

U of RPG

Meliora - Online

V. 1.0

BY

| | |
|---|---|
| Alex Hankin – Author and Co-Head of Art | Graeme McGuire – Head of Game Mechanics |
| Santiago Loane – Head of Art | Hayden Shiff - Head of Web |
| Bradley Beyers - Head of AI | Aaron McClure - Head of UI |
| Jacob Niebloom - Head of Backend | Naropa Perez - Head of Animation |

10/11/14

Introduction

An RPG really? Is there something about role playing games that makes you cringe? Why is making an rpg the go to project for 8 young programmers to tackle? The answer is simple; role-playing games allow players to escape reality and become something more. What's more, we have the opportunity to turn our own lives here at the University of Rochester into an online game, and have the community we live in play this game aswell. This us a unique opportunity, which explains why when someone said "hey guys lets make an rpg game," a light lit in our hearts. The purpose of U of RPG is to create a virtual reality of the University of Rochester, and make it gorgeous, entertaining and iconic. We want perspective students who are exploring Rochester to stumble across this game and know that first semester freshman made this as a class project. Indeed, why we chose Rochester because not many other schools would allow a group a freshman walk in, throw money at them, allow them to make a semester long project and give them credits for it. Ultimately this game is a gift to the institution that gave us this privilege.

Game Design

Plot

Purpose:

Provide a story while going through the game (i.e. events, dialogue). Does not have to be complex, but it should provide incentive for the player to want to continue the game

How it links to other sections:

- Progression and Plot are very entwined and are thus going to be constantly working with one another.
Examples: How achievements relate to the story, dialogue for progressing through the game
- Game Mechanics and Plot are going to need to talk to one another to make sure the story incorporated the usage of the game mechanics
Examples: Acquisition of certain abilities should be plot related, the types of mechanics nvolved should impact the direction the story
- Plot will also be influencing certain aspects of art and animation to make sure parts of the story are implemented

Requirements in order for plot to function:

Most of the plot will be written down and executed using art and animation. However, art and framework are going to need to be contacted o make things such as dialogue boxes.

Planning:

- A. Wait for a majority of Game Mechanics to be completed.
- B. Write a synopsis and summary of a planned story
- C. Start making drafts of a script that includes dialogue for main quest and side quests
- D. Write Final draft for story and dialogue
- E. Work with art and animation to create specific animation and dialogue boxes to fit story
- F. Implement dialogue into the game

Art

Purpose: Make the game gorgeous, consent, and simple. Can make/break the game, truly key.

Planning:

1. NPCs/Player creation
2. TILE map creation
 - ***GENERIC MAPS FIRST, so animation can function.
 - must be flexible tiles that easily work together to make a really elegant map
 - a finite definite list of all tiles needed
 - need to be spaced and carefully planned. There is a massive amount of tiles that need to be created since we chose to make these.
3. improve Animations that work with game mechanics, such as pause/text screen

Style

We decided to make the game 2D and orthogonal. The map layout will be similar to that of Pokémon. In order to keep the art consistent and to achieve our ideal product, we chose to create the tiles ourselves. The main map of Rochester will be a square where most of the game will take place. This is an example art asset of our potential main character.



User Interface

Purpose:

To display all content to the user. This includes everything from menu options to gameplay graphics and dialogs.

How it links to other sections:

- User interfacing displays the work generated from the other groups to the user. Therefore this category is responsible for rendering work from all groups. It is important to communicate with game mechanics/framework.
Examples: Display game logic to user as dictated by game mechanics/framework.
Visually depict what is happening in the game.
- User interfacing also must communicate with animation.
Examples: how animations are displayed, buffered, rendering priority, framerates, frequencies, iteration, etc.

Requirements in order for User Interface to function:

All sections need to easily be able to interact with the User Interface. UI must standardize methods/api for interacting and outputting. Input must be accurate and functional in order to display to the user.

Planning:

- A. Setup containers (via EaselJS library) for switching between outputs.
- B. Create standard components (buttons, labels, etc.)
- C. Creation of menu system and setup website layout, look and feel, etc.
- D. Loading of game and required assets/libraries
- E. Rendering of game. Correctly display gameplay and information to user
- F. Respond to user actions/input

Programming/Division of Labor

Animation

Purpose: To elegantly update sprites for gameplay. The animations should be clear to understand, sharp and smooth. Also, animations should be generic, easily re-used, and adaptable.

How it links to other sections:

- Game mechanics is essentially going to dictate how animation works.
Examples: how text screens work, where should they appear how fast should the character walk across the map
- Other section also need to easily interact with Animation.
Examples: Coordinate on map where events for should take place. Placement of items/east bunnies

Requirements in order for Animation to function:

Generic map that marks collision areas, and sets a finite coordinate system on the map, and a player, represented by a rectangle to animate on generic map

With these , even if all art assets are lost, animation can keep its functionality. Also, other sections aren't affected as well

If Animation didn't exist, then the game should be able to run complete text based.

Planning:

- A. Making player moveable on map.
 - B. Making player unable to move on certain tiles
 - C. Generic text animation that take input and elegantly presents it on screen *game mechanics*
 - D. Pause Screen *game mechanics*
 - E. Easily switch map/screens.
 - F. all minigames and ulterior generic maps
- *changes as game develops and not yet planned at all*

AI / Enemy

Purpose:

To make the game more interesting by making enemies behave dynamically based on the situation.

How it links to other sections:

Game mechanics will decide what behaviors each enemy should have, and then backend will use the routines we create to make enemies behave the way we want.

Resources Required:

Basic framework from backend so we can easily create many different enemy types to create their behavior.

Details from game mechanics about what kinds of behavior we should be creating.

Planning:

- A. Wait on game mechanics to give us the descriptions of what each enemy should do.
- B. Work through each description and translate the prose to code.
- C. Apply these routines to the actual game code to make enemies behave like we envisioned.

AI / Enemy could go down without the group being totally lost, since enemies could just be programmed to behave randomly. It would just be a lot less fun that way.

Framework

Purpose: To bind together the frontend and backend elements to form a cohesive game experience. Essentially the glue to make sure everything works out smoothly.

How it links to other sections:

All other sections will eventually be brought together by framework in order to create the final product.

Resources Required:

More or less all the things that come in from other groups will eventually go through framework to be assembled.

Planning:

Take the elements as they come in and put them together. Tough to define a timeline independent of the other groups since it's so closely tied to them.

Backend:

Purpose:

To provide a fluid gameplay, a structured database, keep the server running, updating the server, having fluidity and responsively throughout gameplay and throughout retrieving and manipulating data.

How it links to other sections:

- It is the backbone of every section. Without a backend, there is no way to display a front end.
- Each section must rely on the backend to elegantly connect each and every section of the project together.
- The backend will also deal with the database that stores all the information about the game and each individual player,Âs stats.
- This section is simply the structural integrity for the rest of the project.

Requirements:

Server
Database
Frameworks
A Front End
Domain

Planning:

- A. Create accounts table and credential storage.
- B. Delivery game to user via website.
- C. Serve up the game. (Prototype.)
- D. Store user data of users playing.
- E. Create the backend for whatever idea that will inevitable come up throughout this game making process.

Deadlines for Check Points

Oct 25-Functioning map/player on website

Nov 1-Function UI, Skill tree, Quests and online ranking

**Nov 22-Functional product for Beta Testing
Dec 6-hand in**

Prospective Problems

The number one prospective problem is the fact there are eight of us. As one of us said, “I’m worried that too many cooks ruin the brew.”

- The beauty of the rpg project is that as a work of art it is never finished. The eight of us have more than enough work to do, and can work on the project straight up through the due date.
- Organization is key. Hopefully this proposal gives a sense at how much emphasis we are putting into organization. I believe we successfully split this project up into multiple parts. Together these parts will create the ideal product, but even if one section malfunctions, the whole will still exist. We stressed how to make each major section generic for this purpose.
- Having eight of us can be seen as a problem, but we also see it as our greatest strength. We are never short of imagination and ideas. What’s more is the diversity within our group. At one corner we have a person who enjoy sitting down working on website, and at the other we have another member that exclaims, “I WANT TO MAKE ALL THE TILES.” We are all equally passionate for the separate parts of the project, that together, I believe we will create an amazing product that would surprise everyone.
- Our framework section’s purpose is to prevent any problems specifically due to this issue.

Is the project changing enough? Is there anything that would be too difficult that it would affect our product? Essential have we found a balance between pushing our limits and not attempting the impossible?

- There a lot of dark spots that we still do not know how to make; however, we are sure that with time and effort we can pull through and solve these problems. I believe these are the characteristics of having a sufficiently challenging project. I believe that this project meets this requirement to the fullest.