University of Rochester Playing Game
Proposal v2.1.0

Member	Team
Bradley Beyers	Framework, AI, and Backend
Alex Hankin	Art, Plot, and AI
Santiago Loane	Art and Progression
Aaron McClure	Framework, UI, and Animation
Hayden Schiff	Framework, UI, and Backend
Graeme McGuire	Progression
Jacob Niebloom	Backend, Framework, Sound Design, and UI
Naropa Perez	Framework, Animation, and Sound Design

11/5/2014

Preface

This proposal's purpose is to serve as follow-up to our previous revised proposal and the comments received. Upon receiving the comments of our revised proposal, the U of RPG project group realized that we were unaware of what was expected of us for our revised proposal, and have decided to write a fully revised proposal where we provide fresh corrections and an updated look at the size and scale we want our project to have. We apologize for the misunderstanding when writing our previous revisions.

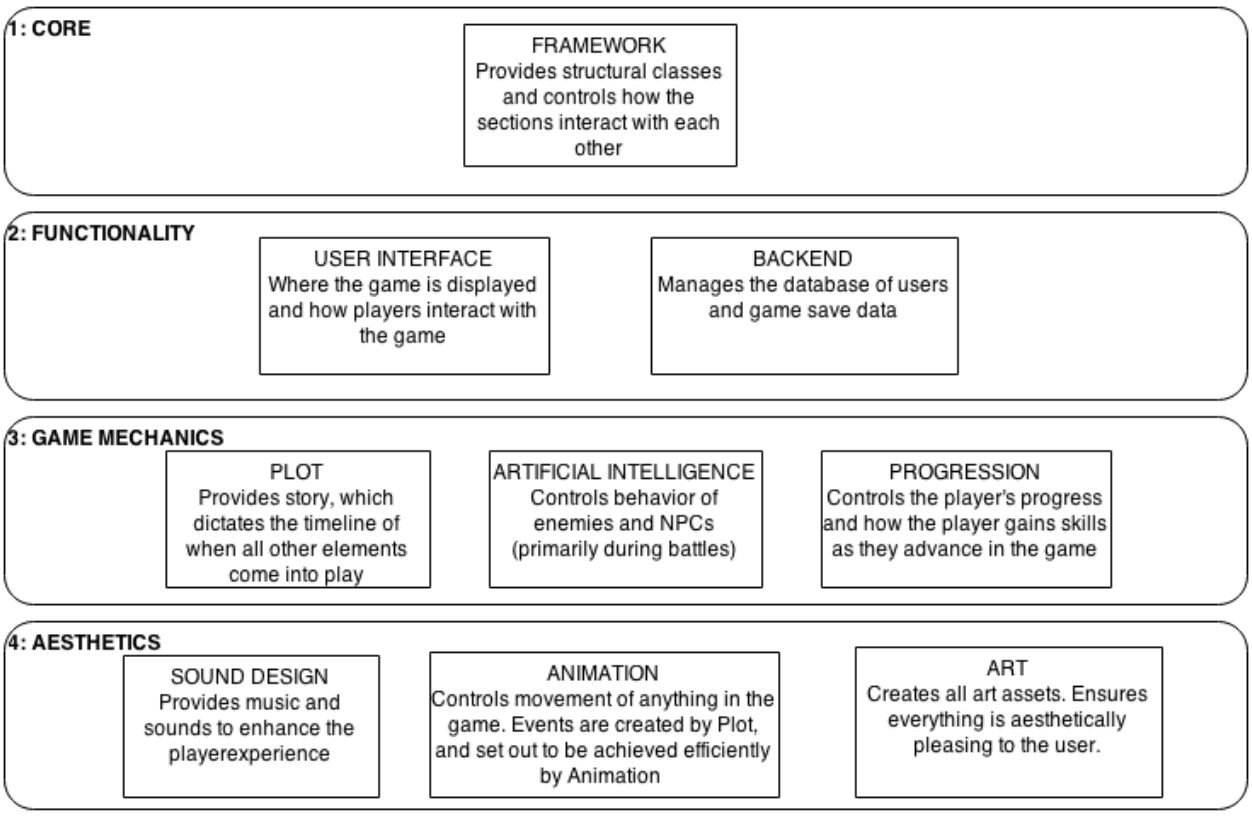
Introduction

The goal of this group is to make a role playing game that takes place on the University of Rochester campus. The game will be a 2D top-down game with simple activities and objectives the player must complete in order to progress in the game (see “Plot” section for more information). Due to time restrictions, we have fall-back plans to lower the complexity of the game in order to have a complete, working game by the deadline (see “Potential problems” section).

This game (as a work in progress) can currently be accessed at uofrpg.com. We are utilizing a private Github repository where we collaborate on code (see email attachment for sample). We have collectively written thousands of lines of code to date (11/5/14) and have a functioning website, database, and map/player generation system implemented. We have spent 10 hours each weekend working together as a group on this project as well as countless hours individually during the week. We are optimistic about the outcome of this project, however, we are working hard to create a simple “working game” before adding more ambitious additions.

To create our game, we are utilizing several different libraries. EaselJS is the main library we are using to create the game itself. EaselJS is a very primitive library; however, it does provide us with a few very useful functions to avoid “reinventing the wheel.” Additional libraries include jQuery, Bootstrap, Parse, etc. Each of these libraries are integrated throughout our codebase to provide us functional methods of manipulating data.

Groups



Animation

Team: Aaron and Naropa
Purpose: Controls movement of anything in the game. The primary forms of movement are on the map due to plot related events random, npc movement, and fighting scene animations. Takes input from plot to make events happen on the main map. The main priority is to make everything at least functional for game play. After this, everything else is extra for memorable and convenient gameplay.

Progress: Currently, player movement and map generation has been Animation's highest priority. These are the most essential animation pieces that are necessary for game play, and luckily both of these goals have now been successfully completed. Using the tile system, any one tile on the map can easily be referenced, and checked for collision, which is going to be crucial for plot development of the game.

After this, the battle scene is the next priority. We must have dynamic and interesting animations for the fighting scenes. Currently we have plans of making the fighting scenes look something like this:

Final Fantasy™ Dimensions owned by Square Enix

Animation should create functional and entertaining, yet generic animations for different moves.

If unable to do this, we can always set the bar lower by just drawing the sprites we already have on the screen, and just the selection process working, which we know we can 100 percent do. This way only numbers and health points would be updated, while the ai and player take turns playing.

Requirements to get started: All animation needed was a couple generic art assets. The black and white map has been available since the first week. Animation has everything it needs. There is no excuse for it to be hindered by any other group.

Framework

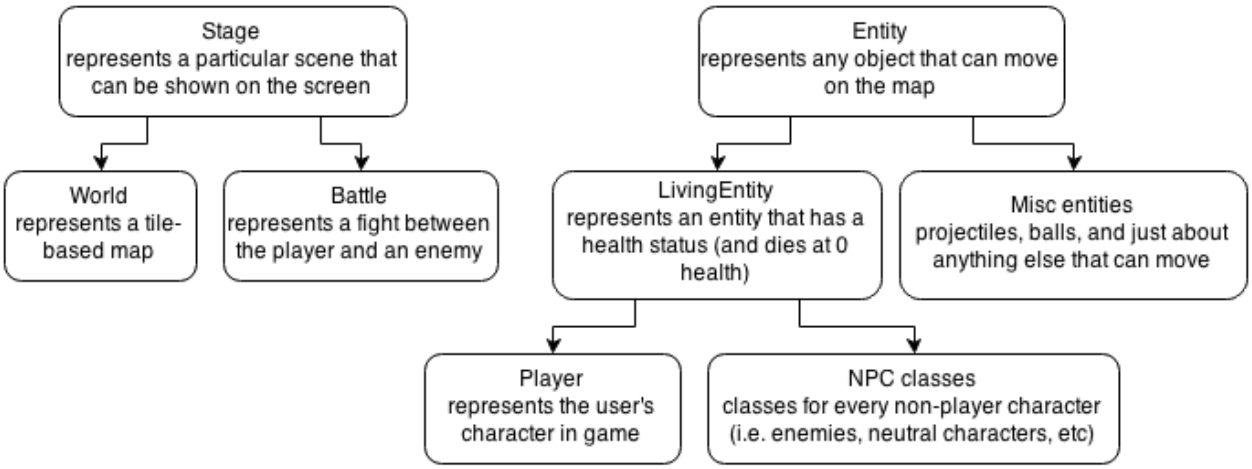
Team: Hayden, Bradley, Jacob, Naropa, and Aaron
Purpose: Framework serves to provide a structure for the rest of the game to build upon. Framework involves creating classes and interfaces that make up the basic building blocks of the game.

Progress: Framework has been progressing very excellently. Originally a lot of our code was very rough and quick, hardcoded just to get a proof of concept functioning. However, we have now put time into building a proper framework. We now have an Entity class that represents any object that can move around on the map, and a Player class that extends this.

We also have assembled the basic structure for allowing a map of square tiles to be displayed, with the viewport centered on the player. There is also collision, so the player can only move in some tiles but not others. Once we have additional maps created (i.e. building interiors and such), the map code will be moved into a World class that will allow maps to be easily switched when the user, say, enters a building. Most of our code is already largely written modularly, so

the transition to using a class should be easy.

The following diagram shows our current planned classes for the framework. Arrows represent inheritance, and point from superclasses to subclasses.

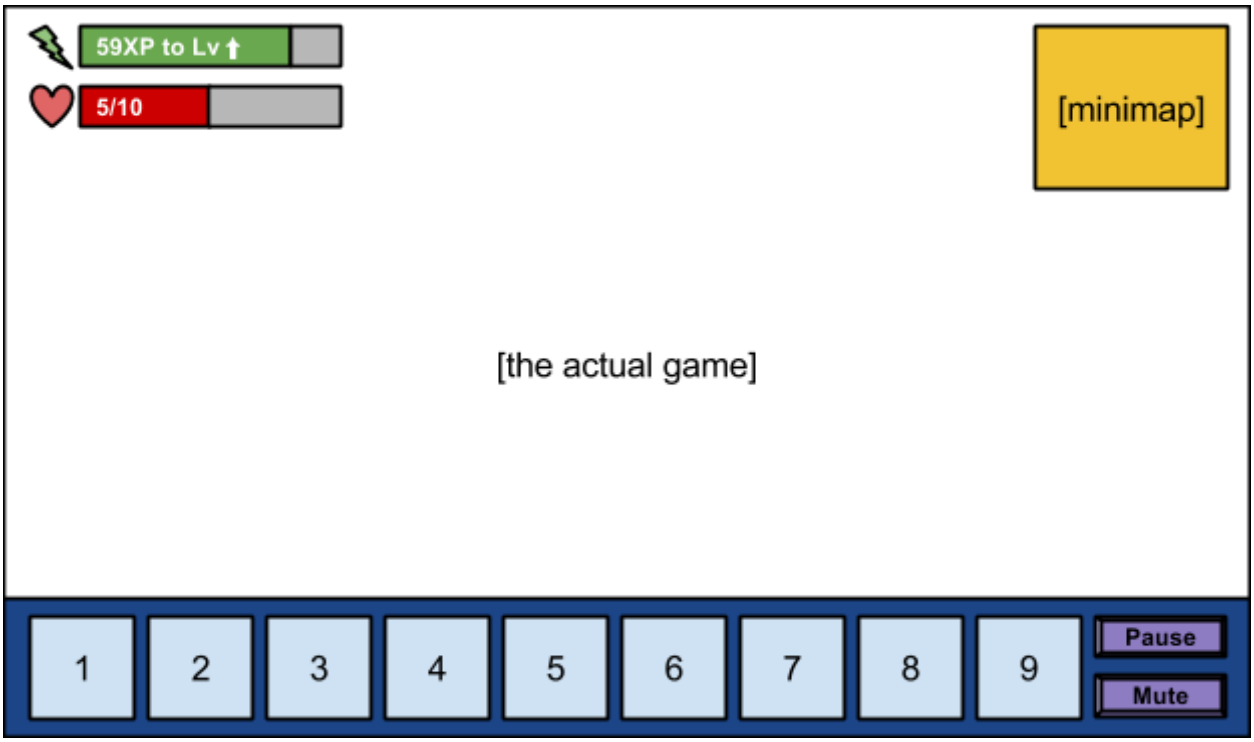


User Interface (UI)

Team: Aaron, Hayden, and Jacob
Purpose: The goal of UI is to provide the means through which the player can interact with the game. Everything displayed on the screen is a result of the UI team. UI must be able to interact with each of the other groups in order to display the correct information on the screen.

Progress: The user interface is starting to take shape. We already have the interface for logging in and registering pretty much worked out. We also have the basic controls for movement worked out. The player moves with the arrow keys (will likely also permit WASD soon), and can sprint by holding Shift. There will also be an action key (the default will probably be E or Space) to allow the player to interact with whatever they're currently looking at in the game (e.g. for reading signs, talking to NPCs, opening doors, etc).

Our team's focus now is to create a HUD (heads-up display). Below is a mockup of the game's primary HUD:



In the top left corner, we have an XP (experience points) meter and a health meter, showing the player's character's current status (the experience meter is just for sake of example, and will probably change based on input from the Progression team). In the right corner, we have the minimap, giving the user a zoomed-out birdseye view of the area around them. At the bottom, we have the inventory. There are nine inventory slots, which will be bound to the number keys 1-9 (i.e. if you want to use a sword that you have in slot #4, you just press the 4 key to switch to it). There's also a pause button to pause the game (the Esc key will also do the same thing) and a mute button to toggle game sounds/music.

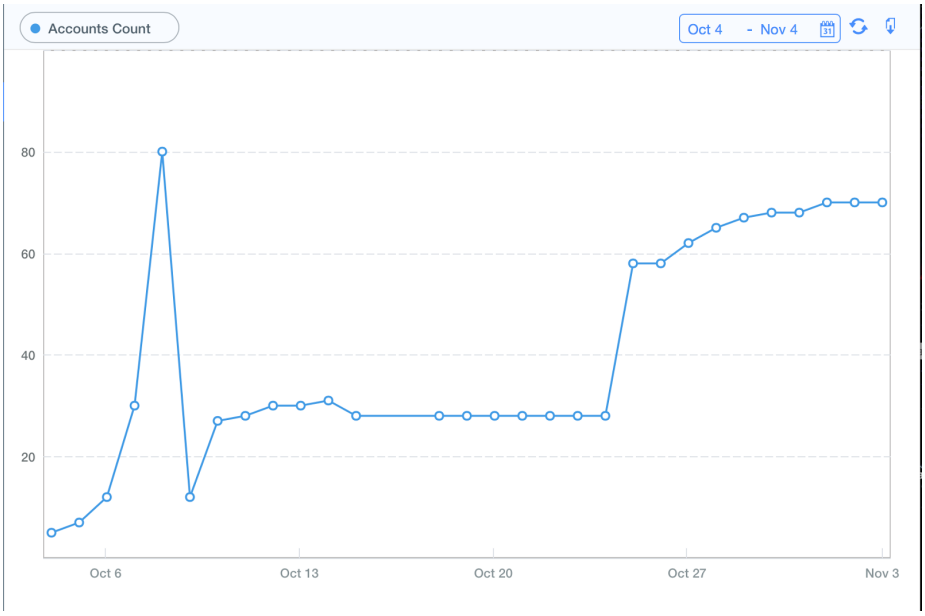
Backend

Team: Jacob, Bradley, Hayden
Purpose: The backend is the server-side element to our otherwise client-based game. The server backend provides a system for users to log in to personal accounts and access their game save data and any other personal data relevant to their game progress.

Progress: At present, we have a basic backend in place for managing user accounts. Users can register for an account and login with their credentials. The account data is stored in a database using the service Parse. This functionality needs a few tweaks to be complete (e.g. password hashing, etc) but is currently operational at a basic level.

In addition, the backend needs to host user data. The user will be able to save their game via a button in the pause menu. Saving the game will upload the user's data (e.g. current health, inventory, location, progression, etc) will be uploaded to Parse. They will also be able to view the leaderboards, which will contain the score and bits of game data from all other users.

Here is a graph showing the number of accounts that have been created in the database over time. (the fluctuations occur due to occasions when the database was modified or purged of invalid/ unused accounts)



Artificial Intelligence (AI)

Team: Bradley and Alex
Purpose: To create interesting enemies for the player to fight. Essentially, planning out what each enemy is going to do in plain English and then translating that into code. This will involve writing routines that make the enemies behave in intelligent ways, making the game more challenging and interesting for the player.

Requirements:

- Basic framework so we can actually put enemies into the game (Completed)
- Art assets to use for the enemies

Progress: AI will also possibly be involved in giving characters random responses to the player when talked to to make the world a little more dynamic and alive. It will also be responsible for the "random" paths characters will wander around in.

Once we have just a little bit more groundwork as far as making combat function in the game, AI will be able to begin its main function as a way to spice up the game a little bit in a multitude of small ways that add up to have a large impact on player enjoyment of the game.

The actual requirements for AI are surprisingly low, since we're basically building all of this from the ground up. In addition, in the absence of AI, we can substitute randomness (have enemies randomly select an action from a list of possible actions instead of having them choose intelligently) for intelligence in the unlikely event that we never get around to actually creating AI.

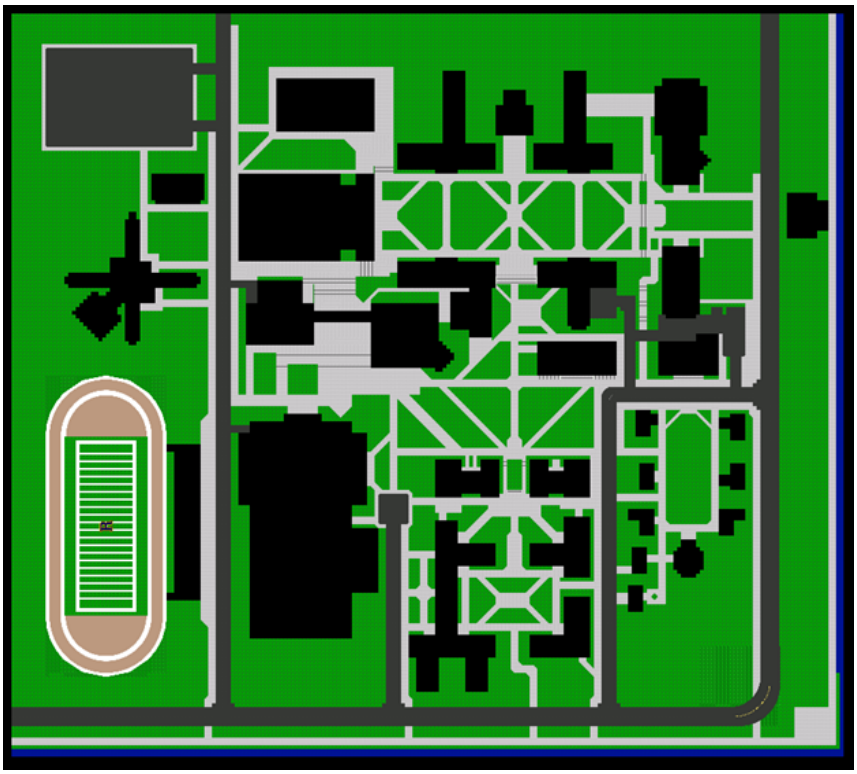
Art

Team: Alex and Santiago

Purpose: Supply art assets for the game. This includes making tiles to make objects as well as maps and creating sprites, some that will and will not be animated.

Progress: Art has been going very smoothly. Two members of our team have been consistently working on aspects of art. One member has made considerable progress on our sprites as shown in our previously submitted proposal:

Our other art team member has recently completed a basic terrain-filled overworld map using around 65 original tiles, and will be implemented into our website shortly.



We have decided upon using 32px x 32px tiles, this gives a reasonably expansive landscape, while at the same a manageable size to create. All NPCs will be 2 tiles high and 1 tile wide. Art for us is clearly not an issue. Thanks to the creation of the overworld, we have a sense of spacing of the world we created, and our map template gives a sense of scale for how big we want our other environments to be such as the inside of buildings. We have been able to produce sprites efficiently, and we believe animation will not be too difficult because of the simplistic style of the sprites. We have plans to add in new NPCs with the template we have right now. With a template ready, it will take very little time to create many different NPC's. We have shown that we are capable of map editing and can create enough locations and tiles to meet our project needs. Now that we have learned the basic techniques for map editing, creating and implementing tiles has become a noticeably more efficient process. As long as art remains the high priority we have given it, and our scope of the game remains at a reasonable size (which is something we have realized may need to be smaller than originally intended), the project's art assets will have no problem being completed.

Plot

Team: Alex

Purpose: Plot provides the story behind the game. This gives a logical progression of events and provides the basis for the rest of the game mechanics to flow together.
Progress: Plot is currently not one of our main priorities, as the plot will need to wait until we have completed the final framework of our project. We have currently begun reworking what we are capable of in the time given. While plot will definitely not be a "last-minute addition" we have decided to wait until the finalization of our Mechanics and our Framework. Our purpose remains the same however, to give the player a reason for continuing, and a story to follow.

Sound Design

Team: Jacob and Naropa
Purpose: Music and sound augment the game experience by setting the mood or tone of the current scene. This enhances the user experience and deepens the player's immersion into the game.

Progress: As of now there is not much going on in music. Until we have a real idea of what the game will feel like, we do not plan on beginning the composition and compilation of Music to the game. The music group is starting to gather some smaller sound clips for certain sound piece we know will be in the game; such as rain, bumping into something, eating, and walking/ running.

Progression

Team: Santiago and Graeme
Purpose: Each member of the group will appear as a playable character. Each character has a different "major" to them, which corresponds to a different class that one would find in a typical adventure RPG. Furthermore, each student will have to take "classes" as chosen by the player, which will be a means of leveling up their skills along certain branches and unlocking new abilities.

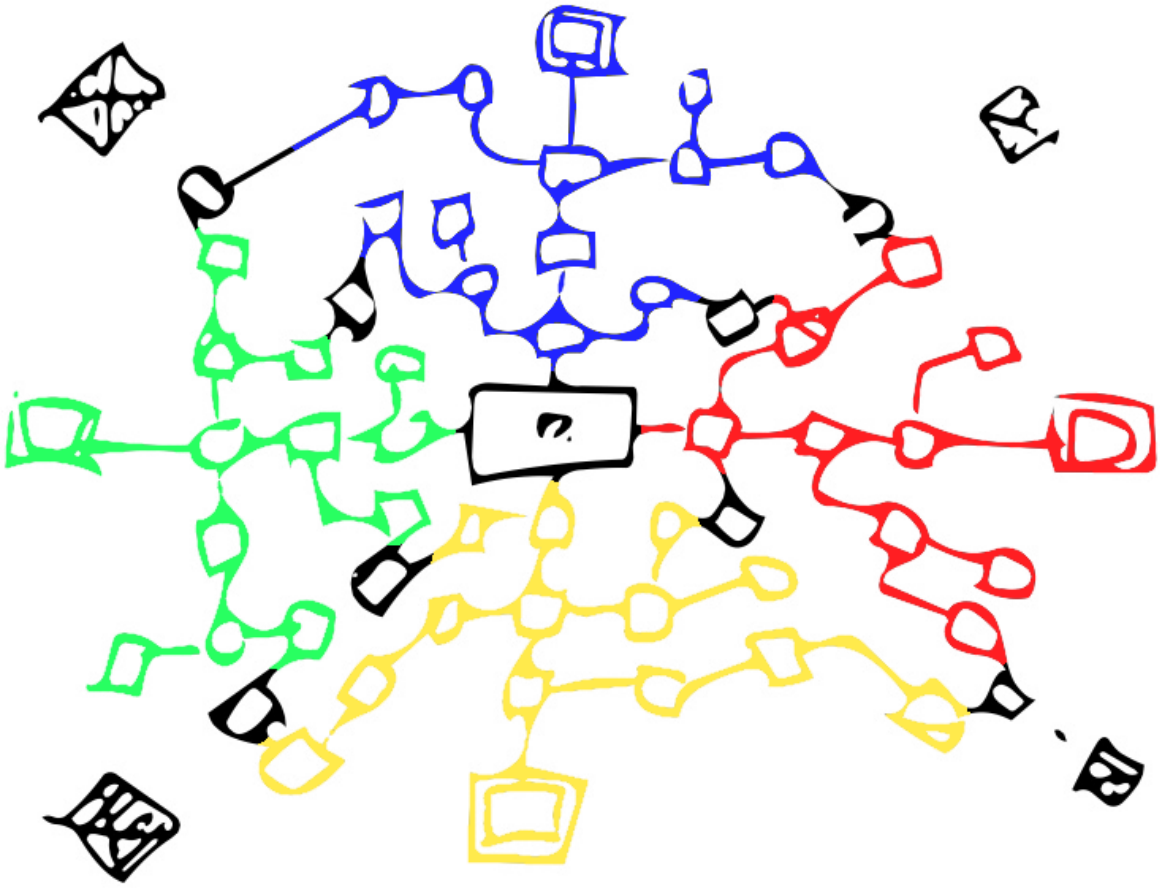
As appears in most RPGs, we will be following a general "skill tree" archetype in the structure of this game. This means our characters will be progressing incrementally by levels (achieved by receiving certain amounts of experience), and the player will be able to choose which character archetype they choose to follow as they progress. Players will be therefore have the option to fulfill one of a large variety of skill paths, which will lend an important element of customization to the game. The progress needs to be saved and sent to backend and translated to a total score for ranking purposes.

RPG CLASS	CORRESPONDING MAJOR	PLAYER	SPECIALIZATIONS: STR | DEF | CHA | INT

Warrior	Business	Naropa	Berserker | Knight | Monk | Warlord
Ranger	Math	Alex	Hunter | Dual-Wielder | Beastmaster | Sharpshooter
Mage	Physics	Santiago	Battlemage | Warlock | Druid | Psychic
Healer	Pre-Med	Hayden	Paladin | Necromancer | Illusionist | Cleric
Gunner	Mechanical Engineering	Aaron	Assault | Heavy | Sniper | Engineer
Alchemist	Chemistry	Jacob	Bomber | Wall | Shaman | Trapmaker
Rogue	Psychology	Brad	Pirate | Thief | Scout | Ninja
Bard	Music	Graeme	Battlebard | Dancer | Singer | Buffer

A potential distribution of classes and sub-classes depending on skill distribution

Progress: We know all players will have health points, charisma, physical ability and intelligence whose total can be increased according to which skill the player chooses to level. These are the four skills the player can choose from to progress or level up throughout the game. Each side has different perks that have not yet been decided, but should make sense with the class's name. A player can have perks from two branches next to each other, but not to the extent as if he were to purely level up towards one of them. As such, players can hybridize their preferred "skillset" in order to achieve a more diverse character at the cost of having a less specialized one and barring progress to higher-tier skills.



Mockup Skill Tree

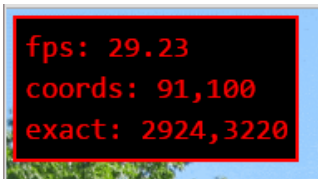
Miscellaneous

Team: Everyone
Purpose: In addition to all the group-based work, there is inevitably some work that doesn't really fit into any team's scope, but isn't substantial enough to justify the creation of a new team. Everyone does a little bit of this type of work. There's a lot of miscellaneous code that has to be written to make each individual team's code function in the larger project as a whole.

Progress: There are a lot of tools that need to be created that don't factor into the final product but help along the way. To this end, we have created a debug mode for developmental purposes. Debug mode can be enabled by putting "#debug" at the end of the URL (i.e. http://uofrpg.com/#debug) or by pressing the grave/tilde key.

Debug mode offers several features. If you use the "#debug" technique, the login screen is automatically bypassed (so we don't have to login every time we test something). Once in the game, debug mode adds a box that gives the current FPS (frames per second; used to monitor how smoothly the game's running) and player coordinates on the map (expressed both in terms of the tile grid and in terms of pixels). If you hold Ctrl with debug mode on, collision will be disabled, allowing the player to pass through walls.

Here's what debug mode's informational box looks like:



Future milestones

- Get a functioning prototype of the game, with at least a slight element of all areas of the game implemented. A “snapshot” of the game, in essence. Alpha - 11/19/14
- A Beta version of the game, where everything is fully implemented but rough around the edges. A “rough draft”, in essence. Beta - 12/1/14
- Final version of the game - 12/6/14

Potential problems

- **Size.** We are aware that perhaps the scope of our project may be a bit too large for the time we are given. However, we believe that for the most part, everything we want to achieve is feasible for the time we are given. If within the next couple weeks we discover that what we need to cut certain aspects, we will add this to our progress.
- **Differing Coding Languages.** There was an initial concern that certain languages of our project such as the map editor, game editor, and HTML files wouldn’t be compatible with one another, however we have since made sure that this would not become an issue.