|  |
| --- |
| IMAT 2103 – Progressive Game Engines |
| Lab 2A: Introduction to Blueprints |

***Part 1:***

In this exercise, you will practice executing some basic steps in the Blueprint Editor by creating a simple Blueprint that will print a message on the screen.

**Directions**

1. In the Content Browser, click the green Add New button and select “Blueprint Class” to create a new Blueprint class.
2. In the Pick Parent Class window, choose “Actor” as the parent class. Name the Blueprint “BP\_Hello”.
3. Double-click on BP\_Hello to open the Blueprint Editor.
4. In the Event Graph, drag a wire from the BeginPlay event’s output execution pin and release the mouse button to open the context menu.
5. Search for “print” and choose “Print String”. Write the string to be displayed, such as “Hello Blueprints” (see Figure 1).
6. Press the Compile button.
7. Drag and drop the Blueprint from the Content Browser into the Viewport or World Outliner to add it to the Level.
8. Press the Play button in the Level Editor.

**Outcome**

After pressing the Play button, you should see your message displayed in the Viewport and in the Output Log window.

Graphical user interface, text, application

Description automatically generated

*Figure 1: Print String*

***Part 2:***

In this exercise, you will modify the Blueprint created in the previous exercise so that it uses a Text Render component and a variable to display a message. You will use the Components panel, Details panel, Viewport, Event Graph, and My Blueprint panel.

Directions

1. Double-click on BP\_Hello (the Blueprint created in the previous exercise) to open the Blueprint Editor.
2. In the Event Graph, remove the Print String action, but keep the BeginPlay event.
3. In the Components panel, click the Add Component button, search for “text”, and choose the Text Render component.
4. Select the Text Render component. In the Details panel, set the Z value of Location to “100” and change Text Render Color to a green color (see Figure 1).
5. In the My Blueprint panel, click the “+” button in the Variables category to add a new variable.
6. Select the variable created, and in the Details panel change Variable Name to “Message” and Variable Type to “Text”. Compile the Blueprint, and then under the Default Value category set the Message variable’s default value to “Hello”.
7. In the Event Graph, drag out the Text Render component and drop it near the BeginPlay event. Then drag and drop the Message variable and choose the Get option.
8. Drag a wire from the Text Render component and release the mouse button to open the context menu. Search for “text” and choose “Set Text”.
9. Connect the BeginPlay event’s output execution pin to the Set Text action’s input execution pin, and connect the Message variable to the Set Text action’s Value pin (see Figure 2).
10. Compile the Blueprint, and then drag and drop it from the Content Browser into the Viewport or World Outliner to add it to the Level.
11. Press the Play button in the Level Editor and move your character to where the Blueprint was placed.

**Outcome**

After pressing the Play button and moving your character to where the Blueprint was placed, you should see the message “Hello” as 3D text in the Level.

Graphical user interface

Description automatically generated

*Figure 1: Viewport*

A screenshot of a cell phone

Description automatically generated with low confidence

*Figure 2: Set Text*

|  |
| --- |
| IMAT 2103 – Progressive Game Engines |
| Lab 2B: Programming Concepts |

***Part 1:***

In this exercise, you will create a Blueprint with variables of various types that can be modified in the Level Editor.

## Directions

1. In the **Content Browser**, click the green **Add New** button and select “**Blueprint Class**” to create a new Blueprint class.
2. In the **Pick Parent Class** window, choose “**Actor**” as the parent class. Name the Blueprint “**BP\_Variables**”.
3. Double-click **BP\_Variables** to open the **Blueprint Editor**.
4. In the **My Blueprint** panel, click the “**+**” button in the **Variables** category to add a new variable.
5. Select the variable created, and in the **Details** panel change **Variable Name** to “**BooleanVar**” and **Variable Type** to “**Boolean**”. Check the **Instance Editable** property.
6. Create an **Integer** variable named “**IntegerVar**” and check **Instance Editable**.
7. Create a **Float** variable named “**FloatVar**” and check **Instance Editable**.
8. Create a **Text** variable named “**TextVar**” and check **Instance Editable**.
9. Create an **Integer** variable named “**ArrayInteger**” and set **Variable Type** to “**Integer**”. Click the button to the side of “**Variable Type**” and choose “**Array**”. Check **Instance Editable**.
10. See Figure 1, which shows the variables created.

Graphical user interface, text

Description automatically generated

*Figure 1: Exercise 1 Variables*

1. Press the **Compile** button. In the **Level Editor**, drag and drop the Blueprint into the Level.
2. Select the instance of the Blueprint in the Level and edit the values of the variables in the **Level Editor**’s **Details** panel (see Figure 2).

Graphical user interface, application

Description automatically generated

*Figure 2: Editable Variables*

## Outcome

After adding Actor instances of **BP\_Variables** to the Level, it is possible to set the values of their editable variables using the **Details** panel. Each Actor instance can have unique values.

***Part 2:***

In this exercise, you will create a function that calculates an attack value based on an input parameter. Operators and the Branch node will also be used.

## Directions

1. In the **Content Browser**, click the green **Add New** button and select “**Blueprint Class**” to create a new Blueprint class.
2. In the **Pick Parent Class** window, choose “**Actor**” as the parent class. Name the Blueprint “**BP\_Attack**”.
3. Double-click **BP\_Attack** to open the **Blueprint Editor**.
4. Create an **Integer** variable named “**Strength**” and check **Instance Editable**.
5. In the **Details** panel under the **Variable** category, set the **Value Range** property from “**1**” to “**10**”. Compile the Blueprint and then under the **Default Value** category set the variable’s default value to “**5**”.
6. In the **My Blueprint** panel, click the “**+**” button in the **Functions** category to create a function. Change the name of the function to “**Calculate Attack**”.
7. Use the **Details** panel for this function to create an input parameter named “**Strength Value**” of **Integer** type and an output parameter named “**Attack Value**” of **Integer** type.
8. The function will calculate the following expression:
9. Double-click the function to edit it and create the expression seen in Figure 1.

*Attack Value = (Strength Value x 5) + 10*

Graphical user interface, text, application

Description automatically generated

*Figure 1: Function Calculate Attack*

1. In the **Event Graph**, add a **BeginPlay** event and create the nodes seen in Figure 2. These actions will calculate the attack value based on the value of the **Strength** variable. If the attack value is equal to or greater than 40, then the message “STRONG ATTACK” will be printed on screen; if it is less than 40, then the message “WEAK ATTACK” will be printed.

A screenshot of a video game

Description automatically generated with medium confidence

*Figure 2: Event BeginPlay*

1. Compile the Blueprint. In the **Level Editor,** drag and drop the Blueprint into the Level.
2. Modify the value of the **Strength** variable using the **Details** panel in the **Level Editor**. Play the Level to see the message.

## Outcome

After pressing the **Play** button, you should see the message “STRONG ATTACK” or “WEAK ATTACK” displayed in the **Viewport**. The message displayed depends on the value of the **Strength** variable.

***Part 3:***

In this exercise, you will use random number functions to simulate the throw of two dice to calculate the value of an attack. The Append node will be used to create the resulting message that will be printed on screen.

## Directions

1. Create a new project or use an existing one.
2. Click the **Blueprints** button on the **Toolbar** and select “**Open Level Blueprint**”.
3. Create an **Integer** variable called “**Attack Value**”.
4. Add the **Enter** keyboard event.
5. Add the actions shown in Figure 1.

Graphical user interface, application

Description automatically generated

*Figure 1: Using Random Number Functions*

1. When the user presses the **Enter** key, two random integer values between “1” and “6” will be generated and added together. The resulting value will be set in the **Attack Value** variable. The **Append** function concatenates the message “ATTACK: ” with the value of the **Attack Value** variable.
2. Play the game and press the **Enter** key to see the attack value.

## Outcome

When playing the game and pressing the **Enter** key, you should see the message “ATTACK: ” displayed in the Viewport, with a number following the colon indicating the attack value.