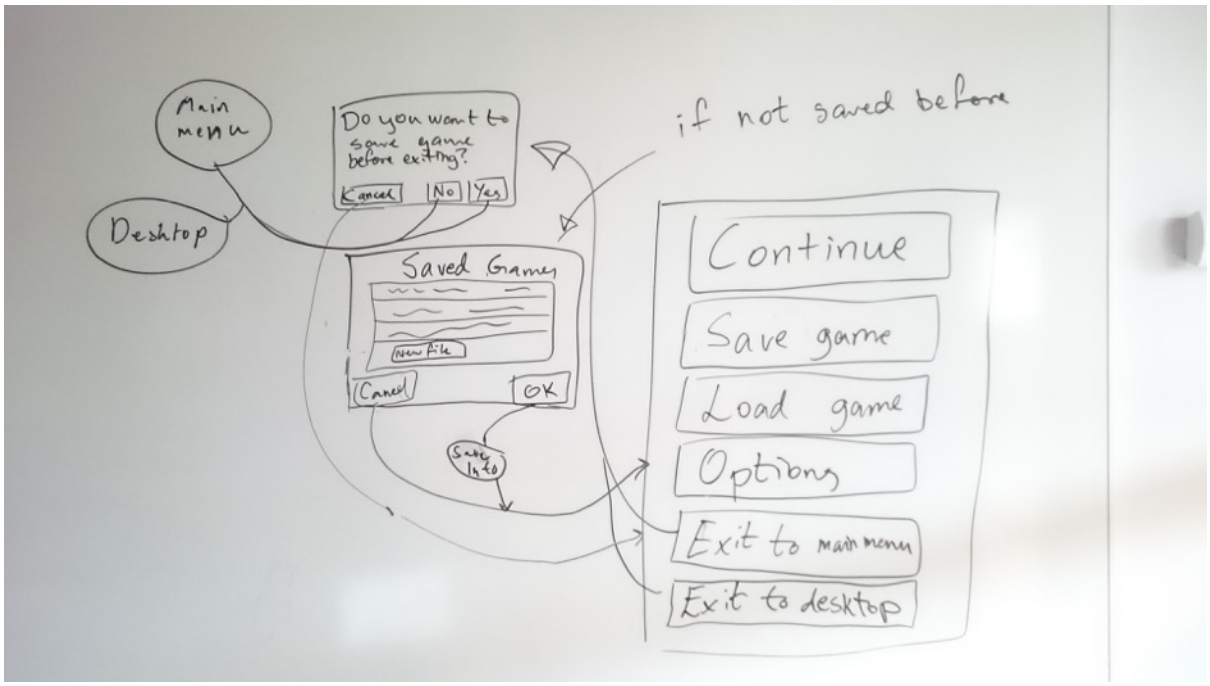

Figure 2. In-game menu

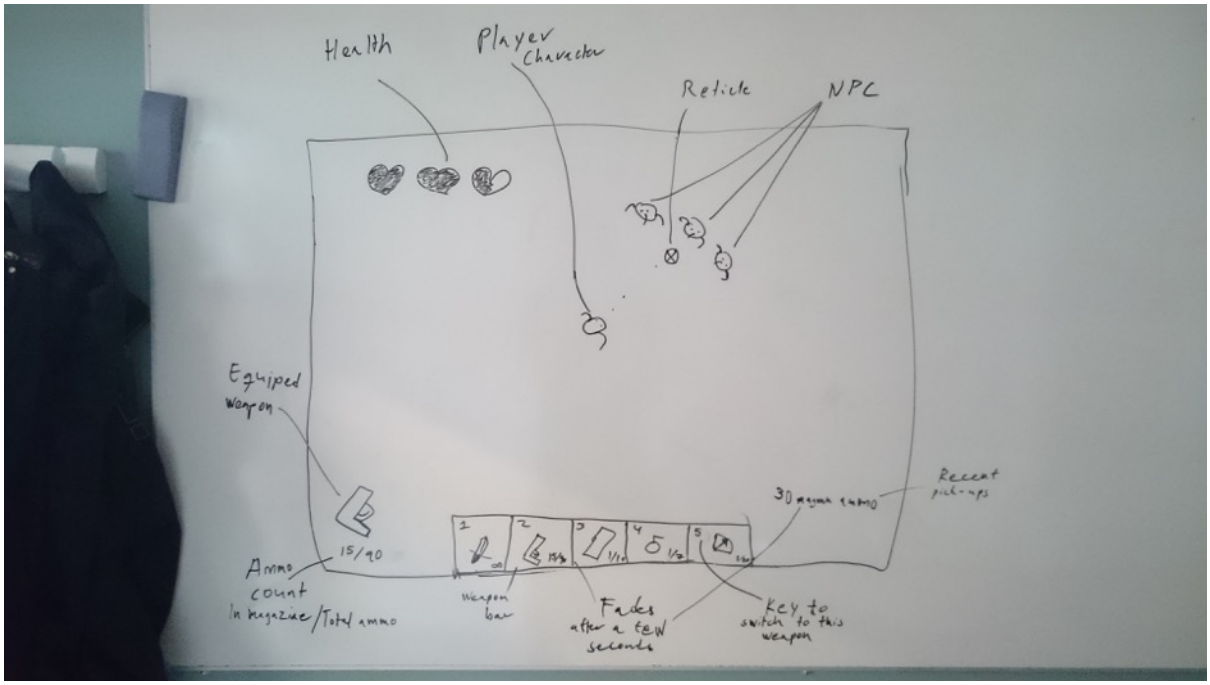

Figure 3. In-game GUI

Domain model

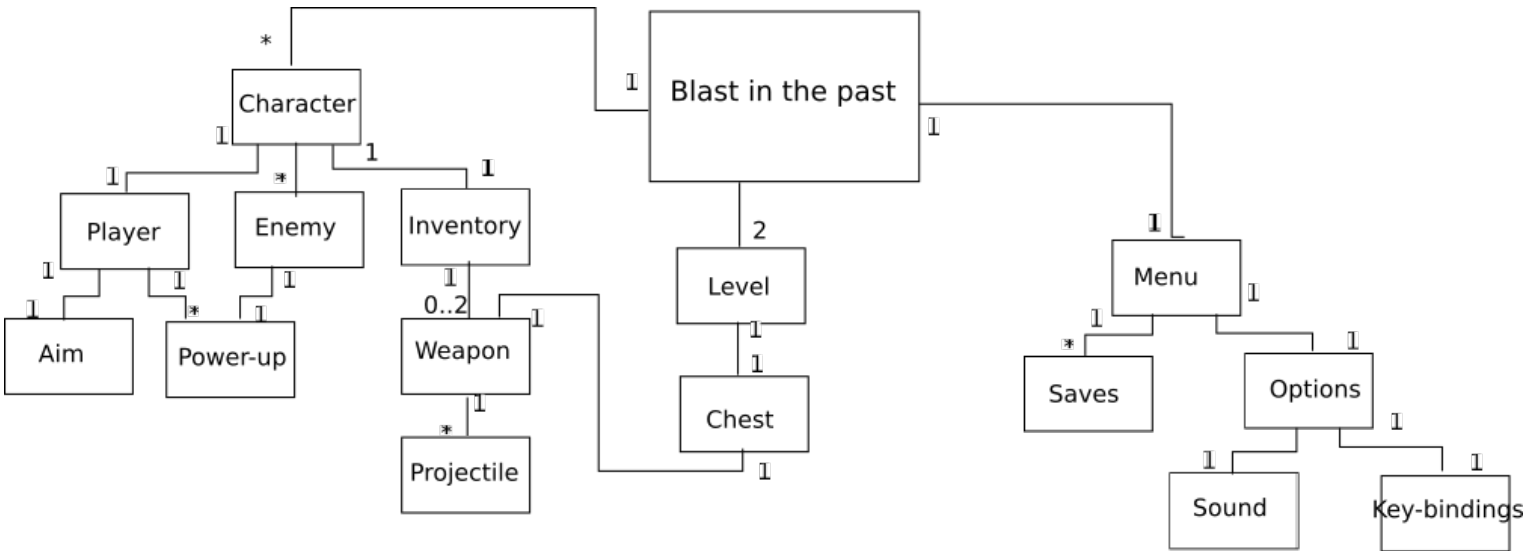

Figure 4. Domain model

Use case texts

Use case: Change keybindings

Summary: The user changes the key binding to move forward from the in-game menu.

Priority: Low

Participators: User and system

Normal flow

User	System
1. Presses ”Menu”-button	
	2. Pauses game, opens menu
3. Clicks options	
	4. Opens options
5. Clicks ”Key bindings”	
	6. Switches tab to ”Key bindings”
7. Clicks ”W”	
	8. Highlights W
9. Presses UP_ARROW	
	10. Changes W to UP_ARROW
11. Clicks OK	
	12. Saves and closes.

Use case: Loot from chest

Summary: User moves towards a chest and loots it.

Priority: High

Participators: User and system

Normal flow

User	System
1. Presses movement-key	
	2. Character moves to chest
3. Presses interact-key	
	4. Character opens chest + sound effect
	5. Adds weapon to inventory
	6. Equips weapon

Use case: Save game

Summary: The user saves his progress in the in-game menu.

Priority: Mid

Participators: User and system

Normal flow

User	System
1. Clicks ”Save game”	
	2. Opens ”Saved games”
3. Clicks ”New file”	
	4. Creates a new save file row
	5. Sets default name & data
	6. Highlights name so user can change it
7. Writes ”Save 1”	
	8. Prints ”Save 1”
9. Clicks ”OK”	
	10. Save file
	11. Gives feedback

Use case: Fire weapon

Summary: Fires a weapon towards an enemy.

Priority: High

Participators: User and system

Normal flow

User	System
1. Aim at enemy	
2. Press ”shoot”-button	
	3. Check if weapon has ammo in magazine
	4. Launch projectile, lower ammo count in magazine
	5. Checks if enemy is hit, decrease enemy HP

Exceptional flow

User	System
1. Aim at enemy	
2. Press ”shoot”-button	
	3. No ammo in magazine
	4. Reload

Use case: Pick up power-up.

Summary: PC picks up and activates power-up.

Priority: Mid

Participators: User and system

Normal flow

User	System
1. Presses movement keys	
	2. Move PC over power-up
	3. Power-up activates