**Design and Document – Feasibility Study**

**Problem Definition**

The program we have chosen to develop is a role-playing video game, commonly known as an RPG. It will include a top-down view overworld, a side-scrolling view in towns and dungeons, and a turn-based battle system. The end users we expect to use our program include gamers who enjoy video game series such as The Legend of Zelda, Paper Mario, Final Fantasy, etc.

**Problem Analysis**

We don’t expect hardware to be an issue, as almost all modern computers should be able to run our game. User Training depends on whether or not the user has player many video games before; if so, user training should only take a few minutes. Otherwise, it could take a while for the user to become comfortable playing. Integration with other software shouldn’t be an issue, as Love2D, the game engine we are using, supports Windows, Mac, Linux, Android, and iOS. Research requirements will also not be a major issue, seeing as we have some experience making video games. However, Israel is not familiar with Love2D, so some learning will be required. We have around 3 weeks to complete this project, which will probably be our largest constraint, as it greatly limits the amount of content we can include.

**End User Requirements/Recommendations**

The end user needs at least a 32-bit dual core PC, which must be running

**Software Project Plan**

Statement of work:

* **Intro Screen (Pseudocode):**

Draw logo and title screen

If user clicks play button:

Set up new game

Enter overworld state

If user clicks load button:

Get save file name

Load save file

If user clicks exit button:

Shut down game

* **Overworld state (Pseudocode):**

Draw 2D overhead view of the world map, centered on the player

If user hits menu key:

Enter menu state

If player enters event tile:

Run event

Get movement input state

Move player from movement input state

* **Sidescrolling state (Pseudocode):**

Draw 2D side view of the map, centered horizontally on the player

If user hits menu key:

Enter menu state

If player enters event tile:

Run event

Get movement input state

Move player from movement input state

* **Menu State:**

Draw background

Draw menu options

Draw player statistics

If user hits up key:

Set current option to the one above it

If user hits down key:

Set current option to the one below it

If user hits interact key:

Run the current option

* **Battle state:**

Draw player and enemy

Draw HUD and options

If user hits up key:

Set current option to the one above it

If user hits down key:

Set current option to the one below it

If user hits interact key:

If current option is “Attack”:

Attack enemy

Enemy.Attack player

If player health <= 0:

Gameover

If enemy health <= 0:

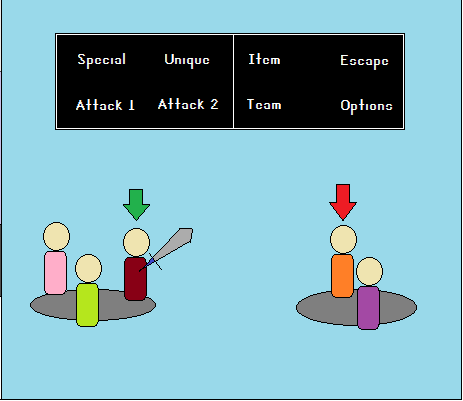
Leave battle

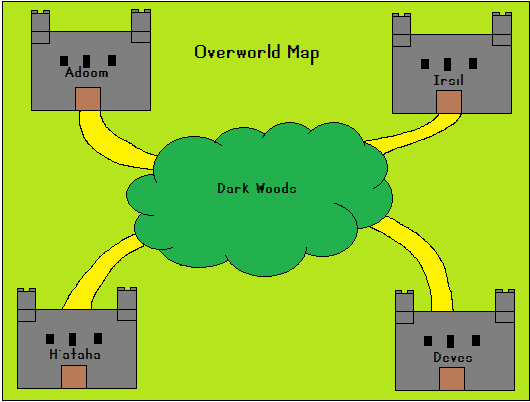
If current option is “Flee”:

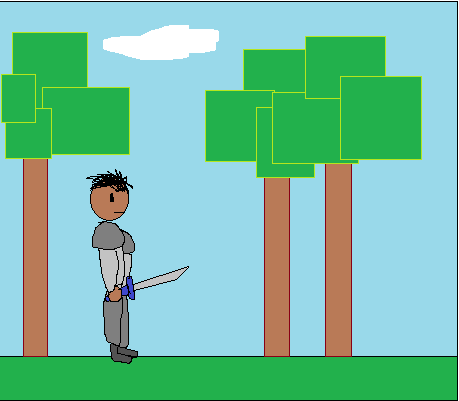
Leave battle

**Storyboards:**









**Resource List:**

People: Adam Vandolder, Israel Fowler, Davanjit Sandhu  
We are all available pretty much all the time.

We each have laptops, which are available all the time.

**Work breakdown:**

Intro state: ½ day

Overworld state: ½ day

Sidescrolling state: 1 day

Menu state: 1 day

Event system: 2 days

Input manager: ½ day

Player systems: 1 day

Graphics: 4 days

Content (maps, events, etc.): 4 days

**Project schedule:**

On Gantt chart.

**Risk plan:**

There are a variety of risks we are taking with this project. If we don’t have enough time, we may have to cut content, lowering time for events/maps/graphics.