Requirements Bubble Trouble

Group 32

Tim Buckers
Isha Dijcks
Boning Gong
Floris List
Naomi de Ridder

September 8, 2015

Contents

1 Functional requirements

For the game Bubble trouble, the requirements that describe functionality and service are organized under the Functional Requirements. We divided the functional requirements into four sub groups. They are divided using the MoSCoW model. give a priority to the requirements.

1.1 Must haves

- When the game is opened, a starting screen is shown.
- The starting screen has a start button.
- A player must be able to start the game.
- The game starts with 'Pedro' (the guy who the player controls) and a big ball somewhere in the map.
- The player has 5 lives.
- Somewhere on the map, the lives of the player are shown.
- The map has borders where the player and the balls can't go through.
- If the ball hits the wall it bounces back.
- If the player hits the wall it simply can't move further to that side.
- The player must be able to move left and right with Pedro.
- The player must be able to shoot a rope, when there are no ropes in the game.
- The rope disappears when it hits the roof.
- The rope disappears when it hits a ball.
- The ball should pop into 2 smaller balls when the rope hits it.
- There are four different sizes for the balls.
- The ball bounces.
- A ball has a maximum height which stays the same.
- A small ball has a higher maximum height than a large ball.
- If the ball has the smallest size and the player hits it, then it will disappear.
- The largest ball in the game can be popped a maximum of 3 times into smaller balls.
- If the ball hits the player, the player dies and loses 1 life.

- If the player dies the level restarts.
- If the player runs out of lives the game is lost.
- If all balls are disappeared then the game is won.
- When a game is won, the player will go to the next level.
- When the last game is won, the player has won the game.
- The balls in the game are visualized as circles.
- The player in the game is visualized as a rectangle.

1.2 Should haves

- The player shall be able to start a new game
- The player shall be able to pause a game of Bubble trouble that is currently in progress
- If the game is paused the player should be able to start the game again.
- The game ends when the player stops it
- The game shall assign a color to each bubble depending on its size.
- When the game starts the player score is 0.
- Score increases when the player pops a ball.
- Score increases more when the player pops a bubble before the first bounce.
- Every level should be finished in a certain amount of time.
- If there's no time left, the player dies and loses 1 life.
- Score increases when the level is won and there is time left.
- When the player dies the score of that level resets.
- When the bubble hits the roof the entire ball disappears and the players score increases.
- In the interface next to the board is the current score shown.

1.3 Could haves

- You could be able to play the game with two players.
- The start screen has a button which, if clicked, lets you play with two players.
- If the game is played with two players, different controls are used for the two players.
- A local leaderboard of scores.
- Continuing in the last level where you left.
- The game could have power-ups.
- If the player walks over the power-up, the power-up is activated.
- There is a power-up 'time' which gives you more time to finish the game if activated.
- There is a power-up 'freeze rope' which makes sure that the rope doesn't disappear when it hits the roof if activated.
- The power-up 'freeze rope' is deactivated when a ball hits the rope and then the ball pops and the rope disappears.
- There is a power-up 'walk faster' which lets the player walk faster for a few seconds, if activated.
- There is a power-up 'more points' which increases the score more, if it is activated, when a rope hits a ball.
- There is a power-up 'shield' which makes sure that you don't lose a life if you are hit by a ball, if it is activated.

1.4 Would/Won't haves

- Creating own levels.
- More than 2 player modes.
- Online leaderboard of scores.

2 Non-functional requirements

It is also useful to provide a list of constraints.These requirements will not tell what the system should do, but instead indicate the constraints that apply to the system or the development process of the system.

- The game shall be implemented in Java.
- A basic functionally game should be delivered on Friday September 11, 2015 before 23:55.
- The final product shall be delivered October 30, 2015 before 23:55.
- The game shall be playable on Windows.
- We will use the Scrum methodology to create this game.
- We will use the tools Trello, Checkstyle, PMD, Maven, Travis, Eclipse, JUnit and github.
- Our group consists of only 5 people.