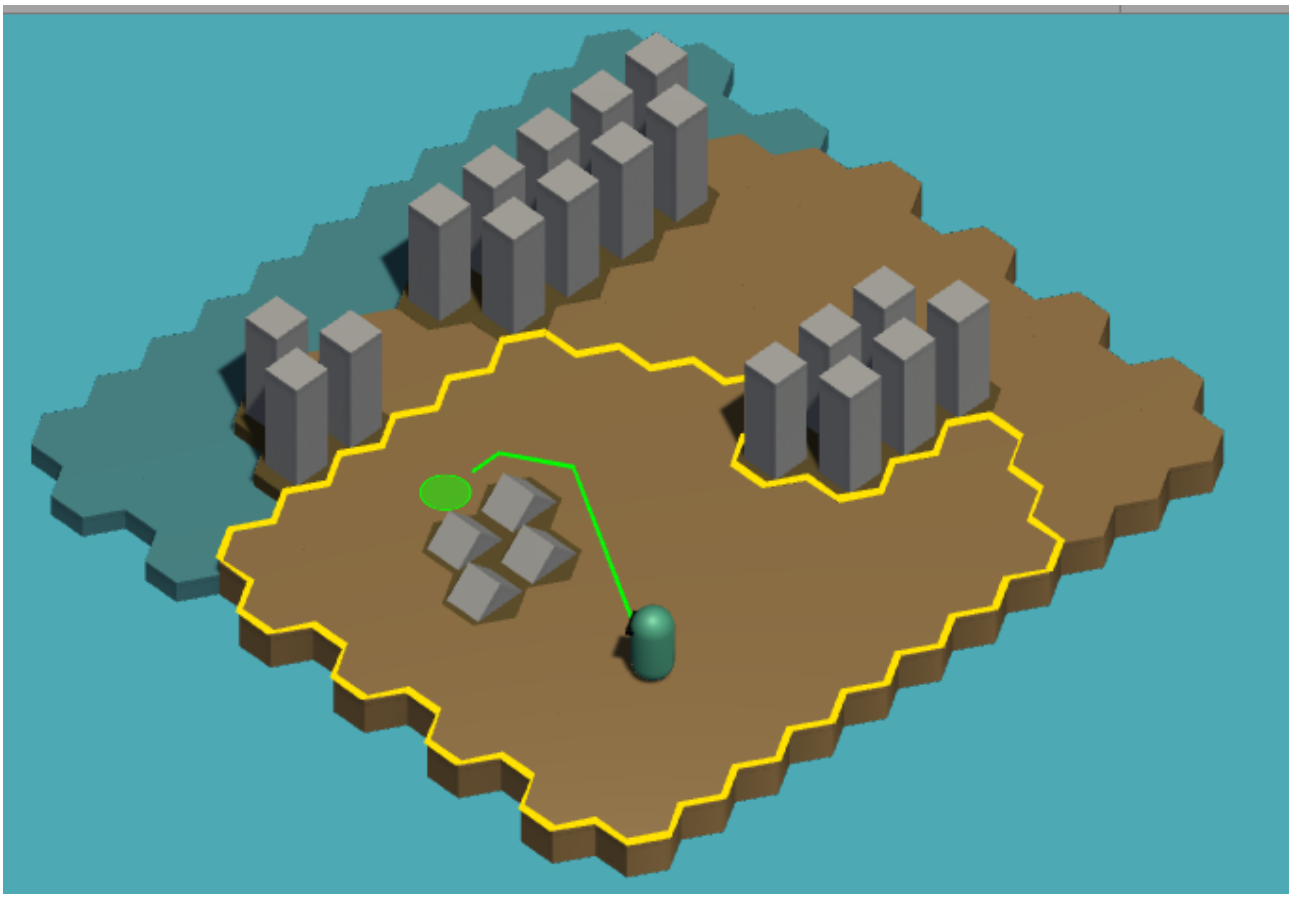
ProtoTiles

Documentation



Contents

Join our Discord

Overview

This asset is developed to speed up prototyping process of game developing.

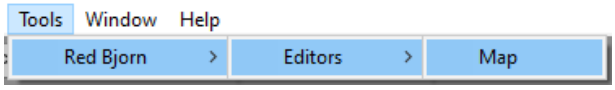
How? Just by creating helpful data with an easy tool

By the meaning of helpful data, we understand an asset (Map Settings) which contains information about bunch of tiles which represent a field where any kind of gameplay can be created. It is supposed that turn-based games use such data more often than other genres.

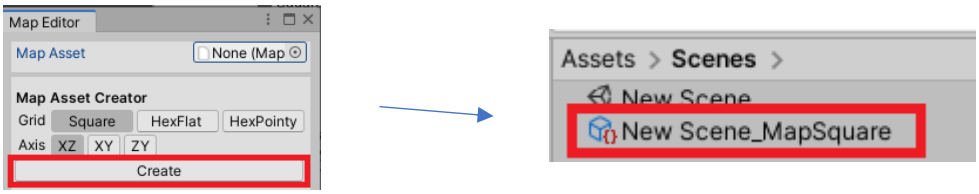
Easy tool means than we have clear way of creating data and it is called Map Editor. Simple steps will lead you to your first map

Quick Start

- Open Map Editor from Tools menu

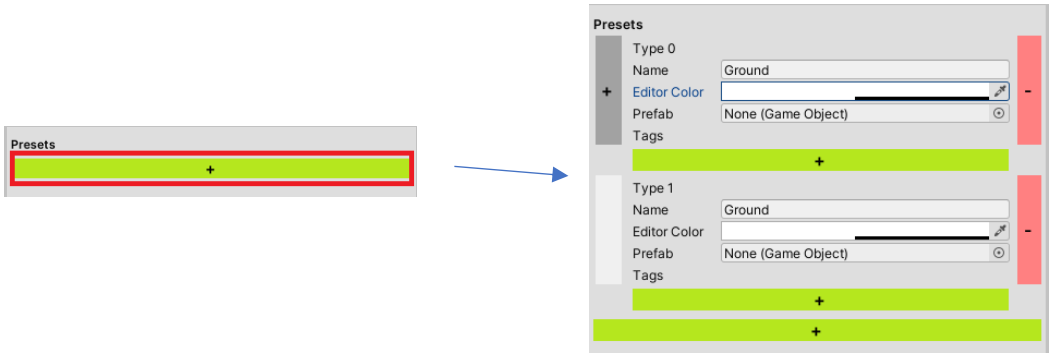


- Define which grid type (Square or Hex) and axis type (XZ, XY, ZY) should be used and press Create

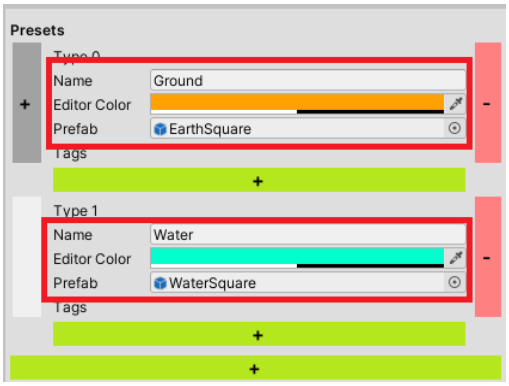


MapSettings asset will be created at the folder where active scene asset is located

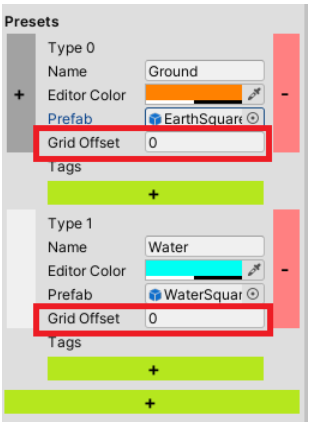
- Add two (or at least one) tile presets by pressing green button with plus sign.



Assign Name, Editor Color and Prefab (more information in Prefabs section for all presets



- (Optional) Define approporiate Grid Offset.



For example, our water tile prefab's (WaterSquare) top surface is lower than zero on the Y-axis. So, setting offset to -0.1 will make it looks better. Compare Fig.1 and Fig.2

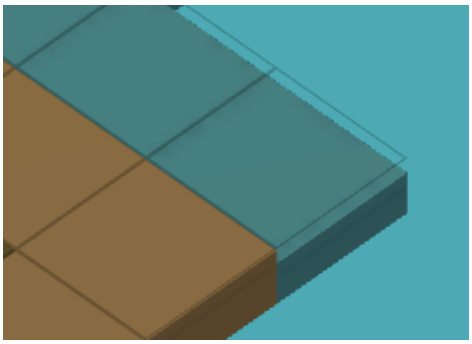


Figure 1 WaterSquare Grid Offset = 0

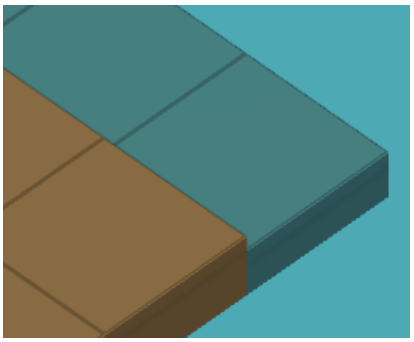
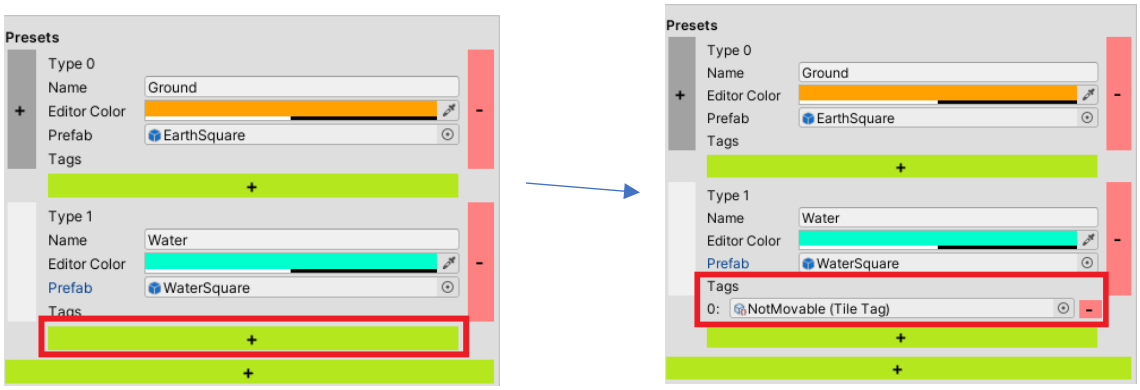


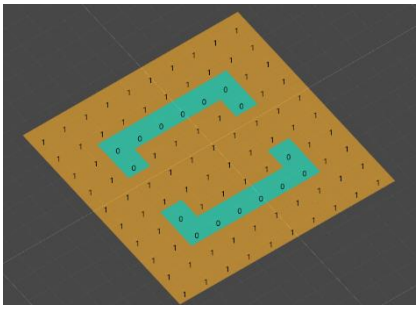
Figure 2 WaterSquare Grid Offset = -0.1

- Choose which preset will be marked as NotMovable by adding tag with same name

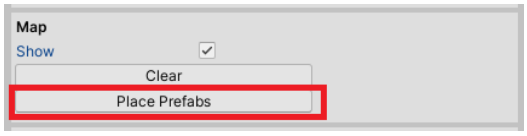


- Select brush tool

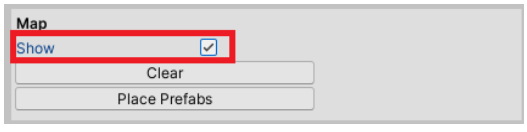
- Paint your map at Scene view

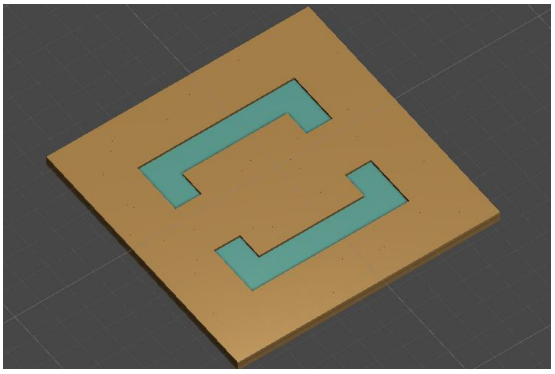


- Place Prefabs (from tile presets to scene view)



- Disable map

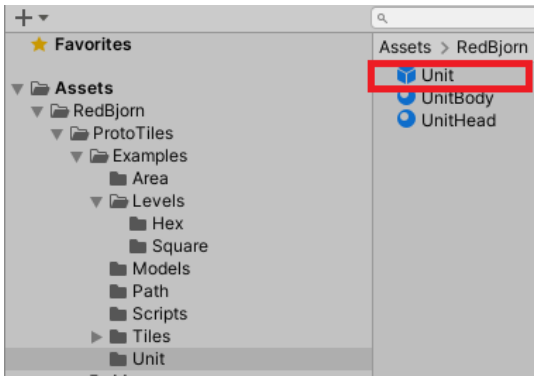




Join our Discord

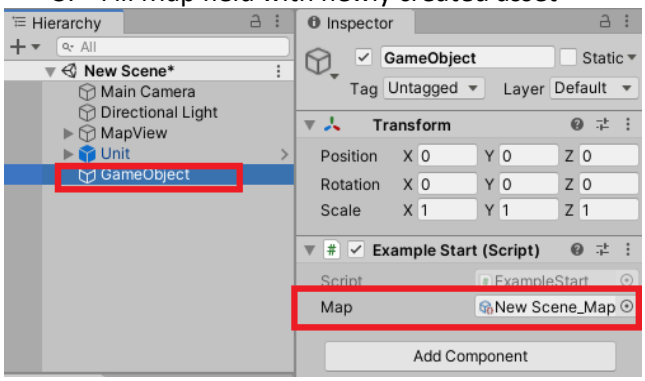
Additionally (For quick test)

Add Unit (Unit prefab from Examples)



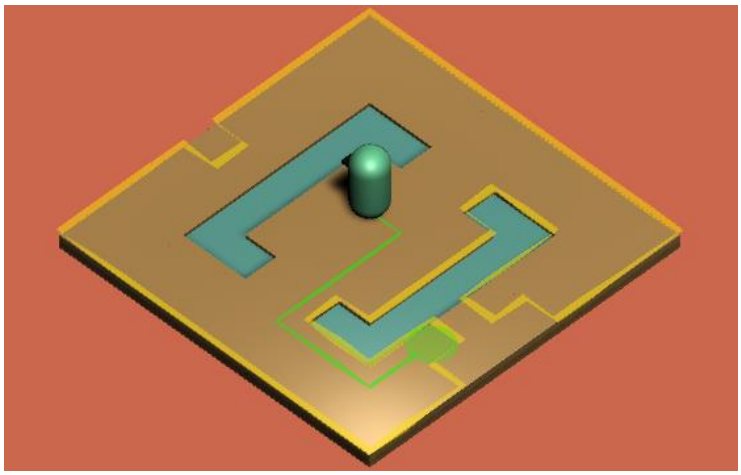
Add MapSettings link

1. Create new GameObject.
2. Add Example Start component.
3. Fill Map field with newly created asset



- Click playmode.

Ta-dam! Congratulations, you have created your first map with walkable information. Move Unit by clicking left mouse button to test how does it work



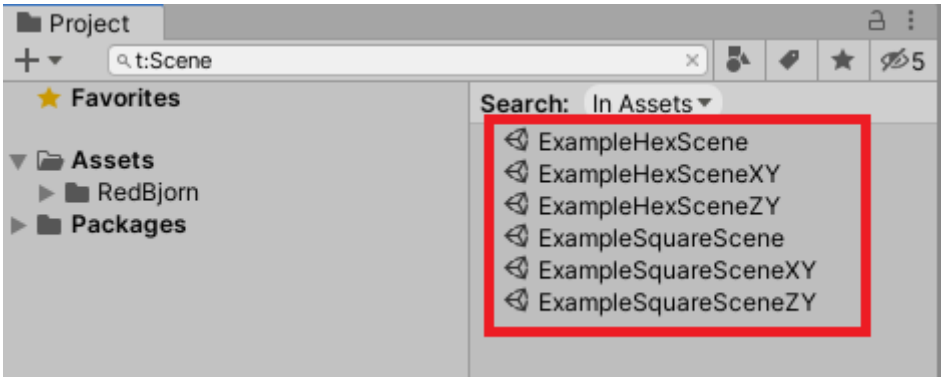
Essentials

Let’s have a look at folder structure in extraordinary way: from bottom to top

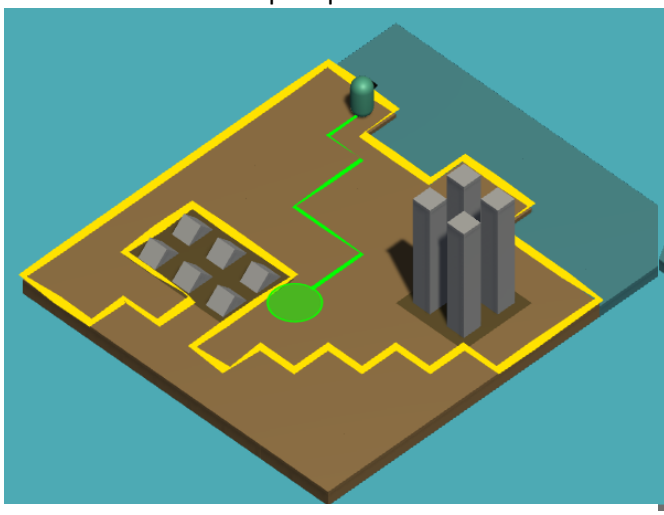
Utils	Folder includes plenty of useful extensions and methods for speeding up the developing process
ProtoTiles -> Map Editor	Folder contains scripts for drawing Map Editor window and logic of Brush tools at the scene View. Also, it includes several resources for UI of Map window
ProtoTiles -> Map -> Rules	Rules implemented as scriptable objects which specify logic for identifying walkable tiles
ProtoTiles -> Map -> Scripts	Includes wrappers for initials settings which usually contain Entity word. Also, the main script MapSettings with its dependencies is located here.
ProtoTiles -> Map -> Tags	Only one tag named NotMovable. It should be added to tile presets which will stop considering tile as vacant for pathfinding algorithm
ProtoTiles -> Examples	Contains 2 example maps with square and hexagonal tiles. Details at the Examples section

Examples

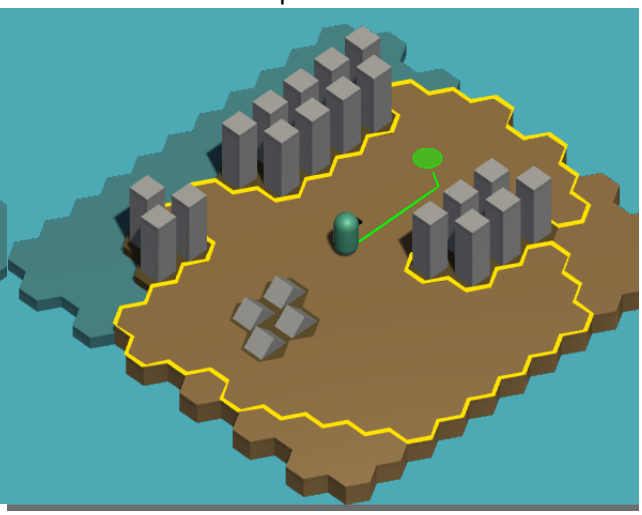
Package contains six example scenes



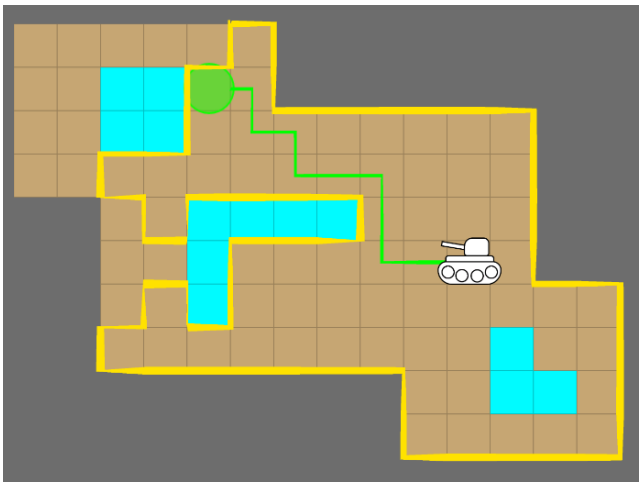
ExampleSquareScene



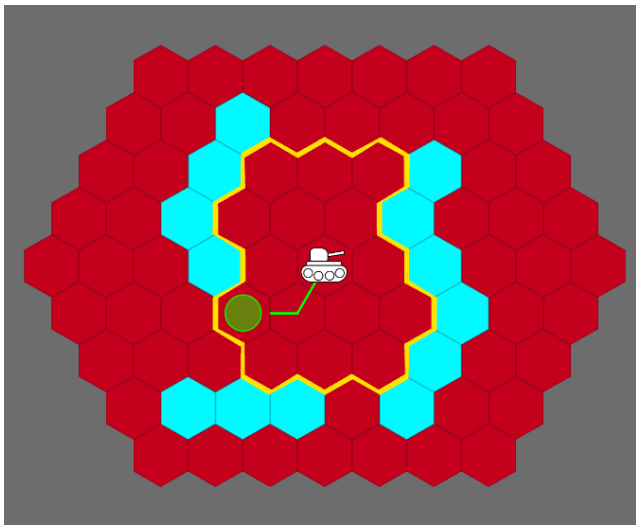
ExampleHexScene



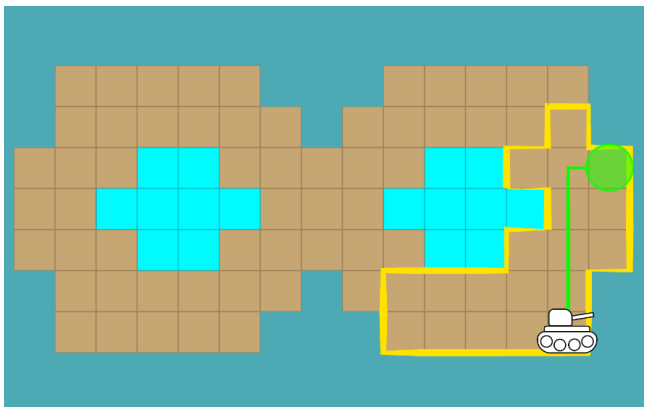
ExampleSquareSceneXY



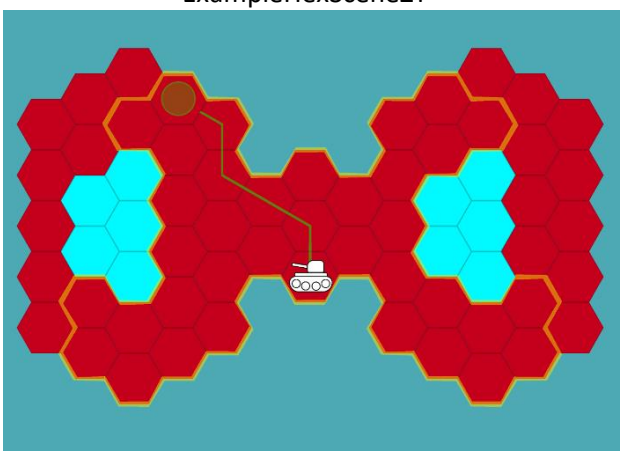
ExampleHexSceneXY



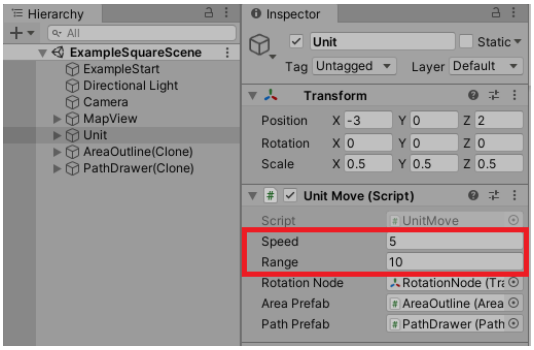
ExampleSquareSceneZY



ExampleHexSceneZY



- You can tune Unit's speed and range



- Unit moves by pressing up right mouse button inside area bounded yellow line
- Camera could be dragged by holding right mouse button and moving the mouse
- ExampleStart.cs script contains important injection of MapView and UnitMove classes (screenshot below) **Please, consider this fact when you change example scenes' logic.**

```
using UnityEngine;

namespace RedBjorn.ProtoTiles.Example
{
    public class ExampleStart : MonoBehaviour
    {
        public MapSettings Map;
        public KeyCode GridToggle = KeyCode.G;
        public MapView MapView;
        public UnitMove Unit;

        public MapEntity MapEntity { get; private set; }

        void Start()
        {
            if (!MapView)
            {
                MapView = GameObject.FindObjectOfType<MapView>();
            }
            MapEntity = new MapEntity(Map, MapView);
            if (MapView)
            {
                MapView.Init(MapEntity);
            }
            else
            {
                Log.E("Can't find MapView. Random errors can occur");
            }

            if (!Unit)
            {
                Unit = GameObject.FindObjectOfType<UnitMove>();
            }
            if (Unit)
            {
                Unit.Init(MapEntity);
            }
            else
            {
                Log.E("Can't find any Unit. Example level start incorrect");
            }
        }
```

Tips

- Create “Entity” classes for “Settings” classes during playmode. It prevents from modifying predefined map data and provides an opportunity to reset to default values

  MapEntity for MapSettings

  TileEntity for TileData

  It could be easily done by calling constructor. Example could be observed at ExampleStart.cs

- Most useful methods are located at MapEntity class
  Converting tile coordinates to world space coordinates and vice versa

```
152     /// <summary> Get world space position of center of tile entity
157     public Vector3 WorldPosition(TileEntity tile)...
161
162     /// <summary> Get world space position of center of tile bt it's coordinates
167     public Vector3 WorldPosition(Vector3Int tilePos)...
171
172     /// <summary> Get world space position of center of tile which is located at wor ...
177     public Vector3 TileCenter(Vector3 worldPos)...
```

  Calculating distance in tile space

```
198     public float Distance(TileEntity tileA, TileEntity tileB)...
202
203     /// <summary> Distance between two tiles located at corresponding world space po ...
209     public float Distance(Vector3 worldPosA, Vector3 worldPosB)...
```

  Calculating paths

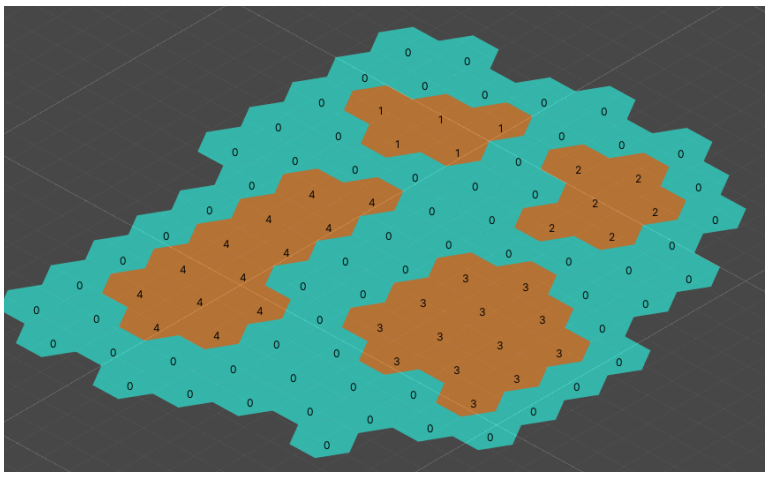
```
268     /// <summary> Get path that consist of tile entities
275     public List<TileEntity> PathTiles(Vector3 from, Vector3 to, float range)...
288
289     /// <summary> Get path that consist of world space positions
296     public List<Vector3> PathPoints(Vector3 from, Vector3 to, float range)...
```

  Outline of walkable border

```
334     /// <summary> Get positions of border of walkable area
340     public List<Vector3> WalkableBorder(Vector3Int tilePosition, float range)...
345
346     /// <summary> Get positions of border of walkable area
352     public List<Vector3> WalkableBorder(Vector3 worldPosition, float range)...
357
358     /// <summary> Get positions of border of walkable area
364     public List<Vector3> WalkableBorder(TileEntity origin, float range)...
```

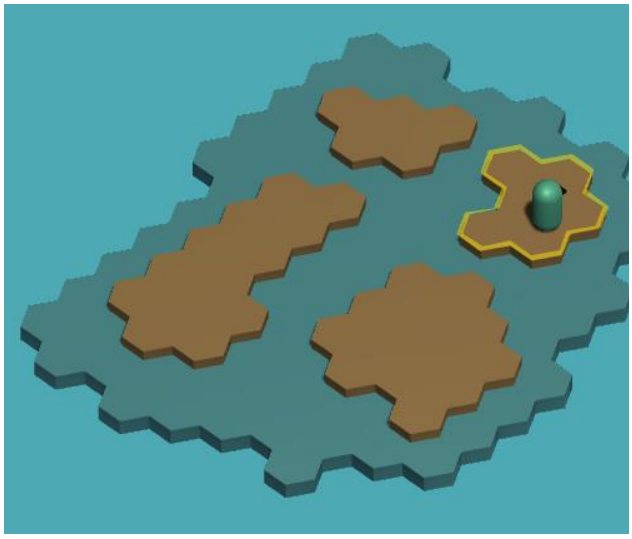
  Etc..

- Numbers which are located inside tiles during painting mode at Scene view indicate separate move zone areas
  0 index – predefined for NotMovable zone
  Indeces greater 0 - different movable zones

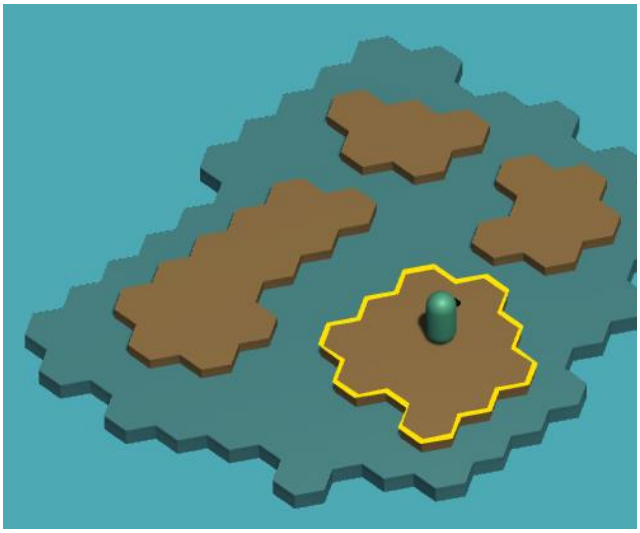


For example.

There is no walkable path from zone 2 to zone 1,3 and 4



There is no walkable path from zone 3 to zone 1,2 and 4



Prefabs

Right now, Map editor window helps to draw a map with the size of tile's border equal to 1 unit. Other border values will be implemented in future updates.

Knowing that information, let's have a look at some restrictions

Square Map

Common Unity primitive (cube) or square sprite can be the base of square tile.

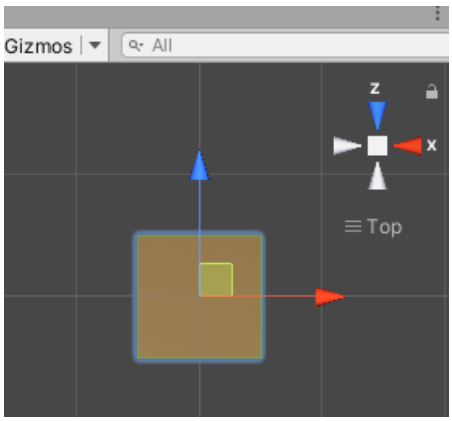


Figure 3

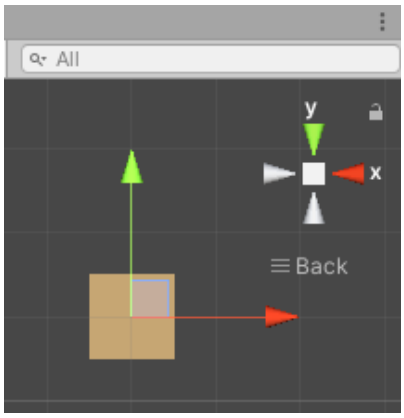


Figure 4

Tile sides should be only parallel to the corresponding map axis (Figure 5-6). Orientation doesn't matter as square (or cube) is symmetric along these axes. I recommend creating empty object as a parent of whole prefab for every tile because some tiles could have more than one object inside.

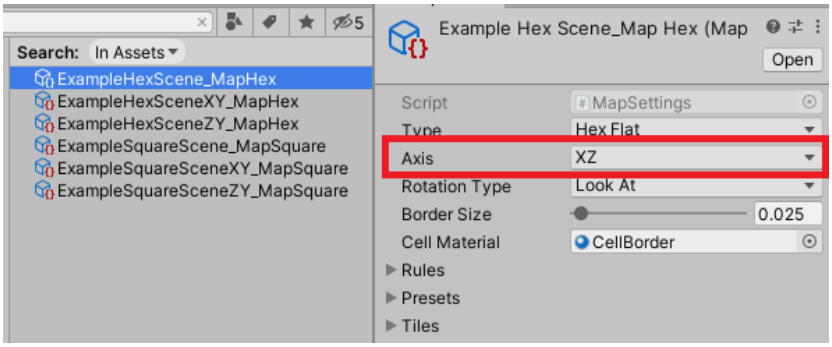


Figure 5

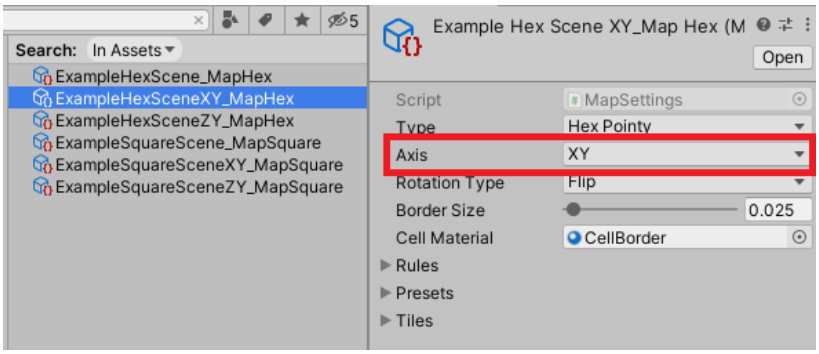
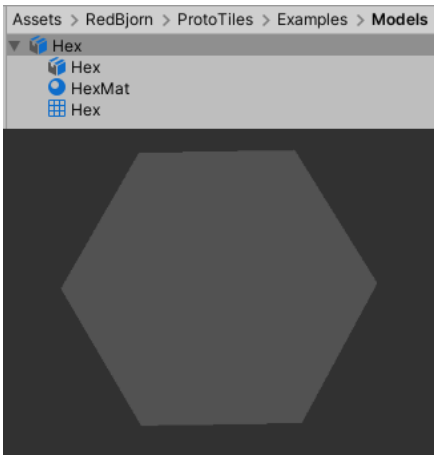


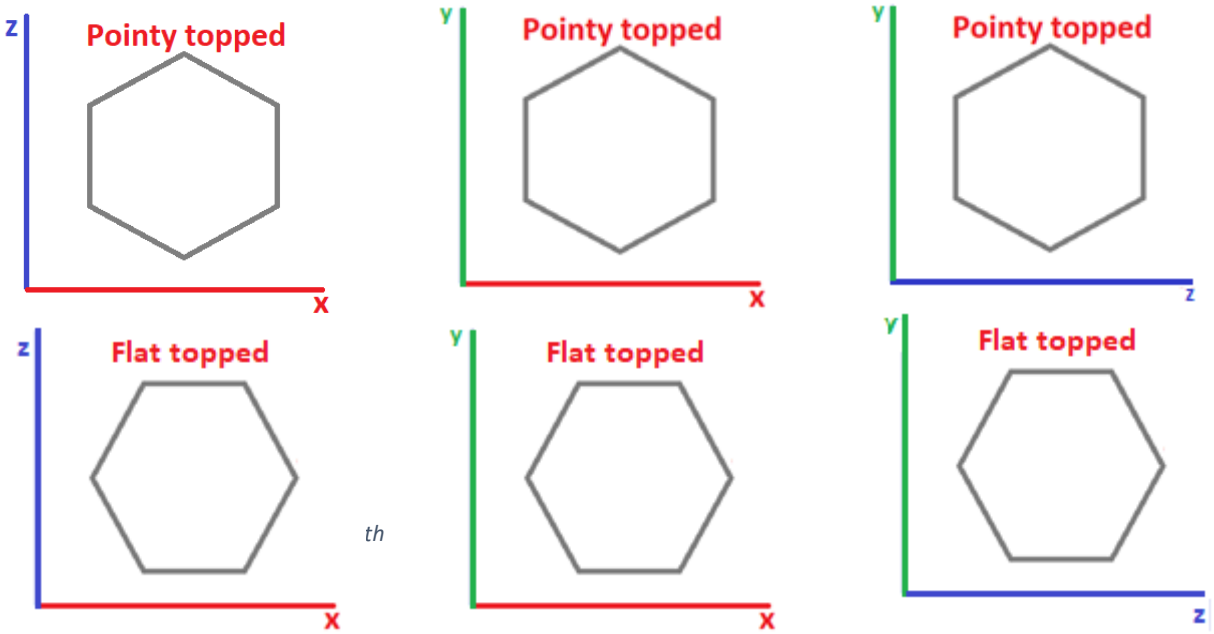
Figure 6

Hex Map

Hex have no Unity primitives out of the box, so feel free to use ProtoTiles hex.



But first let's define main orientational concepts: Pointy topped and Flat topped hexes.



th

If you prefer to use any other hex model, you could make it works by creating an empty gameObject as a parent of prefab and specify appropriate rotation.