Requirements and Analysis Document (RAD) for Killer Kidz

Table of Contents:

Version: 1.0
Date: 31st of May 2015

Author: Kim Berger, Oscar Beronius, Matilda Horppu, Marie Klevedal

This version overrides all previous versions.

1 Introduction

This section gives a brief overview of the project.

1.1 Purpose of application

The purpose of this application is to provide entertainment to people. The player will be able to play it in short periods of time since it is based on separate levels.

1.2 General characteristics of application

The application is an action game with a graphical interface, played on computers.

The goal is to protect a toy store from greedy children who want to invade it. Each level will permit a certain number of children to enter the store before failing. The player throws candy at the children to make them disappear and gain money. If the number of allowed children in the toy store hasn't been exceeded when there are no more children, the player completes the level and can continue to the next one, after visiting the candy shop.

The player has several attacks, represented by different types of candies he can throw. When a level is finished, the money collected by the player can be spent to upgrade or buy new attacks. To create variation and make the game more fun, the player has different options in how he can upgrade his candies. The idea is that each candy has four different upgrades that can be bought. Buying one of these upgrades unlocks another specific upgrade for that branch. The further down in a branch, the more expensive the upgrade. The branch extends to 2-4 new upgrades. Upgrading in several branches is possible. Unused money will be saved to the next level.

Different kids will react differently to different kinds of candy. Some candy might make them more happy and make them disappear at once, while another candy sort will make them go berserk (become killer kidz).

1.3 Scope of application

The game will have four levels, four kinds of children and three kinds of candy. All candy sorts will not be available from the beginning, but the player will be able to buy new ones as he makes progress in the game. The estimated time to complete the game is thirty minutes.

The following features are not part of the game but are suggesions on how it could be extended:

- Multi-player (parts of the game are already implemented to support this)

- Sound effects and music in the game. The volume of these can be modified from a main menu.
- When completing the game, the amount of money left will be saved in a high score table. The player (or other players) can then choose to play the game again, to try to beat the current high score.
- The game will be saved if interrupted, and if the player fails a level, he will be able to play it again without having to start over from the first level.

1.4 Objectives and success criteria of the project

There will be a high replay-value (see Definitions) of the game because the player will not have money to upgrade all CandyTypes.

There will be “randomness” witihin the game.

1.5 Definitions, acronyms and abbreviations

- Levels - the game is consisted of multiple levels which you progress through to finish them all to complete the game
- Waves - each level will exist of a certain number of waves which will arrive within certain timeframe for each level.
- Candy - a projectile you can throw at kids to nullify them
- CandyType - there exist a number of candy types the player can buy to unlock and even upgrade, and each type have unique attributes you can spend money to upgrade to alter them to stronger types.
- KidTypes - there exist different kids with unique attributes (not all of them will be easy to stop) which also will have a minor randomised speed to add a more interactive touch to the game (they are not robots).
- Replay-value - after finishing the game, you could replay the game multiple times with different playstyles and encounter.
- KeyBindings - The set of keys you use to play the game (arrows and Space as default)
- CandyShop - a window(area) you will arrive after completing each level, where you can spend money to unlock and upgrade your candy, and also see your current balance, see KidTypes which will arrive next level.

2 Requirements

In this section we specify all requirements

2.1 Functional requirements

The player(s) should be able to:

1. Read an ingame help how to play the game
2. Start a new game
3. Play a game until the player wins or lose
4. During the action-part of the game, the player can:
    a. Move the Player around in the levels
    b. Choose candyType the Player have unlocked
    c. Throw chosen candyType
5. Exit the application.

2.2 Non-functional requirements

2.2.1 Usability

Usability and interaction is a high priority. Anyone should be able to understand and play the game. There will be information on how to play the game in the HelpWindow reached from the MainMenu.

2.2.2 Reliability

The game shouldn't have any bugs that we know of at least.

2.2.3 Performance

Any action initiated by the players of the game should be experienced as if they happen instantly. All moving objects on the playfield should without any hinder and maintain a 60 FPS (frames per second) with an everyday computer.

2.2.4 Supportability

The game supports the operating systems Windows, IOS and Linux.

2.2.5 Implementation

The application will be implemented using the Java environment in order for it to be able to run at different platforms. Every device where the application will be run needs to have

JRE(Java Runtime Environment) installed.

2.2.6 Packaging and installation

NA.

2.2.7 Legal

NA.

2.3 Application models

2.3.1 Use case model

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

2.3.2 Use cases priority

MM at the start of the name means that the use case takes place in the main menu.
CS at the start of the name means that the use case takes place in the candy store.

High priority use cases

- MM_StartNewGame
- MovePlayer
- ThrowCandy

Medium priority use cases

- ChangeCandy
- StartWave
- CandyHitsKid
- GainMoney
- KidEntersStore

Low priority use cases

- GoToCandyShop
- CS_Move
- CS_ChangePlayer
- CS_ChangeCandy
- CS_BuyUpgrade
- CS_StartNextLevel
- MM_Move
- MM_ViewHowToPlay
- MM_Quit

2.3.3 Domain model

To begin with, there will only be one player. However, the game will be implemented in a way that makes it easy to add more players, which also will be done if there is time.

See appendix for a graph of the domain model.

2.3.4 User interface

The goal is to make the user interface easy to understand and a natural part of the application, i.e it should not take unnecessary attention from the game itself. Since it's an entertaining game there will be bright colours and a quite childish design on all components.

The GUI will consist of a main menu (enclosed file MainMenu) with three alternatives you can choose from. Which you can navigate to with the arrow-keys and select current object with the enter key. And the alternatives are:

- Play - where you start a new game.
- HowToPlay - information how to play the game
- Quit - close the game.

When playing the game there will be a static background with the toy store guard (the player) and the children moving around infront of it (enclosed file Playfield). On the left side the toy store, which the player is supposed to protect, will be and the children will be running into the screen from the right. In the top part of the screen there will be a bar of different information. In the top left corner is an box which tells the current-level the player is on, and below that is an number which tells how many more kids you are allowed to pass into the toy store, before loosing. Beside the level information all the candy sorts that you have bought will be shown, each one in a box. In each box you will be able to see which key to press in order to choose that specific candy. There will be an orange border around the selected candy in the candy bar. This is becase it should be easy to see which candy you will throw. In the top right corner you will be able to see how much money you have earned at this level so far.

When you have completed a level you will get to the candy shop where you can upgrade your candy (enclosed file CandyShop). Under the title, there will be a line telling you what player is currently shoping and under that a line with the current candy for which you are purchasing upgrades. When marking any of those lines, you will be able to switch player/candy with left and right arrow keys. Below that, there will be a table of upgrades. Each upgrade is colour coded so you can see if you already have it, can buy it or can't buy it. At the top left corner in the shop, there is information on what the colours mean. At the top right corner there will be information on how much money you have got. You perchase an upgrade by pressing enter while marking it. Below the upgrade table there will be a line saying "Start Next Level", which you mark (and press enter) when you have bought all you want/can. At the bottom of the

screen, information will pop up when the acndy shop wants to tell you something. If you fail a level, you will start over.

2.4 References

NA.

APPENDIX

GUI

The figures of the GUI are enclosed in folder GUI.

Domain model

The Domain model is enclosed as a separate file.

Use case texts

The use case texts are enclosed in a folder UseCases.