Link to Unity 2D Game Creation Tutorial
https://unity3d.com/learn/tutorials/s/2d-game-creation
Tilemap tutorials are near the bottom of the page

Grid: Component which controls properties of the grid that gets drawn on. Tilemaps are children of a Grid.
Tilemap: GameObject with component tiles get painted onto, similar to a Photoshop layer.
Tilemap Renderer: Component which controls how tiles are rendered including sorting, material and masking.
Tile Palette: An asset which holds a collection of Tiles we can select from.
Tile: An asset which holds a reference to a Sprite, a Color Value, and a Collider type.

Scriptable Tiles: Tiles can be scripted in C# to create custom behavior which executes when the Tilemap is refreshed, for example by drawing more tiles. Generally contain rendering and collision info.
Scriptable Brushes: Brushes can be scripted in C# and can execute whatever code you want when someone paints. Brush code runs when painting and will not update when the Tilemap refreshes. Users must repaint to make changes.

- Unity has examples of brushes and tiles - https://github.com/unity-technologies/2d-extras

There are 2 ways to add tiles to the Tilemap

1) One tile piece at a time (can click and drag)
2) Using a Script to auto change tiles for you

When in Unity:

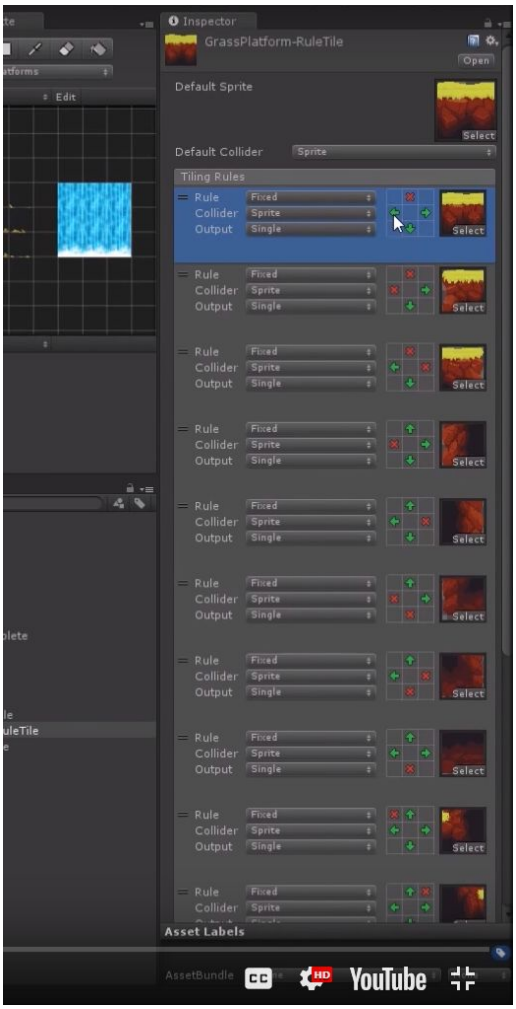
- Add a Tilemap (Create -> 2D Object -> Tilemap)
- Change the tile palette in “Tile Palette” (Click the current tile palette’s name to change tile palette)
    - Creating a new “Tile Palette” (Click current palette’s name -> Create New Palette -> Name it -> Click Create)
    - Choose which Unity folder to save it in (Probs a “Palettes” folder, make one if one doesn’t exist, this will create a new asset, it will be nested under a grid GameObject)
- Add tile sprites
    - Recommend to slice sprites in the sprite editor and make sure sprite resolution and the “Tilemap” grid resolution are the same
    - Click and drag tile sprites to the “Tile Palette” window (Shift-Clicking multi sprites works well here)
    - Folder window opens (move into “Tiles” folder, make folder if not there. Make a new folder for the tiles you are adding and save into that folder)
- Tiles are now ready to use for drawing

can use the [ ] keys to rotate tiles

can select multi tiles to add at once, ie a whole row of tiles from the menu

- Adding colliders
    - Add “Tilemap Collider 2D” component to the Tilemap GameObject (NOT the grid it’s nest under but the tilemap itself)
    - ***this adds colliders to ALL tiles that get placed.*** This can be optimized by adding the “Composite Collider 2D” component (Add component -> Physics 2D -> Composite Collider 2D)
    - ***^^Continued*** Set the Tilemap GameObject’s “Rigidbody 2D” body type to static, then in the “Tilemap Collider 2D” component click the box next to “Used By Composite” ***This makes it so the Collides are only applied to the outside tiles not every tile that is on the screen, ie middle tiles that won’t get touched* By default with these setting you can edit the tiles while in play mode and the tile colliders will update instantly**
    - You can drag the Tilemap’s collider down if other sprites (ie character) is floating (Select your Tilemap -> Go to Tilemap Collider 2D -> Change Offset ‘Y’ value)

- Using Rule Tiles (Scriptable Tiles) *detects what title should be used and changes the tile accordingly* is not always 100% correct
    - The Rule Tile is a Scriptable tile (C#) an example of this can be found here: https://github.com/Unity-Technologies/2d-extras/tree/master/Assets/Tilemap/Tiles/Rule%20Tile) example ->
    - **Can look into this more if we want to use this**



- Random and Animated Tiles (Scriptable)
    - This is great if you have little details you want to overlay on tiles without making a whole new tile sprite for it. Unity will pick from the sprite list at random. They used flowers and extra grass as their example.
    - They used the example of waterfalls for animation
    - **Can look into these more as well if we want to use them.**

Link to other Unity examples for Scriptable Tiles : LINK

- We can add many tiles over/behind each other. This will cut down the amount of sprites that will need to be made. If you adjust the Tilemap layers you can choose what the player can walk behind/in front of, as well as what they can interact with (ie jump on)
- If you move colliders and sprites around it makes it a little harder to erase them. You just have to play around with it. Ie

if you click on something and it doesn’t erase try moving the cursor up or down and see if that works.

I can look into any of this more if there is something we want to try. I included links to where I found the information.