**\* = rough idea**

**\*\* = need to expand on / not finished**

**Idea 1:** **Game that Teaches Web Development**

Genre Ideas: Dungeon Crawler, *Insert any others here*

* Level-Based

**Requirements**

1. Define Learning Objectives
   1. HTML: Tags, Attributes, structure of an HTML document\*\*
   2. CSS: Styling, selectors, box model\*\*
   3. JavaScript: Variables, functions, DOM manipulation\*\*
2. Game Concept Development
   1. Narrative and Settings
      1. Simple storyline
      2. Each level/set of levels represents different web dev concept/set of learning objectives\*\*
   2. Game Mechanics
      1. Determine how players will interact with the game\*\*
      2. Example: solve a CSS puzzle to unlock a door\*
   3. Education / Curriculum Integration:
      1. Plan how the levels are structured to teach specific concepts\*\*
3. Game Design
   1. Level Design:
      1. Each level should incorporate puzzles and/or challenges that start simple and gradually increase in complexity
   2. User Interface:
      1. Design interface that includes both game screen and a code editor\*
      2. Player should be able to write code and see how it affects the environment
4. Development Tools
   1. Game Engine: choose engine that’s suitable for beginners and supports integrations of web technologies
      1. Look into: Godot, Unity (with plugins), HTML5 frameworks (maybe Phaser)
      2. Learn/Plan: How to integrate web coding into the game. Could look into Ace or CodeMirror
5. Development Phases
   1. Prototype:
      1. Create a low-fidelity prototype (such as one-level, one puzzle)
      2. Focus on core functionality: code editor, mechanics, input to output for game changes
   2. High-Fidelity Prototype / Mixed-Fidelity Prototype:
      1. Attempt to adapt low-fidelity prototype into high/mixed to reduce development timeline
      2. Make any core changes early on if necessary (game engine, etc.)
   3. Test and Feedback:
      1. Ongoing testing and feedback on builds
      2. Determine testing methodology
6. Learning Resources
   1. In-Game Guidance: provide hints, tips, or links to resources
   2. Progressive Difficulty: ensure each level builds on concepts and that guidance follows accordingly
7. Time Management
   1. Project Plan: Create detailed project plan with milestones
      1. Given time constraints we need to aim for MVP (minimum viable product) that covers the essentials.
   2. Scalability: Design game so that adding levels and concepts can be added later if time allows
8. Documentation
   1. User Guide: create simple guide on how to play the game and learn from it
   2. Reflective Documentation: Ongoing documentation of development process, challenges faced, and how we overcame them.

Notes (Dylan 2/1/24):

* Should focus on just windows compatibility
* Choose IDE (weigh VScode vs other options)
* Find tiled map editor (Sherwood Project uses Aseprite [freeware])
  + Need animation previews (assuming most have this)
* Browser console window will be useful for debugging and object behavior
* Powershell script to spin up temp server for ensuring all assets load correctly
* We should sync to a git repo
* Brush up on JS
* Tutorial Playlist Sherwood refers to (build an RPG from scratch with HTML, CSS, and JavaScript): <https://www.youtube.com/playlist?list=PLcjhmZ8oLT0r9dSiIK6RB_PuBWlG1KSq_>
* Two layers for graphics
* Mechanism to control timing of the game to account for refresh rate
  + Part of this needs to be the camera (what draws to the canvas container)
  + Need to avoid executing events for every step
  + State of objects must be defined (static or moving), if not, frame becomes undefined and it creates list of errors
  + Tldr Drawing to canvas has the possibility for a lot of errors

- console error handling (obvious)

- OOP involved for game flow (level advancement, story flags [persistent & non persistent])

- A tile map editor should output XML files

- Sherwood pastes numerical values (assigned to all parts of the map) into text file before running python script that translates the values into characters

- reduces time inputting map layout manually

- Research Observer pattern (OO concept)

Notes (Ashlyn 02/07/2024):

* I really like the idea of creating the game, I have not done anything like this before so I definitely look forward to doing it should everyone want to move forward.
* I watched the Youtube video that was included in your notes above and I really like that he literally goes step by step in the playlist so if we were to get stuck we can reference that video and maybe work through our issues.
* I use Visual Studios for all of my code work so I definitely have that as a preference, but again it wouldn’t be too hard for me to use something else im sure.
* I will be slightly out of my element and comfort zone for something like this other than knowing how to code in HTML and CSS to benefit the project. However, I am willing to watch all of those videos by Sherwood to help aid us in completing this project.