U of RPG
Meliora - Online

V. 1.0

BY

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

10/11/14

Response to Criticism

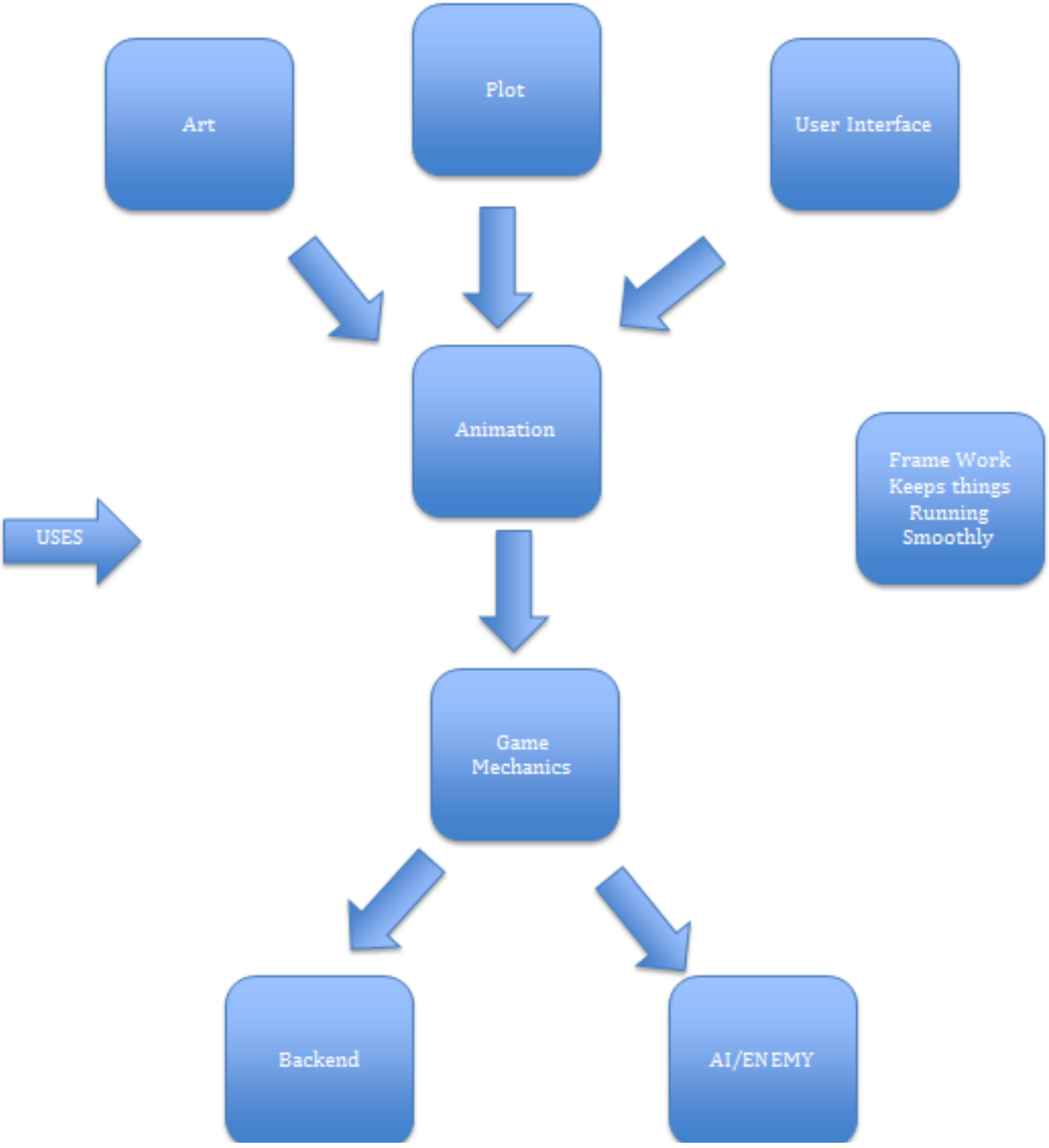
Most criticism to our present proposal was brought about over concerns of the complexity of our project. While we recognize that our ambitions are high, we are confident that we will be able to reach them with such a large team. Much progress has already been created as we have spent 10+ hours each weekend working and collaborating as a group. It is important, however, to always have fall back plans as mentioned if certain parts fail/lack required functionality.

As a result, we have already created a basic structure to prevent an overall disaster. We have a working website, database system, and structured game mechanic engine already established. We believe that a bit of underestimation occurred, for we are beyond the point of going anywhere near creating a text based game. Please visit uofrpg.com, and you can see that we have come a long way. We have a functional moving object with a virtual model of the campus. As for the art assets bellow we have the progress that we have made. Adding the art assets in is a quick step because of the flexibility of our code. We will build the game with the minimum functionality before adding additional features. If our existing framework were to somehow fail, we do have systems established to replace complex functionality with text-based (lower level) input. As most structural issues have already been resolved, we do not see this being an issue.

You also expressed concern of certain groups becoming overwhelmed with work. We have made several adjustments as certain groups have assumed mutual responsibilities to balance the workload. During our weekend collaboration sessions, we are able to scale our resources collectively toward the "chokehold" points you mentioned. We have set an additional co-head of UI (Hayden) to help manage much of the work this job entails.

(sample art asset modeled after each of us, we also have modeled over 30+ other tiles)

(see flow diagram next page)


Art
Plot
User Interface
Animation
Frame Work
Keeps things
Running
Smoothly
USES
Game
Mechanics
Backend
AI/ENEMY

Introduction

An RPG, really? Is there something about role playing games that makes you cringe? Why is making an rpg game the "go-to" project for 8 young programmers? The answer is simple. Role-playing games allow players to escape reality and become something more. Furthermore, 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 as well. This is a unique opportunity, which explains why when someone said, "Hey guys let's make an rpg game," a light lit up in our hearts. The purpose of U of RPG is to create a virtual reality game 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, this is why we chose Rochester. Not many other schools would allow a group a freshman to 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 involved 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:
UI should 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

Game Mechanics

Purpose:
To carry out the gameplay on all levels, and to identify the structures, paths, functions & choices that will make up the process of actually playing the game. The biggest aspect of this (and the largest distinction) will be the overworld mechanics versus the combat mechanics.

How it links to other sections:

* The game mechanics are in effect what will be implemented by way of the game's other sections. For example, through the user interface and sprite art, we'll visually display the functions of the overworld: moving around, character interactions, activating the functions of different buildings and items on campus. Additionally, it connects to the combat system similarly - the visuals will display the implementation of our combat mechanics.

* The game mechanics are slightly less interdependent with the game's plot in a general sense, but overlap exists, and we will need to make sure the progression and pacing of each one coincide appropriately.

* Some smaller sections, like AI and combat, are completely under the umbrella of game mechanics - we'll divide ourselves into smaller groups to deal with the finer points of these sub-areas when we've established the mechanics on the broadest level.

Planning:

A. Settle on the broadest aspects of gameplay: possibilities for overworld and combat, plus skill trees and character development.
B. Implement broad-level mechanics decisions in code, before focusing on more narrow aspects of issues like enemy behavior, exact techniques/items, etc.
C. Implement specific implications of character choices - particular skill tree routes, etc.

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 sections 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:
Providing a fluid gameplay, a structured database, keeping the server running, updating the server, having fluidity and responsivity throughout gameplay, and retrieving and manipulating game 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's stats.
- This section is simply the structural integrity for the rest of the project.

Requirements:

Server
Database
Framework
s 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 any idea that will inevitably come up throughout the game making process.

Deadlines for Check Points

Oct 25-Functioning map/player on website
Nov 1-Function UI, Skill tree, Quests and online rankingNov 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 people who enjoy sitting down working on the website, and at the other we have others exclaiming, "I WANT TO MAKE THE MAP AND TILES!" We are all equally passionate for the separate parts of the project, that together, I believe we will create an amazing product that will surprise everyone.

Is the project changing enough? Is there anything that would be too difficult that it would affect our product? Essentially, 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.