

# Installation

This Game Engine/FrameWork was written in C#, with no GUI whatsoever to aid in modification or visually representing your work. Hence, It is required to use an IDE of some kind to write your code.

## The IDE

1. This Engine was developed on the [Rider](#) IDE, and so this page will you Rider as its source of reference.
2. However, Visual Studio is also a powerfull IDE for C# and C++, and as long as you can understand the Docs, and do some basic googling, you should also be able to get the Engine Running.
3. Make sure you have [Dotnet](#) 7.0 or above installed.
4. Open your IDE, and made a dedicated Project to start making your game.

## The Dependencies

5. Go to [Release](#) and download the latest release for the DLL required.
6. In the Project, Open up the dedicated [Nugget](#) tab and install *RayLib\_cs*(6.0.0).
  - Incase you cannot use the inbuilt solution, you can try the command line [here](#)
7. In your dedicated IDE, Put the RayGame.dll in your local directory, and setup a reference to it.
  - In Rider, right - click the project, click Add, Add reference and browse to add the dll.
8. And there you go! You can now start Making your game!

# Namespace RayGame

## Classes

### CollisionDetection

Provides a method for collision detection between shapes.

### Engine

The main engine class that initializes and runs the game.

### GameObject

Represents a game object in the Scene.

### Mesh

Represents a 2D mesh consisting of a collection of vertices.

### MeshRenderer

A Renderer that renders a `Mesh` associated with a `GameObject` .

### SpriteRenderer

### Transform

The class that holds all the transformation data for an object

## Interfaces

### IGameComponent

Interface for game components.

### IRenderer

Interface for renderers.