## **6. Tutorials and Examples**

### **Getting Started**

The "Getting Started" section is designed for beginners embarking on their game development journey with Muniffic. Here, you will find information on the basic functions of the engine, learn how to utilize different components, and create simple entities. This foundational knowledge will help you understand the core aspects of Muniffic and prepare you for more advanced topics.

### **Intermediate Tutorials**

Once you are comfortable with the basics of Muniffic, the intermediate tutorials will help you delve deeper into the engine's capabilities. This section covers the use of audio, more complex entities, and advanced entity functions. You'll gain a better understanding of how to leverage Muniffic’s features to create more sophisticated game mechanics and interactions.

### **Advanced Section**

For those looking to push the boundaries of what Muniffic can achieve, the advanced tutorials provide in-depth knowledge on specialized topics. These tutorials cover optimization techniques, networking for multiplayer games, custom shaders, and advanced rendering effects. This section is ideal for developers who want to master the full potential of Muniffic and create highly polished games.

### **Example Project**

In addition to the tutorials, we provide an example project that demonstrates various aspects and functionalities of Muniffic. This project serves as a practical reference, showcasing how to implement different features and mechanics within the engine. It is a valuable resource for understanding how to bring your own game ideas to life using Muniffic.

By following these tutorials and exploring the example project, you'll gain a comprehensive understanding of Muniffic and be well-equipped to develop your own games.

**Getting started**

**Entity Management**

#### **How to Create an Entity**

1. Right-click on the **“Scene Hierarchy”** panel.
2. Select **“Create Empty Entity”** from the context menu.
3. Click on your newly created entity to begin editing it.

#### **Renaming Your Entity**

1. Click on the entity you wish to rename.
2. In the **“Properties”** panel, type the desired name in the input field.

#### **Modifying Your Entity’s Position**

1. Select the entity whose position you want to change.
2. In the **“Transform”** panel, adjust the position values as needed.

#### **Adding Components to an Entity**

1. Click on the entity to which you want to add a component.
2. In the **“Properties”** panel, select **“Add Component”**.
3. Choose the appropriate component. (Learn about different components in the [Components Section](https://chatgpt.com/c/5ae8e485-76d2-4db5-b401-225c81d44578).)

By following these steps, you can create, rename, modify, and enhance your entities within Muniffic, allowing you to build the foundation of your game world.

**Sandbox**

**Showing Physics Colliders**

1. Open **“Settings”** panel
2. Check the **“Show Physics Colliders”** box

**Showing Axis**

1. Open **“Settings”** panel
2. Check the **“Show Axis”** box

**Showing Grid**

1. Open **“Settings”** panel
2. Check the **“Show Grid”** box

By enabling these options in the Settings panel, you can visualize important elements like physics colliders, axes, and the grid in the sandbox, making it easier to design and debug your game environment.

**Project**

**Opening a Project**

1. Click on the **“File”** button in the menu bar.
2. Select **“Open Project”** from the dropdown menu.

Alternatively, use the **Ctrl+O** keyboard shortcut.

### **Creating a New Scene**

1. Click on the **“File”** button in the menu bar.
2. Select **“New Scene”** from the dropdown menu.

Alternatively, use the **Ctrl+N** keyboard shortcut.

### **Saving a Scene**

1. Click on the **“File”** button in the menu bar.
2. Select **“Save Scene”** from the dropdown menu.

Alternatively, use the **Ctrl+S** keyboard shortcut.

### **Saving a Scene As**

1. Click on the **“File”** button in the menu bar.
2. Select **“Save Scene As”** from the dropdown menu.

Alternatively, use the **Ctrl+Shift+S** keyboard shortcut.

**Content Browser**

### **Changing Thumbnail Size**

1. Navigate to the **“Content Browser”** panel.
2. Adjust the **“Thumbnail Size”** slider to your desired value.

### **Changing Padding Size**

1. Navigate to the **“Content Browser”** panel.
2. Adjust the **“Padding”** slider to your desired value.

### **Deleting a Folder/File**

1. Navigate to the **“Content Browser”** panel.
2. Right-click on the folder or file you want to delete.
3. Select the **“Delete”** option.

### **Renaming a Folder/File**

1. Navigate to the **“Content Browser”** panel.
2. Right-click on the folder or file you want to rename.
3. Select the **“Rename”** option.
4. Enter the new name in the **“Enter new folder name”** input (or **“Enter new resource name”** input for files).
5. Click the **“Rename”** button.

### **Creating a New Folder**

1. Navigate to the **“Content Browser”** panel.
2. Click on the **“+”** button.
3. Select the **“Create Folder”** option.
4. Enter the name of the folder in the **“Enter new folder name”** input.
5. Click the **“Create Directory”** button.

By following these instructions, you can efficiently manage your files and folders within the Content Browser panel, customizing the layout, organizing your resources, and maintaining a tidy workspace.

**Resources**

**Assets Management**

For an optimal user experience, it’s crucial to keep your project files and folders well-organized. Place all assets in the main 'Assets' folder within your project directory. Within 'Assets,' categorize files into dedicated subfolders to maintain clarity and efficiency. Here’s a suggested structure:

**Adding Files to Projects**

1. Navigate to the menu in the top left corner.
2. Click on the **Resources** button.
3. Choose the **Add Resources** button.
4. Add the desired resource to the right folder.

**Assets Folder:** Your main directory for all project files.

1. **Audio:** Store all audio files, including sound effects, music, and voice-overs.
2. **Scripts:** Keep all code files organized here for easy reference and debugging.
3. **Textures:** Place all textures and images used in your game.
4. **Subtextures:** Store your subtextures here.
5. **Fonts:** Place all the fonts you want to use in this directory.
6. **Scenes:** Keep all scene files organized for structured level management.
7. **Animations:** Store animation files for characters and objects.
8. **SpriteAtlas:** Keep your sprite atlases here.
9. **Shaders:** Place all user interface elements like buttons, menus, and HUDs.

By categorizing your assets, you not only streamline your workflow but also ensure that your project remains scalable and easier to maintain.

**Components**

#### To modify yours entity component check the “**Adding Components to an Entity”**

In the **“Entity Management”** section.

**Camera**

Adding a camera to your project is necessary for it to work. Add it to your main Entity, so it can follow it.

### **Making a Camera Primary**

By making your camera primary, you tell the engine that it is the main camera of the project so it will follow your entity as you play the game.

1. Navigate to the **“Camera”** section in the **“Properties”** panel.
2. Check the **“Primary”** checkbox.

### **Changing the Projection of the Camera**

To maintain parallel lines of your projection, use **Ortographic** option, but if you want the sense of depth, use **Perspective** projection.

1. Navigate to the **“Camera”** section in the **“Properties”** panel.
2. Select the desired option from the **“Projection”** dropdown menu.

### **Changing the Size of the Camera**

By changing the size of your projection, you set how much of the map player can see.

1. Navigate to the **“Camera”** section in the **“Properties”** panel.
2. Adjust the **“Size”** slider to the desired value.

### **Changing the Near Clipping of the Camera**

By changing the Near Clipping of your camera, you set the closest point relative to the camera that drawing will occur.

1. Navigate to the **“Camera”** section in the **“Properties”** panel.
2. Adjust the **“Near”** slider to the desired value.

### **Changing the Far Clipping of the Camera**

By changing the Near Clipping of your camera, you set the furthest point relative to the camera that drawing will occur.

1. Navigate to the **“Camera”** section in the **“Properties”** panel.
2. Adjust the **“Far”** slider to the desired value.

### **Fixing the Aspect Ratio of the Camera**

By choosing this option you make yours camera aspect ratio fixed.

1. Navigate to the **“Camera”** section in the **“Properties”** panel.
2. Check the **“Fixed Aspect Ratio”** checkbox.

**Removing Camera Component**

1. Navigate to the **“Camera”** section in the **“Properties”** panel.
2. Click on the **“...”** button.
3. Click the **“Remove Component”** button.

By using these settings in the Properties panel, you can configure your camera to suit the specific needs of your project, ensuring optimal performance and visual output.

**Sprite Renderer**

Sprite renderer makes your entity visible, and you can modify its appearance with this component. Use this for rectangle entities.

**Changing the Color of the Entity**

You use this option if you want to change the color of your entity.

1. Navigate to the **“Sprite Renderer”** section in the **“Properties”** panel.
2. Click on the **“Color”** input box and select the desired color or modify the RGB values in the input fields.

### **Changing the Texture of the Entity**

You use this option to change the texture of your entity. But be careful. If you apply a texture to your entity its color will not be visible.

1. Navigate to the **“Sprite Renderer”** section in the **“Properties”** panel.
2. Drag a texture from the **“Content Browser”** panel into the **“Texture”** field.

### **Changing the Tiling Factor of the Entity**

The Tiling Factor option is used to control the repetition of a texture across a surface. It determines how many times a texture will tile or repeat horizontally and vertically on an entity or object.

1. Navigate to the **“Sprite Renderer”** section in the **“Properties”** panel.
2. Adjust the **“Tiling Factor”** slider to the desired value.

By following these steps, you can customize the appearance of your entities, giving them the right color, texture, and tiling properties to fit your game's aesthetic.

**Circle Renderer**

Sprite renderer makes your entity visible, and you can modify its appearance with this component. Use this for circle entities.

**Changing the Color of the Entity**

You use this option if you want to change the color of your entity.

1. Navigate to the **“Circle Renderer”** section in the **“Properties”** panel.
2. Click on the **“Color”** input box and select the desired color or modify the RGB values in the input fields.

### **Changing the Thickness of the Entity**

You use this to change your circle entity’s border thickness. The default value is 1 which is also the biggest value you can set.

1. Navigate to the **“Circle Renderer”** section in the **“Properties”** panel.
2. Adjust the **“Thickness”** slider to the desired value.

### **Changing the Fade of the Entity**

You use this to make your circle entity more faded.

1. Navigate to the **“Circle Renderer”** section in the **“Properties”** panel.
2. Adjust the **“Fade”** slider to the desired value.

By following these steps, you can customize the appearance of your circle entities, including their color, thickness, and fade properties, to suit your game's design needs.

**Rigidbody 2D**

If you want to create physical object you need to add Rigidbody to it. It gives your entity the physical body which can be affected by gravity and collisions.

**Changing the Body Type of the Entity**

You can apply three body types. If you want to create an object that will not be affected by gravity, won’t be moving and you won’t apply collisions to it, use Static body type. It is good for ground, walls and other objects that you don’t want your character to collide with.

If your entity will be entirely driven by scripts and only process collisions with dynamic objects, use Kinematic body type. It is the best type if you want your character to move with more flexibility.

But if your objects are entirely driven by game engine and are affected by force, velocity etc., use Dynamic body type. Physics engine takes responsibility for resolving their collisions. It is the best type for game characters, moving game objects etc.

1. Navigate to the **“Rigidbody 2D”** section in the **“Properties”** panel.
2. Select the desired value from the **“Body Type”** dropdown menu.

**Fixing the Rotation of the Entity**

You use this option to stop your rigid body from rotating.

1. Navigate to the **“Rigidbody 2D”** section in the **“Properties”** panel.
2. Check the **“Fixed Rotation”** checkbox.

These settings allow you to configure the physics behavior of your 2D entities, including their body type and whether their rotation is fixed, to ensure the desired physical interactions within your game.

**Box Collider 2D**

You apply box collider if you want to apply collisions to your entity or other game objects that require collisions. Use this for rectangle entities.

**Changing the Offset of the Entity’s Box Collider**

By changing the offset, you set the position of entity’s collider.

1. Navigate to the **“Box Collider 2D”** section in the **“Properties”** panel.
2. Adjust the **“Offset”** sliders to the desired values.

### **Changing the Size of the Entity’s Box Collider**

By changing the size of your entity’s collider, you set the area that will be sensitive to collisions.

1. Navigate to the **“Box Collider 2D”** section in the **“Properties”** panel.
2. Adjust the **“Size”** sliders to the desired values.

### **Changing the Density of the Entity’s Box Collider**

The density of your entity is used to calculate its mass. The bigger the density value is, the bigger is the mass of your entity.

1. Navigate to the **“Box Collider 2D”** section in the **“Properties”** panel.
2. Adjust the **“Density”** slider to the desired value.

### **Changing the Friction of the Entity’s Box Collider**

The friction of the entity can affect how fast it goes. The bigger the friction, the slower the entity moves.

1. Navigate to the **“Box Collider 2D”** section in the **“Properties”** panel.
2. Adjust the **“Friction”** slider to the desired value.

### **Changing the Restitution of the Entity’s Box Collider**

By changing the restitution, you change how bouncy your entity is.

1. Navigate to the **“Box Collider 2D”** section in the **“Properties”** panel.
2. Adjust the **“Restitution”** slider to the desired value.

### **Changing the Restitution Threshold of the Entity’s Box Collider**

You use it to define a threshold at which your entities will no longer bounce of each other.

1. Navigate to the **“Box Collider 2D”** section in the **“Properties”** panel.
2. Adjust the **“Restitution Thresh”** slider to the desired value.

By configuring these settings, you can fine-tune the physical properties of your entity's box collider to achieve the desired behavior in your game, including its size, offset, density, friction, and restitution characteristics.

**Circle Collider 2D**

You apply circle collider if you want to apply collisions to your entity or other game objects that require collisions. Use this for circle entities.

**Changing the Offset of the Entity’s Circle Collider**

By changing the offset, you set the position of entity’s collider.

1. Navigate to the **“Circle Collider 2D”** section in the **“Properties”** panel.
2. Adjust the **“Offset”** sliders to the desired values.

### **Changing the Radius of the Entity’s Circle Collider**

By changing the radius of your entity’s collider, you set the area that will be sensitive to collisions.

1. Navigate to the **“Circle Collider 2D”** section in the **“Properties”** panel.
2. Adjust the **“Radius”** slider to the desired value.

### **Changing the Density of the Entity’s Circle Collider**

The density of your entity is used to calculate its mass. The bigger the density value is, the bigger is the mass of your entity.

1. Navigate to the **“Circle Collider 2D”** section in the **“Properties”** panel.
2. Adjust the **“Density”** slider to the desired value.

### **Changing the Friction of the Entity’s Circle Collider**

The friction of the entity can affect how fast it goes. The bigger the friction, the slower the entity moves.

1. Navigate to the **“Circle Collider 2D”** section in the **“Properties”** panel.
2. Adjust the **“Friction”** slider to the desired value.

### **Changing the Restitution of the Entity’s Circle Collider**

By changing the restitution, you change how bouncy your entity is.

1. Navigate to the **“Circle Collider 2D”** section in the **“Properties”** panel.
2. Adjust the **“Restitution”** slider to the desired value.

### **Changing the Restitution Threshold of the Entity’s Circle Collider**

You use it to define a threshold at which your entities will no longer bounce of each other.

1. Navigate to the **“Circle Collider 2D”** section in the **“Properties”** panel.
2. Adjust the **“Restitution Thresh”** slider to the desired value.

These settings allow you to fine-tune the physical properties of your entity's circle collider, including its offset, radius, density, friction, and restitution, to achieve the desired behavior in your game.

**Text Renderer**

You use the text renderer to display text in your projects.

**Writing Text to Render**

Use this to set the text you want to render.

1. Navigate to the **“Text Renderer”** section in the **“Properties”** panel.
2. Input the desired text into the **“Text String”** input field.

### **Changing the Font of the Text**

Use this option to change the font of your text component.

1. Navigate to the **“Text Renderer”** section in the **“Properties”** panel.
2. Select the desired font from the **“Font”** dropdown menu.

### **Changing the Color of the Text**

Use this option to change the color of your text.

1. Navigate to the **“Text Renderer”** section in the **“Properties”** panel.
2. Click on the **“Color”** input box and select the desired color or modify the RGB values in the input fields.

### **Changing the Kerning of the Text**

Manipulate this value to change the distance between letters of your text.

1. Navigate to the **“Text Renderer”** section in the **“Properties”** panel.
2. Adjust the **“Kerning”** slider to the desired value.

### **Changing the Line Spacing of the Text**

Manipulate this value to change the distance between lines of your text.

1. Navigate to the **“Text Renderer”** section in the **“Properties”** panel.
2. Adjust the **“Line Spacing”** slider to the desired value.

By following these steps, you can customize the appearance and layout of your text, including its content, font, color, kerning, and line spacing, to achieve the desired look for your game.

**SubTexture Sprite Renderer 2D**

Subtexture sprite renderer makes your subtextures visible in the editor and in the game.

**Changing the Color of the Subtexture**

Use this to change the color of your subtexture.

1. Navigate to the **“SubTexture Sprite Renderer 2D”** section in the **“Properties”** panel.
2. Click on the **“Color”** input box and select the desired color or modify the RGB values in the input fields.

### **Changing the Texture of the Subtexture**

Use this option to change your subtexture's texture.

1. Navigate to the **“SubTexture Sprite Renderer 2D”** section in the **“Properties”** panel.
2. Drag the texture from the **“Content Browser”** into the **“Texture”** field.

### **Changing the Minimal Coordinates of the Subtexture**

Set these values to set the coords from which the subtexutre will start rendering.

1. Navigate to the **“SubTexture Sprite Renderer 2D”** section in the **“Properties”** panel.
2. Adjust the **“Min Coords”** sliders to the appropriate values.

### **Changing the Maximal Coordinates of the Subtexture**

By setting this values you indicate the range for your subtexture to be rendered.

1. Navigate to the **“SubTexture Sprite Renderer 2D”** section in the **“Properties”** panel.
2. Adjust the **“Max Coords”** sliders to the appropriate values.

### **Changing the Tiling Factor of the Subtexture**

The Tiling Factor option is used to control the repetition of a texture across a surface. It determines how many times a texture will tile or repeat horizontally and vertically on an entity or object.

1. Navigate to the **“SubTexture Sprite Renderer 2D”** section in the **“Properties”** panel.
2. Adjust the **“Tiling Factor”** slider to the appropriate value.

By following these steps, you can effectively customize the appearance and behavior of subtextures within your 2D entities, including their color, texture, coordinates, and tiling properties, to enhance your game's visual style and functionality.

**Intermediate Tutorials**

**Audio Integration and Management**

Use this component to add audio to your entities.

**Audio Setup**

1. Add **“Audio”** component to your entity.
2. Choose your Audio file. Ensure your audio file is in **.wav format** for compatibility with our Engine.

**Looping Audio**

You should loop your audio if you want it to play continuously during your game.

1. Navigate to the **“Audio”** section in the **“Properties”** panel.
2. Check the **“Loop”** box.

**Playing Audio from the start**

Check this option if you want your audio to play from the start.

1. Navigate to the **“Audio”** section in the **“Properties”** panel.
2. Check the **“Playing from start”** box.

**Changing the Volume of Audio**

Use this option to change your audio’s volume.

1. Navigate to the **Volume** slider.
2. Choose the desired volume value.

**Child Components and Inheritance**

**Creating Child Entity**

1. Navigate to the component that you want to create child entity of.
2. Right Click on the component.
3. Click on the **“Create Child Entity”** button.

**Stop Inheriting a Component in Child Entity**

1. Navigate to the component that you want to stop inheriting with your Child Entity.
2. Click on the three dots near the component name.
3. Choose the **“Stop inheriting component in children”** option.

**Copy Component to Child Entity**

1. Navigate to the component that you want to stop inheriting with your Child Entity.
2. Click on the three dots near the component name.
3. Choose the **“Copy to children”** option.

**Remove Component from Child Entity**

1. Navigate to the component that you want to stop inheriting with your Child Entity.
2. Click on the three dots near the component name.
3. Choose the **“Remove from children”** option.

**Copy Component with Values to Child Entity**

1. Navigate to the component that you want to stop inheriting with your Child Entity.
2. Click on the three dots near the component name.
3. Choose the **“Copy component with values to children”** option.

**Animation Techniques**

To make your animation you need to add it to your project resources.

**Changing the Amount of Rows and Columns of Animation**

This option divide your spreadsheet to the desired amount of rows and columns.

1. Navigate to the **Column count** and **Rows count** sliders.
2. Change them to your desired values.

**Selecting Tiles for your Animation**

Use this option to select which tiles from your spreadsheet you want to use in your animation.

1. Navigate to the **Begin Column** and **Begin Row** sliders.
2. Choose the desired rows and columns.
3. Navigate to the **Selected Columns Count** and **Selected Rows Count** sliders.
4. Choose the desired amount of tiles from each row and column.

To select all tiles click the **Select All** button. If you don’t want to select any tile click the **Select None** button.

**Playing your Animation**

To play your animation check the **Play** checkbox.

**Looping your Animation**

To loop your animation check the **Loop** checkbox.

**Changing the Frame Rate of your Animation**

Frame rate of your animation indicates how fast slides are changing in your animation. If you want your animation to be fast and smooth, make sure that frame rate value is high.

1. Navigate to the **Frame Rate** slider.
2. Choose the desired value.

**Changing the Name of your Animation**

1. Navigate to the **Animation Name** input.
2. Type in the desired name.

**Saving your Animation**

To save your animation click on the **Save** button. If you don’t want tom save your animation click on the **Cancel** button.

**Adding Empty Animation to your Entity**

1. Navigate to the **“Animator”** section in the **“Properties”** panel.
2. Drag your animation on the **“Drop Animation Here”** button.

**Changing the Speed of the Animation**

Modify this value to tell the game engine how fast to play your animation.

1. Navigate to the **“Animator”** section in the **“Properties”** panel.
2. Select the appropriate value from the **“Speed”** slider.

**Adding Transitions to Your Animations**

By adding transitions between your animations, you indicate the order in which animations will be playing.

1. Navigate to the **“Animator”** section in the **“Properties”** panel.
2. Click on the **“Add Transition”** button.
3. Choose the desired transition

**Remember**, you need to have at least two animations in a single entity to apply transitions between them.

**Building Scenes**

**Adding Entities**

1. **Create Empty Entity:** Start by creating a new, empty entity in your scene. This entity will act as the foundation for your game object. Transform it to the desired size.
2. **Add Components to Your Entity:** Customize the entity by adding the necessary components to bring it to life. For a realistic and functional game object, we recommend adding the following components:
   * **Camera:** Attach a Camera component if this entity will be responsible for rendering the game view.
   * **Script:** Add a Script component to control the behavior and logic of the entity.
   * **Sprite Renderer:** Use the Sprite Renderer component to give your entity a visual representation. This could be a character, object, or any other element in your game.
   * **Box Collider 2D:** Include a Box Collider 2D component to define the physical boundaries of the entity for collision detection.
   * **Rigidbody 2D:** Add a Rigidbody 2D component to enable physics interactions, such as movement and collisions, giving the entity a realistic presence in the game world. Use ‘Dynamic’ body type for entities.

By combining these components, you can create a versatile and interactive entity that forms the building blocks of your game.

**Adding Obstacles and Other Physical Objects**

1. **Create Empty Entity:** Begin by creating a new empty entity in your scene. This entity will serve as the base for your obstacle.
2. **Transform the Obstacle:** Adjust the entity's Transform properties (position, scale, and rotation) to set the desired size and shape of your obstacle.
3. **Position the Obstacle:** Move the entity to the location where you want the obstacle to appear in your game world.
4. **Add Components for Basic Functionality:** Enhance your obstacle with the following components to give it form and functionality:
   * **Sprite Renderer:** Attach a Sprite Renderer component to give your obstacle a visible sprite. This will make the obstacle appear in the game.
   * **Rigidbody 2D:** Add a Rigidbody 2D component to allow the obstacle to interact with physics. You can set its body type to "Static" if you don't want it to move, or "Dynamic" if it should react to forces and collisions.
   * **Box Collider 2D:** Include a Box Collider 2D component to define the physical boundaries of the obstacle. This will enable collision detection between the obstacle and other entities.

With these steps, you create a functional obstacle that can interact with other elements in your game, adding both challenge and realism to the gameplay.

**Advanced Section & Example Project**

**Building a Simple Game**

**Downloading Software**

To build your first game, you’ll need to download the **Muniffic** gameengine. For detailed instructions on how to install it, refer to the **Installation** section of this documentation. There, you'll find step-by-step guidance on setting up Muniffic, so you can start creating your game in no time.

**Project Setup**

1. Launch the **Muniffic** game engine editor.
2. In the **Menu** section, click on the **Create New Project** button.
3. Enter your project name.
4. Click the **Save** button to create and store your new project.

You're now ready to start developing your game!

**Project Structure**

Before diving into game development, ensure your project is well-organized for smooth and efficient workflow. Follow the guidelines in the **Assets Management** section to keep everything structured and accessible, making development easier and more enjoyable.

Your projects will be stored in this path:

**../Muniffic/Engine-Editor/Projects/<project\_name>**

If you can’t locate your project, double-check your **Project Settings** to ensure the correct storage path is being used.

For this example project we will use the following files and structures:

Add these to your project assets using **“Add resource”** button.

**Building the first Scene**

**Adding Static Map Elements**

1. **Create Floor1:**
   * Add an empty entity, name it **Floor1**.
   * Add **Rigidbody 2D** and **Sprite Renderer** components, and assign the **"Dirt" texture**.
   * Set the **Transform values** to:
     + **Translation:** X: 0.00, Y: -0.50, Z: 0.00
     + **Scale:** X: 20.00, Y: 1.00, Z: 1.00
     + Keep **Rotation** at 0.
2. **Create Floor2:**
   * Add another empty entity, name it **Floor2**.
   * Add **Rigidbody 2D** and **Sprite Renderer**, using the **"Grass" texture**.
   * Transform it as follows:
     + **Translation:** X: -7.50, Y: 0.50, Z: 0.00
     + **Scale:** X: 5.00, Y: 1.00, Z: 1.00
     + Keep **Rotation** at 0.
3. **Create Floor3:**
   * Add a third empty entity, name it **Floor3**.
   * Add **Rigidbody 2D** and **Sprite Renderer**, and apply the **"Grass" texture**.
   * Adjust the **Transform** to:
     + **Translation:** X: 5.00, Y: 1.50, Z: 0.00
     + **Scale:** X: 10.00, Y: 3.00, Z: 1.00
     + Keep **Rotation** at 0.
4. **Create Floor4:**
   * Finally, add one more empty entity, name it **Floor4**.
   * Add **Rigidbody 2D** and **Sprite Renderer**, and apply the **"Grass" texture**.
   * Set the **Transform values** to:
     + **Translation:** X: -2.70, Y: 1.80, Z: 0.00
     + **Scale:** X: 1.80, Y: 0.30, Z: 1.00
     + Keep **Rotation** at 0.

With these steps, you’ll quickly set up multiple floors with different textures and sizes for your game map.

**Adding Interactive Map Elements**

Before adding interactive elements to your map, you'll need to create some animations. Follow these steps to set up an animation for your map:

1. **Add New Animation:**

Click the **"Add Resources"** button in the editor to create a new animation. Name your animation **“Lava”**.

1. **Select Lava.png File:**

From the **Tutorial\_Assets** directory, choose the **Lava.png** file as your animation source.

1. **Divide Image:**

Divide the **Lava.png** image into a grid of **3 columns** and **3 rows**, creating a total of 9 frames for the animation.

1. **Set Frame Rate:**

Assign a frame rate of **5 frames per second (fps)** to the animation for a smooth transition between frames.

Once added, this animated lava effect can be integrated into your map, adding dynamic visual elements to your game world.

1. **Create Lava**
   * Add an empty entity, name it **Lava**.
   * Add **Rigidbody 2D, Animator** and **Box 2D Collider** components, then assign the **"Lava”** animation.
   * Set the **Transform values** to:
     + **Translation:** X: -2.50, Y: 0.50, Z: 0.00
     + **Scale:** X: 5.00, Y: 1.00, Z: 1.00
     + Keep **Rotation** at 0.
2. **Create Key**
   * Add an empty entity, name it **Key**.
   * Add **Rigidbody 2D, Sprite Renderer** and **Box 2D Collider** components, then assign the **"Key”** texture.
   * Set the **Transform values** to:
     + **Translation:** X: 6.50, Y: 3.30, Z: 0.00
     + **Scale:** X: 0.50, Y: 0.50, Z: 1.00
     + Keep **Rotation** at 0.
3. **Create Doors**
   * Add an empty entity, name it **Doors**.
   * Add **Rigidbody 2D, Sprite Renderer, Box 2D Collider** and **Script** components, then assign the **"Doors”** texture and **“Doors”** script.
   * Set the **Transform values** to:
     + **Translation:** X: 9.50, Y: 3.50, Z: 0.00
     + **Scale:** X: 1.00, Y: 1.00, Z: 1.00
     + Keep **Rotation** at 0.

**Creating Player and Entity**

Before creating your Player and Entity, you’ll need to set up two animations for your player character. Follow these steps:

1. **Create Player Movement Animation:**
   * Click on the **"Add Resources"** button to add a new animation.
   * Select the **Player.png** sprite sheet from the **Tutorial\_Assets** directory.
   * Divide the sprite sheet into **8 columns** and **8 rows**.
   * For the movement animation, use only the **second row**.
   * Set the **frame rate** to **2 fps**.
   * Name the animation **Player\_Movement**.
2. **Create Player Attack Animation:**
   * Repeat the same process by clicking **"Add Resources"** again to add a new animation.
   * Select the **Player.png** sprite sheet.
   * Divide it into **8 columns** and **8 rows**.
   * For the attack animation, use only the **fifth row**.
   * Set the **frame rate** to **2 fps**.
   * Name the animation **Player\_Attack**.

With these two animations **Player\_Movement** and **Player\_Attack.** Your player will have smooth movement and attack transitions in your game.

Now we can create our entities for the game.

1. **Create Player**
   * Add an empty entity, name it **Player**.
   * Add **Rigidbody 2D, Animator, Script, Camera** and **Box 2D Collider** components, then assign the **"Player\_Movement”** and **“Player\_Attack”** animations and **“Player”** script.
   * Set the **Transform values** to:
     + **Translation:** X: -8.50, Y: 1.50, Z: 0.00
     + **Scale:** X: 1.00, Y: 1.00, Z: 1.00
     + Keep **Rotation** at 0.
2. **Create Enemy**
   * Add an empty entity, name it **Enemy**.
   * Add **Rigidbody 2D, Animator, Script** and **Box 2D Collider** components, then assign the **“Enemy”** script.
   * Set the **Transform values** to:
     + **Translation:** X: 3.50, Y: 3.50, Z: 0.00
     + **Scale:** X: 1.00, Y: 1.00, Z: 1.00
     + Keep **Rotation** at 0.

Now we have the first of two scenes that we need. Save this scene and name it **Game\_Scene**. We need to create the finish screen scene now.

**Building the Second Scene**

Firstly, create new scene. It will be our title Screen. It will consist of two buttons and a text component.

**Adding Text Component to our Scene**

1. Create Empty Entity. Name it **“Ending\_Scene\_Text”**.
2. Set the **Transform values** to:

* **Translation:** X: -2.00, Y: 1.10, Z: 0.00
* **Scale:** X: 1.00, Y: 1.00, Z: 1.00
* Keep **Rotation** at 0.

1. Add the Text Renderer to your Entity. Give it a Text String of “Game Over”.
2. Set the text color to **R:251, G:18, B:18, A:255**.

**Adding Play Again Button**

1. Create Empty Entity. Name it **“Play\_Again\_Button”**.
2. Set the **Transform values** to:

* **Translation:** X: -1.70, Y: 0.00, Z: 0.00
* **Scale:** X: 3.30, Y: 1.00, Z: 1.00
* Keep **Rotation** at 0.

1. Add **Sprite Renderer** component. Change its color to **R: 73, G:69, B:69, A:255**.
2. Add **Script** component and give it a **“Play\_Again\_Button”** script provided in tutorial assets.
3. Create Child Entity of **Play\_Again\_Button**.
4. Name it **“Play\_Again\_Button\_Text”**.
5. Give the Child Entity transform values of:

* **Translation:** X: -3.00, Y: -0.20, Z: 0.80
* **Scale:** X: 0.80, Y: 1.00, Z: 1.00
* Keep **Rotation** at 0.

1. Add **Text Renderer** to Child Entity. Give it a Text String of **“Play Again”** and change colorto **R:255, G:255, B:255, A:255**.

With this setup, your **"Play Again"** button will be properly styled and functional, allowing players to restart the game with an interactive visual element.

**Adding Exit Button**

1. Create Empty Entity. Name it **“Exit\_Button”**.
2. Set the **Transform values** to:

* **Translation:** X: 2.20, Y: 0.00, Z: 0.00
* **Scale:** X: 1.90, Y: 1.00, Z: 1.00
* Keep **Rotation** at 0.

1. Add **Sprite Renderer** component. Change its color to **R: 73, G:69, B:69, A:255**.
2. Add **Script** component and give it a **“Exit\_Button”** script provided in tutorial assets.
3. Create Child Entity of **Exit\_Button**.
4. Name it **“Exit\_Button\_Text”**.
5. Give the Child Entity transform values of:

* **Translation:** X: 1.40, Y: -0.20, Z: 0.30
* **Scale:** X: 1.10, Y: 1.00, Z: 1.00
* Keep **Rotation** at 0.

1. Add **Text Renderer** to Child Entity. Give it a Text String of **“Exit”** and change colorto **R:255, G:255, B:255, A:255**.

Now, your **"Exit"** button is ready with proper styling and functionality, offering users a way to close or exit the game.

Your Game Ending scene is ready now. Save it as **“Game\_Ending\_Scene”**.

Congratulations! You finished the **Example Project** part. Now your game is ready to play. To run your game, press the play button on the top of the launcher window.