THE GREAT ALPACA

Turn-Based Strategy Toolkit

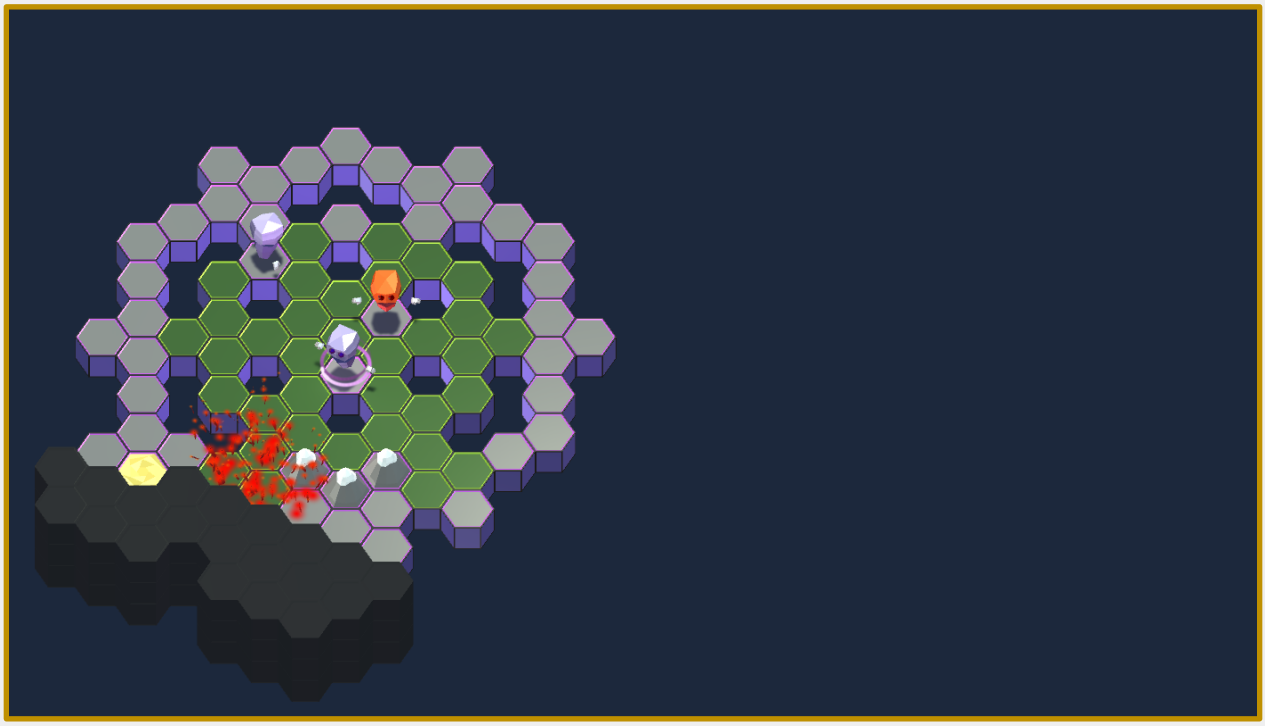
Documentation

Version 1.0.2

Introduction

This system allows you to create 2D, and 3D turn based games with either Square/Cube, or Hexagonal grids. You can define your cell palette and create your own grids with the grid editor. Play against human, or AI units, and choose to either auto place their units, or give the player spawn points to give more tactical decisions to the players. Hide the map to encourage exploration by adding Fog of War, this requires your unit to be a certain distance away for the cell to become visible. You can create custom abilities by stating its shape, who can be affected by it, the parameters(Damage, Heal, Pushback, Spawn, or create your own,..), and ailments(which can apply parameters for a number of turns). Add these abilities to your units and throw them into the game. There are multiple Win Conditions that you can customize, and mix, or create your own with your own rules to fine tune your desired gameplay.

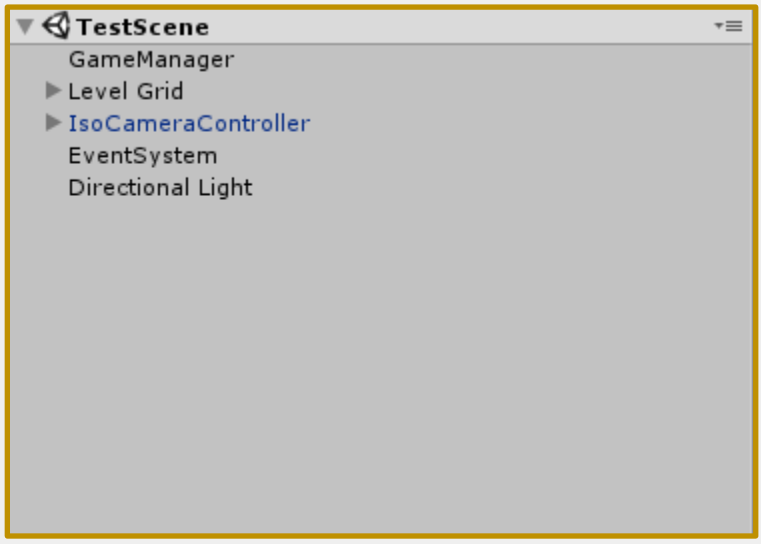
This documentation is best read from start to finish, but if you want to learn about specific parts, check out the Index on the next page.



Index

General Setup

A basic scene contains a Game Manager, Level Grid, Camera Controller, and an Event System. The Event System is important, it is how cell interaction works. Adding a “debug handler” to your scene allows you to kill units, reset team data, and disable audio.

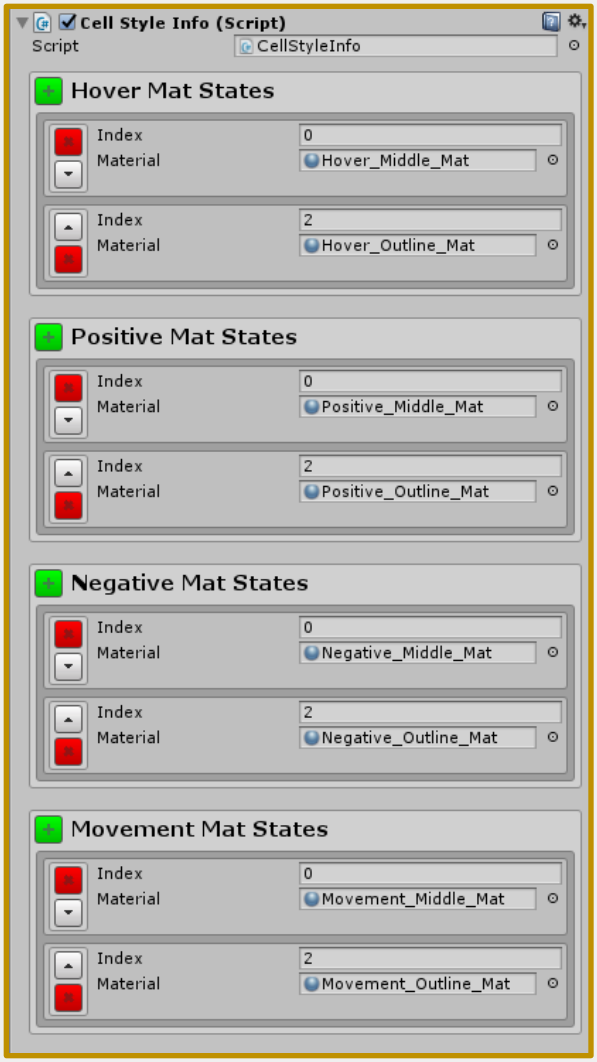


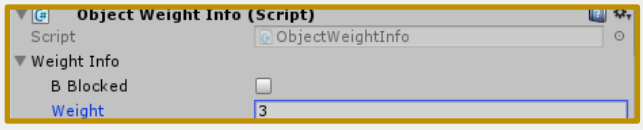
Setting Up A Cell

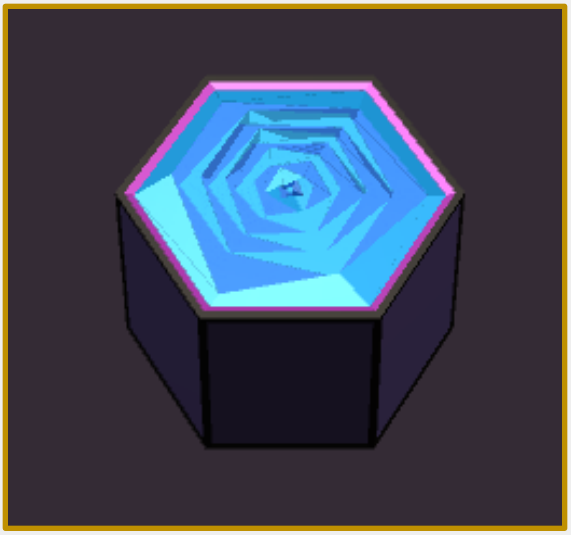
Any object, or sprite can be used for creating Cells. And the setup is quick, and easy.

Steps:

1) Create a prefab of the object
2) Add a collider (must be a 3D one, you can use a sphere/Cube one for 2d grids)
3) Add the “Cell Style Info” component if using a mesh:
    a) Setup the materials for each state based on the Material indexes in the mesh renderer.
4) Add the “Cell Style Info 2D” component if using sprites:
5) Add the “ObjectWeightInfo” component:
    a) This is used to set its weight, and if it’s blocked.

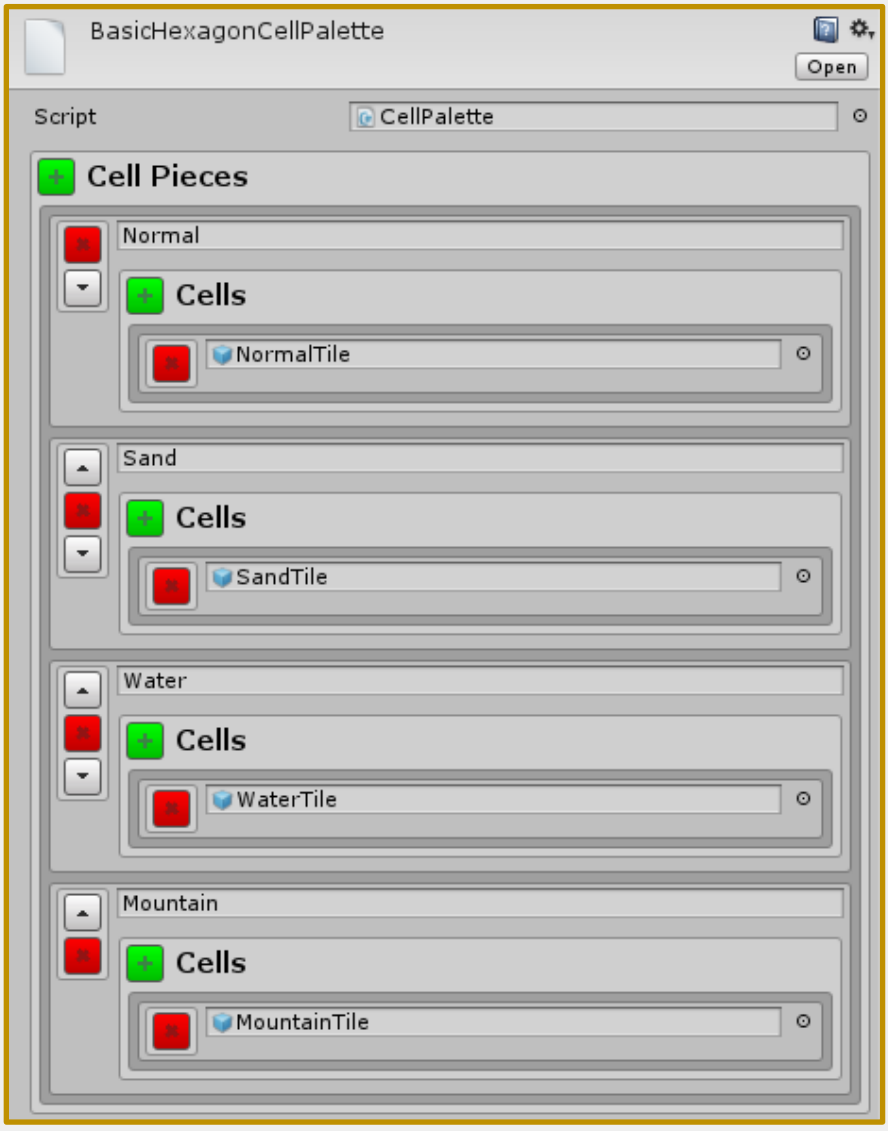






Cell Palette

This is used in the Grid Generator, and the Grid Editor.
If you add multiple Cells to a Cell Piece, it will choose a random one from that list.
More will be explained in the Grid Generator, and the Grid Editor sections.
Right-Click Menu: Create>TurnBasedTools>Create CellPalette



Grid Generator

Grid Size: The number of rows and columns.
Cell Palette: The first cell piece will be used for generating.
Is Grid 2D: Decides if a LevelCell, or LevelCell2D component should be added.

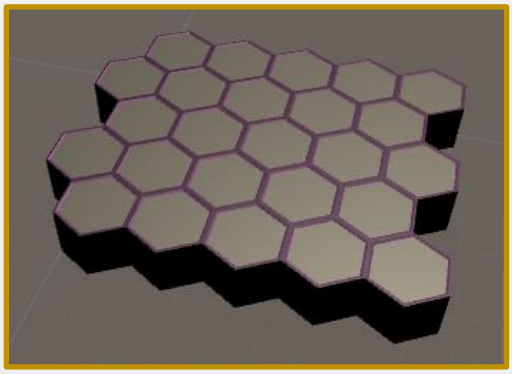
Currently there are two Grid Generators, one for Hexagons, and the other for Cube/Square grids. When you click generate, it creates the grid, and parents all the Cells to an object with a LevelGrid component (Hexagon Grid, or Square Grid), Here you can change the Cell Palette, and this will change what the LevelCellTools Will show.
I will explain more of the uses in the Grid Editor section.

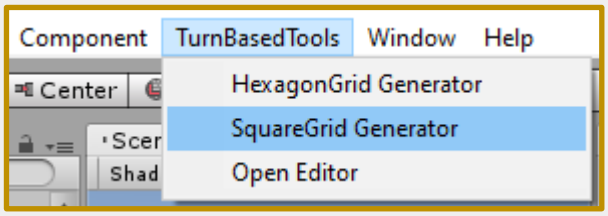
For each Cell, the generator adds:

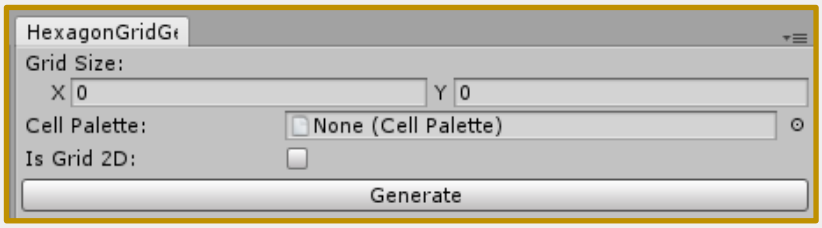
- LevelCellTools component (which allows you to change the cell)
- LevelCell/LevelCell2D (depending on the flag)

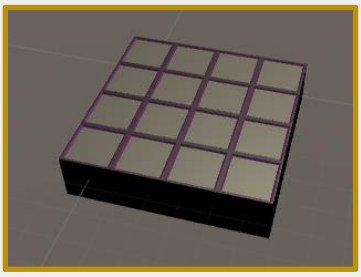
Item Menus:

- TurnBasedTools>SquareGrid Generator
- TurnBasedTools>HexagonGrid Generator





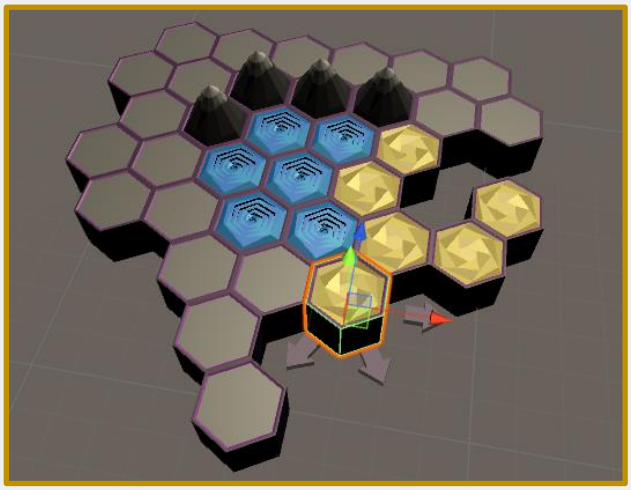


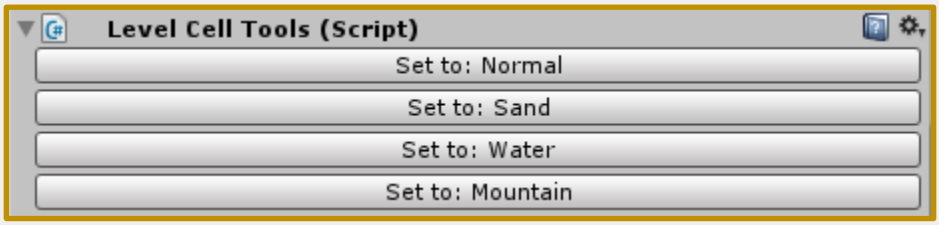


Grid Editor

The grid editor allows you to add and remove cells from a grid. And Change the cell to another from the Cell Palette.
Changing a cell with the LevelCellTools will change what the added cells will be.

To Remove a Cell: Just delete it, the adjacent cells will update themselves for both addition, and removal.
To add cells: Click on a cell, and an arrow selector will appear,
Left clicking an arrow will add and select that new cell in that direction
Right clicking will just add a new cell.

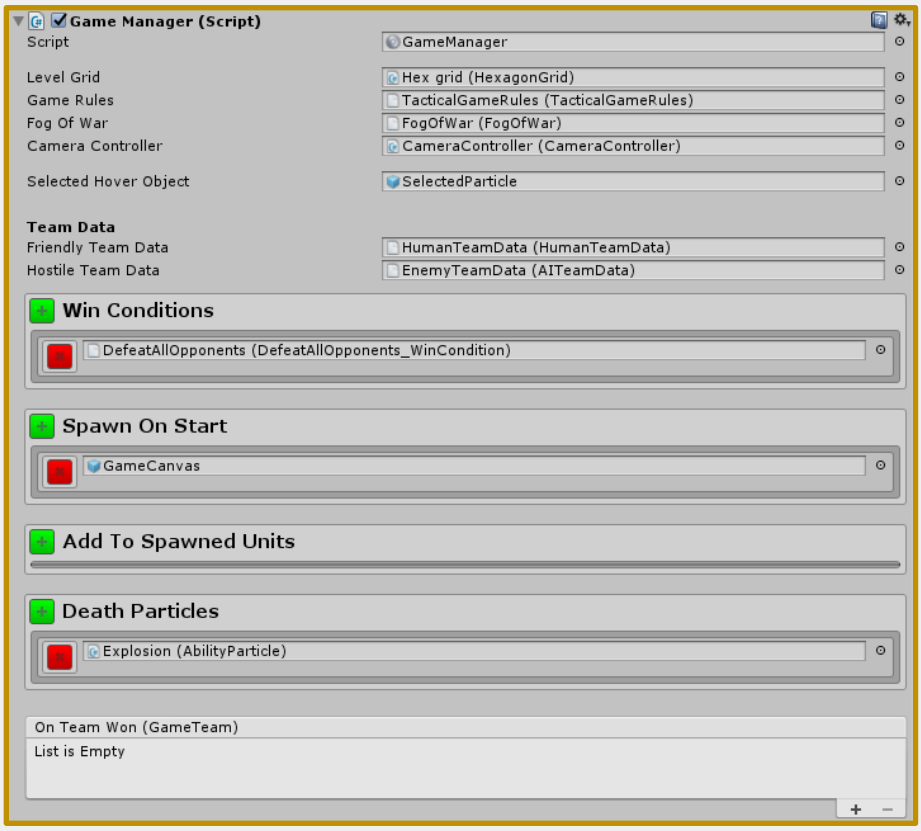




Game Manager

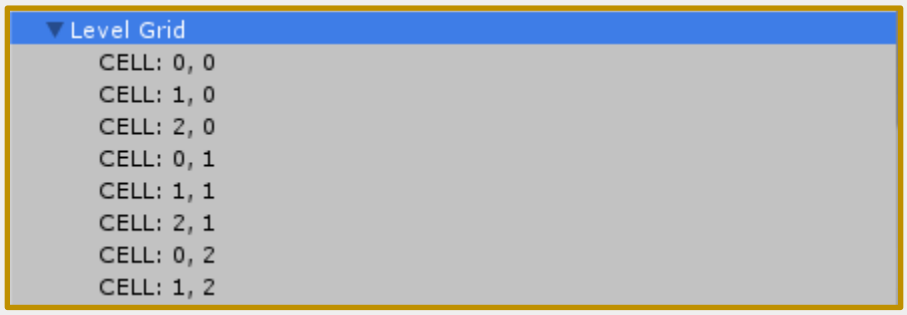
This is the connection to the systems, handles units, and has various other useful functions.

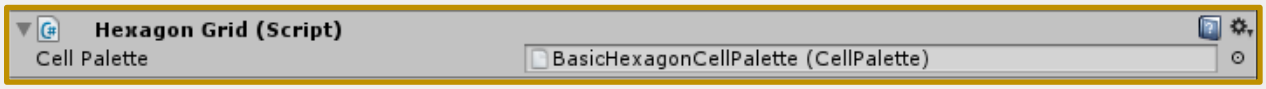
Level Grid: The Grid to do all game logic on.
Game Rules: This handles most of the gameplay logic.
Fog of War: This is optional but shows invisible blocks as you get closer.
Camera Controller: This is used for accessing the camera mainly for UI connections.
Selected Hover Object: This is the prefab that spawns on the currently selected player.
Friendly Team Data: Contains info for spawning Units (must be human)
Hostile Team Data: like Friendly Team Data but allows for AI Team Data.
Win Conditions: This is a list of requirements for how a team wins or loses.
Spawn on Start: This allows you to give prefabs that get spawned at the start of the game, great for things like UI.
Add to Spawned Units: Anything added here will be added to all units spawned, I use it for showing an icon above a targets head in one of the example scenes.
Death Particles: This spawns any number of particles on unit's that die, great for hiding the disappearing of the units.
On Team Won: This gets called only once, when all the Win conditions pass.

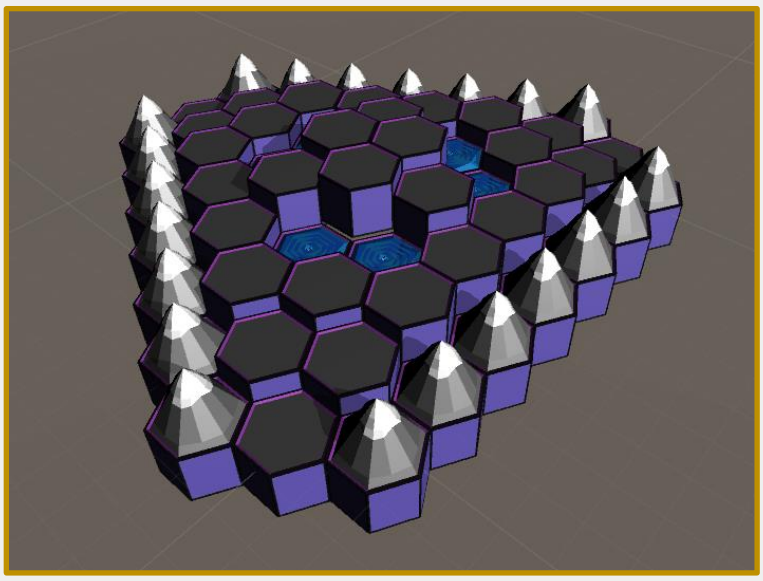


Level Grid

This handles the creation, removal, and accessing of Cells. The Grid Editor handles the creation/Removal and will automatically update the grid. Cells are parented to the Level Grid Object and allows you to change data on each Level Cell.
There are two types of Level Grids: SquareGrid, and HexagonGrid.
The level grids are automatically added with the Grid Generator, both allow you to change the CellPalette, and the Square grid allows you to say if you want to count diagonal cells in movement and abilities.





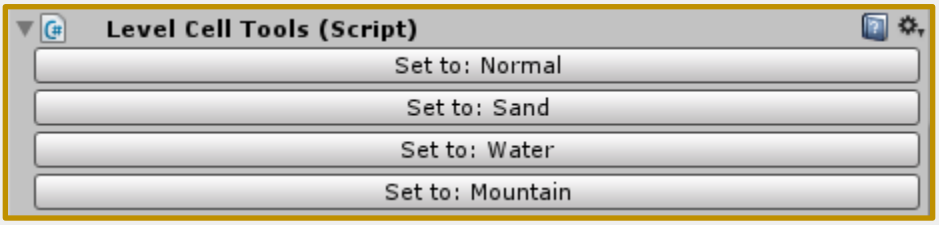


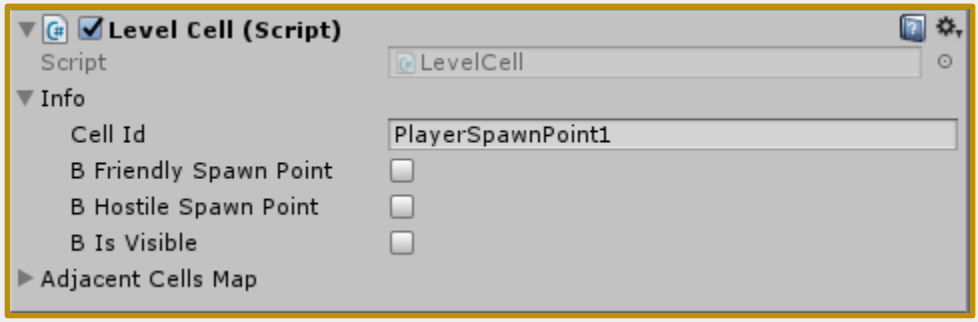
Level Cell

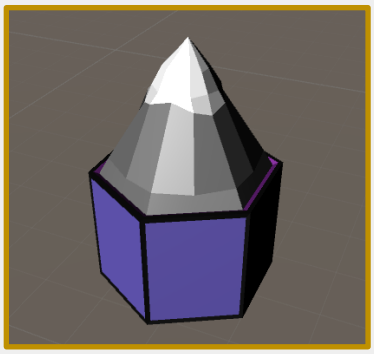
Generating a Level Cell will add these components to it:
LevelCellTools: Allows you to swap the cell for another one from the Level Grids CellPalette.
LevelCell or LevelCell2D (depending on the type of grid): This allows you to set cell specific data.

- Cell Id: This is primarily used for spawning units, but can also be used to get a list of all cells with an Id.
- B Friendly/Hostile Spawn: This is used in the GameRules if you decide to allow players to choose their starting point.
- B Is Visible: this hides the cell, and anything on it, hostile units can still traverse the area. Used with the FogOfWar will allow you to uncover invisible cells.
- Adjacent Cells Map: This is automatically changed with the Grid Generator, and Grid Editor.

Also see Setting Up A Cell for more info.





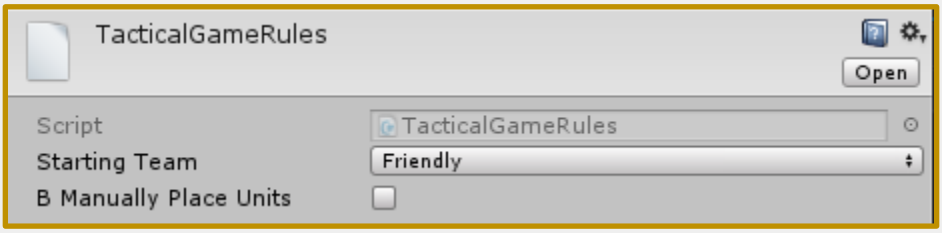


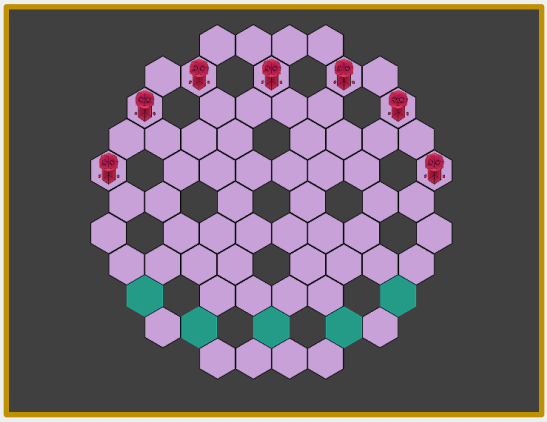
Game Rules

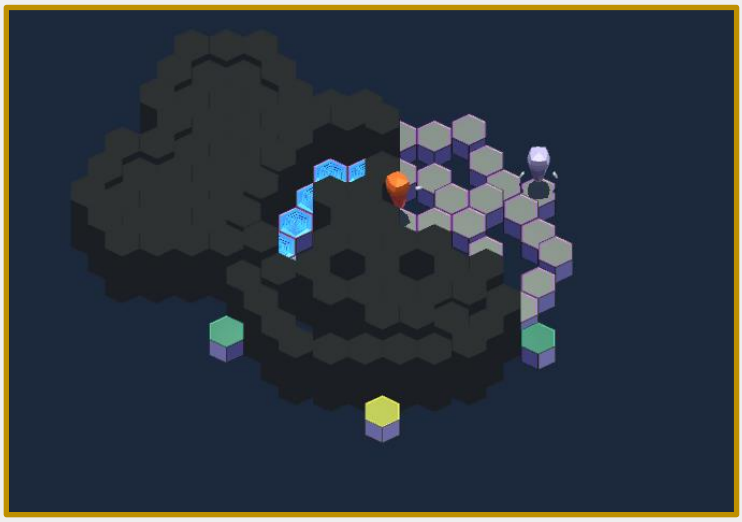
“Tactical Game Rules” has the logic that handles game start, completion, and team start/end turns.

Starting Team: Set’s which team goes first.
B Manually Place Units: If true, then the plays will be able to choose their spawn points defined on the LevelCell.
B Should Hit Anim On Move: If true, this will play hit animations if you are getting hurt while moving.

Asset Right Click Menu:
Create > TurnBasedTools > GameRules > Create TacticalGameRules





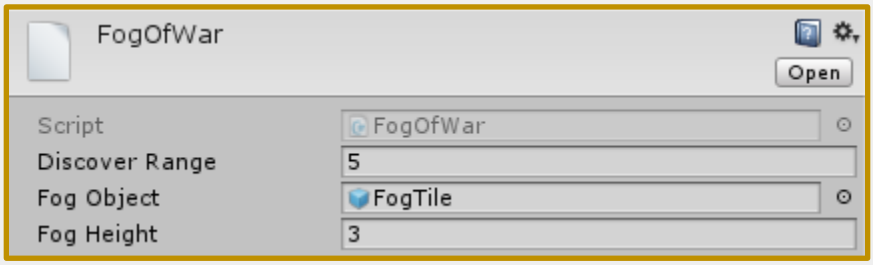


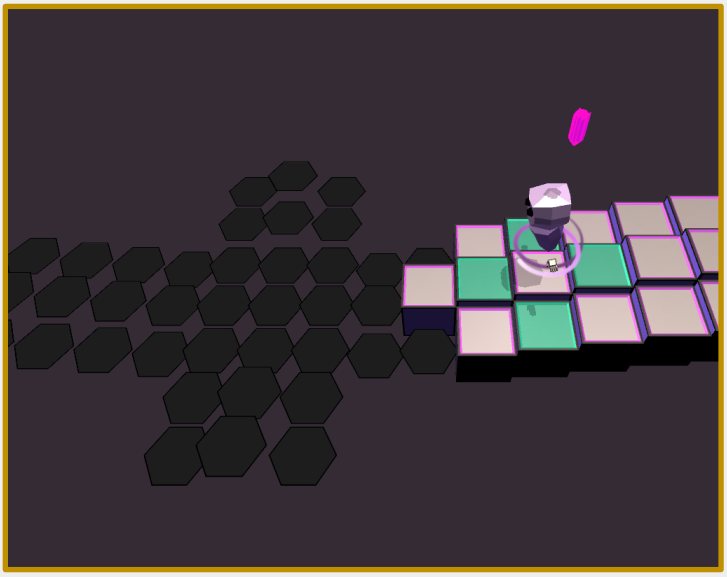
Fog of War

This is optional but allows you to make invisible cells visible when units are placed or moved near them. Enemy units can still move around, and their visibility will change based on the cell they are over.

Discover Range: How far away a friendly unit must be to become visible.
Fog Object: The object placed on invisible Level Cells, removed when discovered.
Fog Height: How many Fog Objects are stacked on an invisible cell.

Asset Right Click Menu:
Create > TurnBasedTools > Create FogOfWar



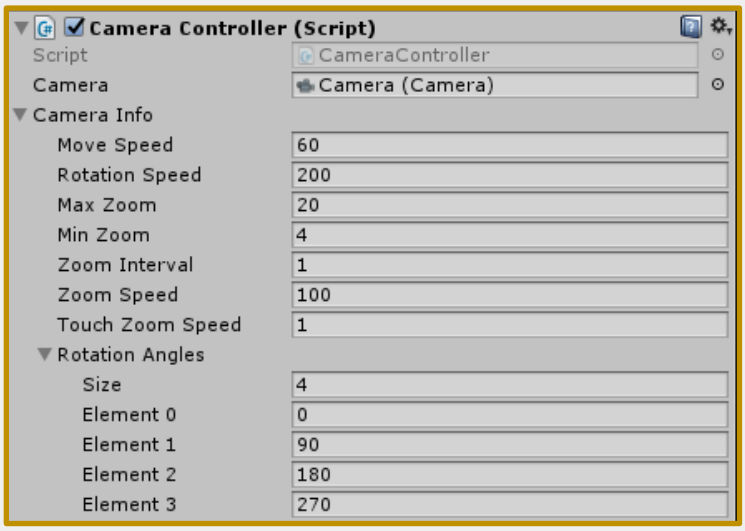


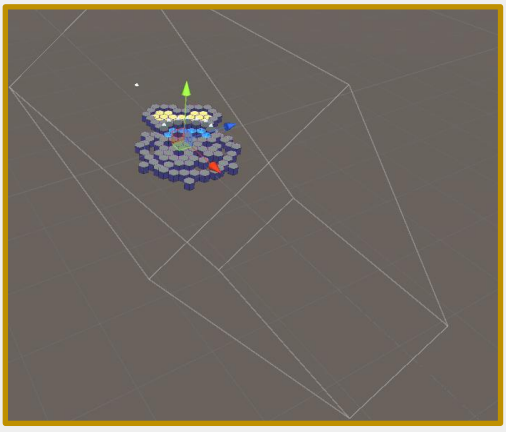
Camera Controller

A camera controller must be placed in scene, I added some pre-made ones in the "_Cameras" folder for 2D, Isometric, and Perspective views.

The Camera Controller also handles {w, a, s, d} for movement, Mouse wheel for zooming, and {q, e} for rotation.

Camera Info:
Move Speed: The panning speed.
Rotation Speed: How quick the rotate is.
Max Zoom: How far out you can zoom.
Min Zoom: How close you can zoom in.
Zoom Interval: How far to zoom each scroll.
Zoom Speed: How quick the zoom is
Touch Zoom Speed: How quick the zoom is for pinch zooming.
Rotation Angles: The angle list for rotating. It will circle around to the start.





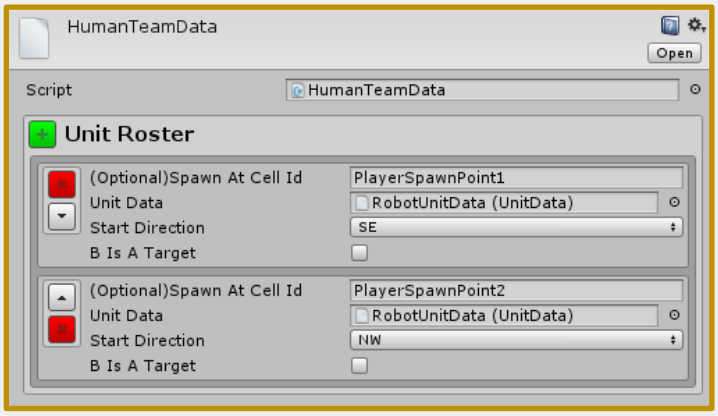
Team Data

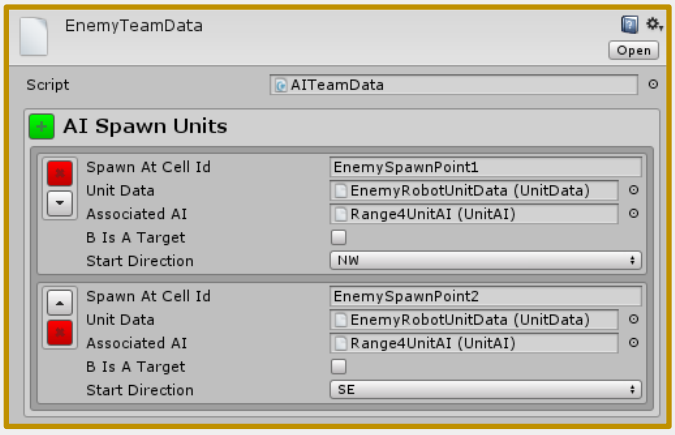
Team Data is used for spawning units.
There are two types: one for Humans, and the other for AI. The main difference is that the AI Team Data requires a UnitAI, and a spawn Id.
For the Human Team Data: if you don't specify a spawn point, then it will choose a random Level Cell that is marked as a friendly spawn point.

All:
Spawn at Cell Id: The Id of the cell that you want to spawn at, see Level Cell
Unit Data: Unit to spawn
B Is A Target: Used in the "Defeat all targets" WinCondition.
Start Direction: The direction the unit will face.

AI Only:
Associated AI: The UnitAI used for the enemy turn.

Asset Right Click Menu:
Create > TurnBasedTools > PlayerTeamData > Create (HumanTeamData/AITeamData)



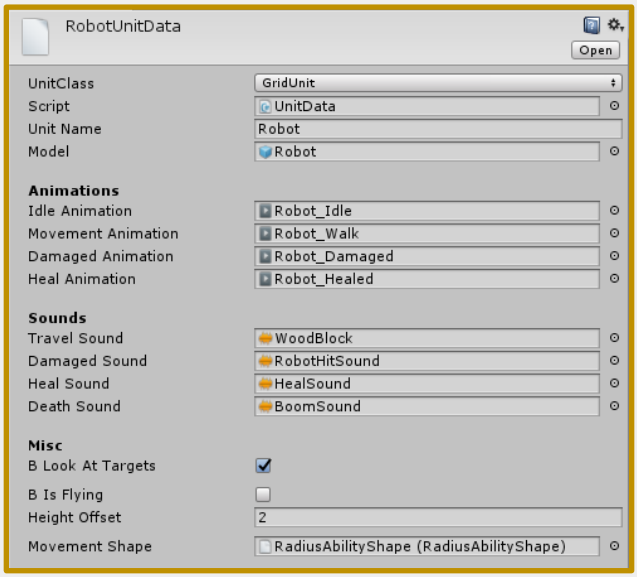


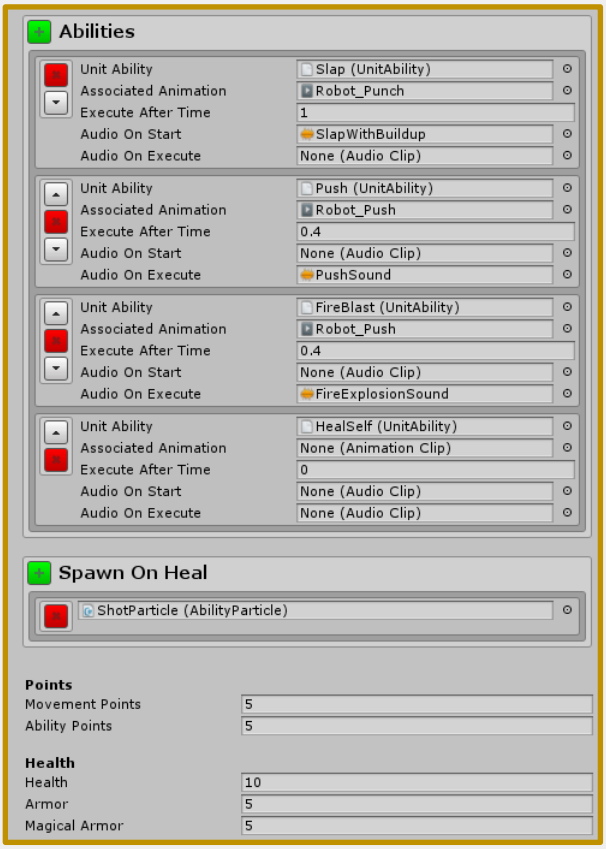
Unit Data

This holds everything about a unit, including model, animation, sounds and Unit Abilities. Animations need to be named the same in their Animator. Movement points, and ability points will be refreshed at the beginning of each turn.

Unit Class: The class derived from GridUnit to be applied on creation.
B Look at Targets: If true, the unit will look at the other unit it is attacking, when it's getting attacked, and when moving.
B Is Flying: If true, the unit will be able to move over blocked cells, See Level Cell
Height Offset: This is how far above the cell the unit will be.
Movement Shape: The Ability shape used for movement.
Abilities: Here you can set the Unit Ability, animation, and the time it takes for the animation to get to the action point.

Asset Right Click Menu:
Create > TurnBasedTools > Create UnitData



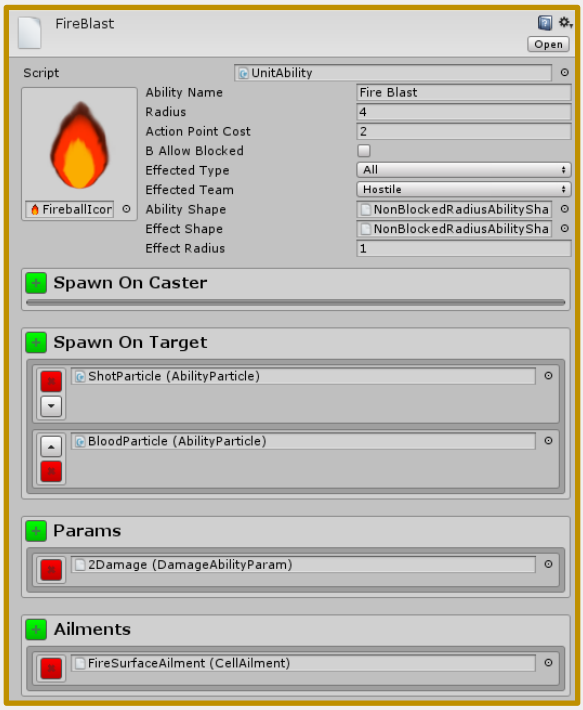


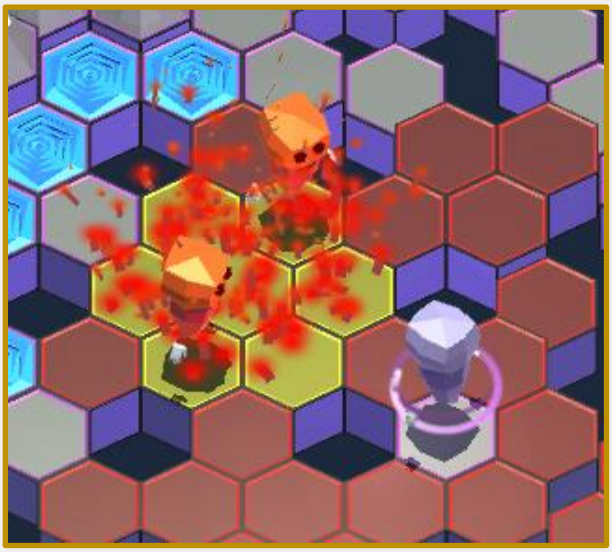
Unit Ability

The unit ability has all info needed for setting up, and executing an ability, and can be added to a Unit Data.

B Allow Blocked: This will allow blocked cells to be affected by the ability.
Effected type: If Flying or Ground, you will only be able to affect cells of that type. “bIsFlying” is set on the Unit Data, if false the unit is seen as ground.
Effected Team: This will block the execution on units of the opposite type, ex.) if set to hostile, you won’t be able to select a cell with a player on your team. This also decides the color of the ability, Red for hostile, green for friendly. None means it can’t be placed on a cell occupied by anyone.
Ability Shape: The Ability shape based on your position and range.
Effect Shape: this is optional and changes the shape of execution.
Spawn on Caster/Target: This will spawn an Ability Particle on either the Target, or Caster.
Params: The Parameters that will be applied to the target LevelCell, and/or target GridUnit. See AbilityParam.
Ailments: The ailment either applied to the target or unit.

Asset Right Click Menu:
Create > TurnBasedTools > Ability > Create New Ability

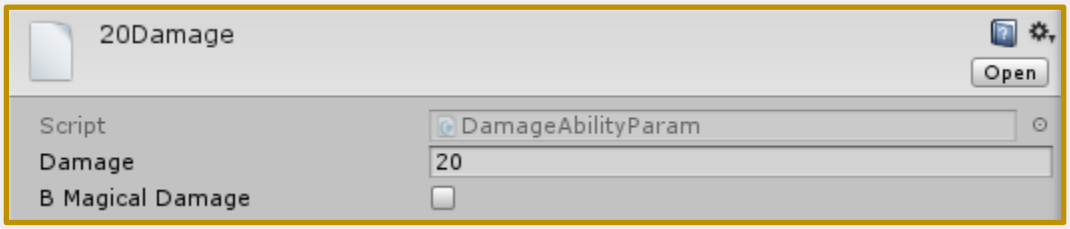


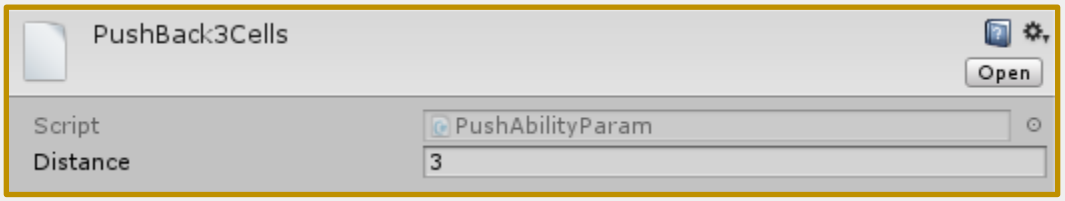


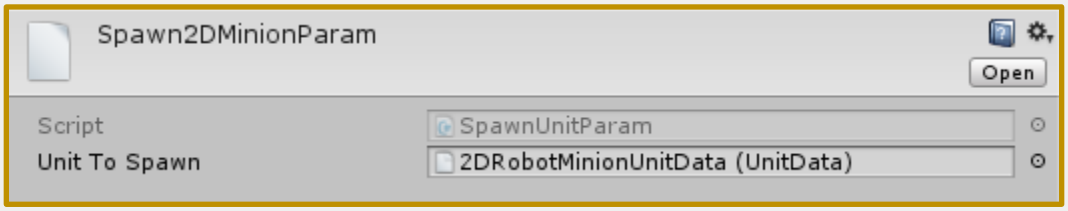
Ability Param

An ability parameter is added to a Unit Ability and can affect either GridUnits, and/or LevelCells depending on the ability. I have included some premade params for damage, healing, pushing, and spawning units. You can easily make your own by deriving a script from AbilityParam, and overriding the “ApplyTo” functions, along with the “GetAbilityInfo” which is used for UI.

Asset Right Click Menu:
Create > TurnBasedTools > Ability > Parameters



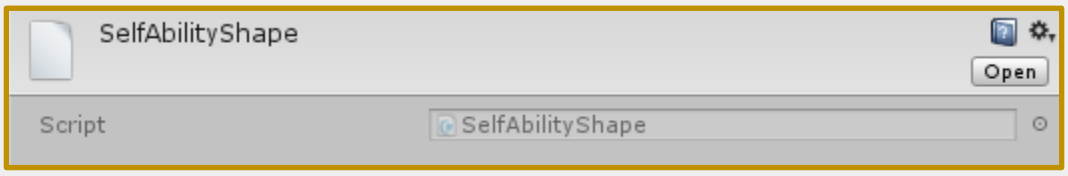


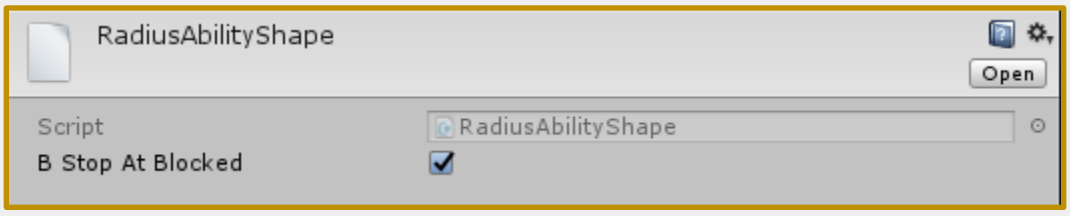


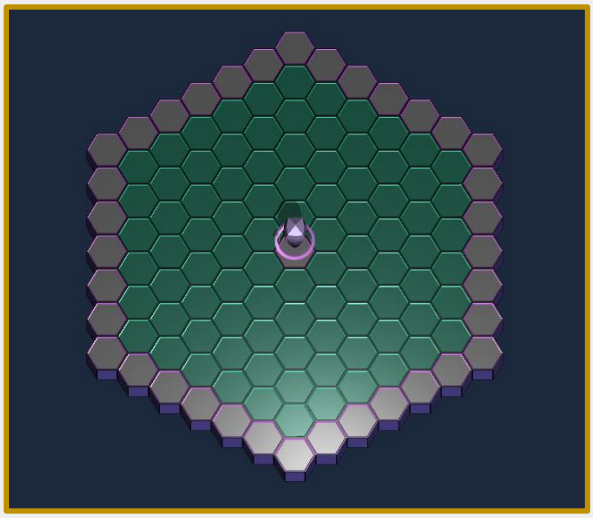
Ability Shape

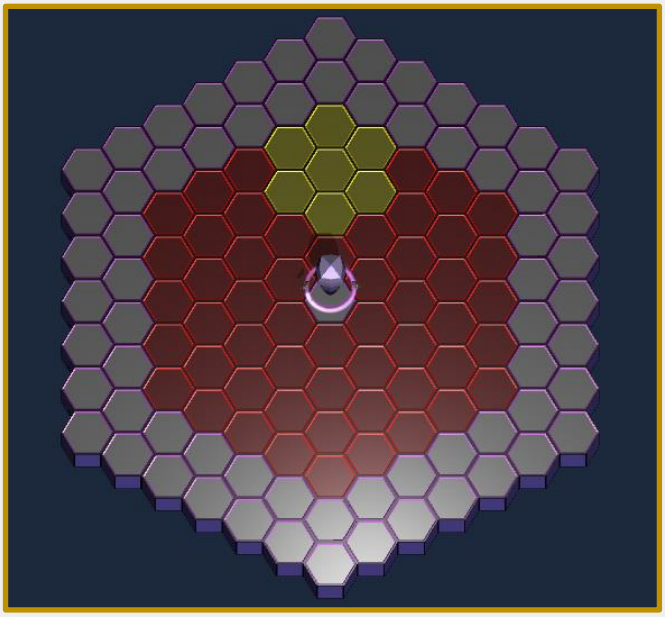
The ability shape is used for movement, ability, and effect. See UnitData, and UnitAbility.
I include some shaped already: Directional, Radius, and Self. You can create your own shapes by deriving a script from “AbilityShape” and overriding “GetCellList”.

Asset Right Click Menu:
Create > TurnBasedTools > Ability > Shapes







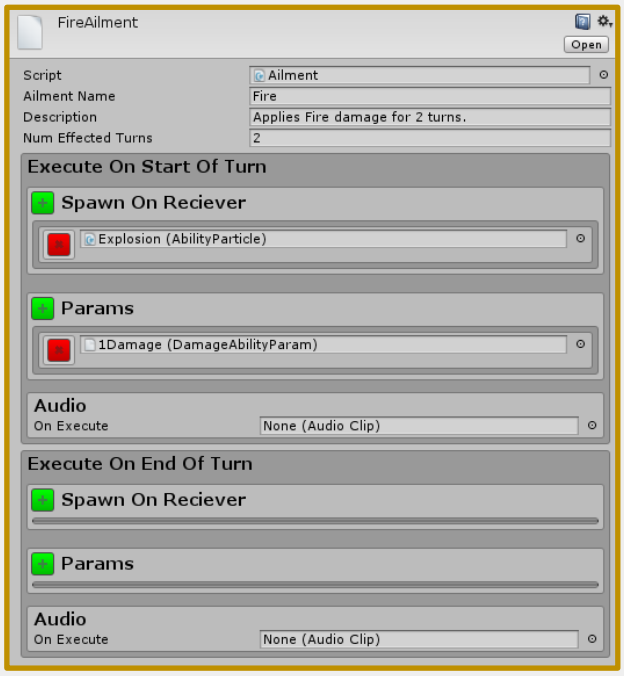


Ailment

There are two types of ailments: Normal ailments that effect units, and CellAilments, that effect LevelCells. Like the UnitAbility, you can also spawn Ability Particles, and play a sound for each event.

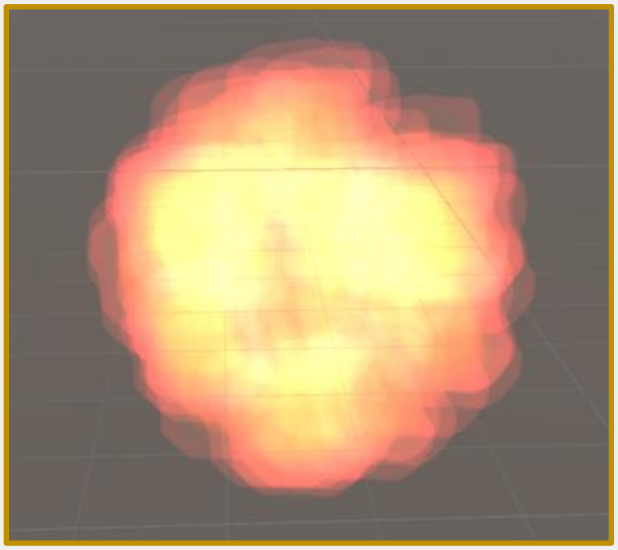
On Both: you can set it up to execute AbilityParams when their turn starts/ends.
On CellAilment: You can add parameters when a unit walks over and change the weight of the LevelCell.

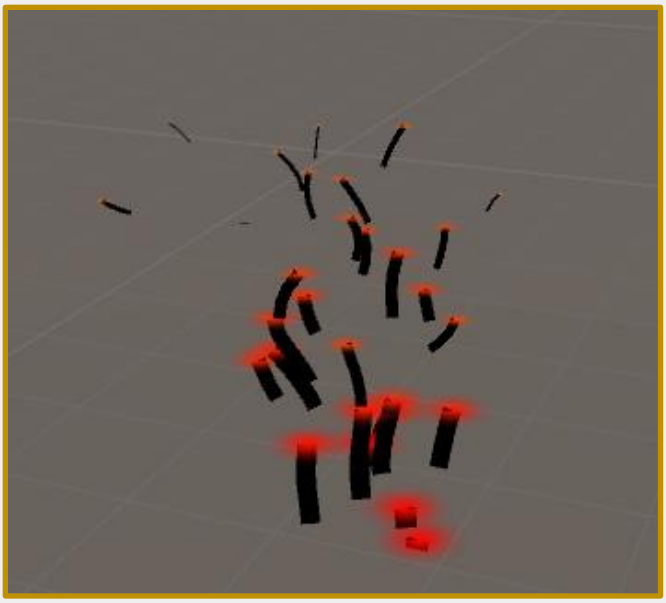
Asset Right Click Menu:
Create > TurnBasedTools > Create New Ailment/Cell Ailment



Ability Particle

Ability particles are used for UnitAbilties, and Ailments.
Any prefab can be an ability particle, to do so, all you must do is add the ability particle script to the prefab and set it's "Delete After Time".

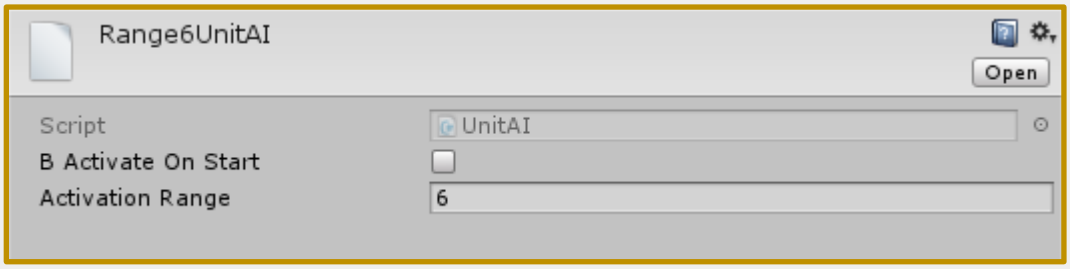




Unit AI

This is how AI units handle their turn. The Unit AI can be specified on an AITeamData. I have added a default UnitAI that moves towards you, and executes an ability based on its Action Points. You can set the activation range on it, or if it should always be active.

Asset Right Click Menu:
Create > TurnBasedTools > AI > Create UnitAIData



Win Conditions

Win conditions are the checks for If you have won or lost. You can add multiple Win Conditions to the game manager, and all conditions will have to pass for it to count as a win, if you fail any, you lose. I have included five Win Conditions, and you can also make your own win conditions.

Included Win Conditions:

- Defeat all opponents: Kill all units on the opposing team.
- Defeat all targets: Kill all opposing target units defined in the TeamData.
- Defeat number of opponents: Kill a specific number of opposing units.
- Discover Area: Requires FogOfWar, you must discover all the invisible cells.
- Survive number of cells: Survive with at least one unit on your team surviving.

Asset Right Click Menu:
Create > TurnBasedTools > WinCondition