CSS 497 Design Specification

**Created by:** Jasper Yeung

**Quarter:** Summer 2025

Part 1▾Basic Info

A Virtual Tabletop for Game Masters to Create Immersive Experiences

Part 1.1▾Description

The capstone project will be a virtual tabletop (VTT) web application tailored to game masters (GM) to intuitively manage their tabletop role-playing games (TTRPGs). The VTT will provide a user-interface menu to switch between battle maps or role-playing scenes, each with their own unique features specific to each TTRPG system such as Dungeons & Dragons. These battle maps and scenes will be created by GMs to immerse their players in the world they are playing in the TTRPG.

Part 1.2▾Goals

* Create an intuitive user interface for GMs to easily manage their TTRPG campaigns.
* Provide GMs with useful tools for arbitrating over TTRPG rules quickly and correctly.
* Provide GMs with audio and visual tools to create immersive experiences for their players to encounter.
* Design a system architecture that is modular that can support multiple different TTRPG systems.
* Design a system architecture that is maintainable which allows for new features to be added to specific TTRPG systems.

Part 1.3▾Results



Figure 1: O. Oliver. Mar 21, 2013, <https://app.roll20.net/forum/post/94944/pc-token-portrait-border-colors-important-to-you>. Accessed July 7, 2025.

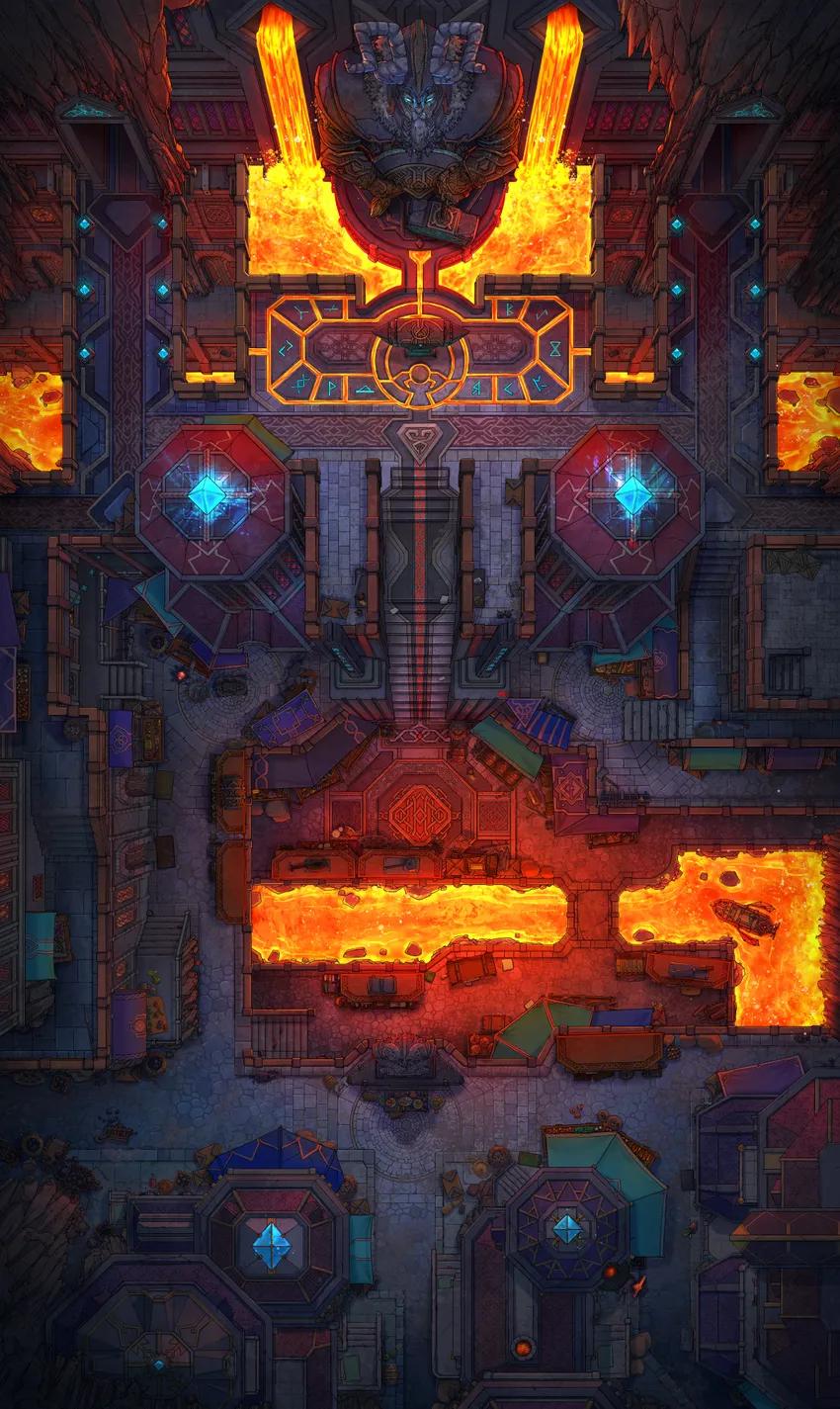


Figure 2: Czepeku “Underground Dwarven City Centre.” <https://www.czepeku.com/fantasy/maps/underground-dwarven-city-centre/original-day>. Accessed July 7, 2025.

The prototype of the project is a dynamic web application where a user can create a “campaign” with multiple battle maps, roleplaying scenes, and tokens which can be saved in a file. This file can then be uploaded to the web application which keeps the progress found in the file. Battle maps allow a GM to upload an image and place tokens as seen in *Figure 1* which represent characters onto the image as seen in *Figure 2* to represent a battle. Tokens contain game statistics called “stat blocks” that follow a chosen TTRPG system’s rules. Battle Maps help a GM to organize and arbitrate rules during a battle. Role-playing scenes allow a GM to upload a video or image to help immerse their players in the scene the GM is creating.

The prototype should be in a state where the battle maps and role-playing scenes are able to functionally work for their intended purposes, and the architecture is able to support multiple TTRPG systems. To support multiple TTRPG systems, the project must showcase well-designed and modular software architecture through object-oriented programming.

Part 1.5▾Relation to Future Career & Competencies

At this stage, the project will use tools like TypeScript and React which will be used to create an intuitive and manageable user interface. Frontend development is a fast-paced world where new tools need to be learned. Learning these two tools in a short time to use in the project will be able to be applied in future frontend development positions.

Furthermore, this project will require working with JSON files for saving user progress and API calls to a cloud storage service. JSON files and APIs are the foundation of backend development. Learning and demonstrating these backend and frontend skills will greatly help in future backend development positions and highlights the learning by doing competency.

The academic competencies that will be shown are as follows:

* Requirements Definition and Analysis will be developed during the project by clearly outlining the requirements in a Software Requirements and Specifications (SRS) document which will clearly define the requirements of the project to be completed and their development priorities. The SRS will contain functional and nonfunctional requirements while also outlining the features they are a part of.
* Software Architecture Design will be developed during the project through architectural diagrams including a class and component diagram. The diagrams must specify the relationships between components and classes as well as how the overall architecture functions together which helps accomplish the project’s goal of accommodating multiple TTRPG systems with useful features.
* Object-Oriented Programming will be shown in the implementation of the project. The project must have modular architecture which will be accomplished through inheritance and overloaded methods. Having multiple child classes also requires a factory class that will be able to create instances of classes without explicit definitions.
* Testing Methodologies will be developed during the project by implementing detailed testing plans including unit, functional, integration, and usability tests. The project requires robust software architecture and codebase which will inevitably lead to many issues and bugs during development. Testing logs written with test cases, results, and issues found will help validate the functionality of features in the project and help ensure the prototype works as intended.
* Technology Evaluation and Selection will be developed during the project by evaluating the tools found from various sources based on the merits and demerits of their use in the project. The selection of tools requires an understanding of the needs of the project using the SRS document and the developer’s circumstances. Technology should be selected and provided with justification for their use in the *Tools and Hardware* section of this document.
* Learning by Doing will be developed during the project by learning to use new tools found during technology evaluation while implementing the project. Tools such as libraries and frameworks should be learned by looking at relevant documentation for use cases that are immediately applicable to the project’s development.

Part 2▾Resources

Part 2.1▾Experts

The expert chosen for the project will be someone in a computer science field who has had current or previous experience working in full stack development, especially on web applications. Two meetings will be held with the expert with an additional code review in the second meeting.

The first meeting will be scheduled for week 4 in the summer quarter 2025. This meeting will primarily discuss the design specifications document and present additional artifacts related to the software architecture and tools used for the project. The expert will be asked to review the artifacts and provide feedback on these artifacts to be used in additional revisions of the artifacts.

The second meeting will be scheduled for week 9 of the summer quarter 2025. This meeting will present any additional artifacts to the expert for feedback and a code review of the project so far. If the project is in the prototype stage, the demo should be presented to the expert for feedback.

Part 2.2▾Stakeholders

The primary stakeholders are game masters who will use the VTT in their own TTRPG groups to enhance their game experience. GMs will be interviewed individually either in-person or online with a questionnaire and open-ended questions which will depend on the interview.

The first interview will ask about their experience as a GM, what difficulties they have when running a TTRPG, and what features they want or expect in a VTT. The first interview will be conducted after the design document is completed and planned to be in week 4 of the summer 2025 quarter.

The second interview will be combined with a demo of the functional prototype to the same game master of the first interview. This demo will first provide a brief tutorial by the developer of features included in the VTT and then have the GM try it and construct a hypothetical TTRPG they would run. After this demo, the GM will be asked what they liked about VTT, what they disliked, was the UI intuitive, and were the features varied and useful. The second interview will be conducted before the capstone colloquium, which is expected to be week 9 of the fall 2025 quarter.

Part 2.3▾Tools and Hardware

Part 2.3.1▾ Software Tools

The following list contains the software that will be used for this project:

* TypeScript
* HTML
* CSS
* React
* GitHub
* Visual Studio Code
* Fabric.js
* Jest

Because the project is a web application, TypeScript, HTML, and CSS are the basic tools required to create an intuitive web page and dynamic functionality. While I have experience with HTML and CSS, TypeScript is a brand-new tool for me to use. However, TypeScript is very similar to JavaScript with all JavaScript syntax working in TypeScript. As such, the learning curve and time will likely not take much time.

TypeScript was chosen over JavaScript because of its emphasis on object-oriented programming which will be required for the project to accomplish its goal of modularity for multiple TTRPG systems with different features. React is being used alongside TypeScript because it provides a library of frontend components to create an intuitive web page for the project. I will have to learn how to use React and find which components to use and modify them into fulfilling the requirements of the project.

GitHub and Visual Studio Code are the software development tools that will be used. GitHub will be used for storing the code base of the project while providing version control, release management, source control, and branching. I have some basic experience with GitHub but may need to learn about more advanced functionality to best utilize it for the project. Visual Studio Code meanwhile is an IDE that I have a lot of experience using which will not require additional time to learn. It is useful because of the many extensions that can connect to applications like GitHub and many programming languages including TypeScript.

Fabric.js is the fundamental library which the project will be built from. It is a popular JavaScript library providing interactive object models on canvas elements. Because the Battle Map and Roleplaying Scene features significantly utilize moving objects, panning, zooming, and other such functionality, Fabric.js is the perfect choice to fulfill the project’s requirements.

Jest was chosen as the testing framework for the project because it was made specifically for unit testing React applications. Additionally, Jest has built-in compatibility with TypeScript while other popular testing frameworks like Mocha and Jasmine must have an additional transpiler to function. Jest also has additional features like code coverage reports and mock functions which could be useful during testing.

Part 2.3.2▾Hardware Tools

The hardware required for this project will be a computer to access the software for the project and a server hosting the web app. My own personal computer will be used to access the software developing the project. The hosting server will be provided by Cloudflare, which is a popular cloud web hosting service. Cloudflare was chosen over other options because it provides a free tier for basic web application hosting while providing some security features. However, Cloudflare is a new tool for me, and I may need to allocate some time to learn about its features.

Part 3▾ Detailed Specifications

Part 3.1▾Software Architecture

A diagram of a diagram

AI-generated content may be incorrect.

Figure : Class Diagram

The class diagram as seen in *Figure 3*, provides a detailed structural representation of the project and its key features. The diagram demonstrates the relationship between features such as the Battle Map and RP Scenes as well as lower-level classes that implement functionality to those features like the StatBlock class. These classes together allow a GM to create immersive TTRPG experiences for their players.

The highest-level class in the diagram is the Campaign class. Every class relates to the Campaign class in some way because the Campaign class contains all the data created by the GM to be saved in a file and later uploaded back to resume the GM’s progress. The classes directly related to the Campaign class are classes that are associated with specific features with their own functional requirements with the only exception being the TurnMenu class. The BattleMap and RPScene classes are the most prominent examples as these two features are fundamental to the project’s functionality. Most of these classes except for the TabletopRoleplayingGameSystem class have composition relationships because these classes are the classes being actively used and modified by the GM with unique data.

Most classes have a composition relationship with other classes. These relationships are seen throughout the diagram because the GM has great control over what class instances created or deleted. A GM who creates a token creates an instance of the Token class which creates further instances of the StatBlock, Resource, and DiceRoll classes. If the token instance were to be deleted by the GM, all those instances would cascade and be deleted.

The two generalization relationships are between abstract classes and their child classes. The abstract classes are the classes that are kept track of by a higher level class so that their children are able to be stored but have unique functionality by overloading inherited methods. The TabletopRoleplayingGameSystem class being in a generalization relationship helps facilitate the overarching goal of the project to support multiple TTRPG Systems while having useful features.

The relationship cardinality seen in the diagram are either one-to-one, one-to-many, or many-to-one. One-to-one relationships are common for features that a GM has little input on. For example, a Campaign will always have a GameLog and TokenMenu instance which cannot be deleted or duplicated. The one-to-many relationships are also common for features the GM has significant control over. GMs can create many tokens, battle maps, and RP scenes which is why their classes have one-to-many relationships. The outlier cardinality is the many-to-one relationship only between two class pairs: Token and StatBlock, and Campaign and TTRPG. Some Tokens can share the same StatBlock and by extension their resources. For TTRPG and Campaign, the TabletopRoleplayingGameSystem class is an abstract class with predefined child classes where one is chosen by GMs for each Campaign.

A diagram of a component diagram

AI-generated content may be incorrect.

Figure : Component Diagram

The component diagram as seen in *Figure 4,* provides a higher-level perspective of the software architecture and the groupings of related components. The diagram demonstrates the relationships between components whether through aggregation, composition, or interfaces which all integrate to allow the GM to create immersive TTRPG experiences.

The three groups in the diagram are the representation logic, campaign elements, and tokens. The representation logic is the highest level of components that receive all the data provided by the other two groups and displaying them in a user interface for GMs. The campaign elements are all the components which contain data that the GM will interact with to create campaigns. The token group is also a part of the campaign elements group but contain subcomponents that are all required to be implemented for full token functionality.

Crucially, the diagram contains aggregation, composition, and interface relationships. Aggregation and composition relationships are similar to the relationships outlined by the class diagram in *Figure 3* but the interface relationships are unique. First, is the interface between the Battle Map, Token, RP Scene, and Audio List to the OneDrive server. The interface is provided by the OneDrive API to retrieve the image, video, or audio files set by the GM for the associated components. OneDrive servers were chosen but other cloud storage services could be implemented the same way with their APIs.

Another interface relationship is between all the representation logic components and campaign element components. These interfaces are necessary so that separation of concerns can clearly be established so that the representation logic components do not have access to any data they do not need.

The last interface relationship is between Dice Roll and Game Log components. This relationship is unique as they are both campaign elements. Dice roll results are sent to the Game Log but that does not mean the Game Log keeps the Dice Roll component itself. Instead, a message is generated by Game Log based on how the Dice Roll component calls Game Log’s provided interface.

Part 3.2▾Testing

The testing for the project will be in multiple stages of unit, integration, functional, and usability testing. Through all these tests, a test log will be created with a parameter of the test, the intended result, the actual result, the date of the test, cause of error, and debugging status.

For unit testing, each component will test the classes and methods used for correct returns and behaviors which can vary by component. The edge cases are more straightforward, with invalid or out of range parameters that may return valid information or error messages. However, some methods are inherently going to require the completion of other component classes, which will be more thoroughly testing in integration testing. The integration with a cloud storage API like OneDrive will need to be tested at this stage to correctly access the desired files and prevent providing error handling for inaccessible files. All unit testing will be performed using the Jest testing framework.

The integration testing, especially in regard to the addition and removal of components contained in other components, will be intensely tested. Many of the components have a composition relationship with their container components which may cause issues in the system. For example, an instance of a Battle Map may contain a Turn Menu and Tokens. When the Battle Map is deleted, the Token instances must also be deleted but the Turn Menu may refer to the same Token which may cause an error.

Another crucial part of the integration testing process will be the addition of multiple TTRPG child classes that will have different Features that need to be accessible with a general method. These TTRPG class Features must be functional where the Campaign class can easily call their methods without explicit parameters for a specific TTRPG.

The functional testing is where the user-interface of the web application will be tested alongside the rest of the system. As such, any components that have a display method will wait to be tested until this stage. Functional testing is the stage to test unusual user behavior which could include uploading a JSON file with an incorrect format or another source, changing the file location in OneDrive while using the web application, and deleting a base Token from the TokenMenu while not deleting Token instances in BattleMaps.

Usability testing will be a non-functional test performed alongside system testing. This will be done to validate whether the non-functional requirements have been accomplished in terms of the overall user experience. This testing focuses more on the frontend implementation rather than backend unless there are significant performance issues that have to be addressed by the backend. The testing will be conducted by a survey related to each non-functional requirement and given to stakeholders to experience the prototype and give their thoughts and possible recommendations.

Part 3.3▾Metrics

To measure the success of the project, metrics will be divided in terms of development metrics and a user survey. The development metrics will include cycle time, code coverage, and bug rate. Cycle time was chosen as a metric because there are multiple features that are able to be implemented individually which cycle time will accurately gauge the progress made throughout the project. Unit testing and integration testing will be performed in this project so using code coverage as a metric gives a good approximation of how much of the project has been tested.

Bug rate will also be used during testing that will help indicate what features are likely to produce errors requiring further testing and debugging. The absence of a bug rate could also indicate that the tests perhaps may not be stringent enough.

Finally, a survey will be given to GMs to complete after interacting with the prototype. The survey will be helpful to measure the success in regard to the non-functional requirements and possibly catch errors and bugs that were not found during the original testing process.

Questions would include:

* Is the user interface intuitive to navigate?
* Are the features provided helpful for you as a GM?
* Is creating new tokens understandable and intuitive?
* Is creating new battle maps understandable and intuitive?
* Is creating new roleplaying scenes understandable and intuitive?
* Is navigating between the battle maps and roleplaying scenes fast and easily accessed?
* How do you feel about the experience of uploading images and videos to a cloud storage service in order to use them in the VTT?
* Do you feel like different TTRPG systems chosen by different campaigns provide unique features?
* Do you feel like you can create immersive experiences for your players with this VTT?
* Have you discovered any errors or odd behaviors when interacting with this VTT?
* What device are you accessing the VTT from? Does it work well on the device?
* What new features would you want from this VTT?

Part 4 ▾ Schedule

First Quarter

| **Wk** | **ACTIVITY** | **RESULT** | **EST TIME** |
| --- | --- | --- | --- |
| 2 | Design Specification | Complete Draft | 7 |
| Status Report | Complete Week 2 Status Report | 1 |
| Meeting | Attend Class | 2.5 |
| Tools Research | Choose Frameworks, Libraries, and File Types | 2 |
| Learn New Tools | Learn Tools to Usable Level | 8.5 |
| 3 | Design Specification | Complete Final Draft with Revision Feedback | 2 |
| Status Report | Complete Week 3 Status Report | 1.5 |
| Showcase Slide | Complete Showcase Slides | 2 |
| Capstone Contract | Start Draft | 2 |
| Meeting | Attend Class | 2.5 |
| Librarian Meeting | Complete Meeting with Librarian | 1 |
| Instructor Meeting | Complete 1st Instructor Meeting | 1 |
| Project Implementation | Implement Battle Map | 10 |
| 4 | Capstone Contract | Complete Draft | 2 |
| Expert Meeting | Complete 1st Expert Meeting | 1 |
| Stakeholder Interaction | Complete 1st Stakeholder Interview | 1 |
| Status Report | Complete Week 4 Status Report | 1.5 |
| Career Preparation | Complete 1 Career Preparation Assignments | 1 |
| Project Implementation | Implement Battle Map and Role-Playing Scenes | 9.5 |
| Testing | Unit Test Battle Map | 4 |
| 5 | Capstone Contract | Complete Final Draft | 2 |
| Instructor Meeting | Complete 2nd Instructor Meeting | 1 |
| Status Report | Complete Week 5 Status Report | 1.5 |
| Meeting | Attend Class | 2.5 |
| Project Implementation | Implement Tokens | 13 |
| 6 | Capstone Portfolio Paper | Begin Drafting | 6 |
| Status Report | Complete Week 6 Status Report | 1.5 |
| Career Preparation | Complete 2 Career Preparation Assignments | 2 |
| Testing | Unit Test Tokens | 4 |
| Project Implementation | Implement Tokens | 6.5 |
| 7 | Status Report | Complete Week 7 Status Report | 1.5 |
| Capstone Portfolio Paper | Complete Draft | 4 |
| Showcase Slides | Complete 2nd Showcase Slides | 2 |
| Meeting | Attend Class | 2.5 |
| Project Implementation | Implement GM and Streaming Mode | 10 |
| 8 | Status Report | Complete Week 8 Status Report | 1.5 |
| Capstone Portfolio Paper | Revise Draft with Feedback | 3 |
| Project Implementation | Implement GM and Streaming Mode as well as Game Log and Dice Rolling | 12.5 |
| Testing | Unit Testing GM and Streaming Mode | 3 |
| 9 | Status Report | Complete Week 9 Status Report | 1.5 |
| Instructor Meeting | Complete 3rd Instructor Meeting | 1 |
| Expert Meeting | Complete 2nd Meeting with Expert | 1 |
| Showcase Slides | Complete 3rd Showcase Slides | 2 |
| Stakeholder Interaction | Complete 2nd Stakeholder Interview | 1 |
| Capstone Portfolio Paper | Final Draft | 3 |
| Usability Testing | Complete Usability Tests for Prototype | 3 |
| Code Review | Complete Code Review with Expert | 5 |
| Meeting | Attend Class | 2.5 |

Summer Gap

| **Wk** | **ACTIVITY** | **RESULT** | **EST TIME** |
| --- | --- | --- | --- |
| 1 | Debugging | System Wide Debugging | 10 |
| Project Implementation | Implement Turn Menu | 10 |
| 2 | Project Implementation | Implement Turn Menu and Audio Controller | 10 |
| Testing | Unit Test Audio Controller and Turn Menu | 10 |
| 3 | Project Implementation | Implement Campaign File Saving | 15 |
| Testing | Integration Test Campaign File Saving | 5 |
| 4 | Project Implementation | Implement Campaign File Saving | 15 |
| Testing | Integration Testing and System for Completed Features | 5 |

Second Quarter

| **Wk** | **ACTIVITY** | **RESULT** | **EST TIME** |
| --- | --- | --- | --- |
| 1 | Status Report | Complete Week 1 Status Report | 1.5 |
| Debugging | Debug System Wide | 8 |
| 2 | Status Report | Complete Week 2 Status Report | 1.5 |
| Project Implementation | Implement New TTRPG System | 6.5 |
| 3 | Status Report | Complete Week 3 Status Report | 1.5 |
| Showcase Slides | Complete 4th Showcase Slides | 2 |
| Testing | Unit Test TTRPG Exclusive Features | 6 |
| 4 | Status Report | Complete Week 4 Status Report | 1.5 |
| Testing | Integration Test TTRPG System Features | 8 |
| 5 | Status Report | Complete Week 5 Status Report | 1.5 |
| Project Implementation | Implement Voice Effects | 8 |
| 6 | Status Report | Complete Week 6 Status Report | 1.5 |
| Project Implementation | Implement Voice Effects | 8 |
| 7 | Status Report | Complete Week 7 Status Report | 1.5 |
| Showcase Slides | Complete 5th Showcase Slides | 2 |
| Testing | Unit Test Voice Effects | 6 |
| 8 | Status Report | Complete Week 8 Status Report | 1.5 |
| Poster | Draft Poster | 6 |
| Presentation | Draft Presentation Slides and Rehearse | 8 |
| 9 | Status Report | Complete Week 9 Status Report | 1.5 |
| Poster | Complete and Print Poster | 8 |
| Presentation | Complete Poster | 8 |
| 10 | Status Report | Complete Week 10 Status Report | 1.5 |
| Showcase Slides | Complete 6th Showcase Slides | 2 |
| Presentation | Rehearse Presentation | 8 |
| Finals week | Capstone Presentation | Present Project Poster and Presentation at Capstone Colloquium | 1.25 |
| Presentation | Rehearse Presentation | 9.25 |

Part 5 ▾ Contingency Plans

If unexpected events occur that prevents work from being completed, the missed hours will be completed on a later day or days in the same week if possible. For example, if on Tuesday I become sick and miss 4 hours of work, those hours will be distributed over Wednesday, Thursday, Friday, and even Saturday which keep the weekly activities at 20 hours logged. However, if reallocating work hours is not possible, the planned activities will both be pushed by a week and distributed over the following weeks.

If the event causes activities to be completed but taking more time than originally anticipated, then the activities need to be pushed back and adjusted by the hours greater than the estimate. Some activities may have been estimated times greater than they may actually need and in those cases, those activities can have reduced time allocated to them. If there is a brand-new unanticipated activity that needs to be completed, a similar process of pushing back subsequent activities and reducing possible overestimated activities needs to be performed.

If certain activities and especially feature development take a significantly greater portion of time than anticipated that severely throw off the schedule then the priority of features should be considered. Certain features such as the audio controller and voice effects were included in the schedule to be developed. However, they are low priority and may be excluded from development over these two quarters.

Glossary

**Battle Map:** A visual image illustrating a scene occurring in the story of a TTRPG. The image of a Battle Map is typically from a top-down perspective. GM can drag tokens onto the Battle Map to indicate the location characters are occupying in a scene.

**Dice Number:** The number of faces an individual dice has with different equally likely results. The notation for referring to dice number is a single “d” followed by the dice number. For example, a twenty-sided dice is denoted as d20.

**Dice Roll:** A random number generator emulating a physical dice roll that creates results based on static modifiers, dice number, number of dice, roll method, number of rolls, and modifier dice rolls.

**DM:** Abbreviation for Dungeon Master.

**Dungeon Master:** Is a synonym to the term Game Master. Primarily used to refer to Game Masters using any edition of Dungeons and Dragons as their TTRPG System.

**Game Log:** A visual log that contains messages of game events that have occurred. Example game events may include, dice rolls made, a condition being applied, and reminders to perform an action.

**Game Master:** An individual who oversees a TTRPG by orally narrating details of scenes and characters of a story while also arbitrating the rules of their chosen TTRPG System.

**GM:** Abbreviation for Game Master.

**GM Mode:** A mode where the GM can switch between Battle Maps and RP Scenes separately from what is shown in Streaming Mode. The mode also allows GMs to see every token and element, even ones that were specified to be hidden by the GM.

**Modifier Dice Roll:** A type of dice roll that cannot have a static modifier and affects the result of another dice roll whether by adding or subtracting from the result. For example, a regular dice roll of a d20 has an outcome of 10 but has an additive modifier dice roll of a d4 with an outcome of 2 which causes a result of 12. A regular dice roll can have multiple modifier dice rolls of the same or different dice numbers.

**Number of Dice:** The number of dice of the same dice number being used in a dice roll. The notation for referring to the number of dice is a number followed by a single “d” and then followed by the dice number. For example, a dice roll with two ten-sided dice is 2d10.

**Number of Rolls:** The number of times dice in a dice roll or used to get that same number of results. For example, a dice roll with four six-sided dice with 2 rolls will have two outcomes of 4 and 24.

**Roleplaying Scene:** A visual image or video illustrating a scene occurring in the story of a TTRPG. The image or video of an RP Scene is typically from a first-person perspective. GM can add visual effects to the RP Scene to further immerse their players in their story.

**Roll Method:** The method a dice roll uses to determine the result. The main two methods are taking the highest result or taking the lowest result. These two methods are sometimes referred to as Advantage or Disadvantage.

**Roll Type:** The method in which a dice roll’s result is determined when there are

**RP Scene:** Abbreviation for Roleplaying Scene.

**Stat Block:** A textual description of a character’s abilities that follow a TTRPG System’s rules. Stat blocks can often include actions, resources, ability scores, and ability modifiers.

**Static Modifier:** The static number that is added to the outcome of a dice roll after the random number has been determined. For example, a static dice modifier of 5 is added to a d20 roll with the outcome of 10 and the final result of 15.

**Streaming Mode:** A mode where the VTT displays Battle Maps, RP Scenes, and Tokens specified by the GM.

**Tabletop Roleplaying Game:** A type of roleplaying game in which there are players that roleplay a character and a GM who plays any non-player characters and arbitrates the rules of the game.

**Token:** A small visual representation of a character in a TTRPG. Tokens often take the form of an image inside a circular shape. Tokens contain information on a character’s stat block, resources, applied conditions, size, and visibility. Tokens can be placed onto Battle Maps by a GM.

**TTRPG:** Abbreviation for Tabletop Roleplaying Game.

**TTRPG System:** A set of predefined rules used to play a TTRPG for GMs and players to follow. TTRPG Systems are often commercialized products and contain rules that try to provide specific types of experiences. Examples include Dungeons & Dragons 5th Edition, Call of Cthulhu 7th Edition, and Pathfinder 2nd Edition.

**Virtual Tabletop:** A tool that aids GMs in creating a TTRPG experiencing by providing the ability to visualize battles and scenes by creating Battle Maps and RP Scenes.

**VTT:** Abbreviation of Virtual Tabletop.