

Procedure Implementation Plan

Brandon Smith, Nieka Gutenberger, Joseph Coppin, Ryan Frazier, Trevor Jewkes

September 10, 2016

1 Proposal

1.1 Requirements

The end goal of the team is to write a Graphical User Interface game which allows the user to play one of two card games; the games being Hearts or Spades. The user will be presented with a gameboard and options to play. The player will be able to choose whether he will play by himself or against other humans. If he chooses to play by himself, computer players will play against him.

2 Structure

Each person in the team has been assigned as the head for a specific role. Each role is described briefly as follows:

2.1 Manager

Trevor Jewkes

- In charge of GitHub account
- Will turn in final versions for group
- In charge of code compilation and merge conflicts

2.2 Documentation

Joe Coppin

- Writes rough drafts of all papers and submit for peer review
- Writes final draft of all papers and pushes them to GitHub account for Manager to submit
- Head person over documents

2.3 Design

Nieka Gutenberg

- In charge of GUI and interface
- Will oversee looks and game design
- Will take minutes from meetings and distribute to team members

2.4 Testing

Ryan Frazier

- Initially gets testing software up and running on GitHub
- Writes tests and helps others write tests
- Reviews tests and makes sure they are adequate
- Test tech support

2.5 Scheduling and Mediation

Brandon Smith

- In charge of scheduling and time management
- Workload Balance
- Maintains as even as possible workload balance between team members

2.6 Other Roles

In addition to these roles, each team member is expected to contribute to the project as needed. This means there is no defined end to any team members' duties. It is the responsibility of each team member to communicate their needs to the team Manager and the head of Scheduling and Mediation in order to allow all team members to be able to contribute and help in a timely manner.

3 Procedures

brief overview of procedures

3.1 Code Reviews

3.1.1 Conflict Resolution Plan

1. Make suggestion for change to the person who's work is subject to be changed
2. If two cannot come to an agreement the discussion goes to the group for a decision
3. If group cannot come to a decision, the Team Leader for the section will have final say in decision

3.1.2 Readability

Each section of code written will be reviewed by at least one other team member before being added to the project. In event of disputes or questions, the conflict resolution plan will be utilized.

3.1.3 Reusability

It is the goal of the team to be specific enough code to solve a problem effectively while being general enough that branching out is feasible and well within the reach of the team in cases of customer requirements.

3.1.4 Improvements and Bugs

Improvements and Bugs will be handled similarly to conflict resolution. The manager needs to make sure each person in the team agrees to a solution to the bug (unless head makes decision).

3.2 Styles and Conventions

The team will follow the Google C++ style guide.

3.3 Testing Framework

We will be using the Catch C++ testing framework. It is a single header file that can be included in the source tree under the Boost Software License with no dependencies and provides a way to organize and automate testing. Source and more information can be found at github.com/philsquared/Catch.

4 Schedule

We will start with developing general game logic design. This falls at the beginning because it is the beginning of the critical path. During this stage we will form the basic structures found in the game, such as the cards, players (including how the players will function with their hands, turn, and scores). After the basic logic is determined, we can go on to the Gameboard. The Gameboard will hold the logic that connects all of the different parts of the game together. This portion is expected to take the first two weeks.

Concurrently, we will work on account information where a user may be created and used to login to a profile in the game.

Directly after each section of game logic is created and decided upon, the GUI counterpart may be created and added so that functionality may be used immediately in gameplay.

Last will come the specific game requirements. Here the team will differentiate between Hearts and Spades with rules and gameplay.

At this point, the project will be ready for deployment. Attached is a PERT chart and timeline that illustrates the plan of action the team will take in order to reach the desired deadline and finish a successful project.

5 Risks

There are several risks associated with different parts of the game. The following table addresses the four most likely parts of the project that could create setbacks for the team.

	Design Logic	GUI	Game Logic	Network
Likelihood	Low	Low	Low	High
Severity	Extremely High	High	Extremely High	Medium
Consequences	No other task is completed-product failure	Will be a text based game instead of a GUI-based game	Gameplay fails, not able to play game-product failure	Unable to use multiple players across different computers-not to customer's satisfaction
Mitigation Strategies	<p>Schedule additional meeting times as a team</p> <p>This will increase meeting times from 2 hours/week to 4-5 hours/week as a team</p>	<p>Schedule additional team members to help with GUI programming.</p> <p>For each day over, an additional team member will be assigned to the task until all team members are working on it.</p>	<p>Will schedule additional meetings as a team to design and make decisions.</p> <p>Will increase team meeting times from 1-2 hours/week to 4-5 hours/week.</p>	<p>Schedule one additional team member every other day until all team members are working on the network portion of the software.</p>