

# TI2206 Software Engineering: Bubble Shooter report

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

September 13, 2014

## Unimplemented features

This small report explains why some features were left untouched and why some things were tested rather poorly. In other words, this document elaborates why some features and testing were left unimplemented and the reasoning behind it.

In the requirements there were several features left unimplemented. One of these features was for instance the *losing* and *winning condition*. While this contributes in a sense to the overall experience the player has when playing the game, it was not a top priority. These features are not critical for the enjoyment of the gameplay, because one could see this game mode as a sort of *endless or zen mode*. During the development phase it was decided to push this feature to the end, in case there was not enough time to implement this. Since this was also the case, this feature was left unimplemented. The main reason why this happened can be explained by going into depth of the development process itself.

During this first *sprint*, of the SCRUM methodology that was being used, this feature was planned to be implemented during one of the last days (as mentioned before, it was pushed to the back). This was done, because the more critical features required a lot more attention and care. Testing also took a lot more time than expected, so in the end there was simply not enough room for this to be implemented and fully tested.

Other features that were listed under the *Won't-have* section were as planned, not implemented. These included pausing a single player game by pressing *Escape* and exiting to the main menu by pressing the *M* key. Simply because these were mostly just features for convenience of the player. This was something that contributed little to none to the actual game experience.

## Difficulties with testing

Testing is usually a difficult thing to do, it takes a lot of time and effort to produce good quality tests and sometimes things just do not go the way you wish to. The latter happened during this *sprint*. This can be explained by reflecting on the design of various classes, such include the different *screens*. Most GUI elements are usually more difficult to test. These all load in files or use classes that load in external files. This caused the tests being forced to include mocking. Although it is of course possible to use mocking, the process of creating good tests in this fashion is exhausting (time-wise). So because of the fact that these classes were changed near the end of the *sprint*, initial test plans were not possible anymore for these classes and had to be discarded. Therefore several classes now remain untested.