Requirements and Analysis Document for Bobo the Panda

Table of Contents

Version: 1.1

Date: 2013-03-26

Author: Sebastian Sandberg, Victor Larsson, Elvira Jonsson, Oscar Muhr

This version overrides all previous versions.

1 Introduction

1.1 Purpose of application

A fun game.A sidescrolling platform singleplayer game, where the objective is to go from start to the end, and find a key to unlock the finish line on each level.

1.2 General characteristics of application

The application will be a desktop, stand alone, single-player application with a graphical user interface for the Mac/Windows/Linux platforms.

The application is multi level gameplay. The player from point A to point B on the screen and tries to avoid obstacles.In order to complete each level, the player must find a key to unlock the door at the end. The only way to finish the game is by completing all levels.

1.3 Scope of application

The application does not include multi-player mode. Nothing is transferred between levels, each level is its own chapter.

1.4 Objectives and success criteria of the project

It should be possible to play the game on any of the given platforms. The game should contain at least two complete levels. A complete level should contain enemies, obstacles and a key that will grant player progression.

1.5 Definitions, acronyms and abbreviations

- GUI, Graphical User interface
- Java, Indepedent programming language
- JRE, Java Runtime Environment
- Player, the user controlled object in the game.
- Level, each independent game environment, where the player must find the key and unlock the door to complete.
- Tiled, the leveleditor used to draw the GUI of each different level.
- Enemies, objects where a collision with the player results in player death.
- Death, when the player dies, he restarts the current level.

2 Requirements

In this section we specify all requirements

2.1 Functional requirements

1. Start the game
2. Click on the start button to start playing
3. Move the character left/right using the arrow keys, to jump - press space bar.
4. Colliding with any lethal object will result in death and level restart
5. Try to get the character to pick up the key
6. After picking up the key, get to the door to start the next level.
7. To exit to main menu, press Escape. This will end all game progress.

2.2 Non-functional requirements

NA

2.2.1 Usability

This is a standard 2D single player game. The game starts with a demo level to introduce the player to the controls. This creates a high level of usability. Anyone can play this game within a few minutes.

2.2.2 Reliability

NA

2.2.3 Performance

The players key presses should give a action response within 150 milliseconds

2.2.4 Supportability

The application is only supported for desktop.
Adding new levels and more obstacles should be easy. The only things necessary should be a sketch of the level and a specification of consequences of collision with any new obstacles.

2.2.5 Implementation

The application is built in Java using Maven as a building tool. The user will need a JRE (Java Runtime Environment) and Maven installed to be able to run the application. The application uses Git as a version handling tool.

2.2.6 Packaging and installation

The application can be cloned from Git-repo
https://github.com/eeayiaia/bobothepanda

2.2.7 Legal

NA

2.3 Application models

2.3.1 Use case model

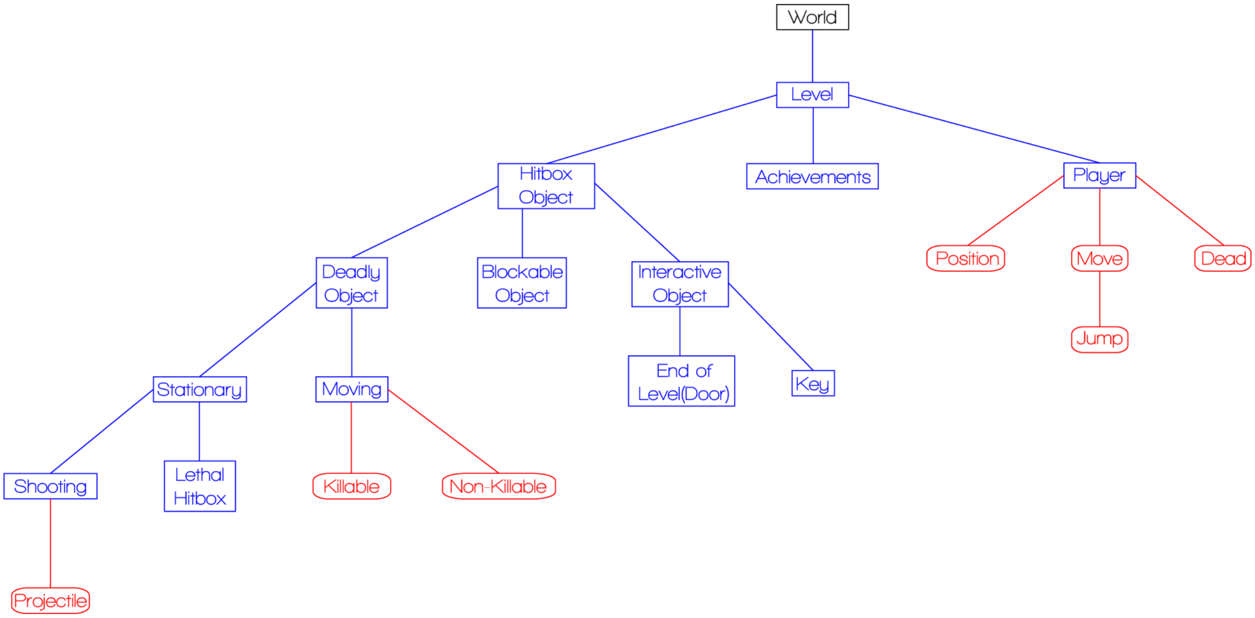
Start level
Move
Jump
Open Door
Grab key
Collision
Achievement
Player death
End of level

UML and a list of UC names (text for all in appendix)

2.3.2 Use cases priority

1. Move
2. Collision
3. Jump
4. Open door
5. End of level
6. Player death
7. Grab key
8. Start level
9. Achievement

2.3.3 Domain model



2.3.4 User interface

Text to motivate a picture.

2.4 References

APPENDIX
GUI
Domain model
Use case texts

Use Case: Move

Summary: The player moves Bobo left/right

Priority: High

Extends: -

Includes: Jump

Participators: Player

Normal flow of events

1. Move Bobo left or right

	Actor	**System**
1	Presses the right or left arrow key	
2		Moves Player across the screen right/left while key is pressed.
3	Releases the arrow key	
4		Stops Player from moving across the screen

Alternate flows of events

1.1 Modifier is held down to run faster

	Actor	**System**
1.1.1	Presses modifier in addition to right or left arrow key	
1.1.2		Moves Player at an increased speed across the screen right/left while keys are pressed.
1.1.3	Releases the arrow key	
1.1.4		Stops Player from moving across the screen

Use Case: Collission

Summary: Player collides with various objects..

Priority: High

Extends: -

Includes: -

Participators: Player

Normal flow of events

1. Player collides with blocking object

	Actor	**System**
1	Player collides with an object	
2		Stops players movement.

Alternate flows of events

1.1 Player collides with static lethal object

	Actor	**System**
1	Player collides with an object	
2		Stops players movement.

2.1. There is terrain in the way blocking Bobos path

	Actor	System
2.1.1		Stops Player from moving across the screen

2.2. Bobo hits lethal obstacle

	Actor	System
2.2.1		Executes Player and the level restarts

Use Case: Jump

Summary: The player makes Bobo jump

Priority: High

Extends: Move

Includes: -

Participators: The player

Normal flow of events

1. Jumping while standing still

	Actor	System
1	Presses the up arrow key	
2		Moves Bobo in an upwards direction with a deceleration.
3		Once peak is reached Bobo starts moving towards the ground.
4		Bobo stops moving once he hits the ground.

Alternate flows of events

1.1 Jumping while moving left or right

	Actor	System
1.1.1	Presses the up arrow key + left/right key	
1.1.2		Moves Bobo in an upwards and a right/left direction with a deceleration.
1.1.3		Once peak is reached Bobo starts moving towards the ground.
1.1.4		

Use Case: Open Door

Summary: Opens a door, enables player progress

Priority: Mid

Extends: Grab Key

Includes:

Participators: Player, Door-object

Normal flow of events

	Actor	System
1	Player grabs key	
2		Sets the door-object status to open

Use Case: End of Level

Summary: Player reaches end of level, next level starts.

Priority: High

Extends:

Includes:

Participators: Player

Normal flow of events
1. Player reaches end of level

	Actor	System
1	Player reaches end of level	
2		End current level
3		Shows a picture with “Completed Level x”
4		Shows a picture with “Level x+1”
5		Starts Level x+1

Use Case: Player death

Summary: Player dies by interactive objects, and restarts current level

Priority: High

Extends:

Includes: Collision

Participators: Player, Interactive Object

Normal flow of events
1. Player dies due to interactive object - Deadly leathal hitbox

	Actor	System
1	Player enters a leathal hitbox	
2		Stops player movement
3		Player status changed to dead

4		Show picture of “You died!”
5		Start game at current level

Use Case: Grab Key

Summary: Player collects an object (a key)

Priority: Mid

Extends: Collision

Includes: Open Door

Participators: Player, key-object

Normal flow of events

	Actor	**System**
1	Player collides with key-object	
2		Indicates that the key has been collected
3		Open Door