**Design and Development**

Gantt Chart Link:

<https://docs.google.com/a/ugcloud.ca/spreadsheet/ccc?key=0Av4PxHJjzXdmdFFHSnB2cFJiTnRTSzZqVkVkWk93M0E&usp=sharing>

Problem Definition:

This project is to be a two-person turn-based strategy game, based on the webcomic Homestuck. In the game two players will take turns moving pieces around a two dimensional grid with the objective of destroying the other player’s “leader” pieces (these pieces are like the commanders of the players’ armies). Pieces can lose health in battle, which begins when one piece enters the square another occupies, and if a piece’s health reaches 0, they are destroyed. The outcome of a battle is determined by weighted chance, meaning that one piece may be more likely to be victorious, but that there is also a degree of luck. When a piece loses a battle, it is not necessarily destroyed, but its health is reduced. A player’s turn will consist of a certain number of actions, which consist of moving or healing a piece. Additional important details are that on certain turns, new pieces enter the game, and pieces possess special abilities of varying potency. Players of this game should ideally be accustomed to playing strategy games, but should be able to learn how to play the game regardless.

Problem Analysis:

1. Minimum system requirements to run the application:

Processor: Intel Pentium III / AMD Athlon MP

RAM: 256 MB

OS: Windows XP

Video Card: NVIDIA GeForce FX 5200 or ATI Radeon Xpress 1200 series

HDD Space: 50 MB

A keyboard and mouse are required to interact with the application.

1. User training;

The user must read the instructions manual included with the game.

1. Recommended age of the user:

8 years and above

1. Time the user will need to learn how to use the program;

Approximately 15 minutes to read the manual.

1. Costs associated with developing the software:

The main cost developing this game software will be in the many hours the developers need to put in.

1. Any user cost to purchase the application:

The application is free to purchase and distribute under the General Public License.

1. integration with other software

This software will be functional on either a *Macintosh* or *Windows* operating system, because the software will be coded in java.

1. research requirements

The members of the project must learn how to develop a game. This includes handling graphics, animation, and user interface to a higher degree than expected in the course. Additionally, someone on the project team should understand what Homestuck is.

1. time constraints

This project must be completed by June 12, 2014.

End User Requirements and Recommendations:

Contact Name: Bradley Cutten

Place of Employment: Centennial CVI

Date of Interview: May 23, 2014

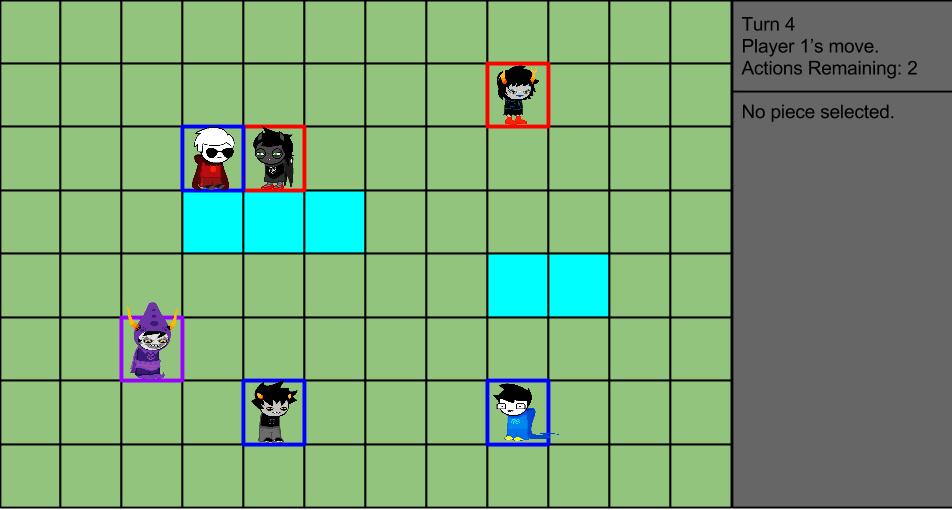
The contact recommended adding a function that would allow the user to save their games, given the long estimated play time. Beyond this, the contact endorsed the plan that the development team deposed.

Software Project Plan:

**Screen Designs**

*Starting Screen:*

This will not be the actual art, as the background belongs to a deviantart user and is being used as a placeholder.



*Main Gameplay Screen:*

Once again, these are placeholder sprites. Additionally, there will be an actual background in the game.

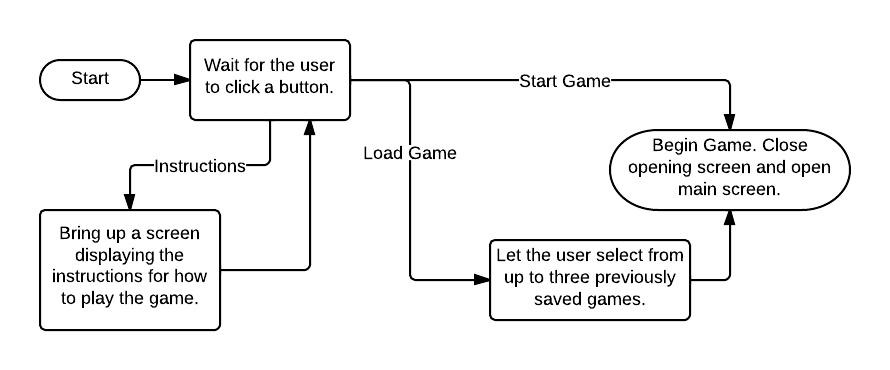


*Battle Screen:*

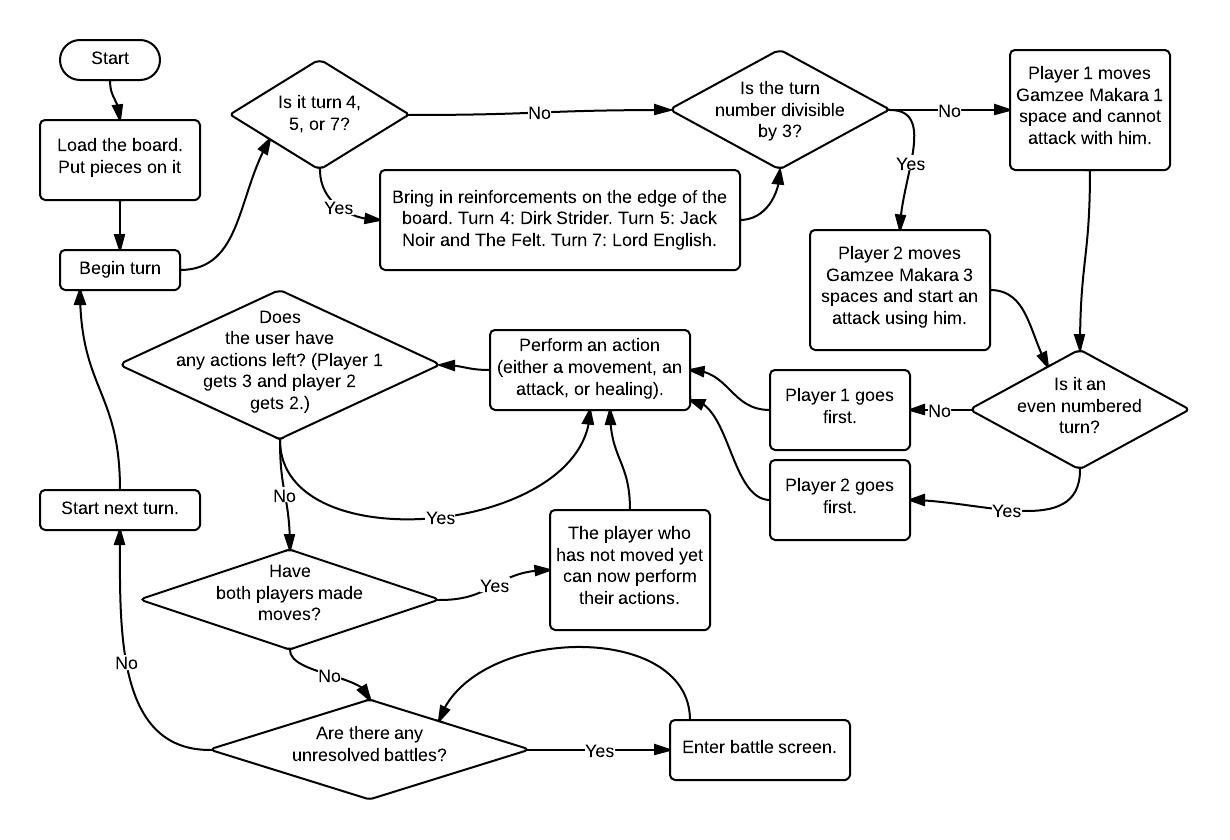
More placeholder images are visible here.

**Application Breakdown**

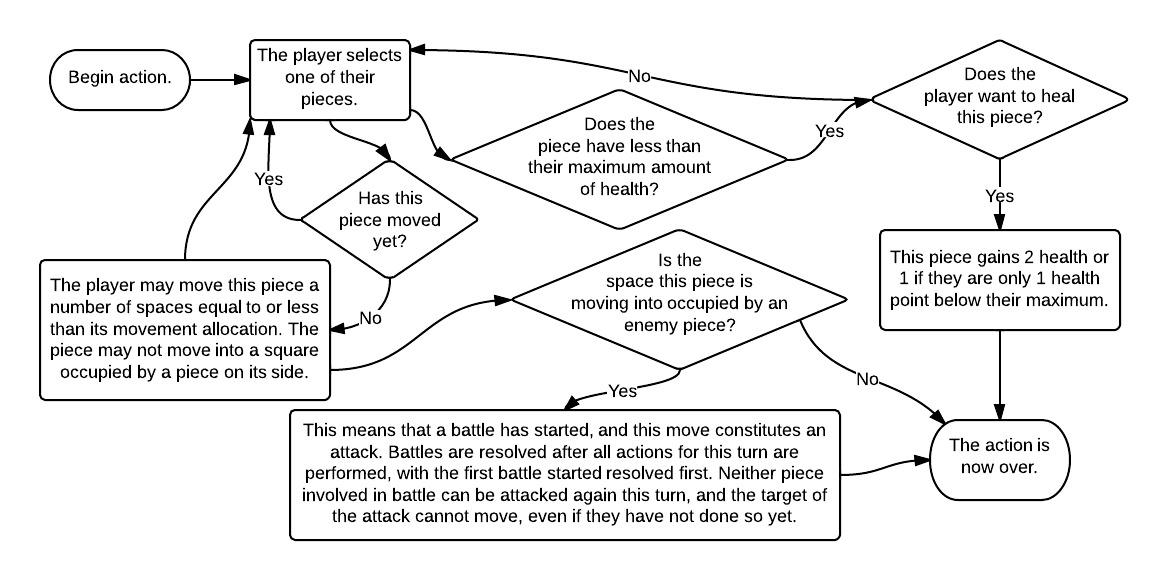
*The opening screen:*



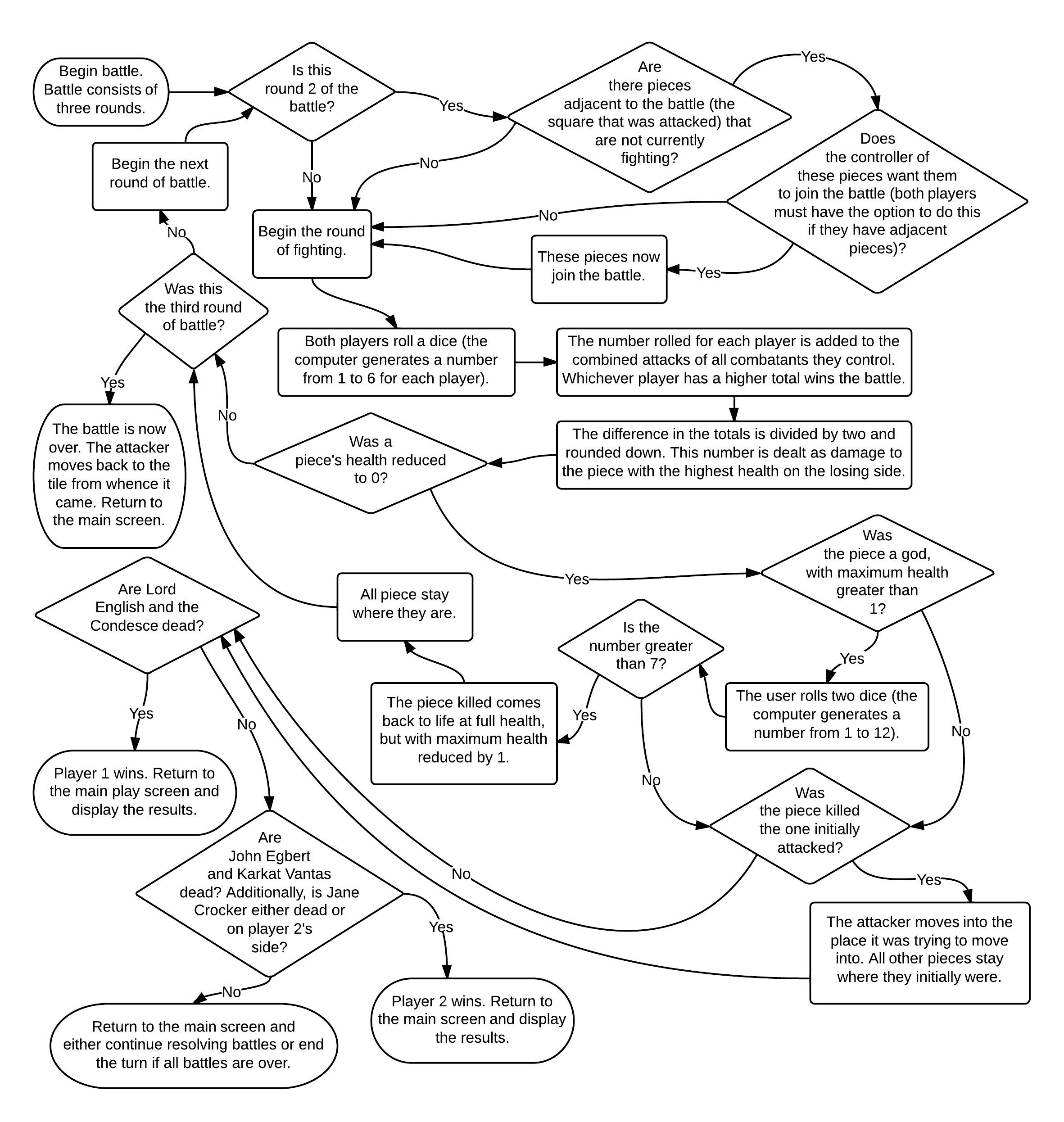
*The main gameplay screen:*



The main screen was already complicated, so some information on actions was left out. It is summarized below.



*The battle screen:*



**Resource list**

Karan, Michael, Mallachie, Maximilian, several computers, NetBeans, several image editing applications, access to the internet, time in class, and time at home.

**Work Breakdown**

Character Design: Mallachie

Board Design: Michael

Game Concept: Maximilian

Code Integration: Karan

Miscellaneous Coding: all programmers (Karan will most likely be in charge of coding for turn progression and Maximilian for battle sequence execution)

**Risk Plan**

Risk #1: insufficient time. In the event that there is insufficient time to complete the project, certain details can be cut out of the game, such as the special abilities of pieces and the computer vs player mode.

Risk #2: difficulty learning new techniques. If the team is unable to learn how certain new programming techniques work, we will first ask Mr. Cutten, then the internet. If neither is able to answer, we will modify our project so we do not need to use this new technique.

Risk #3: inability to integrate code. Should the team not be able to figure out how to put together code made by different people, we will meet in class and all stare at the code whilst taking turns explaining it to a multitude of fictitious rubber ducks.

**Approximate Schedule:**

*June 2, 2014:* have a functioning prototype, complete with moving pieces, health, battle sequences, and victory conditions.

*June 6, 2014:* finish implementing animations, special abilities, and save function.

*June 12, 2014:* finish polishing project and complete instruction manual.