|  |
| --- |
| Team JEM |
| Settlers of Catan |
| Project Plan |
|  |
| **Eric Maxwell; Justin Lynn; Matt Jackels** |
|  |

|  |
| --- |
|  |

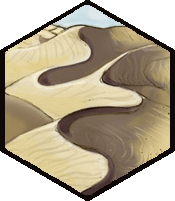


Table of Contents

[1 Revision History 2](#_Toc481420430)

[2 Purpose 2](#_Toc481420431)

[3 Definitions 2](#_Toc481420432)

[4 Software Scope 2](#_Toc481420433)

[5 Gantt Chart 3](#_Toc481420434)

[6 Organization Chart 3](#_Toc481420435)

[7 Tools and Standards 3](#_Toc481420436)

[8 Configuration Management 4](#_Toc481420437)

# Revision History

|  |  |  |
| --- | --- | --- |
| **Date** | **Change Description** | **Changed By** |
| 5/1/2017 | Added scope, Gantt chart, org chart, and tools matrix. | M. Jackels |
| 4/2/2017 | Initial Release w/ purpose and configuration management plan. | M. Jackels |

# Purpose

The purpose of this document is to outline the project plan for the creation of *Settlers of Catan* software by Team JEM. A project plan is intended to give the viewer insight to how the software was engineered and developed to the current working revision.

The scope of this document is to outline the intention for use of the software, define the team hierarchy, explain the timeline, identify the tools and standards used, and outline the configuration management plan.

# Definitions

**Term** – definition

* **Requirements** – the officially approved functionalities to be considered and programmed in the creation of the software.
* **Bugs** – issues or inconsistencies with the program. This can include both front-end (noticeable to an end user) and back-end (noticeable in the source code) issues.
* **Configuration** – The version of the software, as determined by the revision level of the source-code.

# Software Scope

The scope of the JEM project is to plan, design, build, and implement a software version of the game *Settlers of Catan*. This board game popularized by Mayfair offers gameplay for 2-4 players who take turns rolling dice, managing cards, and building settlements. The software will be written in Java and developed to work with the java runtime environment. It will feature a graphical user interface in order for each player to take their turn and perform all necessary actions. The software will initially be a stand-alone game running on a single PC. We will add a dedicated AI and network connectivity in later iterations as time permits.

# C:\Users\mattj\AppData\Local\Microsoft\Windows\INetCache\Content.Word\JEM_SoC_PP2.gifGantt Chart

# Organization Chart

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Role A** | **Role B** | **Role C** |
| Justin Lynn | Testing | Documentation | Programmer |
| Eric Maxwell | Programmer | Architect | Requirements |
| Matt Jackels | Project Management | Documentation | Testing |

Role Descriptions:

* Documentation: responsible for writing the user's manual, documenting umbrella activities such as the project plan, and compiling all documentation in a presentable format. Coders will still document their own code, and designers will still document their own designs.
* Testing: responsible for the test plan, and conducting all tests.
* Programmer: responsible for implementing all designs
* Architect: responsible for designing the software, based on the requirements.
* Project Manager: Advises on ambiguous or conflicting decisions. Creates and manages timeline and status reports.

# Tools and Standards

Below is a table that identifies all the software tools used to create the documents and software. The standards of each created deliverable or product will be defined within the target document.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tool Name** | **General Description** | **Tool Application Description** | **User** |
| Microsoft Word | Text editing software for creating and formatting documents, | Used for a variety of documents, including: Programmers Manual, Planning Documentation, Class Summaries, team messages and partial documentation used as Work-In-Progress files… | All |
| Microsoft Excel | Spreadsheet and calculation software for creating documents that are tables or matrix. | Used for a variety of documents, including: Requirements Matrix, Testing Matrix, Testing Results, … | All |
| Microsoft Project | Project planning software used to create/manage tasks and timelines. | Used to create the Project Plan and timeline for development | Matt J |
| Sublime Text Editor | Text editing software. | Used to view source code and making small notes, and README file viewing. | Matt J |
| Eclipse IDE | Integrated development environment. | Used to create Source Code | Eric |
| MS Visio | Software for creating flowcharts. | Used to draw UML diagrams | Eric |
| Windows command line | Command line interpreter in the windows OS software suite. | Used to create .jar file | Eric |

# Configuration Management

Each development version will be labeled using the standard major.minor.build format and will be uploaded to OneDrive for file sharing. Once a version is deemed “working” by the active developer, they will create an executable .jar file for further testing by the rest of our team. After all testing is completed we will lock that version down and use a copy of it for our next iteration. We plan to store a copy of each iteration as a safety measure and for future reference. During development, we also plan to use GitHub so everyone has quick access to the most updated version. We will use this software as an additional measure to keep track of user history and contributions.

Each release version will be similarly labeled using major.minor, and will be uploaded to OneDrive in a specified “release” folder. These versions will have details of all major and minor updates between the previously released versions. Each version will have undergone rigorous testing as outlined in the testing documentation. Each release will include an executable .jar file for gameplay. We will also use the branch method in GitHub to distinguish our release versions from development versions.

Included with each released version of software loaded into the designated release folder will be documentation of changes to offer clarity to the release. There will be a single document in the release folder titled “Change Matrix” that will track the modifications made to each release. This document should be edited subsequently with the upload of a new release build to the release folder.