Kevin Teynor

**Capstone Proposal**

**Project Motivation:**

The Warcraft III World Editor was the first exposure I had to game development. The tool was at the same time powerful and easy to use, as is evident by the numerous games and game-types it spawned. Because I had such a positive experience with the editor, I find it unacceptable that there are no standalone, platform independent solutions that offer similar functionality today. Because there aren’t readily available solutions, I am proposing to create one for my capstone project—as an extension to the powerful and free-to-use Unity game engine.

**Why Unity:**

I am choosing to use Unity for this project because it already provides much of the core functionality a game needs to be a game. Unity handles graphics, physics, and audio. There is a basic pathfinding system included. Unity is compatible with three scripting languages, and it has a well-documented API. There is an extensive asset store that users can upload resources to and download resources from to use in their projects. Unity has a large and active online community. Unity can deploy to every major operating system for computers, handhelds and consoles.

What Unity doesn’t provide are focused and genre-specific tools and implementations. Unity, in order to be so universally adaptable, stays away from creating any tools that aren’t universally used by many different types of games. None of the core mechanics that make up this project’s tools exist inside of what Unity offers by default. So while I will be making extensive use of existing software, it in no way diminishes the target scope of the project—to build a set of tools that facilitate the development of a specific genre (and its subgenres) of video games. Without Unity, I could not complete this project in 10 weeks, and certainly not with a finished product that I could be proud of.

**Project Success:**

The finished product at the end of the 10 weeks can be considered successful if:

* Novice game developers with no experience using Unity can use the tools I create to build something playable in less than 48 hours spent working on it (not including the time it takes the user to get used to Unity itself).
* Users can create fully functional games without needing to create any scripts themselves.
* There is a complete set of reference materials for all of the tools I create, with code samples.
* I, as the developer, can present 3 small, separate games made using the tools created during the project to showcase the flexibility and power the tools provide:
  + A tower-defense game, with 3 different levels, at least 6 different tower types, and 10 different enemy unit types. The player places towers along a path, which will attack computer controlled enemies (or “creeps”) that try and traverse the path without dying.
  + A hero ARPG (action role playing game) with a single level, a single hero, 4 abilities, and at least 6 different enemy types. The player controls the hero while navigating through a level and fighting different enemies using the hero’s abilities.
  + A pseudo RTS (real time strategy) level with 3 different structures, 6 different friendly unit types, and 3 different enemy unit types. The player will build and upgrade structures which produce friendly units to defend against enemy units which spawn in increasingly difficult waves.

**Target Users:**

The target users of this product are game developers of all skill levels who are interested in creating games that derive from the RTS or ARPG format. Novice developers will be able to create fully functional games without needing much prior experience, and without needing to invest time and money into learning the technology. Experienced developers will be able to prototype and build complete games in as little as a few hours, because they will not have to build the core gameplay functionality themselves.

**User Stories:**

Ordered descending by importance:

1. As a novice game developer, with little to no prior experience using Unity, I can use this tool to build a basic real-time-strategy based game in a quick manner. It will take me no longer than 48 hours to have a fully playable (although not polished) game.
2. As an enthusiastic gamer with no coding experience, I will be able to create a game solely by using the predefined behaviors included in the tool. I will not need to write any of my own code.
3. As a designer, I expect this tool to be fully integrated into Unity. I expect to be able to use every part of the tool through Unity’s GUI.
4. As the map designer for a game, I will be able to specify the dimensions of the game’s grid-based map upon its creation.
5. As the map designer for a game, using the tool I will be able to define exactly where in the game the players may build structures, where each player’s units may go, and what obstacles each player’s units must avoid.
6. As the developer responsible for game-units, I will be able to define different unit types with basic core attributes like Health and movement speed.
7. As the developer responsible for game-units, the tool will provide me with a simple way to build a tech-tree (the upgrade hierarchy for units).
8. As the developer responsible for game-units, provided that I give my units working collision meshes, the tool will take care of the movement and collision details.
9. As the developer responsible for combat for my project, I can give attacks to units and structures. I will be able to specify for each attack: the damage it does, the range of the attack, and the type of attack.
10. As a gameplay designer, the tool will allow me to easily create event-driven gameplay by attaching predefined ‘triggers’ and ‘actions’ to objects. For example, I can have a region automatically produce a new unit each time a timer expires.
11. As a gameplay designer, at the minimum I need the tool to handle the following events:
    1. A timer expires
    2. A unit enters a region
    3. A unit leaves a region
    4. A unit comes within range of something
    5. A unit dies
12. As a gameplay designer, at the minimum I need the tool to have the following actions defined:
    1. Produce a unit
    2. Kill a unit
    3. Start a timer
    4. Build a structure
    5. Set a unit’s property
13. As a game developer, if I get stuck while using these tools, I expect clear and concise reference materials that will help me move forward with my project.
14. As a game developer, I do not expect this tool to include or facilitate in the development of any multimedia asset that is necessary for my game. I will need to either create these for myself or find someone else to create them for me.

**2 Week Plan:**

|  |  |
| --- | --- |
| **Date** | **Description** |
| Week 1 | **Unity Spike** – Set up a ‘scratchpad’ environment to become reacquainted with Unity, learn how to create Editor Extensions, and resource libraries. The scratchpad will be reused as needed to prototype individual parts of the project as it progresses. |
|  | **Map Editor** – Create map class and tile class. Implement random auto generation for maps based on a set of tiles. Create Unity editor-extension for the map editor tool. Write documentation for the map editor. |
| Week 2 | **Trigger Editor** – Define what events and actions are relative to the tools. Create the predefined events/actions as described above. Ensure defining new events/actions is possible. Create Unity editor-extension for the trigger editor tool. Write documentation for the trigger editor. |