

Development Plan

Group 13

September 2018

1 Development Plan

1.1 Team Information

Group members : Gurpartap Kaler, kalerg1,400062310
Freddie Yan, yanz20, 400079138
Sagar Thomas, thomas12, 400054333

1.2 Team Meeting Plan

| Meeting Oder | Date | Location | Topic | Materials | Decisions |
|--------------|---------------------|----------|---------------------|-----------|--|
| 1 | Sept 25th 9:00pm | ITB | Dev. Plan | | Work assigned for each member. |
| 2 | Oct 2nd 9:00pm | Thode | Req. Doc Rev0 | | Part of req. doc. assigned to each member. |
| 3 | Oct 9th 9:00pm | Thode | PoC Demo | | Worked on PoC demo together. |
| 4 | Oct 16th 9:00pm | Thode | PoC Demo Finalized | | Finalized prototype of demo file |
| 5 | Oct 23rd 9:00pm | Thode | Test Plan Rev0 | | TBD |
| 6 | Oct 30th 9:00pm | Thode | Design and Doc Rev0 | | TBD |
| 7 | Nov 6th 9:00pm | Thode | Rev0 Demo | | TBD |
| 8 | Nov 13rd 9:00pm | Thode | Final Demo | | TBD |
| 9 | Nov 13rd 9:00pm | Thode | Final Demo | | TBD |

Rules for meeting:

- 1.Meeting would be held in Thode library every Tuesday after Lab section, every member must arrive before 9:15pm.
- 2.The purposes for those meetings include: (1).review all the sources we have created. (2)contribute ideas for the next deliverable. (3)assign works to each member.
- 3.No phone use during the meeting.
- 4.Meeting duration is about 2 hours.

1.3 Team Communication Plan

Combination of git, email and Facebook messenger will be used for the purpose of team communication.

- 1.Git: git would be used to keep updating the newest version of our project, so

every member can push their contributes into the project.

2.Email: email would be used for the purpose of file sharing.This is the most efficient and secure way to share and save project details.

3.Facebook messenger: messenger is used for daily communication.Since every member is in the same group, we are able to talk about the ideas and ask questions about these project.

1.4 Team Member Roles

There will be no team leader in our team. All decisions will be made together as a team. All tasks will be distributed as a team.

| Name | Role | Expertise |
|-----------------|-----------|------------------------------------|
| Gurpartap Kaler | Developer | Expert on Documentation and Python |
| Freddie Yan | Developer | Expert on LaTeX and Python |
| Sagar Thomas | Developer | Expert on Git and Python |

Table 1: Team Member Roles

All group members have the role of a developer, as we will be working together to create the software. Furthermore, every group member has their own expertise, this helps the team to locate those who are strong in a certain subject.

1.5 Git Workflow Plan

The Git Workflow Plan that we will implement in this project will be centralized and consist of an origin/master and develop branch, as well as feature branches. The master branch will only be used for production-ready code and the develop branch is where all the development will take place. Feature branches will be created as necessary from the develop branch and merged back to the develop branch when complete and tested. Tags will be used to mark important merges/commits such as version numbering for new releases/updates as well as marking milestones throughout the course of this project.

1.6 Proof of Concept Demonstration Plan

1.6.1 Most Significant Risk

The most significant risk of our game, Sokoban, will be having the game crash on the users PC. Another risk is that our game could cause the users PC to hang and not respond, due to keyboard inputs. In order to overcome this risk, we have to make sure that the game is tested thoroughly and rigorously on multiple different operating systems and computers.

1.6.2 Will a part of the implementation be difficult?

Breaking the game down into workable and sizable modules will be difficult, since we are re-implementing the Java applet into Python. I also feel like opti-

mizing the program will be difficult, as Python has different ways to optimize than Java. Furthermore, I feel like our group will be challenged with weighing the importance between performance and readability.

1.6.3 Will testing be difficult?

Unit testing the game will not be difficult because we have worked with Python unit testing in SFWRENG 2AA4, mainly PyUnit. Testing the functionality and logic of the game will be challenging, we will have to let the public play the game and give us feedback. This way we can find bugs with the logic in the program. Another alternative to demoing the game, is creating automated testing, which will be very difficult as none of us have experience with it.

1.6.4 Is a required library difficult to install?

No, the required library is not difficult to install. Our re-implementation will be in Python, and all libraries used will be implanted in our program.

1.6.5 Will portability be a concern?

Portability will not be a concern, due to our group using Python for the re-implementation of Sokoban. Python is available for every operating system (Windows, Linux, and Mac OS). Furthermore, once downloaded initially, our game will be available to the user locally.

1.7 Technology

Programming Language: Python version 3.x.x will be used for this project. The project will not support Python version 2.x.x due to restrictions with the game framework.

Game Framework: We will be using Pygame (latest version) to power the entire game.

Build Tool: Make - a makefile will handle all build, run and test tasks. The makefile will use pip (Python Package Manager) to handle all dependencies.

IDE/Text Editor: Each team member will be using any text editor of their choice. (VS Code, Atom, Vim, Sublime etc.) As all testing, building and running will be done on the terminal, there is no reliance on a specific IDE.

Testing Framework: PyUnit - popular unit testing framework

Document Generation: Sphinx - Extremely popular and simple documentation generator for Python.

1.8 Coding Style

The coding style we will use in this project will be a combination of the [Mozilla Python Coding Style](#) and [PEP 8](#), Python Foundation's own style guide. In order to enforce these principles, we will use [pylint](#) to frequently check the source code for style consistency.

1.9 Project Schedule

The GanttChart file and PDF are available in the [Project Schedule](#) folder. These files will be updated weekly.

1.10 Project Review (for Revision 1)