TI2206 Software Engineering: Bubble Shooter
EWI/EEMCS

Gerlof Fokkema 4257286
Owen Huang 4317459
Adam Iqbal 4293568
Nando Kartoredjo 4271378
Skip Lentz 4334051

September 16, 2014

Functional

Must-have features

- When the user starts the application, a main menu with three buttons will be presented. Inside the user is able to do the following actions:
  - When the user clicks on the *play* button, then a new single player game is started.
  - When the user clicks the *settings* button, then the user will change screens to the options menu.
  - When the user clicks the *quit* button, then the application will terminate. This results in closing the application.
- Given the user has started a new single player game, by clicking on the *play* button in the main menu, the user can do the following actions inside the single player game:
  - When the user presses the *left arrow key* the cannon will rotate to the left, the opposite will be done when the *right arrow key* is pressed.
  - When the user presses the *spacebar* button on the keyboard, the user will shoot a projectile in the direction the cannon is facing.
- Players should be able to play against another player over an internet connection.
- Up to two players have to be able to play together:
  - A screen with two Boards is presented, thus each player has a separate board.
  - Players can shoot Bubbles to the Board of their opponent through a hole.
- The game mode will be [Insert game mode].
  - The player with the highest score after the time limit wins.
  - If a players bubble reaching the bottom of the board within the time limit then the player immediately loses.

Should-have features

- Given the user has started a new single player game, the following game mechanics and events should apply:
  - When the user enters the playfield, the user will be presented a field filled with bubbles.
  - Given that a projectile has been fired and it hits another bubble, the projectile will disappear along with the bubble if the color of both bubbles are the same and **at least** another bubble adjacent to the bubble that was hit, is also of that same color.
  - When the user has fired a projectile and it hits another bubble it will stick to that bubble, if the condition above does not hold.
  - Given that the user shoots a projectile all the way to the ceiling, it will stick to the ceiling when the above two rules do not hold.
  - Given that the user has fired, when the user is able to fire again a bubble of random color (out of a selection of five) will spawn as the new projectile.
  - Given that a bubble was hit and removed, if the adjacent bubbles do not connect to either the ceiling or other bubbles (that ultimately connect to the ceiling), then the bubble is removed.

Could-have features

- Given the user has started the application, clicking on a button will play a sound effect.
- Given the user has started the application, background music will be played right away.
- Given the user clicks on the settings button, the user will be presented the options menu, where the user can do the following actions:
  - When the user clicks the *Volume Up!* button, the background music volume will rise.
  - When the user clicks the *Volume Down!* button, the background music volume will go down.
  - When the user clicks the *SFX Up!* button, the sound effects volume will rise.
  - When the user clicks the *SFX Down!* button, the sound effects volume will go down.
  - When the user clicks the *Back* button, the user will return to the main menu screen.

- Given the player has started a single player game, when a bubble reaches the floor (bottom of the screen) the player loses and the game ends.
- Given the player has started a single player game, when all bubbles are removed the player wins!
- Given the user has started a single player game, when the user presses the *Escape* button, the game will pause
- The players are able to select multiple game modes.

Won't-have features

- Given the user has started a single player game, the user can exit the game (go back to main menu) by pressing the *M* key on the keyboard.

Non-functional

- A simple version should be finished within one week (13/09/14).
- The development team consists of five group members.
- The game must be written in Java using the following supporting tools:
  - maven
  - jUnit
  - git
- The development process (including the sprints) will be using SCRUM, with the help of the following web based tool:
  - ScrumDo
- Meetings
  - Friday 20:00 - 05/09/2014 (Daily sprint)
  - Monday 20:00 - 08/09/2014 (Daily sprint)
  - Tuesday 9:45 - 09/09/2014 (Sprint planning)
  - Wednesday 20:00 - 10/09/2014 (Daily sprint)
  - Friday 20:00 - 12/09/2014 (Sprint review)
  - Tuesday 9:45 - 16/09/2014 (Sprint retrospective + Sprint planning)
  - Wednesday 20:00 - 17/09/2014 (Daily sprint)
  - Friday 20:00 - 19/09/2014 (Daily sprint)
  - Saturday 20:00 - 18/09/2014 (Daily sprint)
- The game must support the following OS:
  - Microsoft Windows
  - Linux
  - OS X