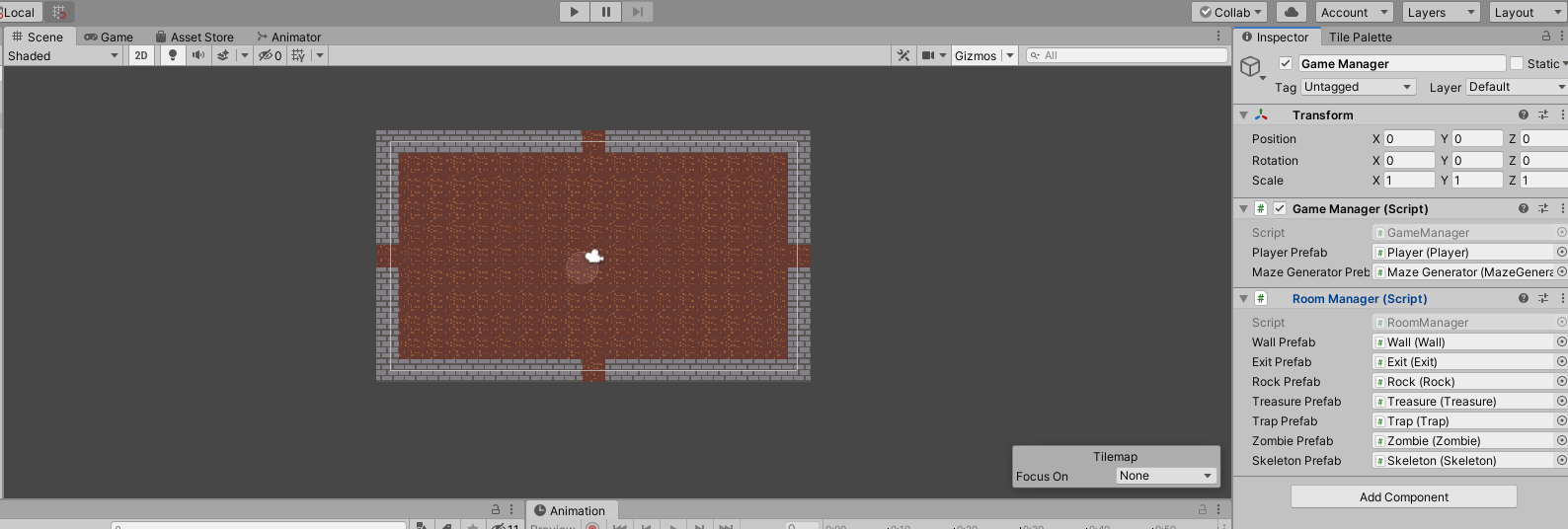
# Tools Used

## Unity

Unity is a well-known game engine that aims to make the development of games easy and fast. I chose to use unity because it handles a lot of work for me including:

* Sprite rendering
* Physics
* Building and compiling
* Asset management
* Memory handling

Unity allows me to build off the core of a game engine and focus more on programming the game logic and algorithms without having to worry about rendering graphics or writing a physics engine.

It also makes visualisation easier, allowing me to see my assets and attach the appropriate components and scripts to them.  


Some limitations I found with unity included the occasional crash with unoptimised code. It also doesn’t integrate well with Git, as I found when adding my project to source control, requiring me to close Unity to add or commit changes. There’s also a lot of noise within the commits, even with a .gitignore file.

## Aseprite

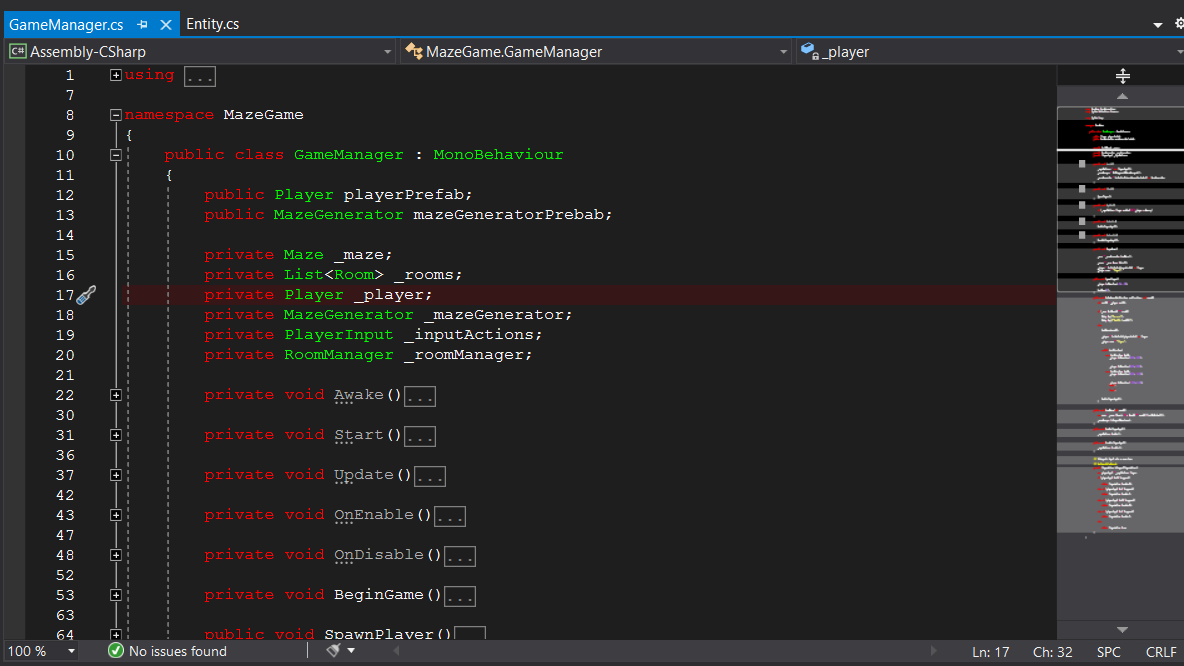
Aseprite is the tool I used to create the sprites in the game:

(image of aseprite)

Aseprite allowed me to create simple graphics for the game and give the retro-look to fit the theme of the project.

## Visual Studio 2019

Visual Studio 2019 was my IDE of choice for writing, debugging and managing my code. Visual studio has a unity extension which when combined with the core package, made developing and debugging easy with features such as code intellisense and breakpoints.



In addition, my language of choice was C# which not only works well with the Unity engine library, but also works extremely well when paired with visual studio’s rich intellisense.

## Other tools used for project

* Git - project source control
* Draw.io - class diagrams
* Visual Studio code - Occasionally used to view code/JSON

# Future Improvements

# User Guide