

# Development Plan

Team 25

N.L.E

Ruoyuan Liu, liur19

Gundeep Kanwal, kanwalg

Zihao Chen, chenz87

September 29, 2017

## 1 Team Meeting Plan

The team plans on meeting at least twice a week, Wednesdays and Fridays from 2:30 p.m. to 3:30 p.m at the Mills Library (or other times if not possible), along with the time the team has during the labs.

## 2 Team Communication Plan

The team will communicate via "Facebook Messenger" and email, various text and Latex documents will be shared with "Google Doc" and "Share-latex". The software for the project will be shared on Git Lab in the team's repository.

## 3 Team Member Roles

Name	Role	Expert On
Ruoyuan Liu	Programmer, Tester, Designer	LaTeX, Java programming
Gundeep Kanwal	Programmer, Log Admin	Documentation, Doxygen
Zihao Chen	Programmer, project manager	Java Programming, Git

## 4 Git Workflow Plan

Our team will use centralized Git workflow plan. It allows the N.L.E team to share and update work consistently while maintaining project modularity. In general, all project documents and source code are supposed to locate in the master branch, which means pushing new versions to git should be discussed during team meetings by all developers in case to make sure the push is valid and the application is still able to compile and run. A temporary branch can be created for new version testing purposes but can never be merged to the main branch.

Tagging is required for each documentation being pushed and a new version of the project is being pushed. When committing changes in source code, commit message should be stated in detail by the developer, consist of sections that changed and its effect.

## 5 Proof of Concept Development Plan

The main task of the project is to provide our users a better version of an simple game doodle jump with attracting modifications and more easy access to the game, which indicates that the architecture of this project is relatively unchanged comparing to the original game. To be more specific, basically all

the original classes and functions will not be affected during our development.

As the implementation for the software itself is quite self-sufficient and complete, new and interesting modifications might be difficult to accomplish at the concept level. Brainstorming and coming up with interesting concepts to improve the game is a significant challenge for our project. However, game itself still has potential space to modify such as character speed boosting. The conceptual difficulty is time-consuming but solvable. On the bright side, the game has working functions, completed user interface and used built-in library in Java only, which makes our implementation much easier.

Testing for the game might be a concern after implementation. Game in general is very difficult to be bug-free. Moreover, Bugs have already existed in the original project as we playing it so far. But since the original project is simple and small, it is possible for us to eliminate most of the bugs. An efficient way of testing the game is to establish access to the user to report the bugs they encountered. We may be able to deliver copies to people who are interested in the project to gather bug reports before pushing final version of our project, and fix the bugs as many as possible.

## 6 Technology

The programming language which will be used is Java. The IDE that will be used is Eclipse. In terms of the testing framework that will be used, the group has decided to make use of Java's JUnit to test the code. For document generation, the team will use Doxygen as the team members have used it previously.

## 7 Coding Style

In terms of the coding style which will be used within the project, the group intends to conform with Oracle's Java Coding Style. To be more specific: - Camelcase naming convention -Two space indentation -Each row is limited to 70-90 characters -Provide meaningful comment for every methods [1]

## 8 Project Schedule

The Gantt Chart for project schedule is located in git repository Documentation/DevelopmentPlan

## 9 Project Review

### References

- [1] Sun Microsystems Inc. Code conventions for the java tm programming language. <http://www.oracle.com/technetwork/java/codeconvtoc-136057.html>. Accessed: 2017-09-29.