TI2206 Software Engineering: Bubble Shooter
EWI/EEMCS

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

October 26, 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 *Single-player* button, then a new game mode may be selected (Arcade/Survival/Zen), these are different single-player games.
  - When the user clicks on the *Multiplayer* button, then a new multiplayer game is started.
  - When the user clicks on the *High Scores* button, then the user will change screens to the high scores screen where all the high scores are listed.
  - 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.
- The following generic game rules will apply, independent of single- or multiplayer mode:
  - When the user enters the playfield, the user will be presented a field filled with bubbles with various colors:
    - Red
    - Blue
    - Purple
    - Yellow
    - Green
  - When the user has fired a projectile and it hits another bubble, the projectile will stick to that bubble.
  - When the user shoots a projectile and it hits the ceiling, the projectile will stick to the ceiling.
  - When a group of at least three adjacent bubbles with the same color has been formed by shooting a projectile, this group of bubbles will disappear and the user's score will increase with:
    *3 * bubbles + 3 * disconnectedgroup - 3* points.
  - Given that the user has fired, when the user is able to fire again a bubble of random color (out of a random selection of five) will spawned 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, and the user's score will increase.
  - When a bubble reaches the floor (bottom of the playfield), the user loses and the game ends.

- Given the user has started a new game, by clicking on the *Single-player* 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.
  - When the user presses the *right arrow key*, the cannon will rotate to the right.
  - When the user presses the *spacebar* button on the keyboard, the user will shoot a projectile in the direction the cannon is facing.

- Given the user has started a new multiplayer game, by clicking on the *Multiplayer* button in the main menu, the user will be presented with a multiplayer game:
  - After starting a multiplayer game from the menu, the user can choose to host a server or to join a server.
    * When the user chooses to host a server, the game will wait until another user connects.
    * When the user chooses to join a server, the user will be presented with a dialog, where he can enter an IP address and join an existing game.
  - Up to two users have to be able to play together.
  - A screen with two Boards is presented, thus each user has a separate board.
  - The game mode will be *sudden death*.
    * The user with the highest score after the time limit wins.
    * If a users bubbles reach the bottom of the board within the time limit then the user immediately loses.

- The game should be logged into a file, given a game has started, including the following elements:
  - When the user has pressed the *left* or *right arrow key* causing the cannon to rotate to the left or right, the new angle after rotation will be logged.

  - When there is a projectile in the field, shot by the user by pressing the *spacebar*, the positions of the projectile will be logged during its travel.
  - When a projectile collides with a bubble on the board, the index of the grid (having a position), to which it will stick to is logged.

- The user should be able to select from multiple game modes.
  - When the user selects Arcade mode, then the user will be able to play through multiple pre-made Bubble Shooter levels (with timer).
  - When the user selects Zen mode, then the user will get a randomized Bubble Shooter level (no timer)
  - When the user selects Survival mode, then the user will get a randomized Bubble Shooter level (no timer).

- Given the user has started a survival game, every seven shots two rows of bubbles will be shifted from the top onto the playfield.

- Unexpected exceptions should all be logged into a file, given that the application has started.

- Given the user has started an arcade game (single-player), when the player loses the score will be recorded into a file:
  - Given the score is above zero, when the score belongs to one of the highest ten scores previously recorded, it will be a new highscore.

- Given the user has started the application, when the user clicks on the *High Scores* button, the top twenty high scores from all the modes will be listed:
  - Given there are high scores to be listed from any mode, the top ranking will be firstly based on the highest level achieved in the game (high score will be displayed on the top then). If there are multiple entries of a high score achieving the same level, the score will be used as a second criterium for determining the highest ranking (higher score means ranked higher).

Should-have features

- Given the user has started a single-player arcade game, when all bubbles are removed the user wins!
- Given the user has started a game, the user should be able to win/lose a game (except when the user plays the Zen or Survival-mode, the user may not be able to win).
- Given the user has started a game (single or multiplayer), when the player rotates the cannon, it must not rotate too fast or too slow (a good sensitivity).
- Given the user has started any game, the score should be displayed on the screen.
- Given the user has started an arcade-game, a timer should be displayed on the screen.

Could-have features

- Given the user has started the application and is in the menu, 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 + button below the *Change Volume*, the background music volume will rise.
  - When the user clicks the - button below the *Change Volume*, the background music volume will go down.
  - When the user clicks the + button below the *Change SFX*, the sound effects volume will rise.
  - When the user clicks the - button below the *Change SFX*, the sound effects volume will go down.
  - When the user presses the *ESC* key, the user will return to the main menu screen.
  - When the user clicks the *Change Theme* button below the *Theme*, the theme of the entire game will change (space/doom theme).
- Given the user has started a single-player game, when a bubble reaches the floor (bottom of the screen) the user loses and the game ends.

- Given the user has started a single-player game, when all bubbles are removed from the level the user wins and proceeds to the next level!
- Given the user has started a game, when the user wins/loses, the user should be able to press *ESC* button resulting in the user being returned to the main menu.
- Given the user has started a game, there will be music played on the background.
- Given the user has started an arcade game. Special bubbles may appear in levels. The following special bubbles may be encountered:
  - Given a Bomb bubble is encountered and the user shoots a bubble to it (causing a collision), and the Bomb bubble is surrounded by other bubbles (color or bubble type does not matter), the Bomb bubble will remove all adjacent surrounding bubbles:
    - **Additional behaviour**: Given two or more Bomb bubbles are within the blast radius of another Bomb bubble, when the bubble is hit, the blast may cause a chain reaction triggering the other Bomb bubble inside the blast radius.
  - Given a Stone bubble is encountered and the user shoots a bubble to it (causing a collision), the bubble will **not** be removed except if it is disconnected from either the ceiling or another bubble.
  - Given a Nuke (also known in-game as a MichaelBay) bubble is encountered and the user shoots a bubble to it (causing a collision), all bubbles are instantaneously removed resulting in passing the level.
- Given the user has started an arcade game, there is a maximum of twenty levels that can be played (in one sitting):
  - Given the user arrived at the 20th level, when the user manages to finish it by winning, the user will have won the game and the game will end.
- Given the user has started the application, when the user goes to the options screen, the user may change the theme of the game. This will change all textures (background, borders, bubbles) in a game accordingly to the selected theme.
- Given the user has started an arcade game, when the user wins a level, the user should feel a rising difficulty when advancing to the next level (from easy to hard).
- Given the user has started an arcade-game, the corresponding level number and name is displayed on the screen.

Won't-have features

- Given a user has started a game, the user can gather and fire power-ups.
- Given the user has started either a single- or multiplayer game and a projectile has been shot, when the projectile collides with a bubble causing it to be removed, a beautiful animation showing the removal will be shown (may be different animations for special bubbles).
- Given the user has started a single-player game, the user can pause the game by pressing the *ESC* key on the keyboard and will be presented with a small in-game menu.
- Given the user has started a multiplayer game, when a game has been found (by connecting with another user), the users may chat with each other in a chatbox.

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 - 18/09/2014 (Daily sprint)
  - Saturday 20:00 - 20/09/2014 (Daily sprint)
  - Tuesday 9:45 - 23/09/2014 (Sprint retrospective + Sprint planning)
  - Wednesday 20:00 - 24/09/2014 (Daily sprint)
  - Friday 20:00 - 26/09/2014 (Daily sprint)
  - Saturday 18:00 - 27/09/2014 (Daily sprint)
  - Tuesday 9:45 - 07/10/2014 (Sprint retrospective + Sprint planning)
  - Wednesday 20:00 - 08/10/2014 (Daily sprint)
  - Friday 20:00 - 10/10/2014 (Daily sprint)
  - Saturday 18:00 - 11/10/2014 (Daily sprint)
  - Tuesday 9:45 - 14/10/2014 (Sprint retrospective + Sprint planning)

  - Wednesday 20:00 - 15/10/2014 (Daily sprint)
  - Friday 20:00 - 17/10/2014 (Daily sprint)
  - Saturday 18:00 - 18/10/2014 (Daily sprint)
  - Tuesday 9:45 - 21/10/2014 (Sprint retrospective + Sprint planning)
  - Wednesday 20:00 - 22/10/2014 (Daily sprint)
  - Friday 20:00 - 24/10/2014 (Daily sprint)
  - Saturday 18:00 - 25/10/2014 (Daily sprint)
  - Tuesday 9:45 - 28/10/2014 (Sprint retrospective)
- The game must support the following OS:
  - Microsoft Windows
  - Linux
  - OS X