for Unity 3d 3.5.x and 4.0.x
by Altered Reality Entertainment LLC
v1.0.2

Overview

What is Game Grid?

Product Description

Game Grid is a Unity scripting package that makes it easy to create games that include grid based gameplay. It supports both grid based games using full 3d art, and games using full 2d sprite art. Game Grid also includes additional features to make creating grid based games even easier, such as automatic navigation data generation and A-Star pathfinding support.

Features

- Easily setup a Game Grid for your 2d or 3d grid based game
- Custom Inspector for easy setup and configuration
- Fast to integrate into your game; only 2 components required
- Automatic directional nav-mesh generation
- A-star pathfinding
- Full C# source code included
- No scripting required
- Full Documentation
- Included example level

Why?

Game Grid was created to make it easier for everyone to make grid based such as a turn based RPG or strategy game. Below is a general overview of all of the components of Game Grid, and exactly what they do:

Key Terms

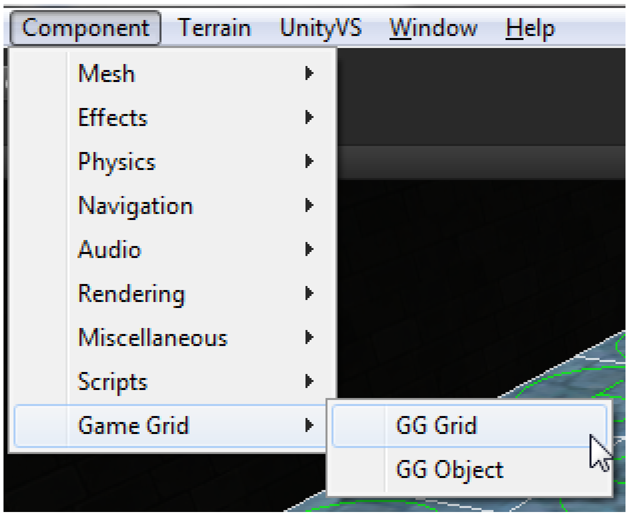
- The ***Grid*** is the main component that controls all of the grid functionality in Game Grid. It contains all of the cell in the grid and all pathability data.
- The **Grid Object** is the component that you put on your game objects that you wish to be included in the Grid.

Workflow

Initial Setup and Creation

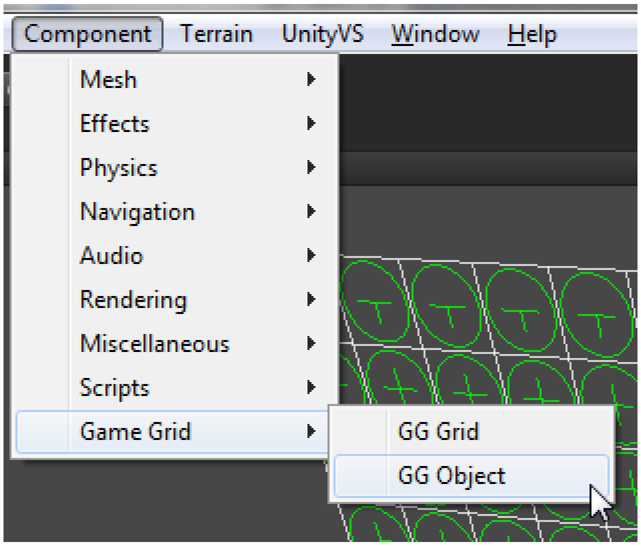
This section will outline how to use Game Grid and set it up for your project.

1. **Create a GG Grid Game Object**
   - Create a new Game Object
   - Select your new Game Object
   - Attach a GG Grid Component to your new Game Object



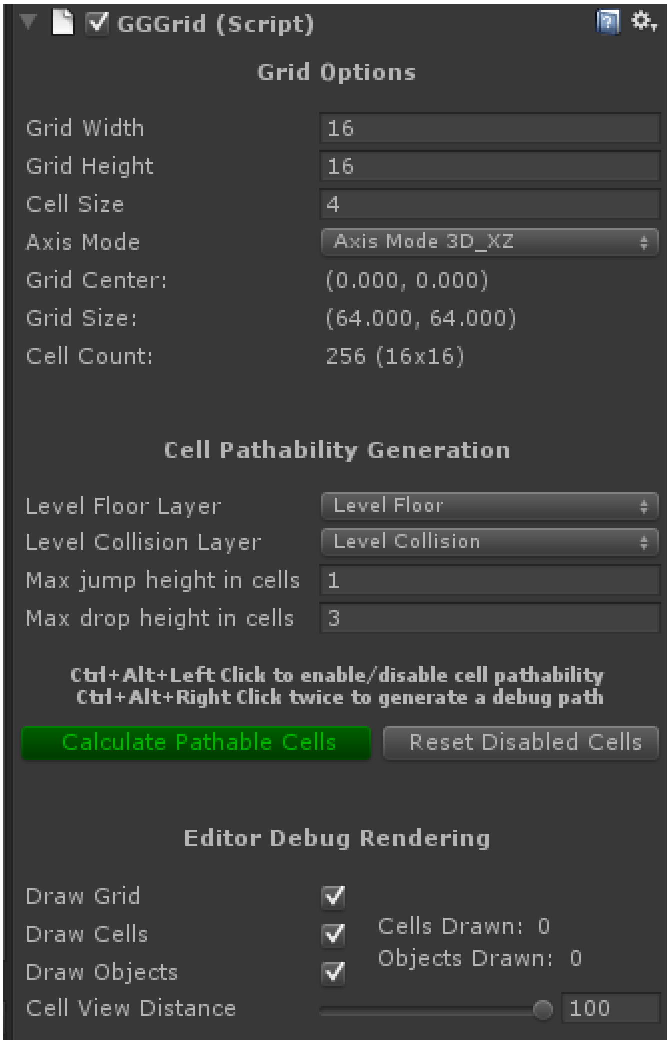
2. **Configure the GG Grid**
    - Make sure your GG Grid Game Object is selected
    - Configure your GG Grid as needed, details on the inspector settings can be found in the next section below.

3. **Add GG Object Components**
    - Select any object you wish to interact with, and be in the GG Grid
    - Attach a GG Object Component to the object



Configuration

This section focuses on how you go about configuring your GG Grid and Grid Objects.



☑ GGGrid (Script)

Grid Options

Grid Width: 16
Grid Height: 16
Cell Size: 4
Axis Mode: Axis Mode 3D_XZ
Grid Center: (0.000, 0.000)
Grid Size: (64.000, 64.000)
Cell Count: 256 (16x16)

Cell Pathability Generation

Level Floor Layer: Level Floor
Level Collision Layer: Level Collision
Max jump height in cells: 1
Max drop height in cells: 3

Ctrl+Alt+Left Click to enable/disable cell pathability
Ctrl+Alt+Right Click twice to generate a debug path

Editor Debug Rendering

Draw Grid ☑
Draw Cells ☑
Draw Objects ☑
Cells Drawn: 0
Objects Drawn: 0
Cell View Distance: 100

Grid Options

- Grid Width
  - The width of the grid in number of cells.
- Grid Height
  - The height of the grid in number of cells.
- Cell Size
  - The size of each cell in the grid.

- Axis Mode
    - This defines if this GG Grid should be laid out for a 3d grid or a 2d grid.
        - 3D X-Z Axis
            - This axis mode is to be used for a full 3d grid based game. This mode aligns the grid along the X-Z axis. This mode includes an advanced feature that will use your level collision to determine the pathable areas in your level
        - 2D X-Y Axis
            - This axis mode is to be used for a 2d grid based game. This mode aligns the grid along the X-Y axis.

Cell Pathability Generation

- Level Floor Layer *(in 3d X-Y axis mode only)*
    - This is the collision layer used for the floor/walkable surface of the level. (i.e. collision in this layer is considered valid to be walked on, and cells will be pathable here)
- Level Collision Layer *(in 3d X-Y axis mode only)*
    - This is the collision layer used for the environment collision for the level. (i.e. the collision in this layer will get in the way of pathability, and cells blocked by this will NOT be pathable)
- Max jump height *(in 3d X-Y axis mode only)*
    - This is the maximum height that is allowed when jumping up from a lower cell to a higher cell. This is used when generating pathable directions between cells.
- Max drop height *(in 3d X-Y axis mode only)*
    - This is the maximum height that is allowed when dropping down from a higher cell to a lower cell. This is used when generating pathable directions between cells.
- Calculate Pathable Cells
    - This button will calculate which cells are pathable in the grid in your level based off of the level floor and collision layers. It also generate which cells can be moved between based on the max jump and drop height. If this button is red, it means your pathable data is invalid, and needs to be generated again. If it is green, that means all of you pathable data is valid. (In 2d axis mode, this just marks all cells as pathable.
- Reset Disabled Cells
    - It is possible in Game Grid to disable the pathability of cells in the world after the pathability is generated. This button will reset all cells to have pathability enabled. They can also be enabled/disabled during runtime for gameplay reasons. In the editor, pathability can be enabled/disabled by holding Alt + Ctrl and left clicking in

the Scene view.

Editor Debug Rendering

- Draw Grid
  - This will draw the selected GG Grid in the Scene view in the editor.
- Draw Cells
  - This will draw the cells in the selected GG Grid in the Scene view in the editor.
- Draw Objects
  - This will draw the objects in the selected GG Grid in the Scene view editor.
- Cell View Distance
  - This is the view distance multiplier that is used to determine whether or not a debug element is too far from the camera to draw.

A* Pathfinding

In Editor

In the editor, it is possible to perform a debug A* pathfinding operation. In Scene view, while holding Alt+Ctrl, right click on a first cell and then a second. A debug A* path will be displayed between the two cells (assuming they can reach each other).

In Game

In game, an A* path can be found between any two cells. To get a path, call GGAStar.GetPath(...), the parameters and return are as follows…

- Param: start
  - The cell the start the path from.
- Param: dest
  - The cell to end the path at.
- Param: ignoreOccupiedAtDestCell
  - This indicates if the path should be considered valid even if the dest cell is occupied.
- Param: allowAnyOccupiedCells
  - This indicates if the path should be considered valid even if the ANY cell is occupied.

- Return: List<GGCell>
  - The list of cells that make up the path, starting with the start cell. An empty list will be returned if no path was found.

Support

Twitter

twitter.com/AlteredR

Email

AlteredRealityEnt@yahoo.com

Website

www.alteredr.com

Forums

forums.alteredr.com

Credits

Lead Programmer and Architect

Andrew Thayer

Special Thanks

Valerie Nunez
Leslie Thayer
Caiden Thayer
Clara Thayer