Project Rome



Bill Munshaw and Jeremiah Trahern

Dominion Alliance Council on Research and Development

Table of Contents

[Vision Statement 3](#_Toc462583892)

[Requirements 4](#_Toc462583893)

[Categories of Users 4](#_Toc462583894)

[Actor-Goal List 4](#_Toc462583895)

[User Stories 4](#_Toc462583896)

[Product Backlog 5](#_Toc462583897)

[Sprint #1 6](#_Toc462583898)

[Review 6](#_Toc462583899)

[Retrospective 6](#_Toc462583900)

[Sprint #2 7](#_Toc462583901)

# **Vision Statement**

Project Rome will be a strategy game that takes place in virtual reality. Unlike normal strategy games such as Civilization and Total War, Project Rome will feel like interacting with a physical board game and offer variety in perspective, allowing users to manipulate the board from an overview and also see their creation in a first person point of view. Project Rome will appeal to fans of strategy games and bridge the gap between table top and computer games.

# **Requirements**

## Categories of Users

**Players** – Participants in the game

## Actor-Goal List

|  |  |
| --- | --- |
| **Actor** | **Goal** |
| Player | Initialize a game |
|  | View and modify available rules and settings |
|  | Start and play the game |
|  | Assume control over the in-game world |
|  | Change view controls and issue orders |
|  | Finish the game |
|  | By quitting or by winning |
|  | Review results with others |

## User Stories

* As a player of this game, I want to be able to personalize the settings of the game.
* As a player of this game, I would want to be able to play with other people so it be competitive and social.
* As a player of this game, I want to be able to easily see what I can do each turn making it easier to manage my empire.
* As a player, I would like to be able to see the results of the game after I am finished such as game stats so I can see how I did.
* As a player, I want a wide range of paths to victory.
* As a player, I want to be able to hide elements of my progress while being able to estimate other player’s progress with in game clues.

### **Product Backlog**

The collection of stories makes up your project’s product backlog:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Story ID** | **Story** | **Story Points** | **Priority** | **Status** |
| S1 | As a player of this game, I want to be able to personalize the settings of the game. | 10 | 5 |  |
| S2 | As a player of this game, I would want to be able to play with other people so it be competitive and social. | 5 | 1 |  |
| S3 | As a player of this game, I want to be able to easily see what I can do each turn making it easier to manage my empire. | 20 | 2 |  |
| S4 | As a player, I would like to be able to see the results of the game after I am finished such as game stats so I can see how I did. | 5 | 6 |  |
| S5 | As a player, I want a wide range of paths to victory. | 25 | 3 |  |
| S6 | As a player, I want to be able to hide elements of my progress while being able to estimate other player’s progress with in game clues. | 15 | 4 |  |

# **Sprint #1**

Sprint Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| **Story ID** | **Story / Task** | **Estimated**  **Hours** | **Actual**  **Hours** |
| S1 | Setup and learn dev environments | 8 | 8 |
|  | Design board and board options | 4 | 4 |
|  | Make controls for viewing board in VR | 4 | 2 |

## Review

* The Unity engine is setup on 3 different machines, and 2 of them are VR ready
* Initial world prototype is developed, but lacking any options or colors/textures
* VR controls have been implemented, and character movement/camera controls are ready, but with some flaws in the scale of the world.

## Retrospective

For our first iteration, we kept things simple as far as coding goes, because developing in unity is unfamiliar to both of us. In terms of how much work we got done, and how much effort we both put in, the tasks we chose turned out to be fairly simple. Designing the board was made simple in the Unity UI, and the world map we used was just a simple google image search away. Oculus’s website also has several Unity integration tools which allowed us to use a player camera and movement controller in just a few steps, rather than making them from scratch by ourselves. All in all, we completed most of what we wanted to, with the exceptions of designing board options.

Project Velocity: 14

# **Sprint #2**

Sprint Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| **Story ID** | **Story / Task** | **Estimated**  **Hours** | **Actual**  **Hours** |
| S1 | Add functionality to board with the player | 12 | - |
|  | Initialize player point system | 10 | - |
|  | Allow player to choose starting location | 8 | - |

## Review

* We starting working on ray casting
* We’ve re-thought how the game will start. Player will no longer choose starting location
* Learned how to modify scripts to set controls for the players

## Retrospective

This backlog was too ambitious and too broad. We didn’t have a clear understanding of what we wanted to do. We also realized that these tasks were too much to handle for this iteration and/or were not able to be done at this time in the project. The order of tasks in creating a game is something we are not familiar with. Things we did work on were not relevant to the tasks in our backlog.

Project Velocity: 0

# **Sprint #3**

Sprint Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| **Story ID** | **Story / Task** | **Estimated**  **Hours** | **Actual**  **Hours** |
| S1 | Make the hex grid for the strategy board | 15 | 16 |
|  | Add ray casting support | 9 | 6 |
|  | Learn how to lock the camera using a hotkey | 6 | 0 |
|  | Research into getting the game working on other VR/AR solutions other than just Oculus Rift | 12 | 0 |
|  | Create and add some more assets | 8 | 4 |

## Review

* Using raycasts to manipulate object positions on the board is much better understood now, and the test object (the player health bar) can be moved around by clicking on the objects around you.
* The hexagonal grid has been created, sans the colliders for each grid position.
* The camera was never locked, because we want the experience to remain first person only, to keep the game feeling like it’s a real table-top experience.
* We moved many of our objects and reorganized them to be in the assets folder under a similar format to eachother.
* In order to increase cohesion of the player controller, we moved some the function in which the player clicks to move the health bar from a sceneController script into the playerController script. Eventually, this code will likely go in a separate script, but it is just a test for now.

## Retrospective

Finally, the backlogs are getting easier to create. While working on these stories, we managed to avoid most roadblocks that weren’t directly related to the story at hand. For example, the only delays while testing raycasting and creating the hexagonal grids involved debugging with those features directly, and acquiring an understanding of the Unity functions involved in making those work. With raycasting, we learned how to use tags, and access game objects correctly, as well as their components from a completely different script. All in all, this sprint got us back on track after a rough 2 weeks during sprint 2.

Project Velocity: 26

# **Sprint #4**

Sprint Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| **Story ID** | **Story / Task** | **Estimated**  **Hours** | **Actual**  **Hours** |
| S1 | Clean up asset files and establish some organization to the project | 1 | 5 |
|  | Create Colliders to be used on the Hex Cells | 15 | 10 |
|  | Making the tiles unique (adding terrain features, or some stats maybe) | 12 | 0 |
|  | Create Game Model(s)/Learn Software for 3d modeling | 10 | 0 |
|  | Use Raycast to center an object on an object. | 10 | 0 |

## Review

* Added assertion and exception code to the scripts for base and healthBar, in order to check the health and radius variables and make sure they are staying in bounds.
* The project has been much better organized, and modifying/locating objects and scripts is now way easier.
* The hexagonal grid has one large collider now, and we’ve found out how to use raycasts to click on the grid and find out the coordinates of the cell we clicked, even though the collider isn’t separated by hexagonal boundaries between cells.
* Looked at some 3D modeling software, but have a while before comfortable with it, and ready to implement models from the software into Unity.
* Didn’t have time for dealing with unique tile generation, although some strategies and code for implementing this was looked at.
* Still only using clicks to reposition the object to where the ray collides with an object, not to the center of whatever collider gets clicked.

## Retrospective

When creating this last backlog, there wasn’t enough focus on exceptions and assertions, but we implemented them as we went, as well as finding methods and to implement a collider to our hexagonal grid. Then, we were able to start a simple script and eventually a prefab for a base object, which the players will own and place on a cell on the grid. For now, however, the base object is just a concept and we will hopefully start looking for events to happen, rather than just adding game objects with little interactivity.

Project Velocity: 15

# **Sprint #5**

Sprint Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| **Story ID** | **Story / Task** | **Estimated**  **Hours** | **Actual**  **Hours** |
| S2 | Allow a second player to join the world. | 20 |  |
| S3 | Decide what first turn for players will look like | 10 |  |
|  | Decide what next turns for players will look like | 15 |  |
| S5 | Choose how players will go about winning the game, such as what different options they will have, and how long it’ll take. | 15 |  |
| S1 | Set terrain types | 12 |  |