GAM200

R00M13BA-OS OWNERS MANUAL

2024-2025

GREY GOOSE

CORE CONCEPT

The main aim of Room13A-OS is to create a **top-down, high octane, single player action** game which aims to play with the player's perspective of a helpless, incapable roomba.

While not straying too far from tried and tested gameplay features, RoombaOS's key selling features are as follows, and will be elaborated more later in the document;

- **Limb Attachment System**
    - Players will be able to attach limbs onto their Roomba, ranging from offensive and utility purposes to allow the player to experience broader gameplay.
- **Sortie System**
    - Players will be able to customize their Roomba at the start of each level, allowing for higher replayability as well as a range of different playstyles to fit each player's needs.
- **One Hit Kill**
    - Players and enemies will only require a single point of damage, leading to higher stakes as well as creating a high octane gameplay environment.
- **Scalability**
    - The sortie and limb system allows players to choose their preferred playstyle.
    - With weapons being easily implementable into the game, more interesting limbs can be created to create a personalized experience.

TEAM ROSTER

RAYNER
PRODUCT MANAGER
PHYSICS/INPUT
CHAMPION

JAZ WINN
TECH LEAD
ENGINE CHAMPION

SEAN
PROGRAMMER
GRAPHICS CHAMPION

JUN JIE
PROGRAMMER
LEVEL EDITOR
CHAMPION

TEDMUND
DESIGN LEAD
LEVEL DESIGN
/MECHANICS
CHAMPION

JOEL
ART LEAD
STORY CHAMIPION

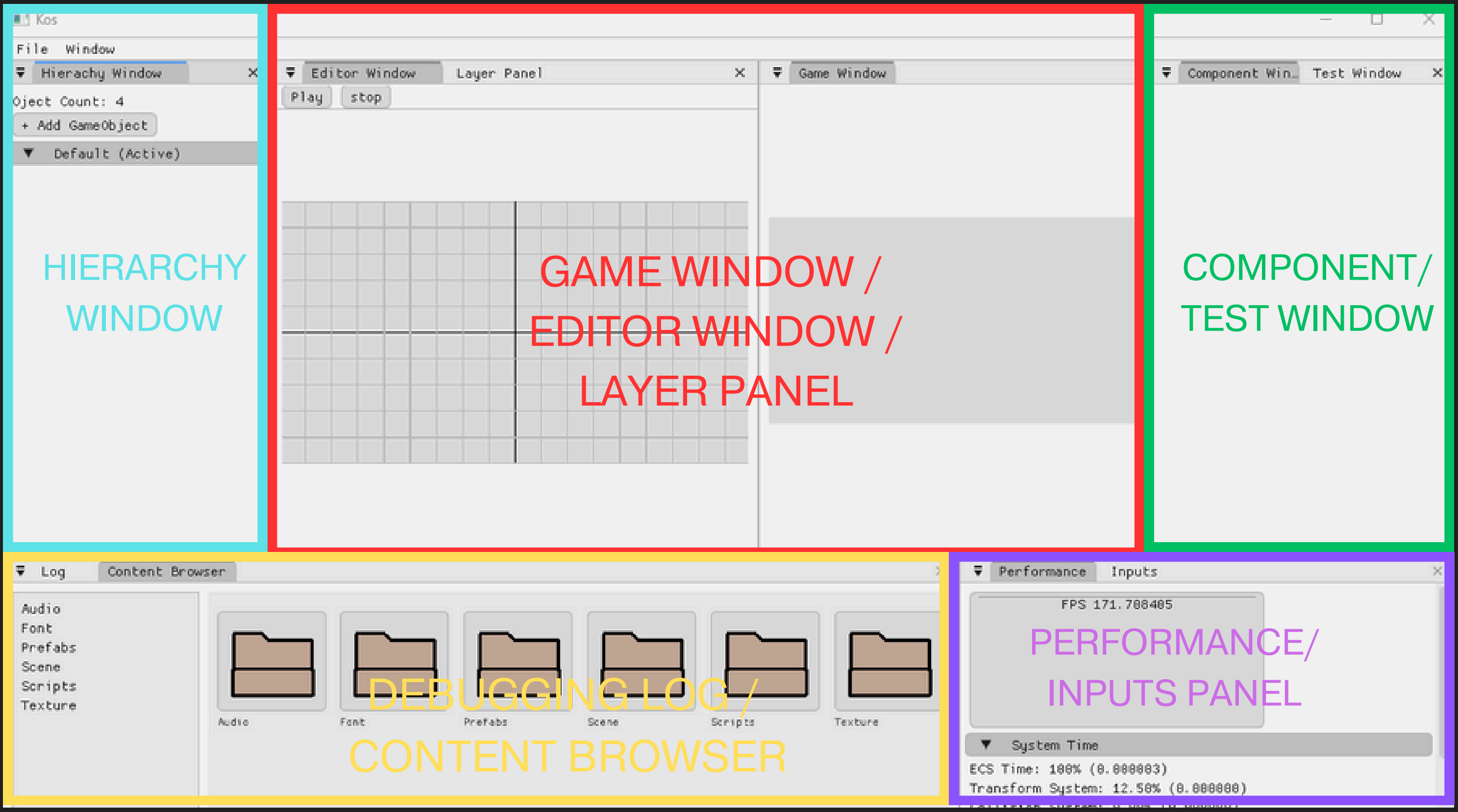
CLARENCE
AUDIO LEAD
GAMEPLAY
CHAMPION

ELIJAH
PROGRAMMER
PHYSICS/INPUT
CHAMION

GREY GOOSE

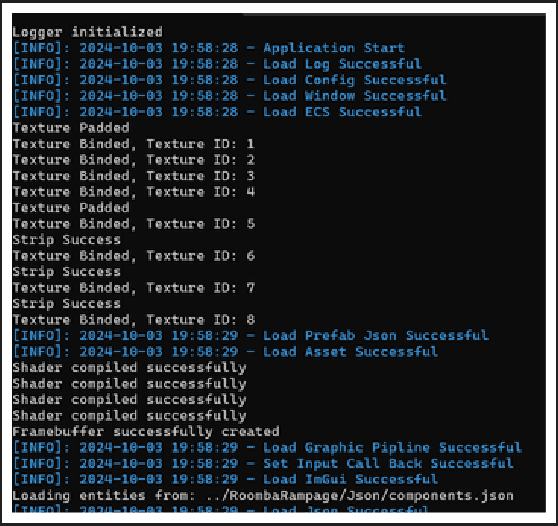
ENGINE/IMGUI COMPONENTS

ENGINE WINDOW



```
Oject Count: 4
FPS 171.708405
ECS Time: 100% (0.000003)
Transform System: 12.50% (0.000000)
```

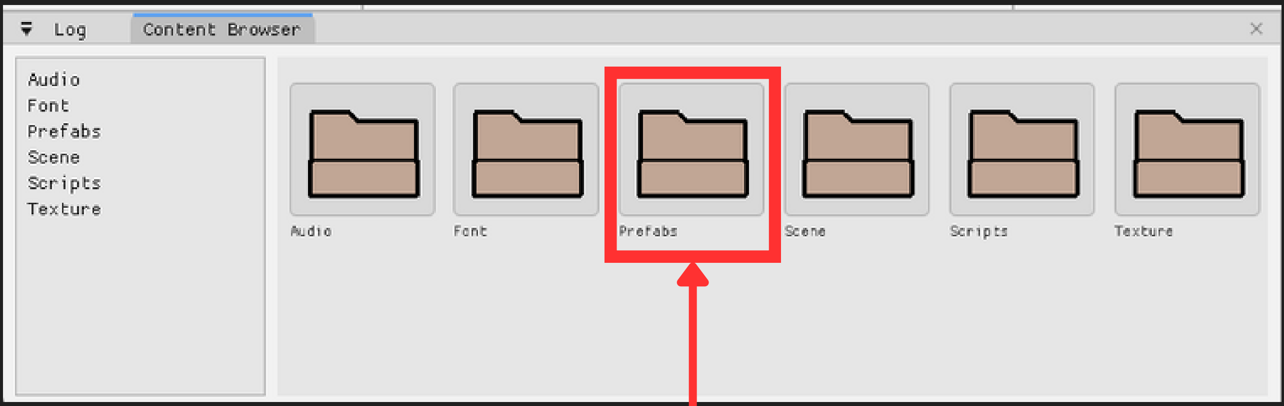
CONSOLE LOG



```
Logger initialized
[INFO]: 2024-10-03 19:58:28 - Application Start
[INFO]: 2024-10-03 19:58:28 - Load Log Successful
[INFO]: 2024-10-03 19:58:28 - Load Config Successful
[INFO]: 2024-10-03 19:58:28 - Load Window Successful
[INFO]: 2024-10-03 19:58:28 - Load ECS Successful
Texture Padded
Texture Binded, Texture ID: 1
Texture Binded, Texture ID: 2
Texture Binded, Texture ID: 3
Texture Binded, Texture ID: 4
Texture Padded
Texture Binded, Texture ID: 5
Strip Success
Texture Binded, Texture ID: 6
Strip Success
Texture Binded, Texture ID: 7
Strip Success
Texture Binded, Texture ID: 8
[INFO]: 2024-10-03 19:58:29 - Load Prefab Json Successful
[INFO]: 2024-10-03 19:58:29 - Load Asset Successful
Shader compiled successfully
Shader compiled successfully
Shader compiled successfully
Shader compiled successfully
Framebuffer successfully created
[INFO]: 2024-10-03 19:58:29 - Load Graphic Pipline Successful
[INFO]: 2024-10-03 19:58:29 - Set Input Call Back Successful
[INFO]: 2024-10-03 19:58:29 - Load ImGui Successful
Loading entities from: ../RoombaRampage/Json/components.json
[INFO]: 2024-10-03 19:58:29 - Load Json Successful
```

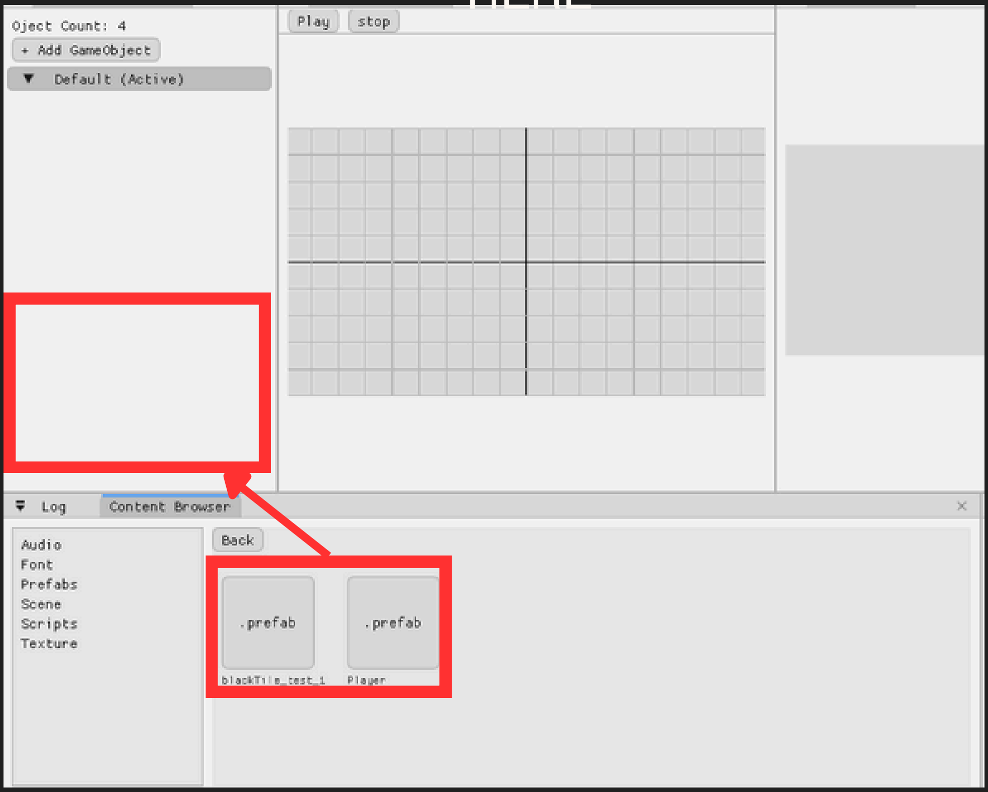
GREY GOOSE

1. USING PREFABS

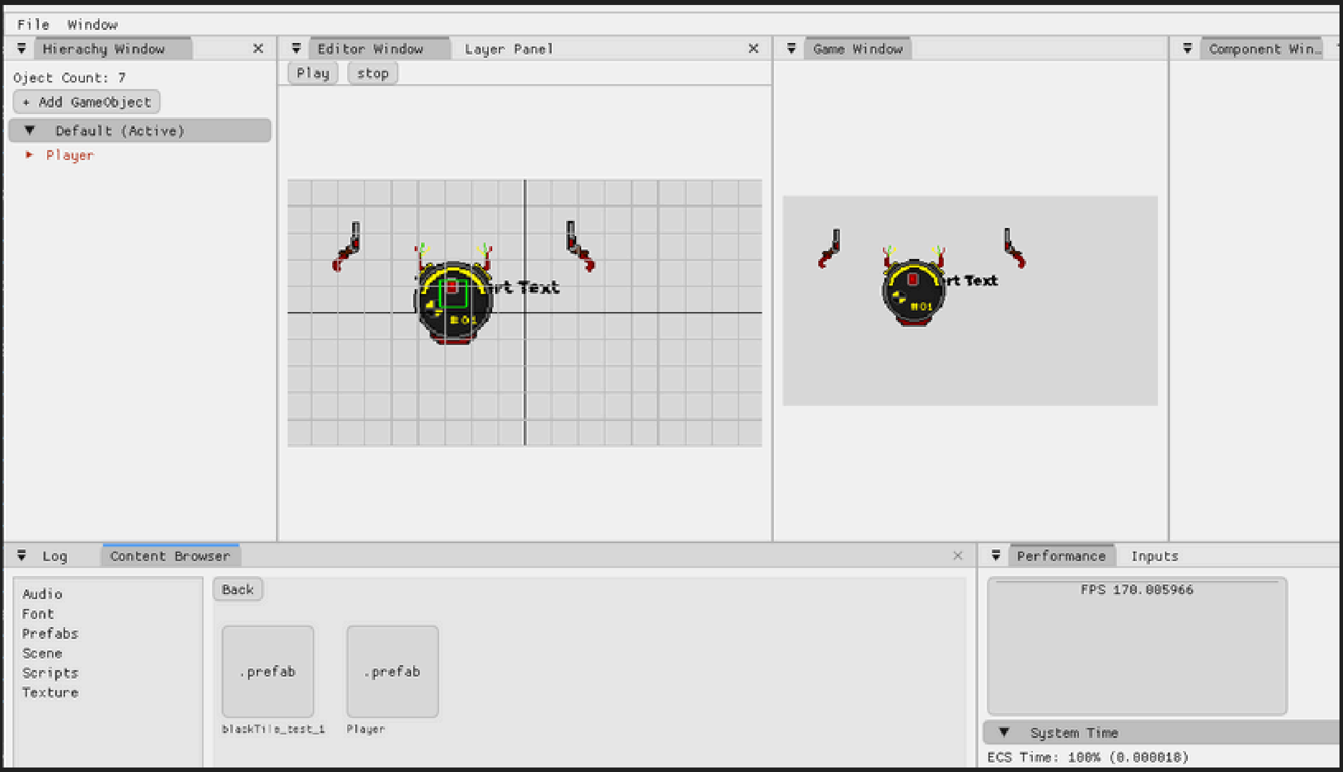


Click on ‘Prefabs’ folder in Content Browser

DRAG EITHER PREFAB AND DROP INSIDE HERE



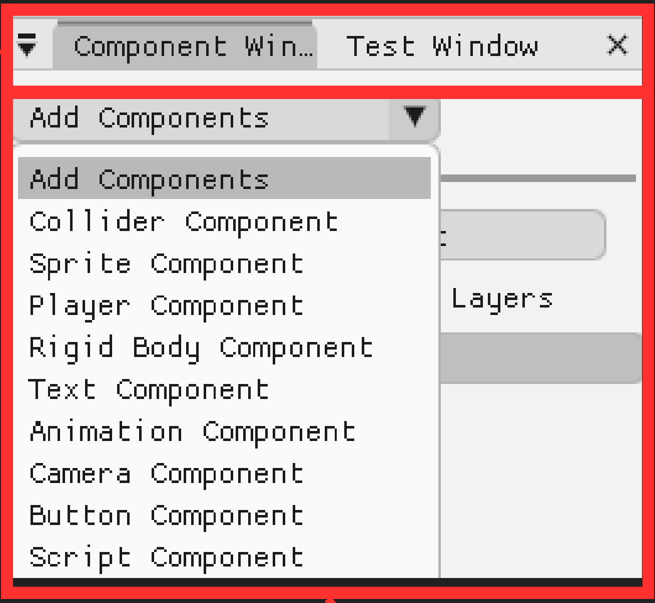
ENTITY WILL BE SHOWN AS BELOW



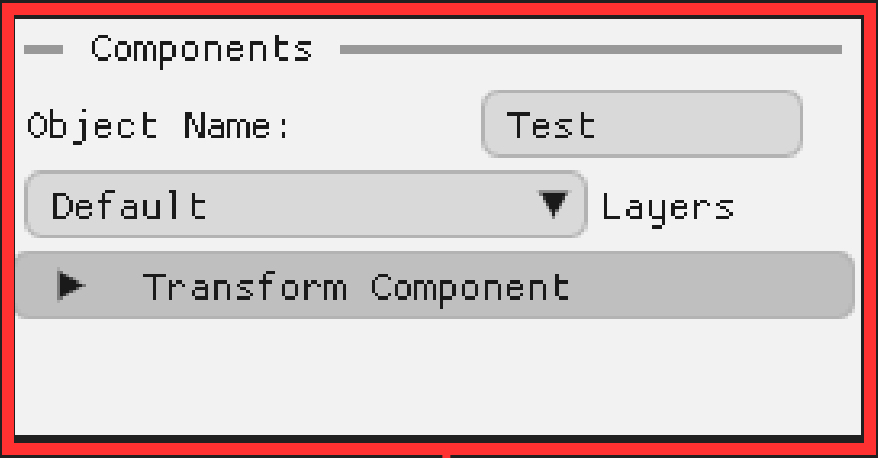
```
FPS 170.005966
ECS Time: 100% (0.000018)
```

2. ADDING COMPONENTS

Click to Open Component List

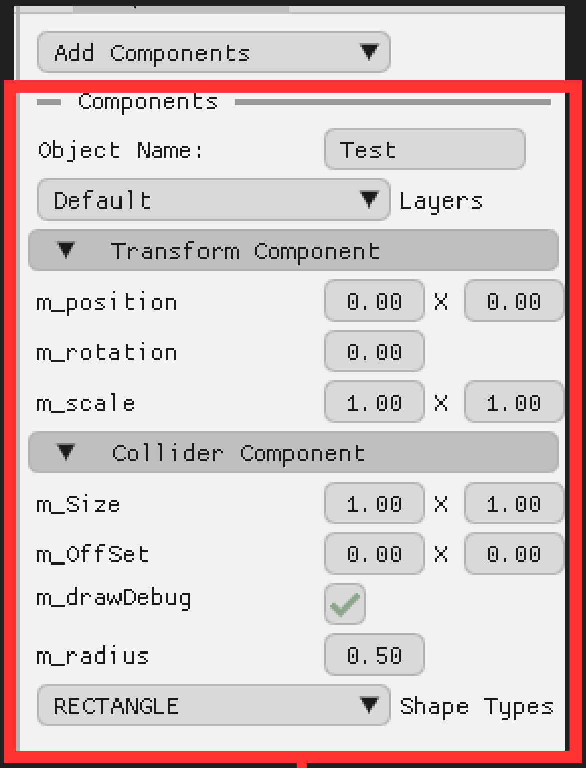


Click to select Component



Check that Component(s) has been added to selected Object

2. ADDING COMPONENTS (CONT'D)



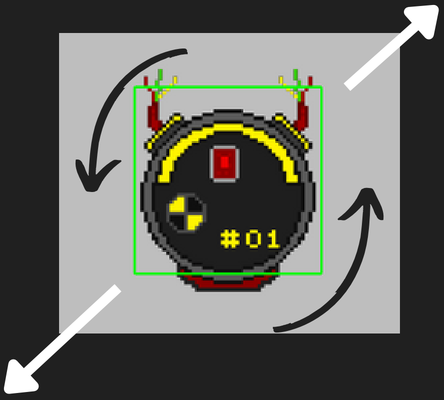
Add Components ▼

Components

Object Name:	Test		
Default ▼	Layers		
▼ Transform Component			
m_position	0.00	X	0.00
m_rotation	0.00		
m_scale	1.00	X	1.00
▼ Collider Component			
m_Size	1.00	X	1.00
m_OffSet	0.00	X	0.00
m_drawDebug	✓		
m_radius	0.50		
RECTANGLE ▼	Shape Types		

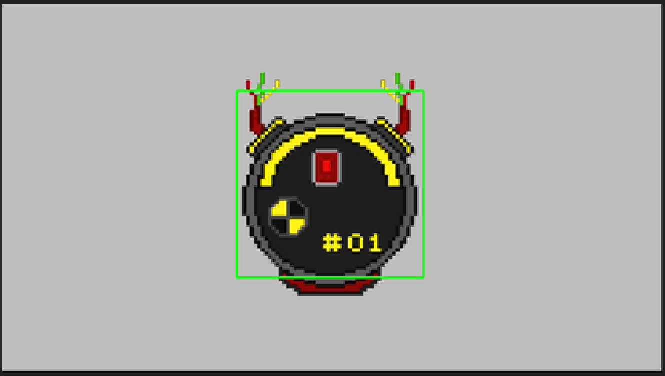
Modify Component Variables

Play around with the various components!
Have fun and experiment!

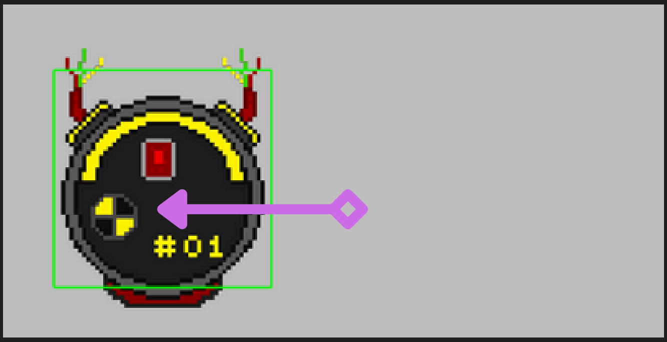


3. PLAYER MOVEMENT

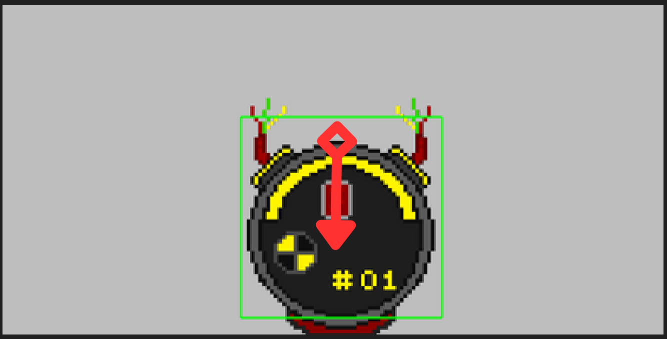
Press 'W, A, S, D' to Move

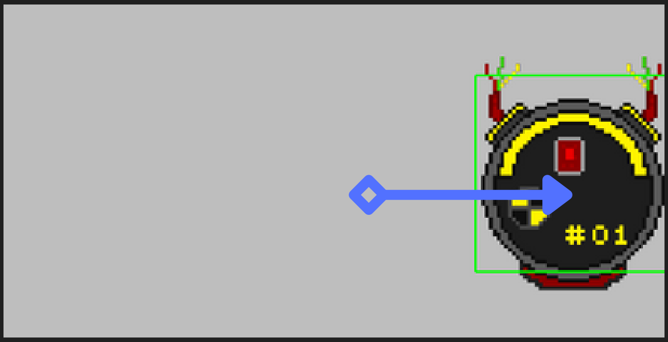


Eg. 'A' key is pressed.

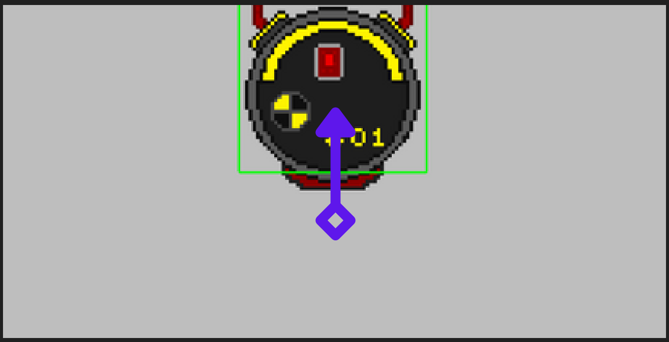


Eg. 'S' key is pressed.



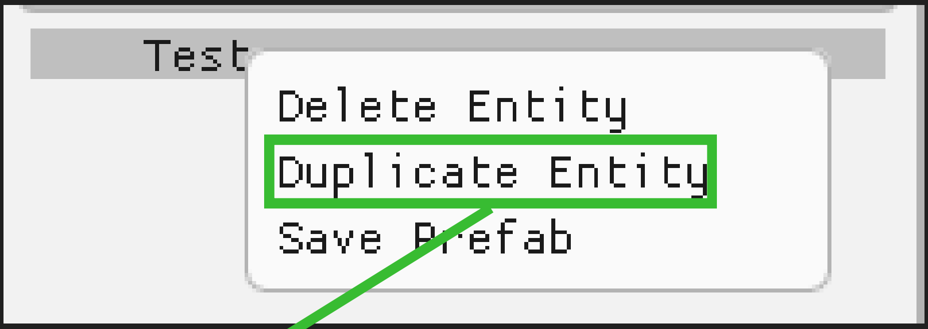


Eg. 'D' key is pressed.



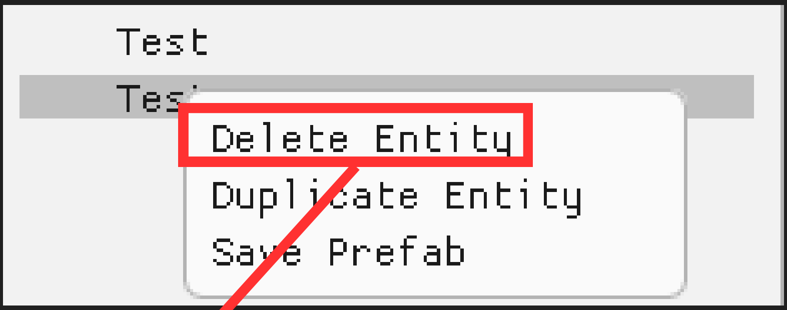
Eg. 'W' key is pressed.

4. DUPLICATING/DELETING OBJECTS

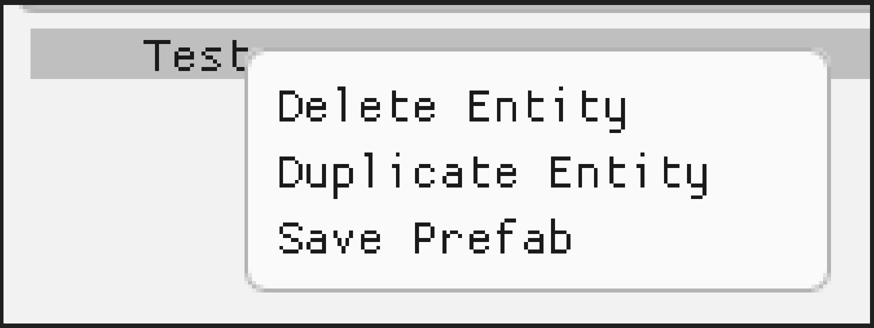


RIGHT CLICK ON GAME OBJECT

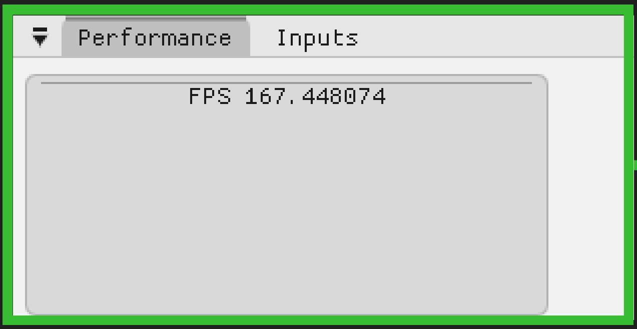
Click ‘Duplicate’ to Duplicate an object incl. all its components/properties



Click ‘Delete’ to Delete an object incl. all its components/properties

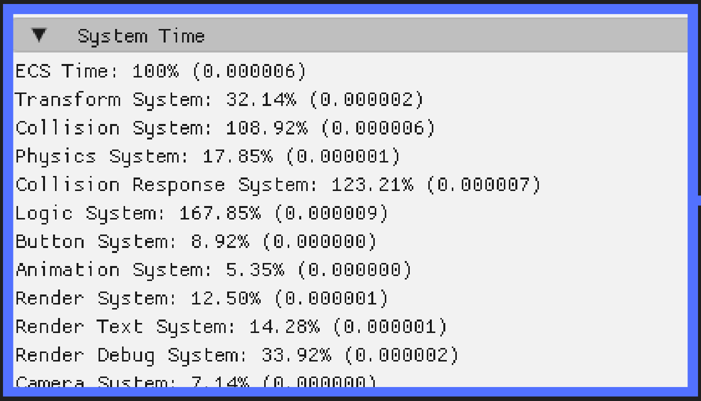


5. PERFORMANCE / INPUT PANEL



```
FPS 167.448074
```

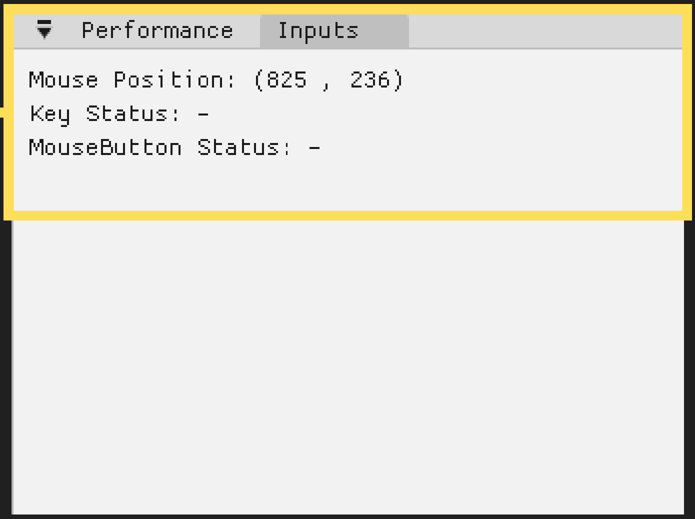
Graph plotter shows FPS Performance of Engine at Runtime



```
ECS Time: 100% (0.000006)
Transform System: 32.14% (0.000002)
Collision System: 108.92% (0.000006)
Physics System: 17.85% (0.000001)
Collision Response System: 123.21% (0.000007)
Logic System: 167.85% (0.000009)
Button System: 8.92% (0.000000)
Animation System: 5.35% (0.000000)
Render System: 12.50% (0.000001)
Render Text System: 14.28% (0.000001)
Render Debug System: 33.92% (0.000002)
Camera System: 7.14% (0.000000)
```

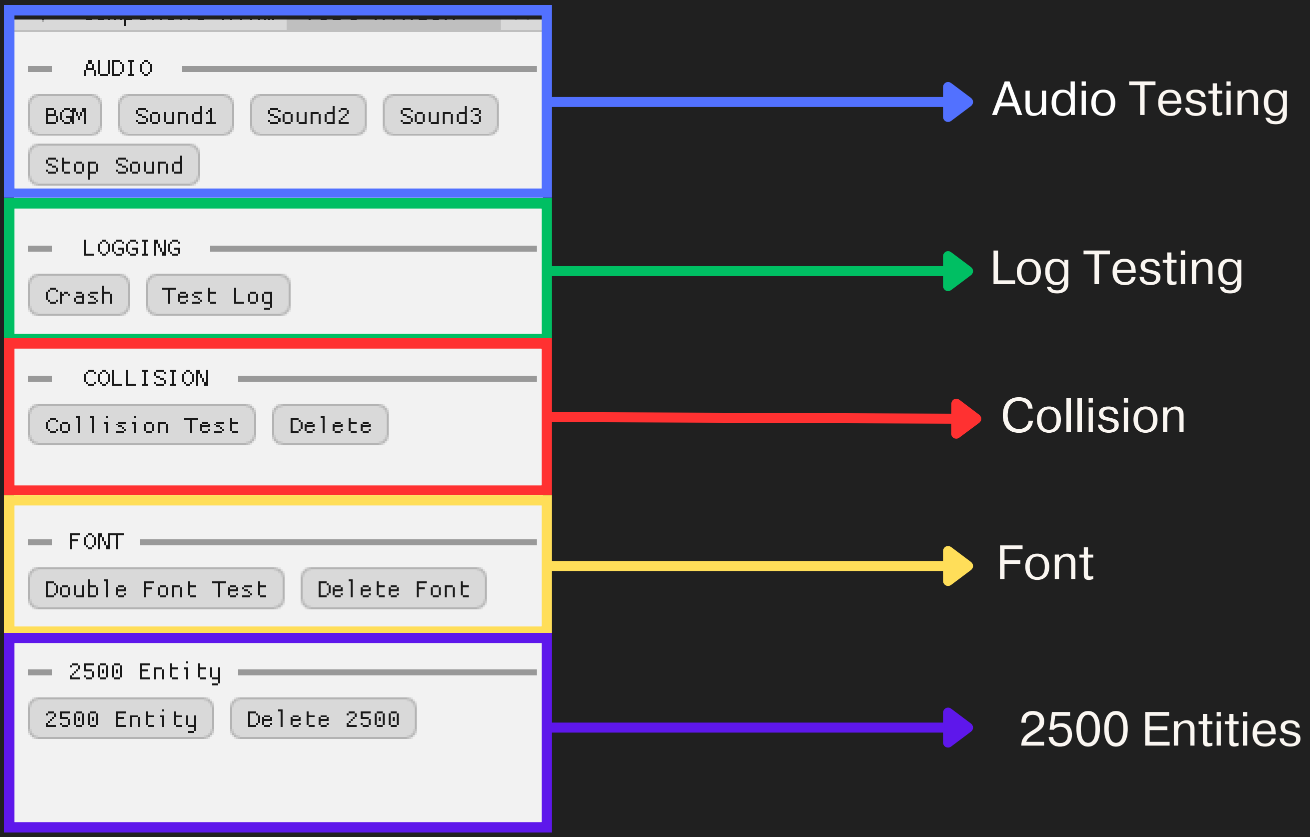
Breakdown of System Performance in relation to Engine Time

Shows Current User Inputs in Engine at runtime



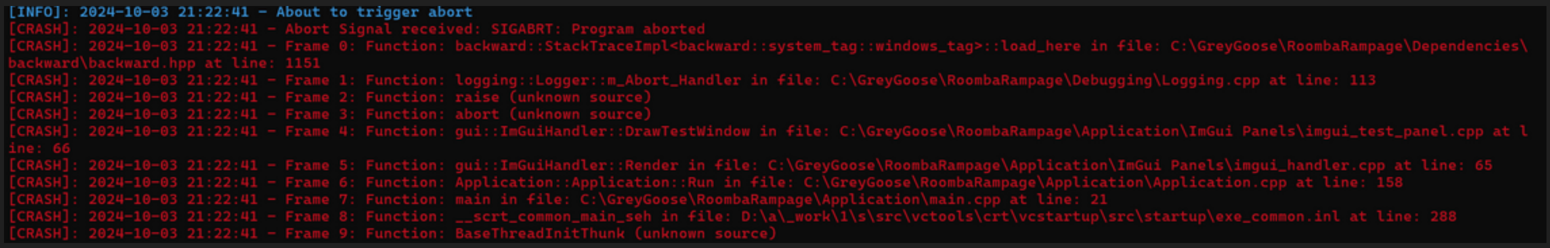
```
Mouse Position: (825 , 236)
Key Status: -
MouseButton Status: -
```

6. TEST PANEL



Crash Testing

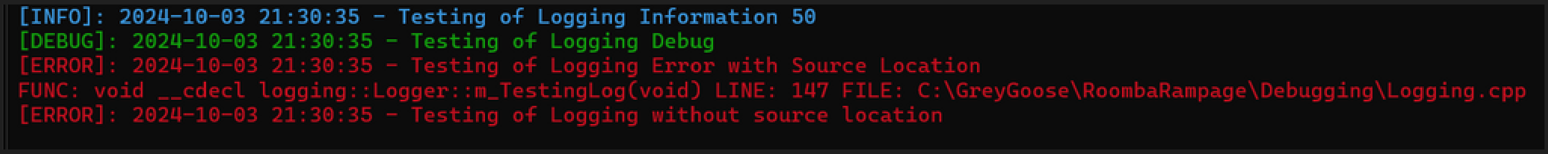
Below shows the log in the console.
It includes the file, function and the line of the code.



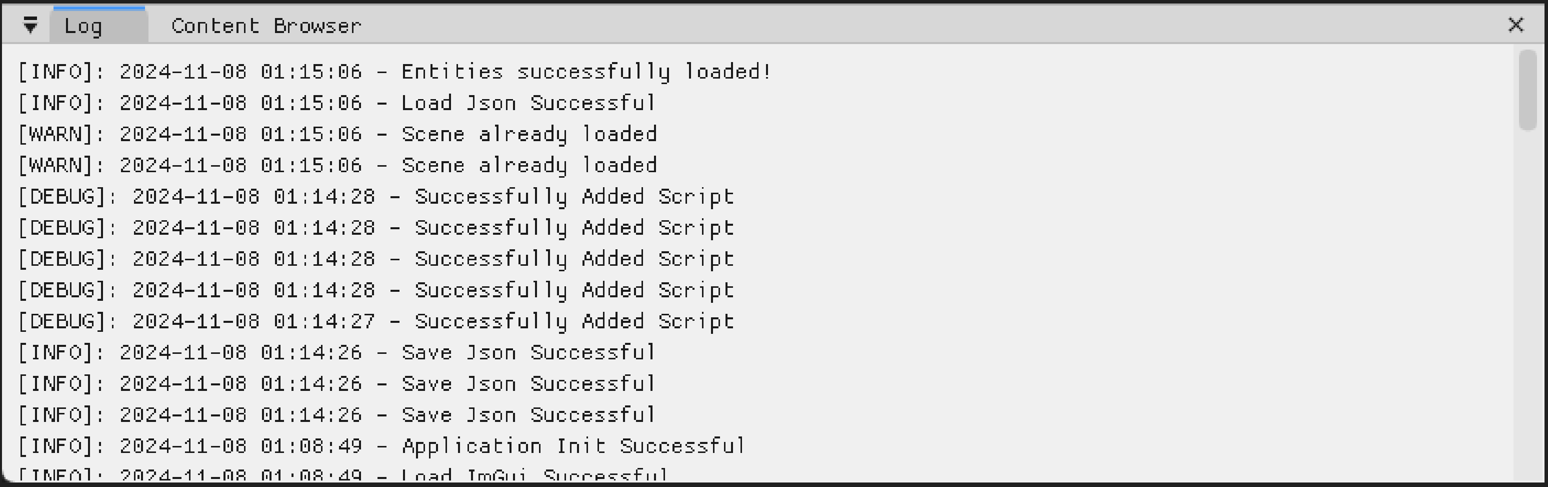
```
[INFO]: 2024-10-03 21:22:41 - About to trigger abort
[CRASH]: 2024-10-03 21:22:41 - Abort Signal received: SIGABRT: Program aborted
[CRASH]: 2024-10-03 21:22:41 - Frame 0: Function: backward::StackTraceImpl<backward::system_tag::windows_tag>::load_here in file: C:\GreyGoose\RoombaRampage\Dependencies\
backward\backward.hpp at line: 1151
[CRASH]: 2024-10-03 21:22:41 - Frame 1: Function: logging::Logger::m_Abort_Handler in file: C:\GreyGoose\RoombaRampage\Debugging\Logging.cpp at line: 113
[CRASH]: 2024-10-03 21:22:41 - Frame 2: Function: raise (unknown source)
[CRASH]: 2024-10-03 21:22:41 - Frame 3: Function: abort (unknown source)
[CRASH]: 2024-10-03 21:22:41 - Frame 4: Function: gui::ImGuiHandler::DrawTestWindow in file: C:\GreyGoose\RoombaRampage\Application\ImGui Panels\imgui_test_panel.cpp at l
ine: 66
[CRASH]: 2024-10-03 21:22:41 - Frame 5: Function: gui::ImGuiHandler::Render in file: C:\GreyGoose\RoombaRampage\Application\ImGui Panels\imgui_handler.cpp at line: 65
[CRASH]: 2024-10-03 21:22:41 - Frame 6: Function: Application::Application::Run in file: C:\GreyGoose\RoombaRampage\Application\Application.cpp at line: 158
[CRASH]: 2024-10-03 21:22:41 - Frame 7: Function: main in file: C:\GreyGoose\RoombaRampage\Application\main.cpp at line: 21
[CRASH]: 2024-10-03 21:22:41 - Frame 8: Function: __scrt_common_main_seh in file: D:\a\_work\1\s\src\vctools\crt\vcstartup\src\startup\exe_common.inl at line: 288
[CRASH]: 2024-10-03 21:22:41 - Frame 9: Function: BaseThreadInitThunk (unknown source)
```

Logging Test

Below shows the different kind of logs.

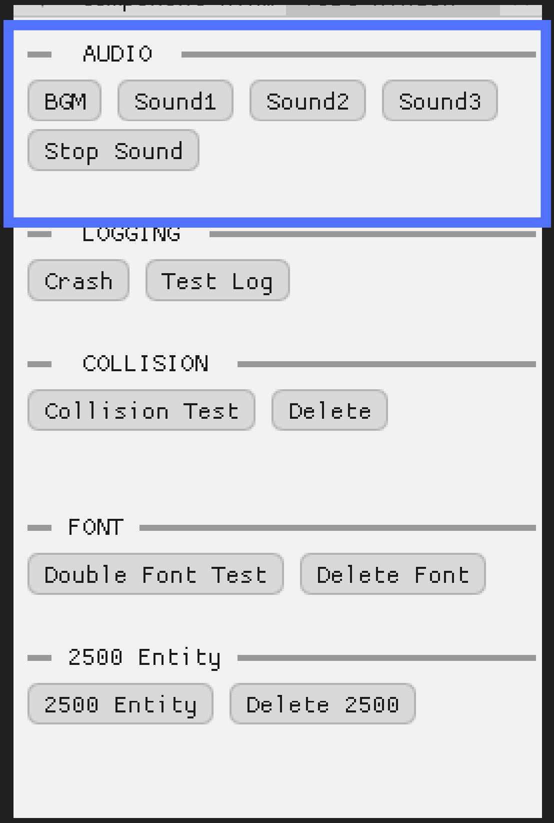


```
[INFO]: 2024-10-03 21:30:35 - Testing of Logging Information 50
[DEBUG]: 2024-10-03 21:30:35 - Testing of Logging Debug
[ERROR]: 2024-10-03 21:30:35 - Testing of Logging Error with Source Location
FUNC: void __cdecl logging::Logger::m_TestingLog(void) LINE: 147 FILE: C:\GreyGoose\RoombaRampage\Debugging\Logging.cpp
[ERROR]: 2024-10-03 21:30:35 - Testing of Logging without source location
```



```
[INFO]: 2024-11-08 01:15:06 - Entities successfully loaded!
[INFO]: 2024-11-08 01:15:06 - Load Json Successful
[WARN]: 2024-11-08 01:15:06 - Scene already loaded
[WARN]: 2024-11-08 01:15:06 - Scene already loaded
[DEBUG]: 2024-11-08 01:14:28 - Successfully Added Script
[DEBUG]: 2024-11-08 01:14:28 - Successfully Added Script
[DEBUG]: 2024-11-08 01:14:28 - Successfully Added Script
[DEBUG]: 2024-11-08 01:14:28 - Successfully Added Script
[DEBUG]: 2024-11-08 01:14:27 - Successfully Added Script
[INFO]: 2024-11-08 01:14:26 - Save Json Successful
[INFO]: 2024-11-08 01:14:26 - Save Json Successful
[INFO]: 2024-11-08 01:14:26 - Save Json Successful
[INFO]: 2024-11-08 01:08:49 - Application Init Successful
[INFO]: 2024-11-08 01:08:49 - Load ImGui Successful
```

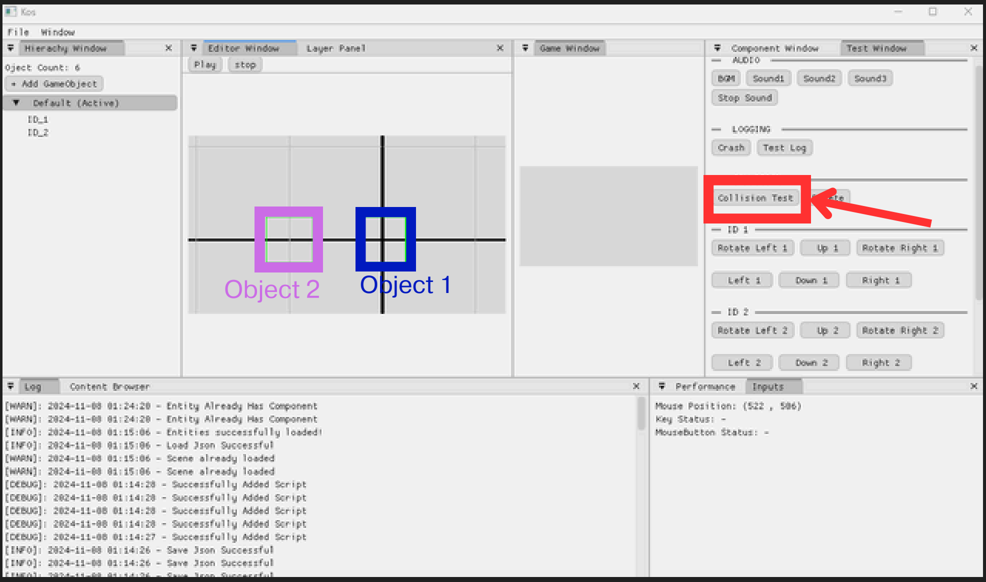
Audio Test



Select the button for the sound to play.

Press stop sound to end.

Collision Test

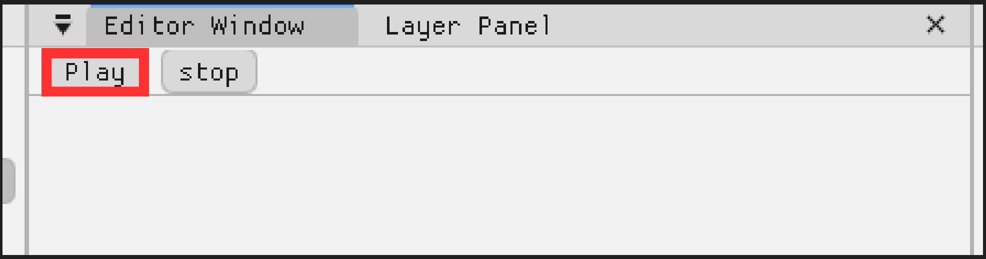


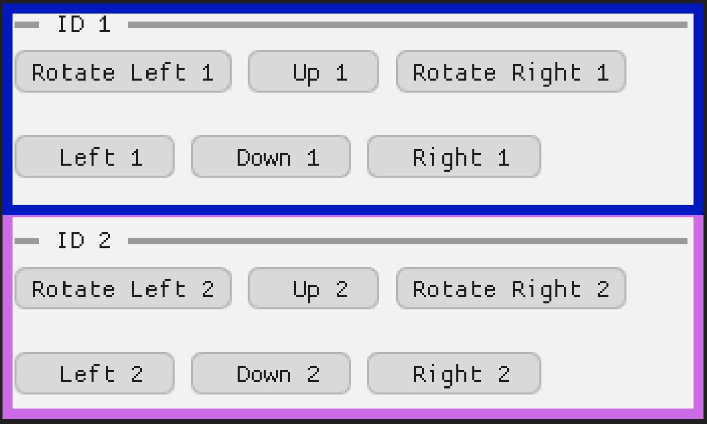
```
[WARN]: 2024-11-08 01:24:20 - Entity Already Has Component
[WARN]: 2024-11-08 01:24:20 - Entity Already Has Component
[INFO]: 2024-11-08 01:15:06 - Entities successfully loaded!
[INFO]: 2024-11-08 01:15:06 - Load Json Successful
[WARN]: 2024-11-08 01:15:06 - Scene already loaded
[WARN]: 2024-11-08 01:15:06 - Scene already loaded
[DEBUG]: 2024-11-08 01:14:28 - Successfully Added Script
[DEBUG]: 2024-11-08 01:14:28 - Successfully Added Script
[DEBUG]: 2024-11-08 01:14:28 - Successfully Added Script
[DEBUG]: 2024-11-08 01:14:28 - Successfully Added Script
[DEBUG]: 2024-11-08 01:14:27 - Successfully Added Script
[INFO]: 2024-11-08 01:14:26 - Save Json Successful
[INFO]: 2024-11-08 01:14:26 - Save Json Successful
```

```
Mouse Position: (522 , 586)
Key Status: -
MouseButton Status: -
```

Click on
"Collision Test"

Click on
"Play"



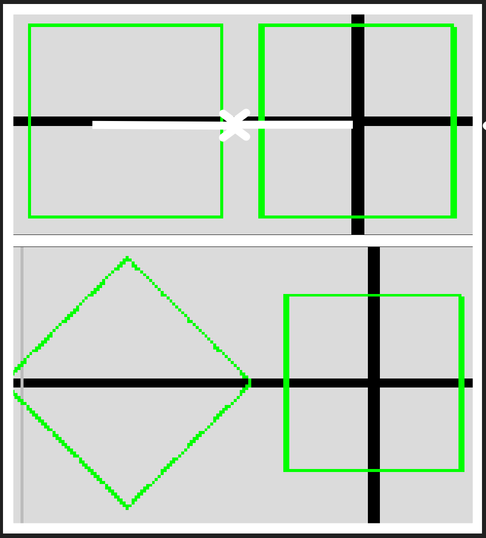


Object 1

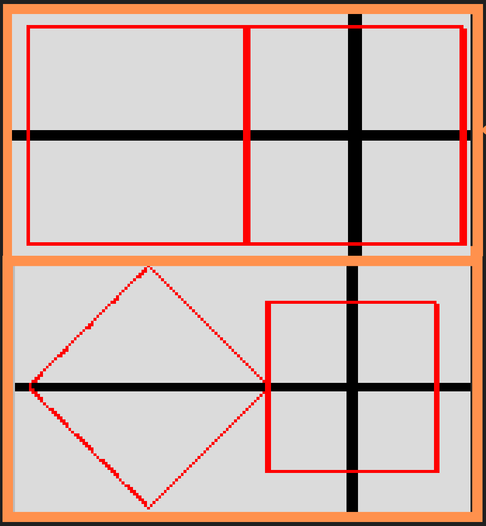
Move the 2 objects
using the buttons

Object 2

Collision Test

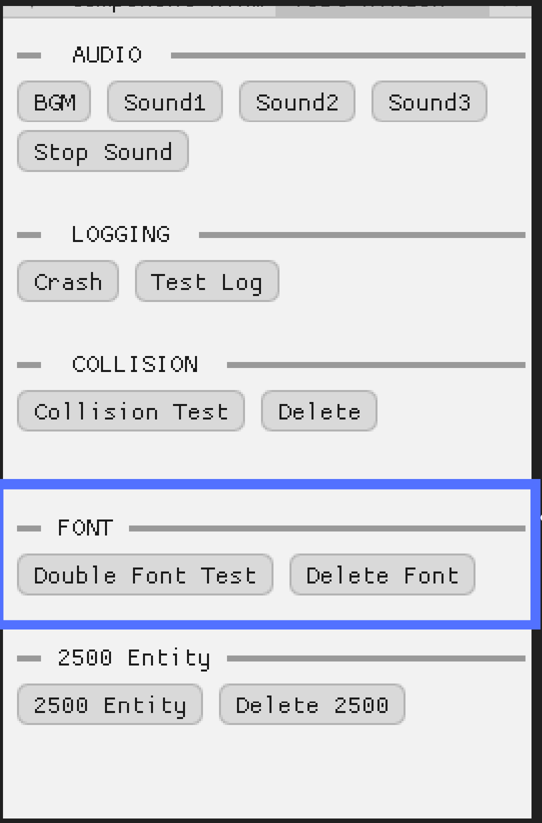


Objects move towards each other.



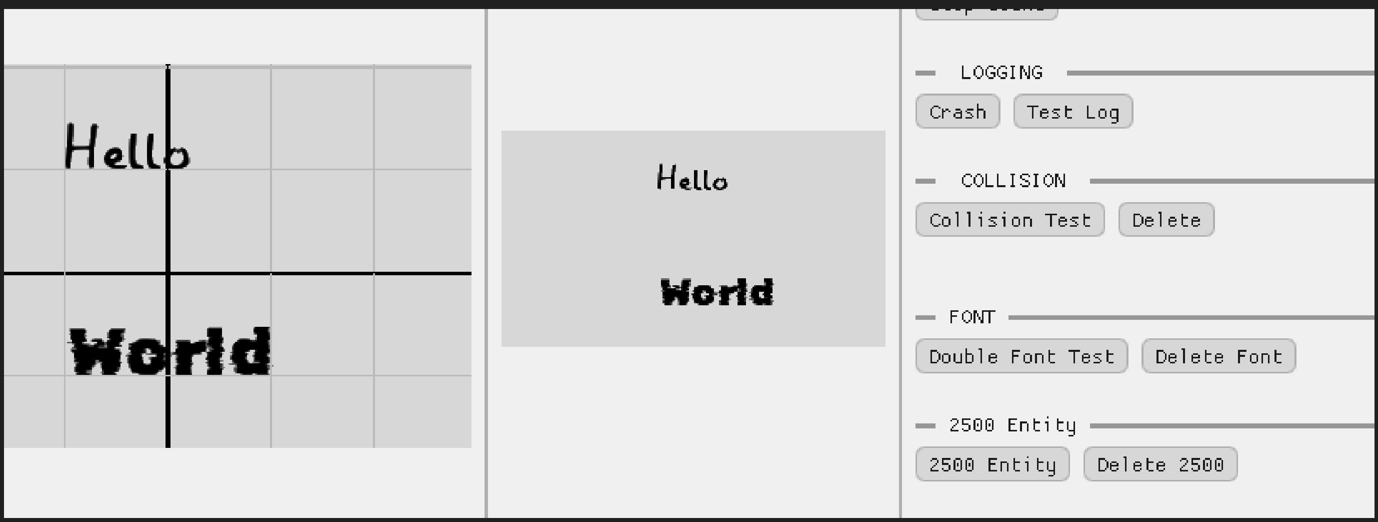
Once collided, both object will stop moving.

7. FONT



Click on Font Test

Shows 2 Fonts


Hello
World
Hello
World
LOGGING
Crash
Test Log
COLLISION
Collision Test
Delete
FONT
Double Font Test
Delete Font
2500 Entity
2500 Entity
Delete 2500