ProtoTiles

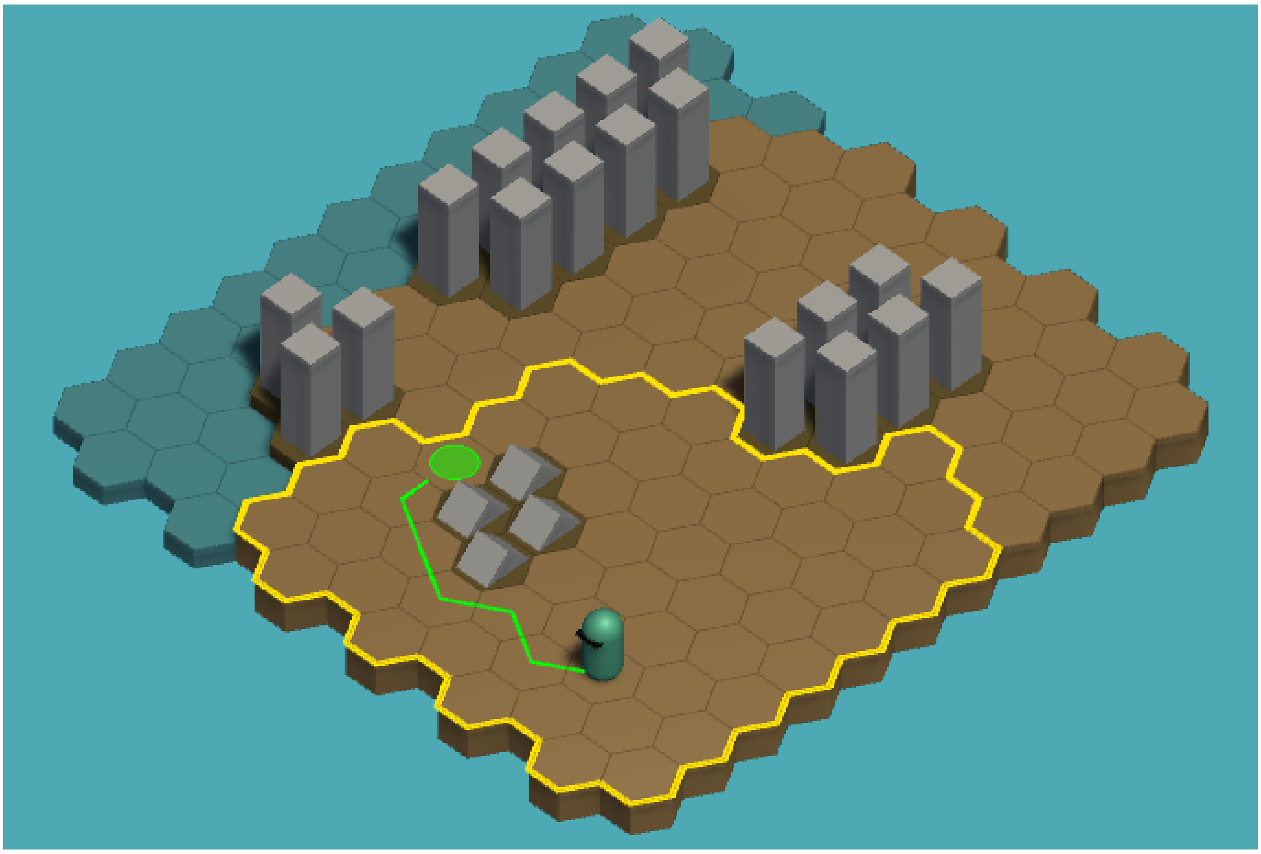


Table of contents

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

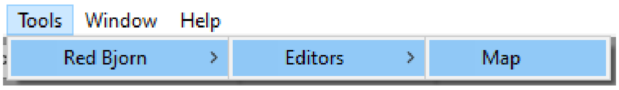
Online-documentation

https://discord.gg/5dTM9SfqME

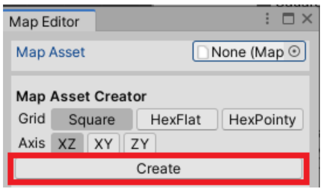
help@redbjorn.dev

Quick Start

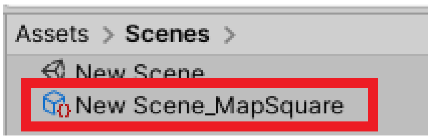
Open Map Editor from Tools menu



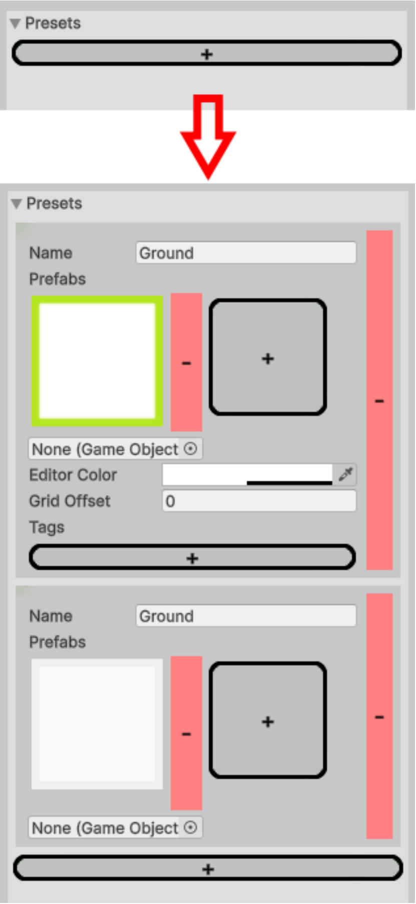
Define which grid type (Square or Hex) and axis type (XZ, XY, ZY) should be used and press Create



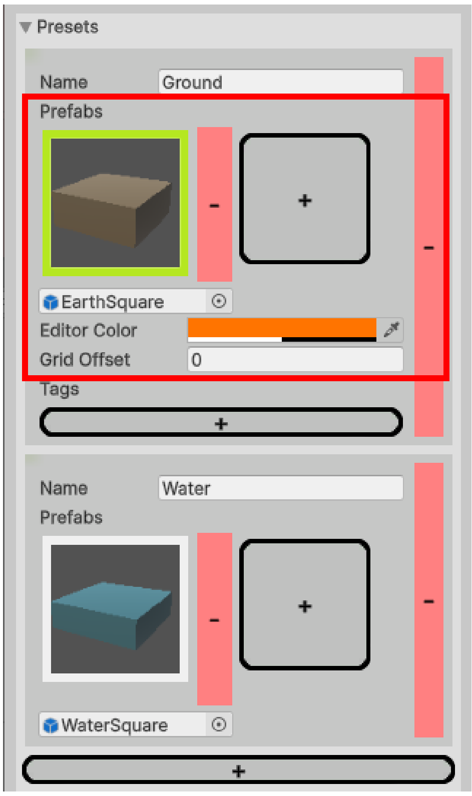
MapSettings asset will be created at the folder where active scene asset is located



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)



Presets

Name: Ground

Prefabs

EarthSquare

Editor Color

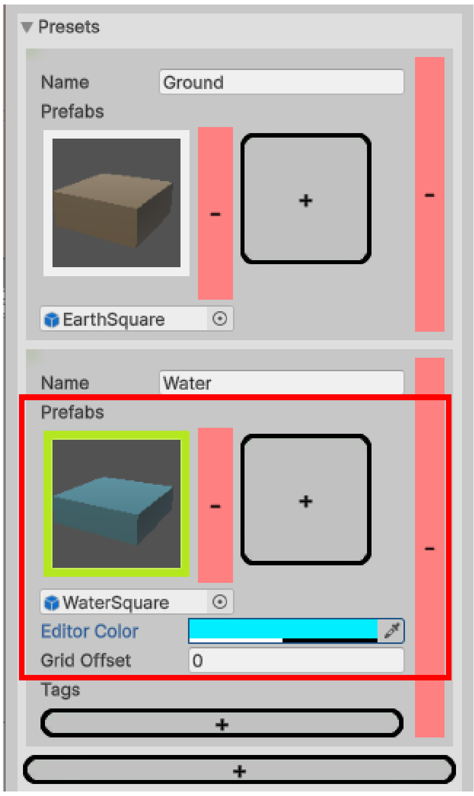
Grid Offset: 0

Tags

Name: Water

Prefabs

WaterSquare



Presets

Name: Ground

Prefabs

EarthSquare

Name: Water

Prefabs

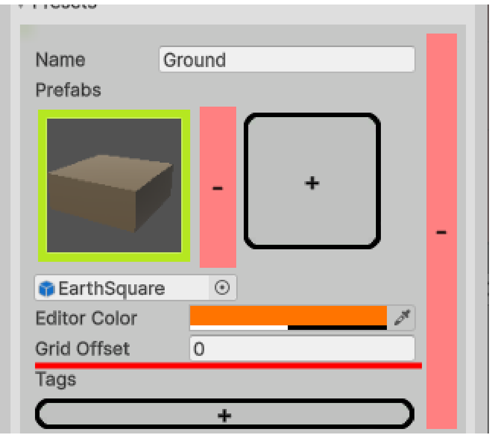
WaterSquare

Editor Color

Grid Offset: 0

Tags

(Optional) Define approporiate Grid Offset.



Name: Ground

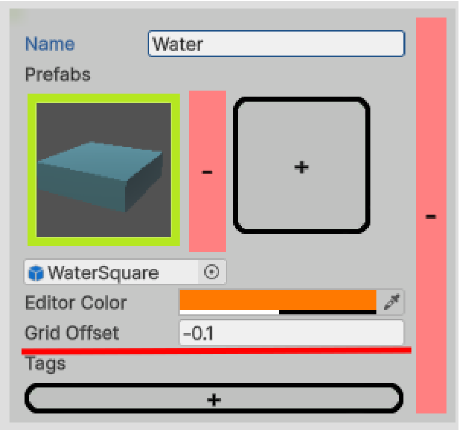
Prefabs

EarthSquare

Editor Color

Grid Offset: 0

Tags



Name: Water

Prefabs

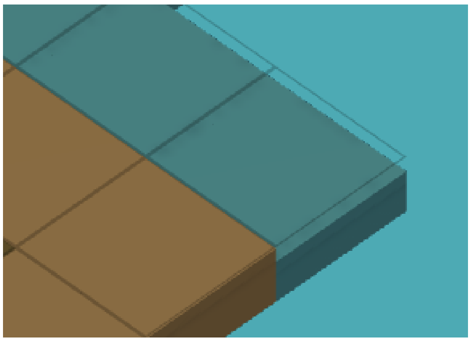
WaterSquare

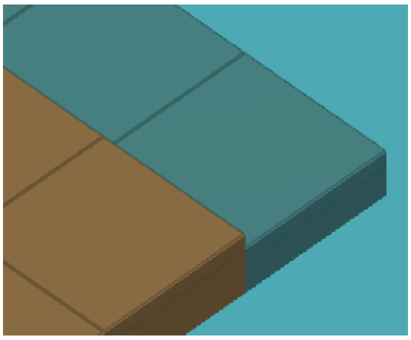
Editor Color

Grid Offset: -0.1

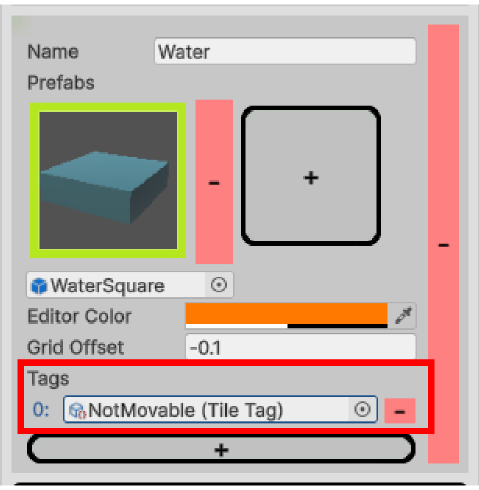
Tags

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





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



Name: Water

Prefabs

WaterSquare

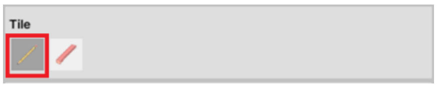
Editor Color

Grid Offset: -0.1

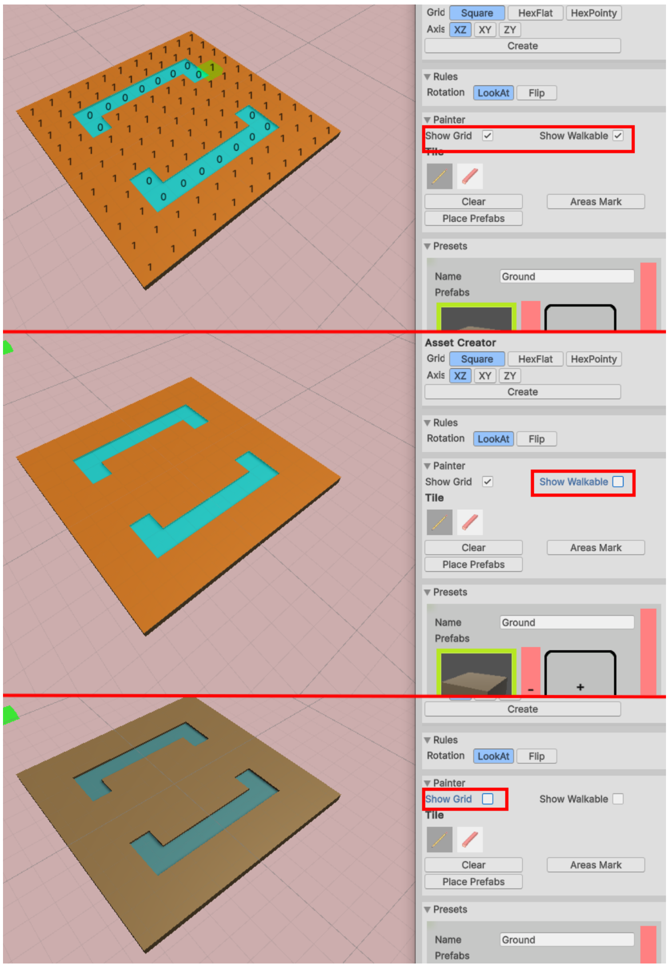
Tags

0: NotMovable (Tile Tag)

Select brush tool



Paint your map at Scene view



Grid: Square | HexFlat | HexPointy
Axis: XZ | XY | ZY
Create

Rules
Rotation: LookAt | Flip

Painter
Show Grid ☑ Show Walkable ☑
Tile
Clear | Areas Mark
Place Prefabs

Presets
Name: Ground
Prefabs

Asset Creator
Grid: Square | HexFlat | HexPointy
Axis: XZ | XY | ZY
Create

Rules
Rotation: LookAt | Flip

Painter
Show Grid ☑ Show Walkable □
Tile
Clear | Areas Mark
Place Prefabs

Presets
Name: Ground
Prefabs

Create

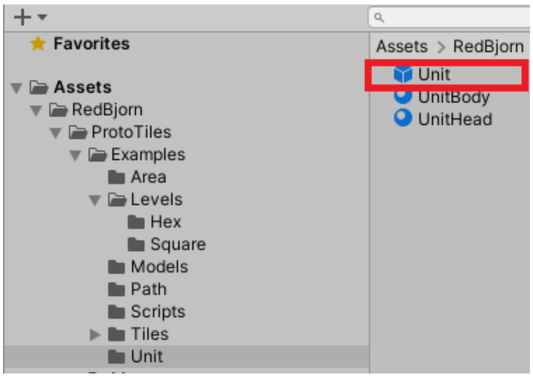
Rules
Rotation: LookAt | Flip

Painter
Show Grid □ Show Walkable □
Tile
Clear | Areas Mark
Place Prefabs

Presets
Name: Ground
Prefabs

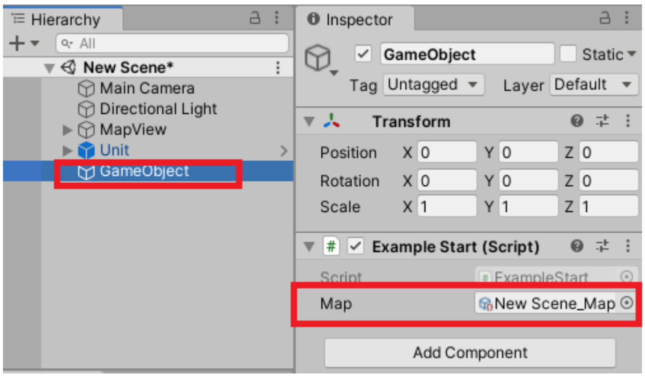
Additionally (For quick test)

Add Unit (Unit prefab from Examples)



Add MapSettings link

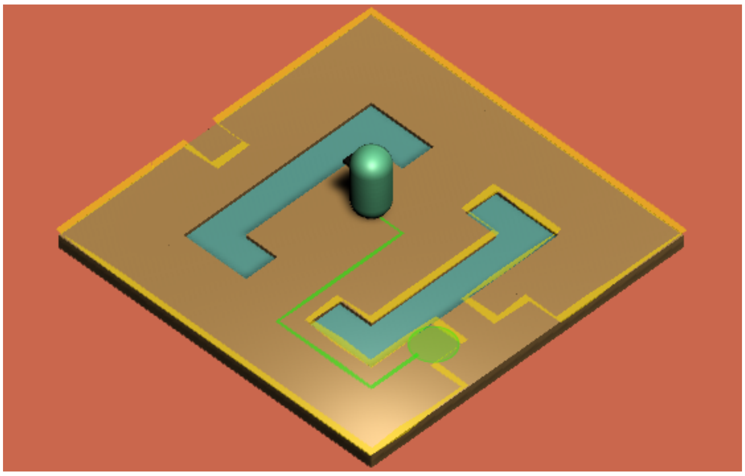
1. Create new GameObject.
2. Add Example Start component.
3. Fill Map field with newly created asset



GameObject
Tag Untagged
Layer Default
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Script ExampleStart
Map New Scene_Map

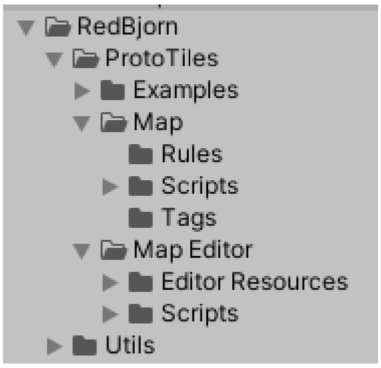
Click playmode.

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

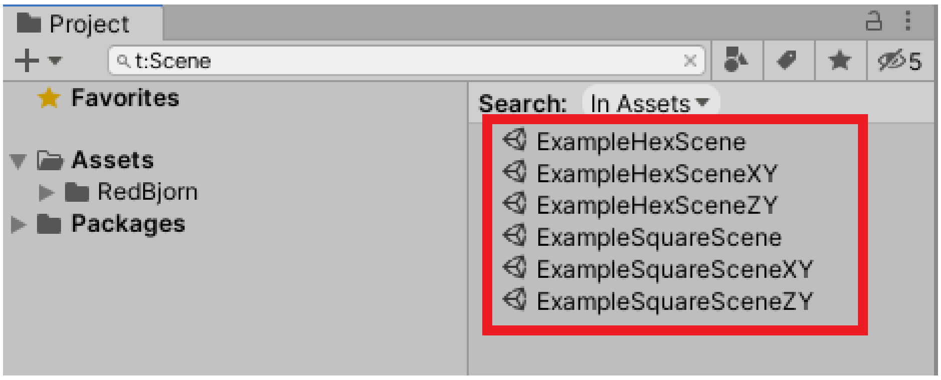


```
▼ RedBjorn
  ▼ ProtoTiles
    ▶ Examples
    ▼ Map
        Rules
      ▶ Scripts
        Tags
    ▼ Map Editor
      ▶ Editor Resources
      ▶ Scripts
  ▶ Utils
```

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

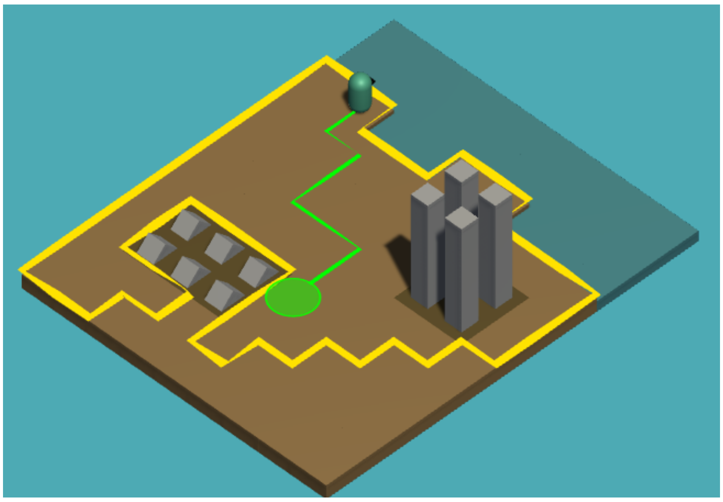
Examples

Package contains six example scenes

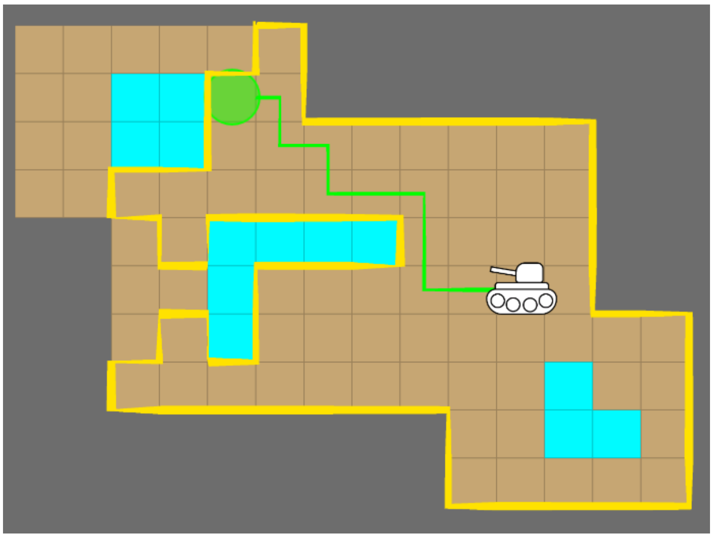


ExampleHexScene
ExampleHexSceneXY
ExampleHexSceneZY
ExampleSquareScene
ExampleSquareSceneXY
ExampleSquareSceneZY

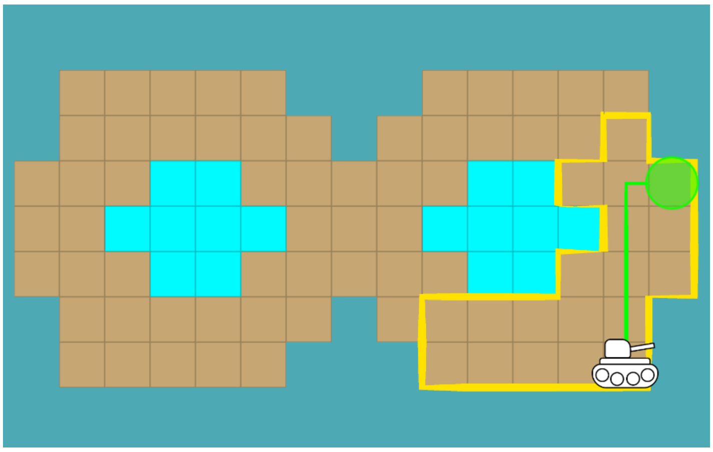
1. ExampleSquareScene



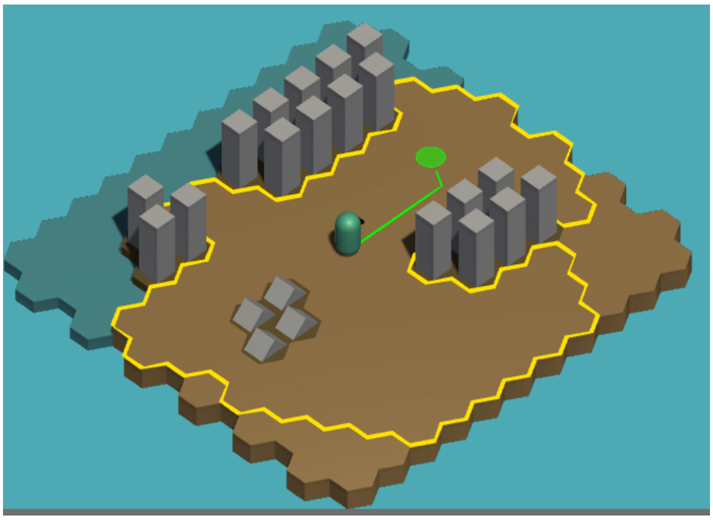
2. ExampleSquareSceneXY



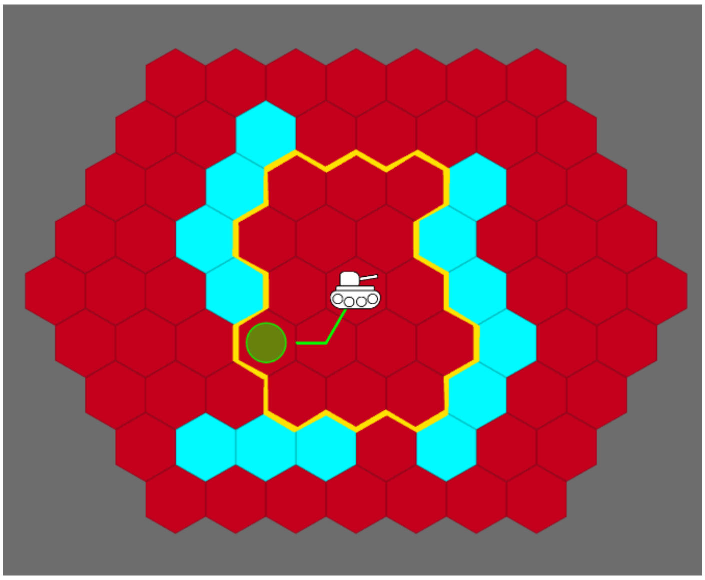
3. ExampleSquareSceneZY



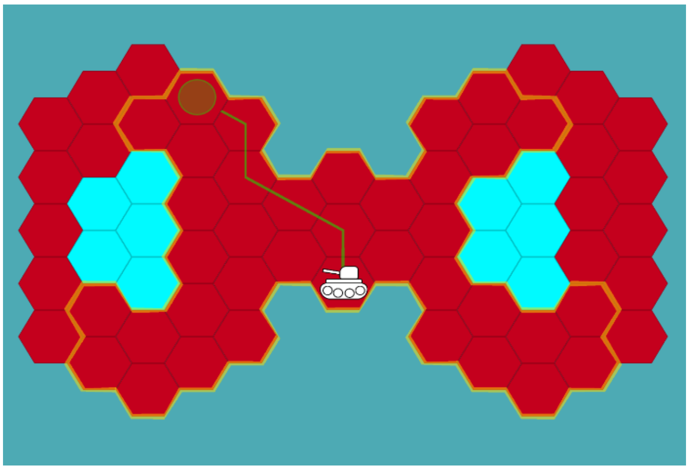
4. ExampleHexScene



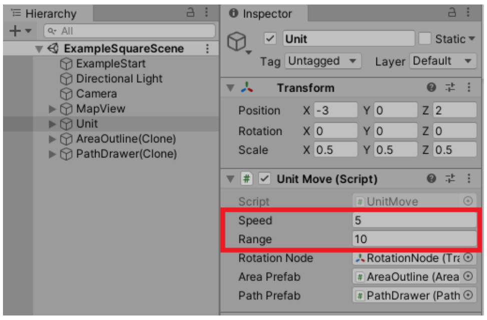
5. ExampleHexSceneXY



6. ExampleHexSceneZY



You can tune Unit’s speed and range



☑ Unit □ Static

Tag Untagged Layer Default

Transform

Position	X	-3	Y	0	Z	2
Rotation	X	0	Y	0	Z	0
Scale	X	0.5	Y	0.5	Z	0.5

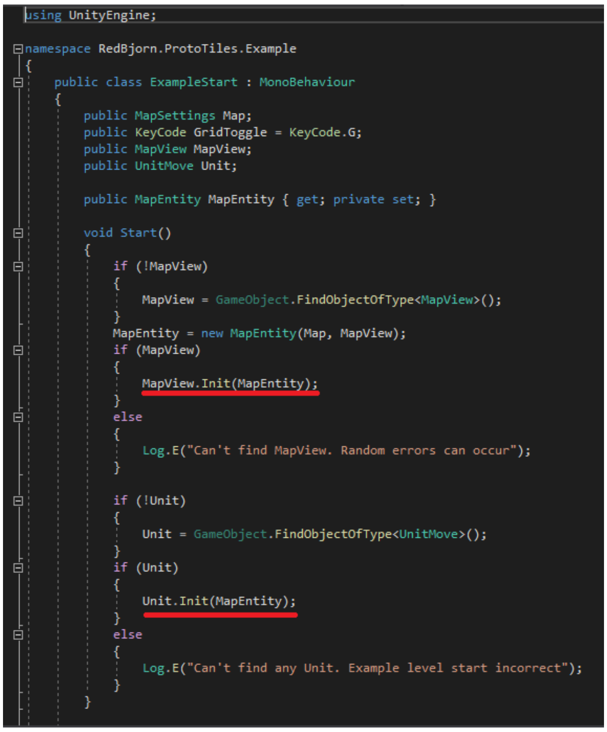
☑ Unit Move (Script)

Script	UnitMove
Speed	5
Range	10
Rotation Node	RotationNode (Tra
Area Prefab	AreaOutline (Area
Path Prefab	PathDrawer (Path

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

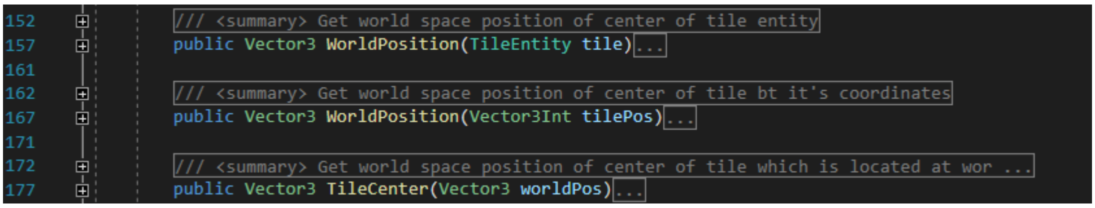
MapEntity for MapSettings

TileEntity for TileData

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

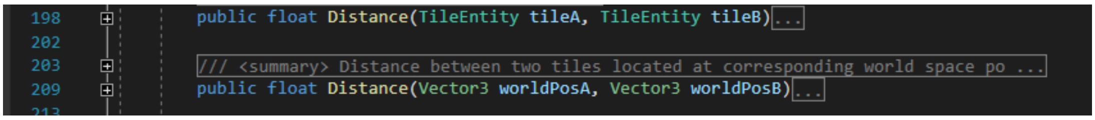
Most useful methods are located at MapEntity class

Converting tile coordinates to world space coordinates and vice versa



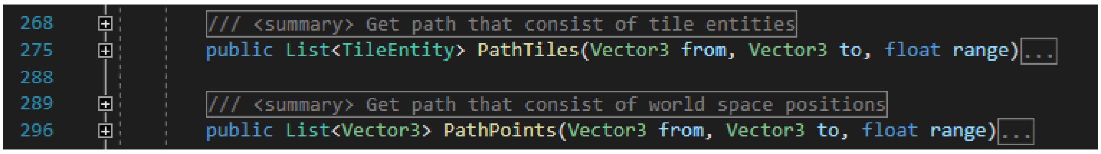
```
/// <summary> Get world space position of center of tile entity
public Vector3 WorldPosition(TileEntity tile)...

/// <summary> Get world space position of center of tile bt it's coordinates
public Vector3 WorldPosition(Vector3Int tilePos)...

/// <summary> Get world space position of center of tile which is located at wor ...
public Vector3 TileCenter(Vector3 worldPos)...
```

Calculating distance in tile space



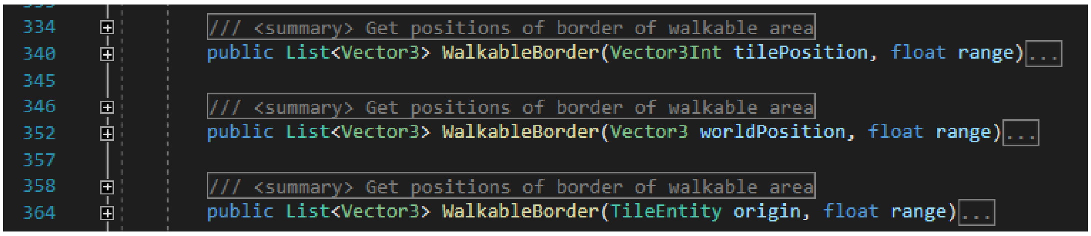
```
public float Distance(TileEntity tileA, TileEntity tileB)...

/// <summary> Distance between two tiles located at corresponding world space po ...
public float Distance(Vector3 worldPosA, Vector3 worldPosB)...
```

Calculating paths



```
/// <summary> Get path that consist of tile entities
public List<TileEntity> PathTiles(Vector3 from, Vector3 to, float range)...

/// <summary> Get path that consist of world space positions
public List<Vector3> PathPoints(Vector3 from, Vector3 to, float range)...
```

Outline of walkable border

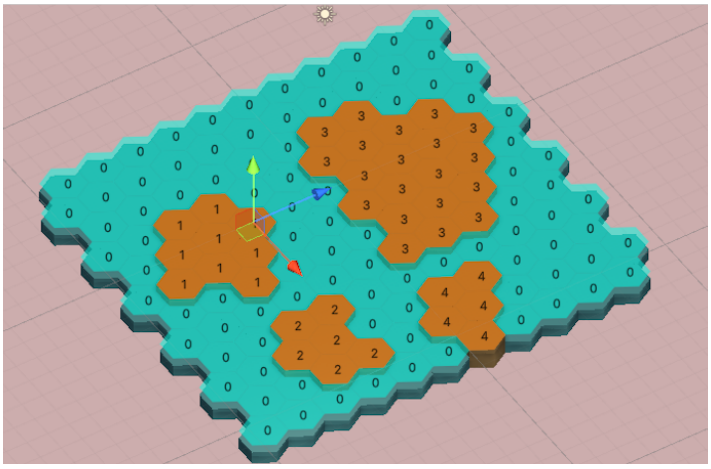


```
/// <summary> Get positions of border of walkable area
public List<Vector3> WalkableBorder(Vector3Int tilePosition, float range)...

/// <summary> Get positions of border of walkable area
public List<Vector3> WalkableBorder(Vector3 worldPosition, float range)...

/// <summary> Get positions of border of walkable area
public List<Vector3> WalkableBorder(TileEntity origin, float range)...
```

Etc..

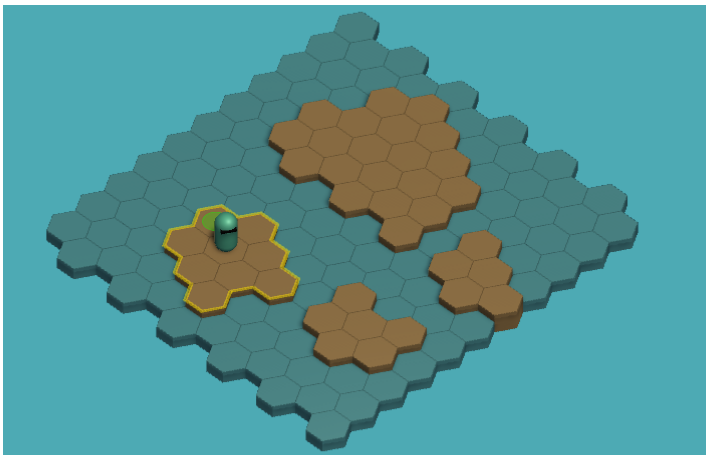
Numbers which are located inside tiles during painting mode at Scene view indicate separate move zone areas

0 index – predefined for NotMovable zone
Indices greater 0 - different movable zones

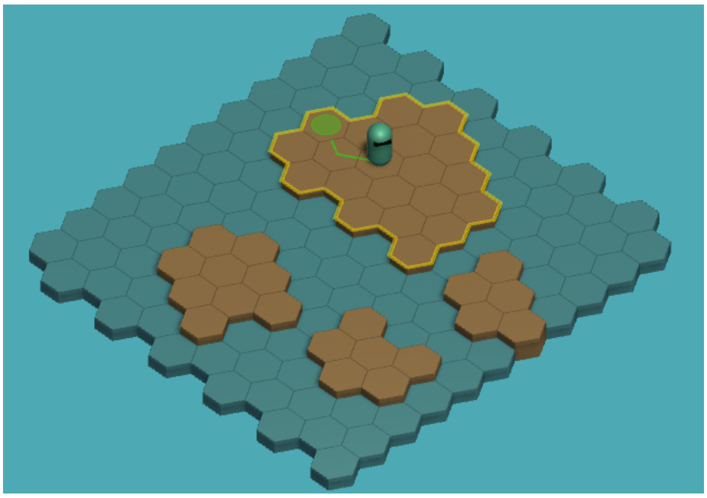


For example.

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



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

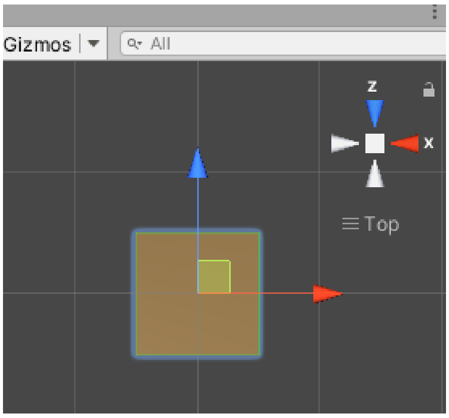


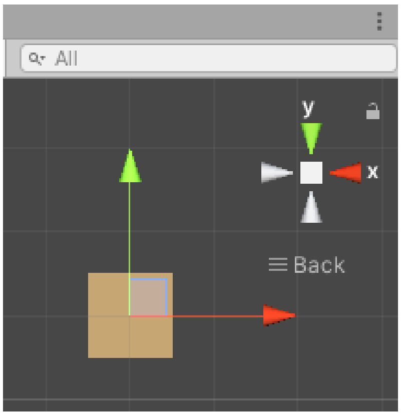
Prefabs

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

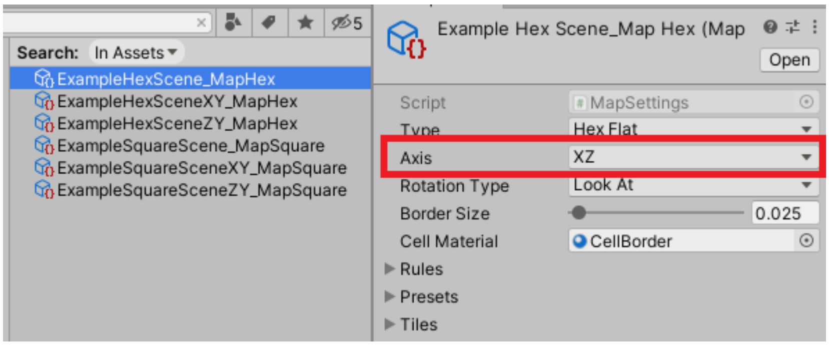
Square Map

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





Tile sides should be only parallel to the corresponding map axis. 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.

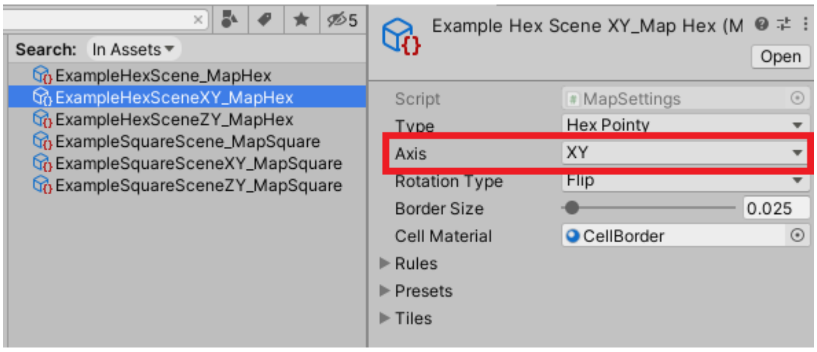


Search: In Assets

- ExampleHexScene_MapHex
- ExampleHexSceneXY_MapHex
- ExampleHexSceneZY_MapHex
- ExampleSquareScene_MapSquare
- ExampleSquareSceneXY_MapSquare
- ExampleSquareSceneZY_MapSquare

Example Hex Scene_Map Hex (Map

Script: MapSettings
Type: Hex Flat
Axis: XZ
Rotation Type: Look At
Border Size: 0.025
Cell Material: CellBorder



Search: In Assets

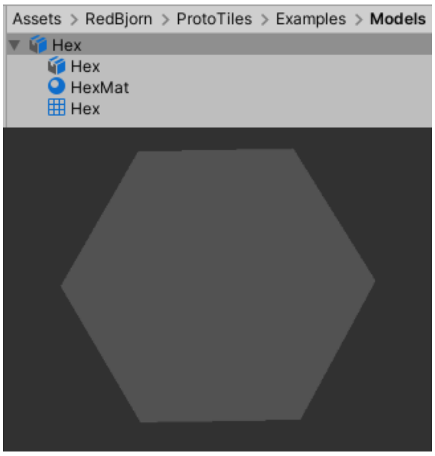
- ExampleHexScene_MapHex
- ExampleHexSceneXY_MapHex
- ExampleHexSceneZY_MapHex
- ExampleSquareScene_MapSquare
- ExampleSquareSceneXY_MapSquare
- ExampleSquareSceneZY_MapSquare

Example Hex Scene XY_Map Hex (M

Script: MapSettings
Type: Hex Pointy
Axis: XY
Rotation Type: Flip
Border Size: 0.025
Cell Material: CellBorder

Hex Map

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

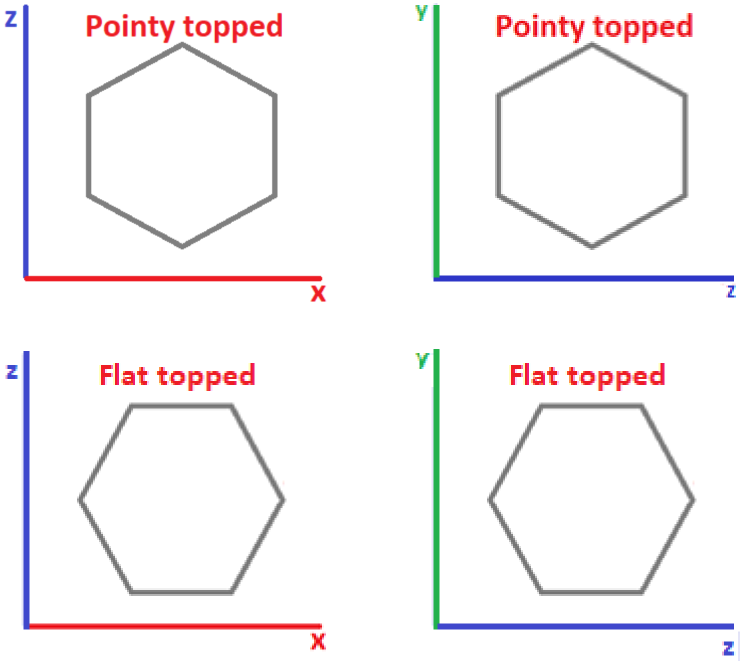


Assets > RedBjorn > ProtoTiles > Examples > **Models**

- Hex
  - Hex
  - HexMat
  - Hex

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

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.



z
Pointy topped
x
y
Pointy topped
z
y
Pointy topped
x
z
Flat topped
x
y
Flat topped
z
y
Flat topped
x