User Manual

Authors:

Jibran Jarwar 21353811
Ivan Shvydchenko 21317061
Supervisor: Prof. Renaat Verbruggen
Date: 01/05/2025

Game Engine Instructions:

Welcome to our User Manual below we will detail the features of our game engine.

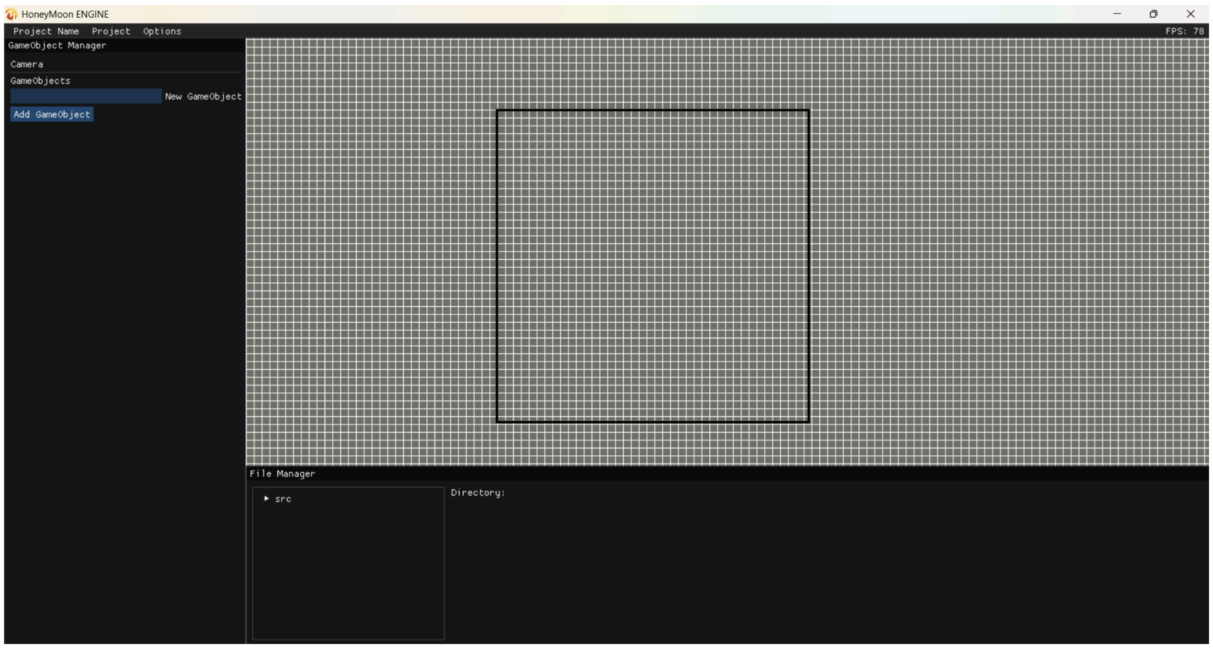
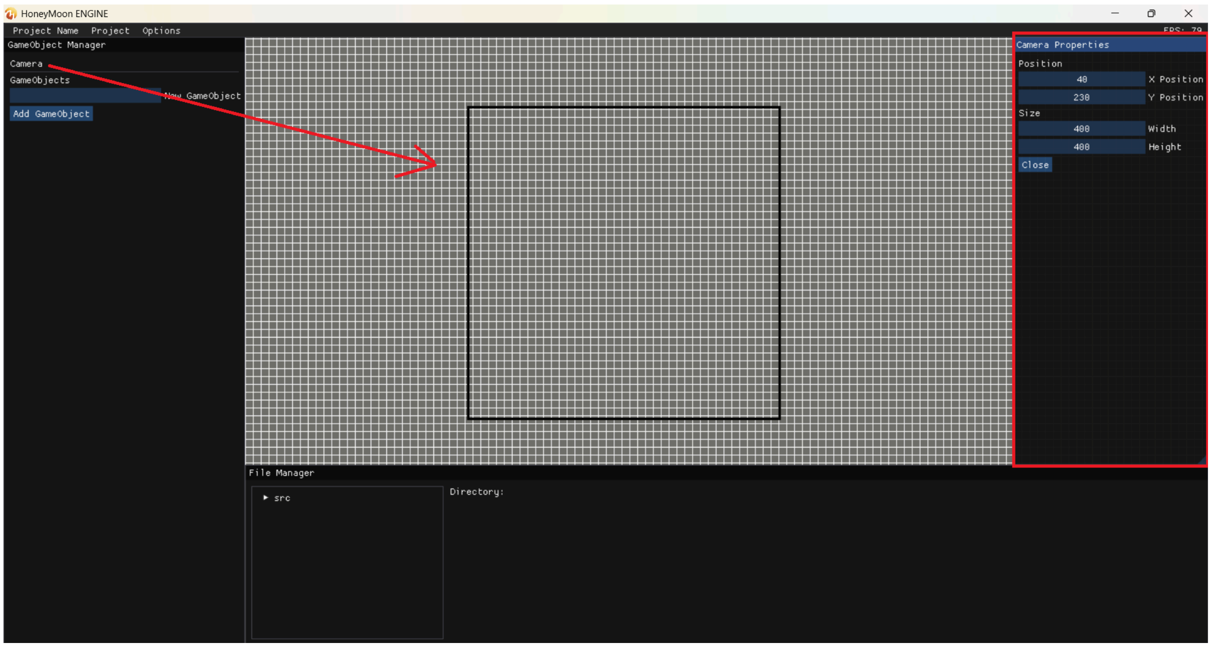


Figure 1

Upon opening the application this will be your home screen. On the left hand side you will see the GameObject Manager handling the creation, deletion and adjustments of game objects as well as the Camera which shows everything that is in its frame. Below we have the File Manager allowing you to browse your local files. On the top left we have feature options for your current project.

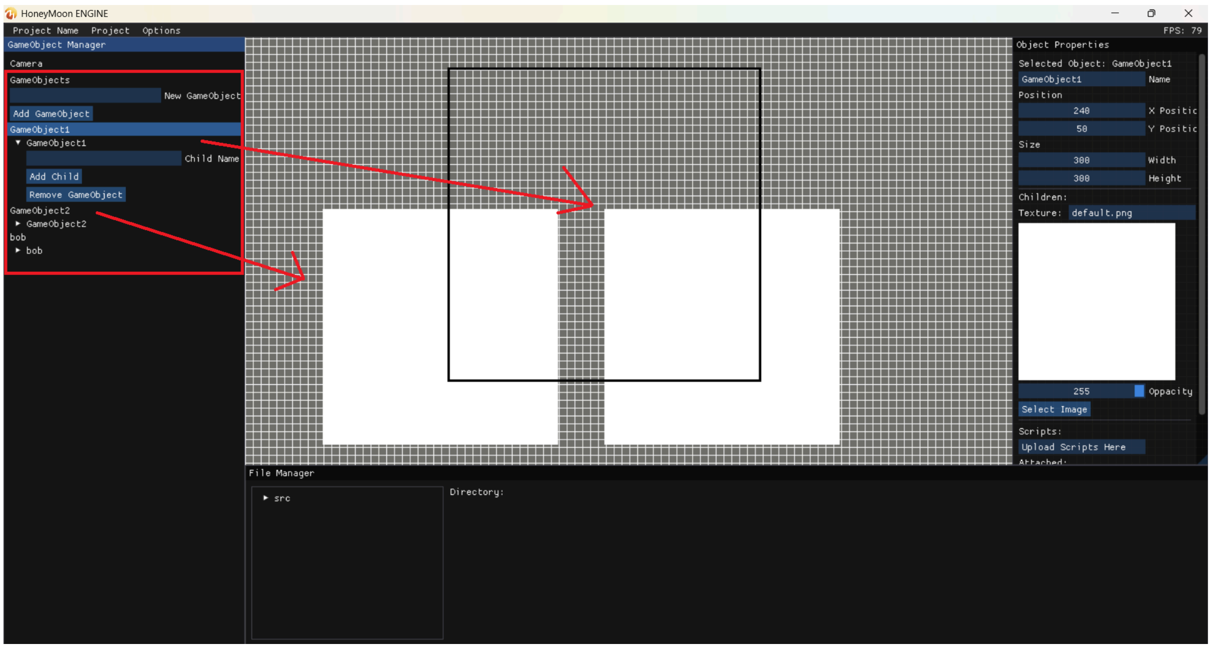


Position
40 X Position
230 Y Position
Size
400 Width
400 Height

Figure 2

The red arrow points from the camera name which functions as a button, it points to the black outline which is the camera itself, the red box on the right opens when the camera name is clicked and opens the camera's properties. With this you can adjust the position and size of the camera.

To adjust these numbers left click and hold the blue box holding the number, then slide left or right to go increase/decrease respectively, You can also double click and type the value you need.



Selected Object: GameObject1
GameObject1 Name
Position
240 X Positic
50 Y Positic
Size
300 Width
300 Height
Children:
Texture: default.png
255 Oppacity
Scripts:
Attached:

Figure 3

The red highlighted box shows our game objects. You can create a game object by either clicking "Add GameObject" which will give a default name of "GameObject1", if that exists then it will default to "GameObject2" and so on. You can also input custom names such as "bob" in the blue box right above "Add GameObject".

Below each created game object we can see the name again this time with a small ">" to the left of it. When clicked this will open a dropdown menu and cause the arrow to point downwards. This dropdown allows you to add a child object to that game object or remove that game object.

The red arrows point to game object 1 and 2 respectively. The game object "bob" also exists but is layered with "GameObject1". This happens as game objects, when created, are created at the same position so layering is done currently in the order gameObjects are created.

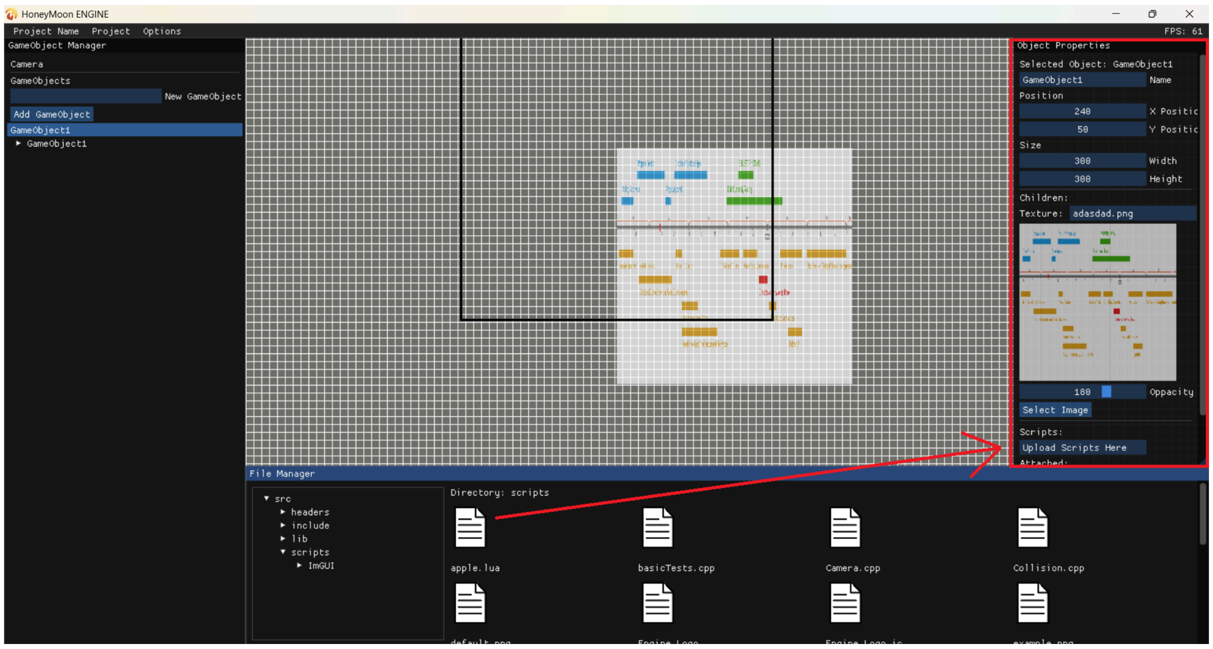


Figure 4

Clicking on "GameObject1" highlights it in blue as can be seen on the left and opens up its properties menu on the right hand side. Within this properties menu we can rename our game object by clicking its name in the blue box beside "Name". We can adjust its position and size. If this game object has children they will be listed under "Children:".
We can also add an image to our game object by clicking select image or drag and dropping from file Manager to the Texture input box, we can then adjust the images oppacity and have the ability to upload a script.

File Manager has also been opened, we navigate my local src folder to reach scripts where i can drag and drop apple.lua into the scripts section to upload it.

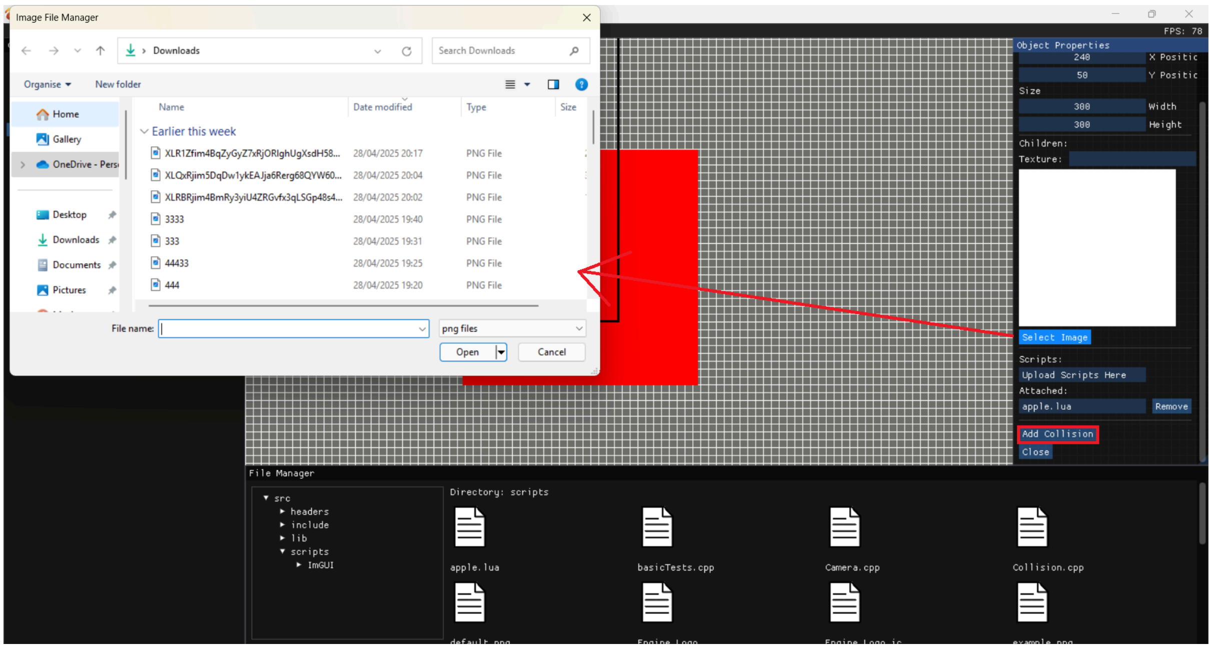


Image File Manager

Downloads

Name	Date modified	Type	Size
XLR1Zfim4BqZyGyZ7xRjORIghUgXsdH58...	28/04/2025 20:17	PNG File	
XLQxRjim5DqDw1ykEAJja6Rerg68QYW60...	28/04/2025 20:04	PNG File	
XLRBRjim4BmRy3yiU4ZRGvfx3qLSGp48s4...	28/04/2025 20:02	PNG File	
3333	28/04/2025 19:40	PNG File	
333	28/04/2025 19:31	PNG File	
44433	28/04/2025 19:25	PNG File	
444	28/04/2025 19:20	PNG File	

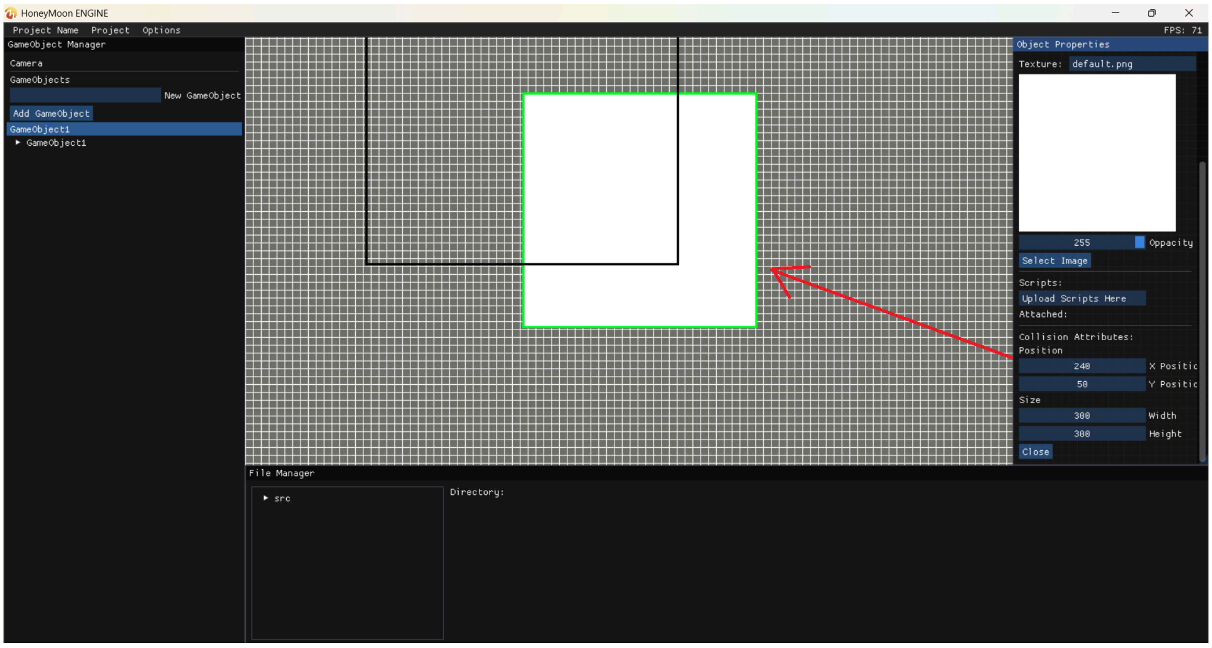
File name: png files

Object Properties
240 X Position
50 Y Position
Size
300 Width
300 Height
Children:
Texture:
Attached:
apple.lua

Directory: scripts

Figure 5

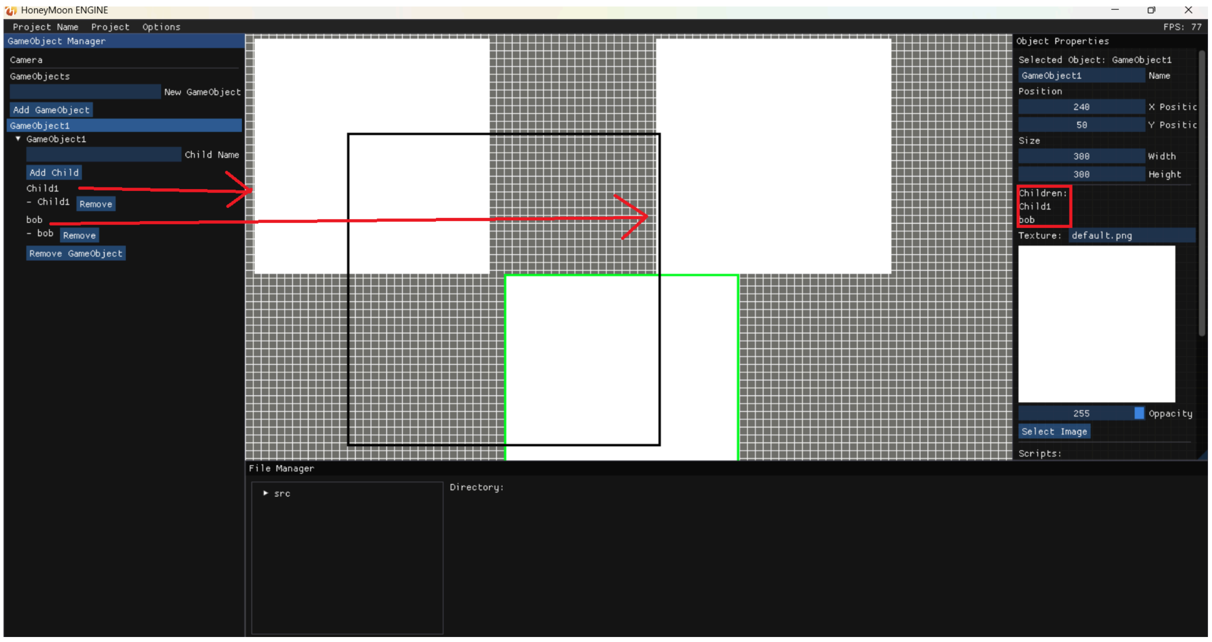
When we click “Select Image” it will open our File Manager so that we can select an image. In the red highlighted box. We also have “Add Collision” this allows us to add collision to our game object allowing them to interact with other game objects. (From the previous Figure we have drag and dropped the apple.lua script)


HoneyMoon ENGINE

Object Properties
Texture: default.png
255 Oppacity
Collision Attributes:
Position
240 X Position
50 Y Position
Size
300 Width
300 Height

Figure 6

By clicking “Add Collision” we get our green collision box as seen by the red arrow. We can adjust the size and position of this box for our “GameObject1”



Selected Object: GameObject1
GameObject1 Name
Position
240 X Position
50 Y Position
Size
300 Width
300 Height
Children:
Child1
bob
Texture: default.png
255 Oppacity

Figure 7

Following game object creation we can also create child objects. They also have default names when “Add Child” is clicked with the blue box beside “Child Name” being empty. Names can also be inputted such as “bob”. Below each child object is its name and the remove button.

Currently on the right hand side we are still in “GameObjects1” properties menu as seen by how it is highlighted on the left hand side, in the red box within this properties menu we can see the children that we have added to “GameObject1”.

If we were to adjust the position of “GameObject1” its children would follow, “GameObject1” would not follow if the children changed their position.

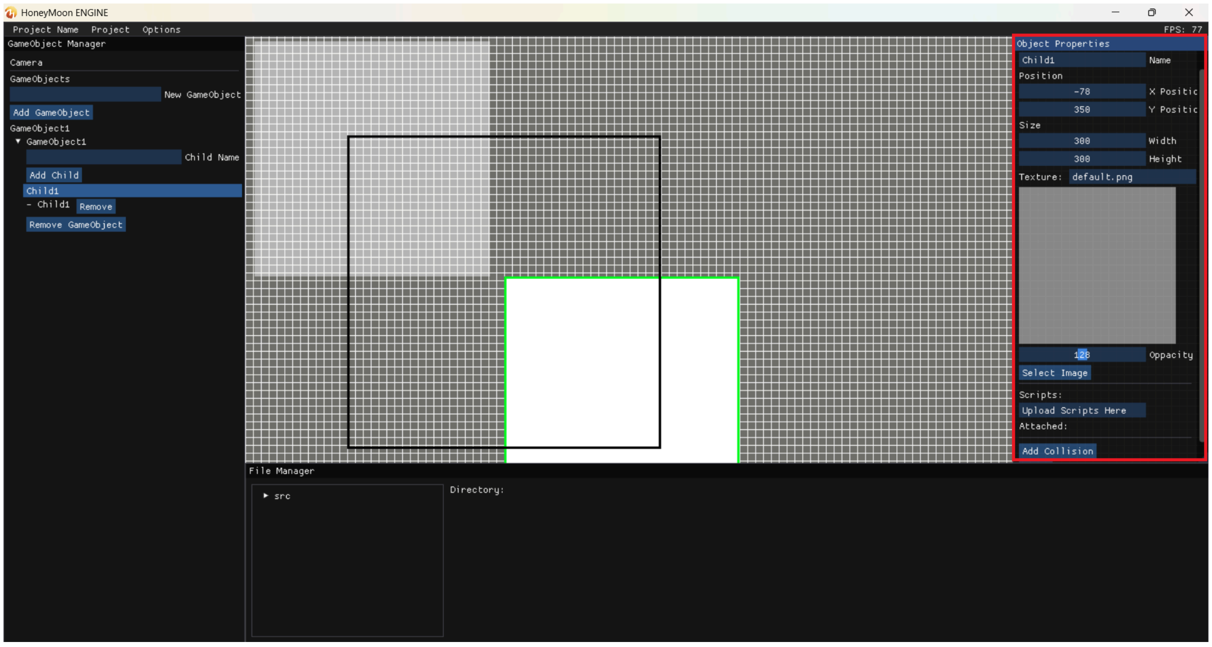


Figure 8

The child object also has its own properties menu with the same properties as a normal game object. Here we can see that moving “Child1” does not affect its parent “GameObject1”.
“Bob” was removed.

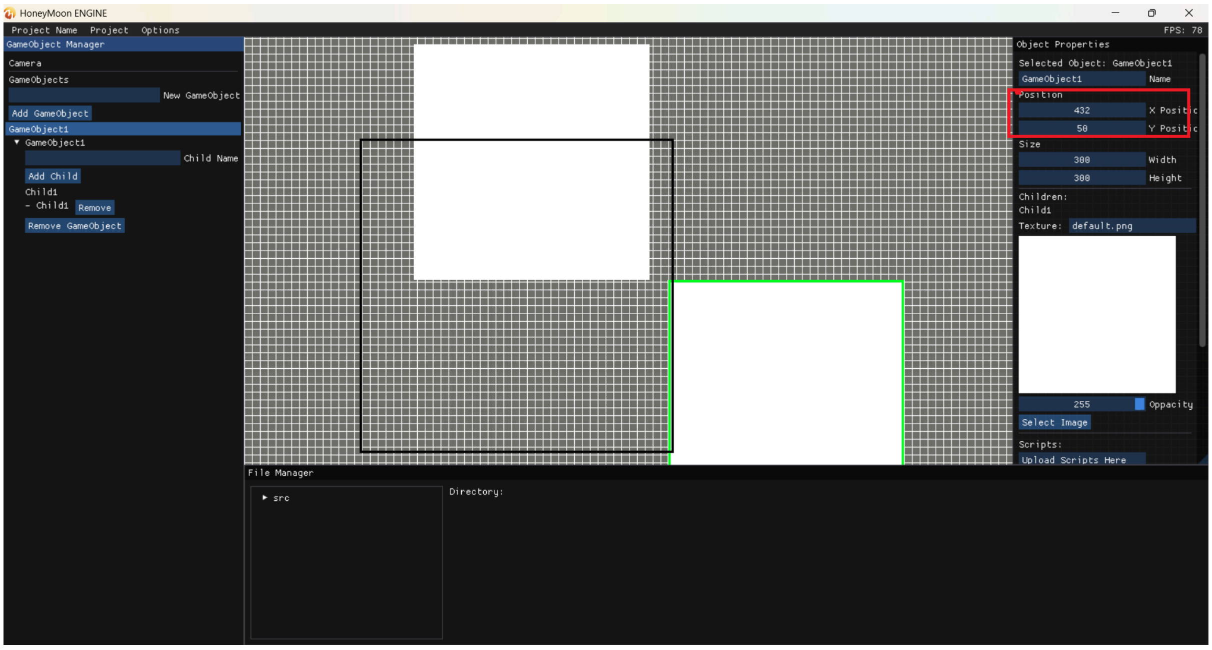


Figure 9

Here we can see adjusting the parents position in the red box causes the child to move with it.

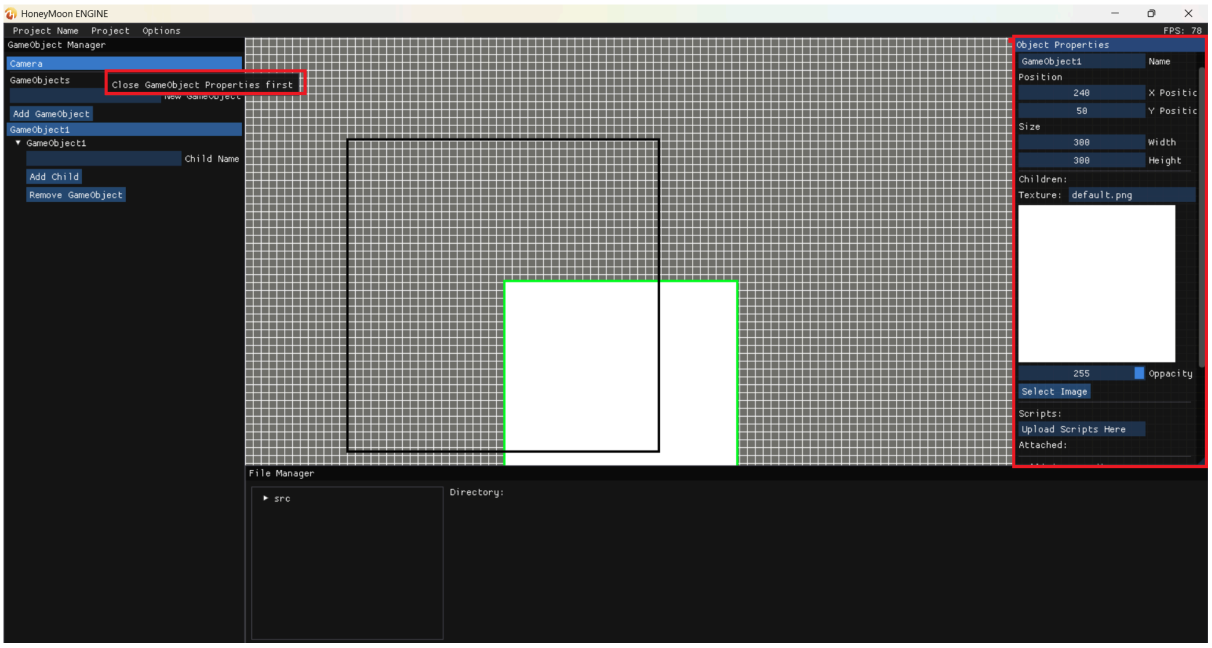


Figure 10

We can not adjust camera properties while we are currently in a game objects properties menu as seen in the red highlighted boxes.

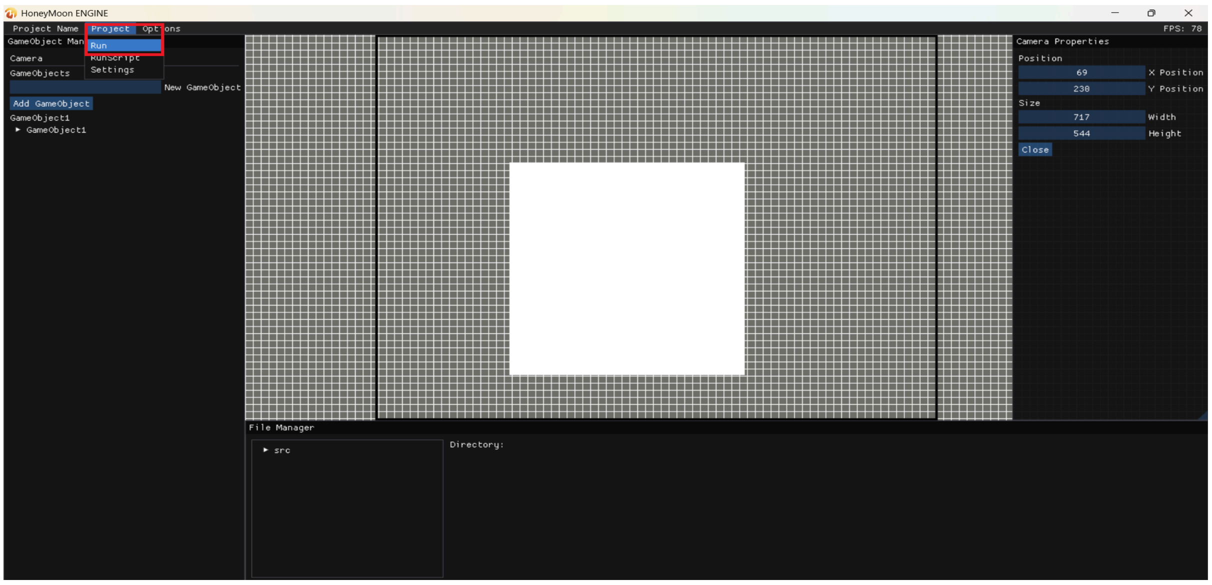


Figure 11

In order to run the Game Engine we can click the “Project” button as shown in the highlighted red box, this will run a preview menu showing everything within the camera.

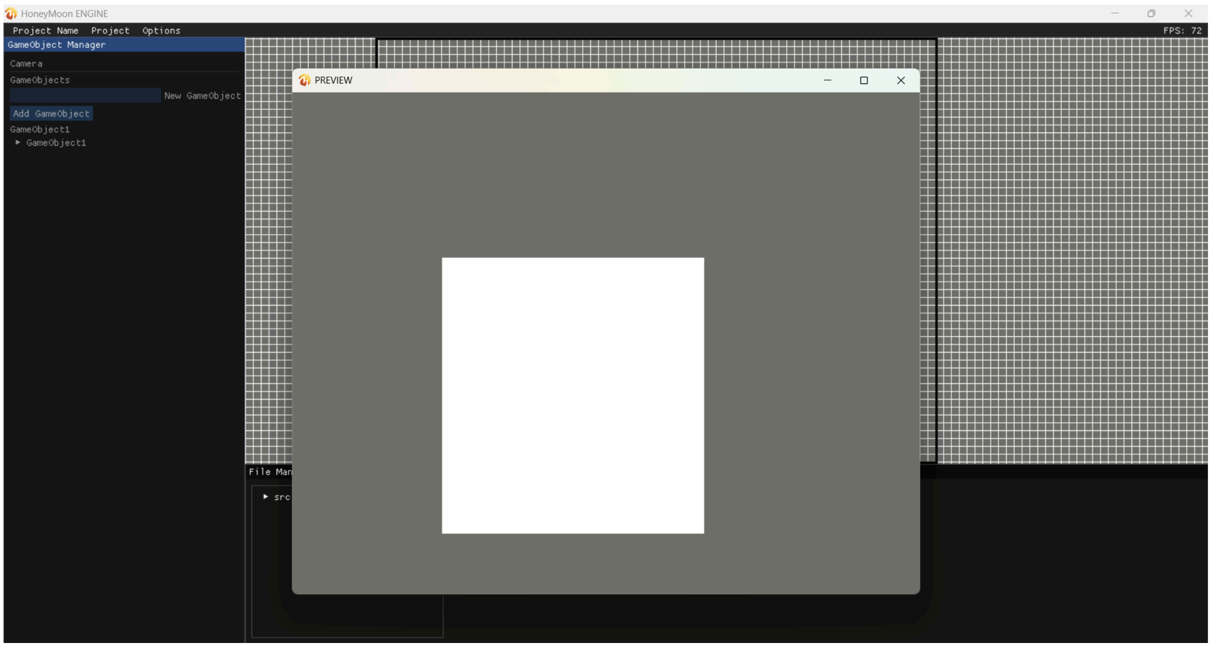


Figure 12

When we hit run this Preview window will open and display the game functionality.

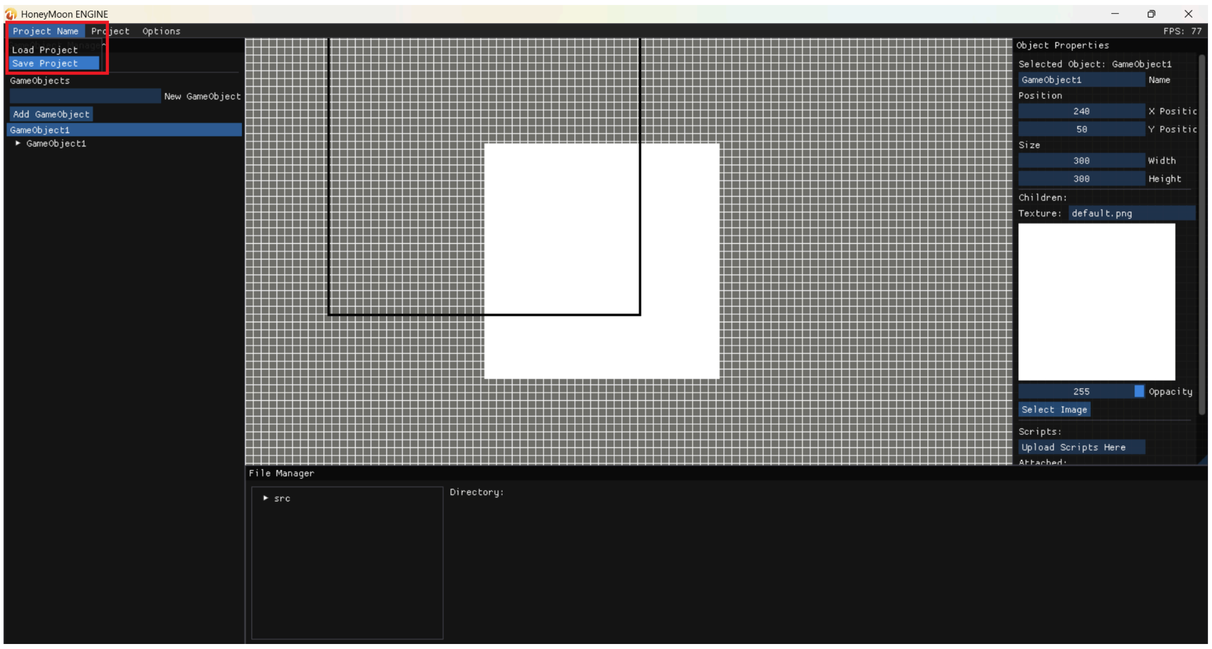
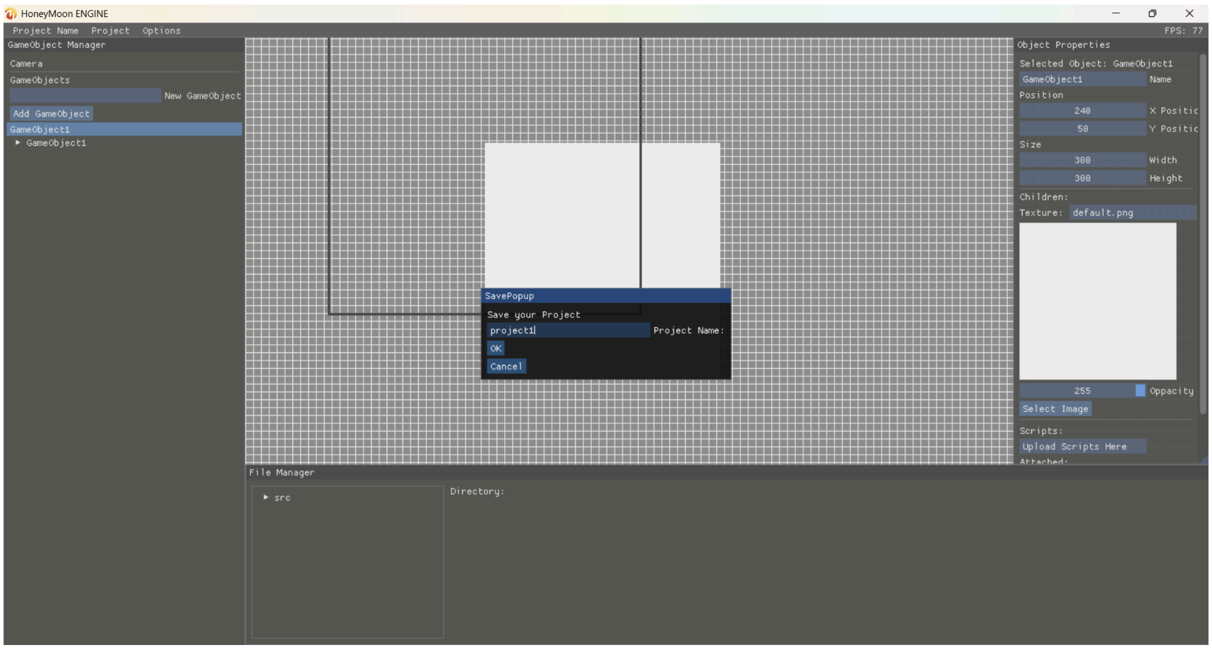


Figure 13

We can also save a project by navigating to the top left and clicking “Project Name” which will show two options, Load and Save project.



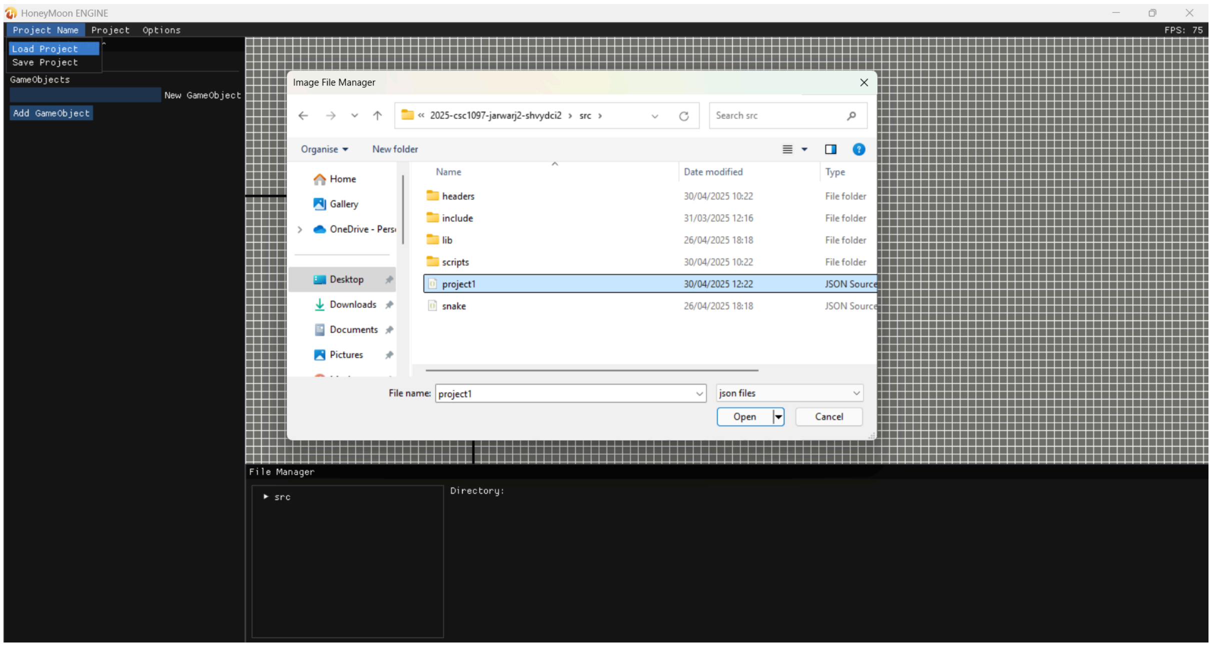
Project Name: project1

Selected Object: GameObject1

GameObject1	Name
Position	
240	X Position
50	Y Position
Size	
300	Width
300	Height
Children:	
Texture:	default.png
255	Oppacity

Figure 14

Clicking “Save Project” will show this popup allowing you to input a name for your project.



Name	Date modified	Type
headers	30/04/2025 10:22	File folder
include	31/03/2025 12:16	File folder
lib	26/04/2025 18:18	File folder
scripts	30/04/2025 10:22	File folder
project1	30/04/2025 12:22	JSON Source
snake	26/04/2025 18:18	JSON Source

File name: project1 json files

Figure 15

Clicking Load Project and navigating to your src folder will let you select and load a project here we can see “project1”.

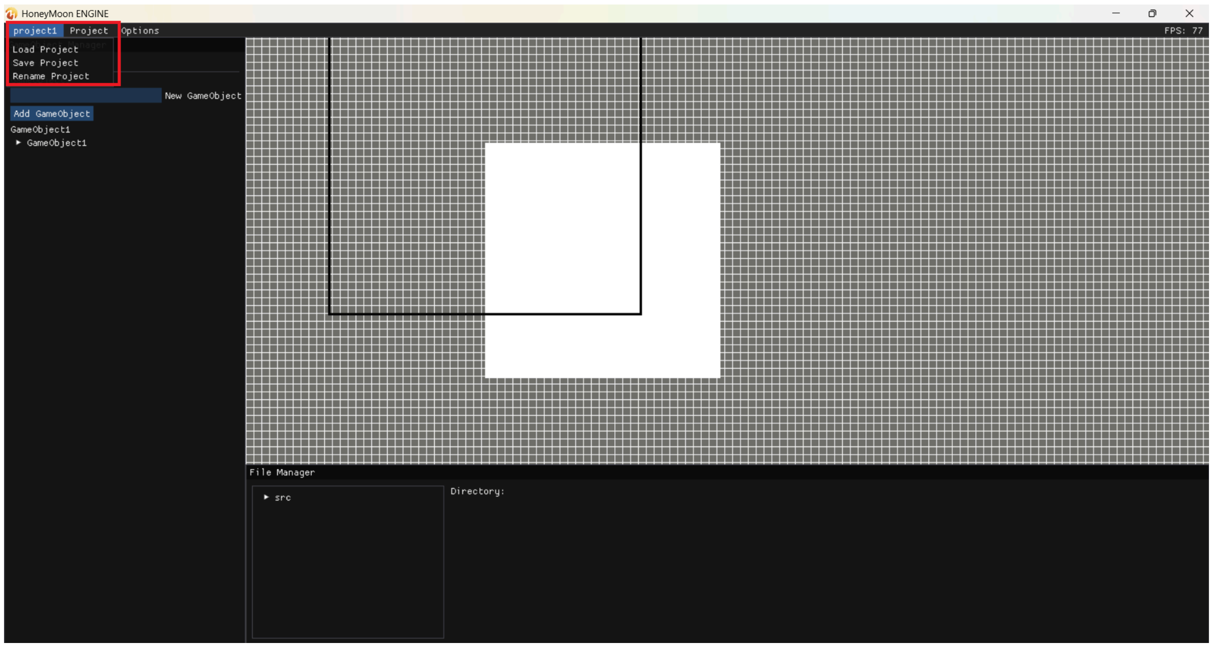


Figure 16

In the highlighted box you can see the loaded project, you are now also given the option to rename this project if needed.

Native Scripting Instructions:

GameObjects:

how you access gameObject in Lua: gameObject[“<name>”]

Properties:

- **x:** The X coordinates of the gameObject (also affects collisionBox position).
- **y:** The Y coordinates of the gameObject (also affects collisionBox position).
- **width:** The width of the gameObject.
- **height:** The height of the gameObject.
- **name:** The name of the gameObject.
- **oppacity:** The transparency of a gameObject.
- **children**: Another table with gameObject children, used like so gameObject[“<name>”].children[“<child_name>”].

Functions:

- **Movement:** Function which provides pre-writen simple movement where you canMove the GameObject with the AWSD keys on the keyboard.

- **AddCollision:** Adds Collision to a gameObject during runtime of the game.

- **OnCollision:** Returns true or false whether the gameObject has collided with any other GameObjects.

- **OnCollisionReturn:** Returns the gameObject you have collided with (use in Conjunction with the “OnCollision” Function).

- **Collide:** Applies push back to gameObject when colliding with another gameObject S so it doesn’t phase or enter through the other gameObject.

- **GetID:** Gets the ID of the gameObject.

- **Copy:** Copies the gameObject that the function is called on and returns the new copy GameObject within the lua script.

Camera:

how you access camera in Lua: Camera

Properties:

- **x:** The X coordinates of the Camera.
- **y:** The Y coordinates of the Camera.
- **width:** The width of the Camera.
- **height:** The height of the Camera.

Functions:

- **FollowObject(GameObject obj):** Camera Follows the centre of the obj passed within the Function.

General:

KeyCode: A_key, D_key, W_key, S_key, Q_key, E_key, F_key, C_key, LEFT_key, RIGHT_key, UP_key, DOWN_key, SPACE_key

Functions:

- **IsKeyPressed(KeyCode):** Checks if the key that was passed was pressed for one frame.

- **IsKeyHeld(KeyCode):** Checks if the key that was passed is held down for multiple frames.

- **DeleteGameObjects(Table objs):** Deleted a table of gameObjects passed as a table to the function (used best for deleting Copy objects during runtime of game).

- **Delay(float value):** returns True or False if a certain amount of time has passed which is given by the value given into the function.