Tech Doc

## About:

This document will contain the technical design, assets to be included and overall process of the development of the game.

This document is to be used in conjunction with the design document in order to gain context of the different mechanics to be implemented.

Folder structure:

Assets

-Sprites

--Characters

---InBattle

---InMap

--Items

---Consumables

---Weapons

----InBattle

----InMenu

--Map

-Effects

-Scenes

-Sounds

-Music

-Xeffect

Programs used:

Word (documents)

Pencil? (prototyping interface)

Inkscape (art)

Visual Studio Community 2015 (code)

Unity3d 2018 (game engine)

Git (version control)

Assets used:

Sound

Sprites

Scripts

Materials

Controls (Keyboard, console controllers (Playstation/Xbox layouts))

WASD/arrow keys, or dpad on supported controllers to navigate menus and move the map cursor around.

Enter key or A/X buttons to confirm menu choices or selecting tiles/units on the map

Backspace or B/Circle to decline menu choices or deselect tiles/units on the map

Escape or Start to open overall game management menu (resume, exit game)

Graphics/rendering:

2D

Player/project etc. settings TBD

Map (map generation, tile data, scenes, in game cursor)

Map is going to be a grid of tiles. Initial grid can be assembled in a pre-defined format through the use of Editor scripts and a tile prefab (quad?) that has relevant components attached. The editor script should allow for X and Y dimensions to be set (where every increment is the size of a tile). The absolute 0 point of the scene is selected as the origin, and tiles will be assembled from top left to top right, where a new tile is placed underneath the first tile once the first row is finished. The new tile repeats the process, and this process continues until an internal counter in the editor script reaches 0 (for the Y attribute)

Once the tiles have been set, the existing game object properties should allow for easy assignment of sprites and attributes (whether units can traverse them normally, are hindered or receive any bonuses if waited on)

The tiles themselves will have the following components: Sprite renderer, sprite, TileAttribute script and the AdjacentTile script. The idea is that the TileAttribute script will refer to a list of sorts to set the attributes of the tile that will affect the gameplay. There will be a dropdown menu of different terrain types: Plain, forest and mountain (enums probably). In the list, the defined attributes of these terrain types will be: Move restriction, tile modifier and sprite path.

Move restriction refers to restrictions set on how units can traverse the tile, tile modifier contains modifiers that affect units occupying the tile and sprite path is just a QoL addition to link to the correct terrain sprite directory by default when selecting a sprite for the tile. The terrain types dictate what the attribute values will be. The move restriction for plain is “free move”, forest is “adjacent move” and mountain is “no move”. Essentially, plains have no restrictions, forests only allow movement if a unit is already next to the forest tile (and may not move more than one space) and mountains are not traversable. Plains and mountains have no tile modifier (null), but the forest has “minor dodge”, which gives a 10% boost to the avoid rate of the unit occupying the space.

The AdjacentTile script will have a 4 item array of all adjacent tiles, if applicable. This is used as a quick reference to see whether a unit can traverse from one tile to another. (although, perhaps this can simply be determined when attempting to move a unit onto a tile?)

UI (start menu, map menus, battle display)

Units (sprites, effects, attribute system, battle calculations)

Battle system (units, weapons, map attributes)

Items (menu, restrictions for usage, effects to affect unit)

Target platform:

PC