Group G:

\* Kyle Pamintuan

\* Pedro Lopez

\* Keith Farwell

CECS 343 - Section 03

Professor Giacalone

Illuminati Project Plan

**Introduction**

Project Scope:

Illuminati is a graphical card game that game lovers can play and enjoy. A user will be able to control up to four players, testing his/her strategic abilities. Since this rendition of Illuminati does not implement networking for playing against other real players, it will be more for people who want to practice the game rather than compete. The user must have a Python IDE installed on their computer in order to operate the game. But the user will not require any knowledge of programming, just basic typing skills.

The project will consist of a graphical user interface that displays sprites of Illuminati cards, dice, and a side bar containing necessary gameplay data.

Inputs:

The user will need to input their choice of menu options which only includes single digit integers.

Outputs:

The program will output illuminati card sprites, dice sprites, a side bar containing necessary gameplay data, and textual menu options.

Performance/Behavior Issues:

There are some free downloadable Python IDE available, but others require a monthly membership or small donation (i.e. Pycharm).

Also, some Python IDEs will require certain versions of Java. And if the Python IDE is a plug-in to another program (i.e. PyDev for Eclipse) it might require a certain version of that program to be able to download the plug-in.

Management and Technical Constraints:

Illuminati has a drop-dead delivery date of May 20, 2017.

**Project Estimates**

Project Resources:

Illuminati will be developed by Group G, a software engineering team containing 3 developers:

Kyle Pamintuan

Keith Farwell

Pedro Lopez

**Risk Management**

Project Risks:

Major risks we have determined for this software are as follows:

* + - Equipment failure
    - Late delivery of software
    - The game will not meet expectations
    - Changes in requirements
    - Deviation from software engineering standards
    - Bugs may still exist after delivery of software
    - Poor commenting of source code

**Project Schedule**

* + - 1. Phase I - Planning

Play the game and get a better understanding of the gameplay

Vision Document

Project Plan

Flowchart / UML

Use Cases

Test Plan

User manual / UI layout

Make notes for game programming:

Data structures

Classes

Class attributes

Class methods

UI plans

Graphics plans

* + - 1. Phase II - Programming

Build program based on the notes made from Phase I

Collaborate using GitHub

Set up meeting times

Basic Gameplay Demo

* + - 1. Phase III - Testing and Efficiency

Version control

Feedback

Bug list and big fixes

Shorten running times

User Interface Demo

Rule Enforcement Demo

* + - 1. Phase IV - Finalizing

Final bug fixes

Documentation

Final Demo

List of Deliverables and Deadlines:

- Vision Document

Due: February 15, 2017

- Project Plan

Due: February 27, 2017

- Flow Chart / UML

Due: March 20, 2017

- Use Cases

Due: April 3, 2017

- Test Plan

Due: March 10, 2017

- User Manual / UI Layout

Due: April 17, 2017

- Basic Gameplay w/ Demo

Due: April 24, 2017

- User Interface w/ Demo

Due: May 1, 2017

- Rule Enforcement w/ Demo

Due: May 8, 2017

- Final Demo

Due: May 17, 2017

Timeline Chart:

**Staff Organization:**

Team Structure:

*Professor Giacalone*

**Project Manager:**

As project manager, Professor Giacalone will be in charge of giving feedback of the project deliverables, making sure progress is going well, and consulting our group as needed.

*Group G - Kyle Pamintuan, Keith Farwell, Pedro Lopez*

**Software Developer:**

All team members are equal ranked software developers responsible for developing the game.

**UI Designer:**

All team members make up the UI design team.

**Game Tester:**

All team members will be game testers and give feedback.

**Documentation:**

All team members will be responsible for documentation of their own share of code.

**Additional Responsibilities:**

All team members will be responsible for completing project deliverables in a timely manner, communicating, and collaborating with each other. Also, all team members will be responsible for their fair share of contribution to the project.

Management Reporting and Communication:

*Mechanisms for Progress Reporting*

Progress is communicated via group text message, GitHub, or Google Docs. All documents and files will be posted on a shared group folder on Google Docs. And all code, commits, and software changes/updates will be pushed to the group repository on GitHub. The communications are done informally, unless special documentation of progress is required.

*Mechanisms for Inter/Intra Team Communication*

Group G will conduct weekly meetings to update other team members on their progress and ask questions that may difficult to answer via electronic communication. All other communication is done electronically. Most team communication is done via the group text message.