|  |
| --- |
| IMAT 2103 – Progressive Game Engines |
| Lab 5: Unreal Motion Graphics |

# Part 1

# Overview

In this exercise, you will create an About screen and see how to switch between the Title screen and the About screen.

## Directions

1. Create a new project or use an existing one.
2. In the **Content Browser**, click the green **Add New** button and in the **User Interface** submenu select “**Widget Blueprint**”. Create two **Widget Blueprint** classes, one for the Title screen and another for the About screen. Name them “**TitleUMG**” and “**AboutUMG**”.
3. Double-click the **TitleUMG** Blueprint to open it in the **UMG Editor**.
4. Add an **Image** or **Text** widget for the Title screen to the **Designer** panel. Add a **Button** widget and place a **Text** widget inside it. Set the value of the **Text** widget’s **Text** property to “**About Screen**” and rename the **Button** widget “**Btn\_About**”. (See Figure 1.)

Graphical user interface, application

Description automatically generated

*Figure 1: About Screen Button*

1. Select **Btn\_About**, and in the **Details** panel expand the **Events** category and add an **OnClicked** event. In the **My Blueprints** panel, create a variable named “**AboutUMGVar**” of type “**AboutUMG**”. In the **Event Graph**, add the functions shown in Figure 2.

A picture containing text, indoor, electronics, iPod

Description automatically generated

*Figure 2: Show About Screen*

1. Double-click the **AboutUMG** Blueprint to open it in the **UMG Editor**.
2. Add an **Image** or **Text** widget for the About screen to the **Designer** panel. Add a **Button** widget and place a **Text** widget inside it. Set the value of the **Text** widget’s **Text** property to “**Return**” and rename the **Button** widget “**Btn\_Return**”.
3. Select **Btn\_Return**, and in the **Details** panel expand the **Events** category and add an **OnClicked** event. In the **My Blueprints** panel, create a variable named “**TitleUMGVar**” of type “**TitleUMG**”. In the **Event Graph**, add the functions shown in Figure 3.

A picture containing text, indoor

Description automatically generated

*Figure 3: Return to Title Screen*

1. In the **Level Blueprint**, create a variable named “**TitleUMGVar**” of type “**TitleUMG**” and a variable named “**AboutUMGVar**” of type “**AboutUMG**”. Add the functions shown in Figure 4. The graph will create the Widget Blueprint objects and save the references to the **TitleUMGVar** and **AboutUMGVar** variables in the **Level Blueprint**. The reference to **TitleUMGVar** is saved in the **AboutUMG** widget and the reference to **AboutUMGVar** is saved in the **TitleUMG** widget.

## Outcome

After pressing **Play**, the Title screen will be added to the Viewport. It contains a button named “**Btn\_About**” with a label that reads “About Screen”. When the **Btn\_About** button is pressed, the Title screen is hidden and the **About UMG** widget is added to the Viewport.

A screenshot of a video game

Description automatically generated

*Figure 4: Add Screens to Viewport*

# Part 2

# Overview

In this exercise, you will use a Widget component to display a UMG screen inside the game world. The Widget component will be used in the Player Character. When the player presses the **Enter** key, a status panel will appear in front of them.

## Directions

1. Create a new project using the **First Person** template.
2. In the **Content Browser**, click the green **Add New** button and in the **User Interface** submenu select “**Widget Blueprint**”. Rename it “**StatusUMG**”.
3. Double-click **StatusUMG** to open the **UMG Editor**.
4. In the **Fill Screen** drop-down menu, select “**Custom**” to change the screen size. Set **Width** to“**500**” and **Height** to“**500**” (see Figure 1).

Graphical user interface, application

Description automatically generated

*Figure 1: Widget Screen Size*

1. Add a **Text** widget to the **Designer** panel. Set the widget’s **Text** property to “**Status Panel**”.
2. Open the **FirstPersonCharacter** Blueprint (path: /Game/FirstPersonBP/Blueprints). In the **Components** panel, click the **Add Component** button and choose the **Widget** component. Rename it “**Widget3D**”. In the **Details** panel, set the **Widget Class** property to “**StatusUMG**” (see Figure 2). Set the **Location** property to “**400.0**, **0.0**, **0.0**” and the **Rotation** property to “**0.0**, **0.0**, **180.0**”.

Graphical user interface, application

Description automatically generated

*Figure 2: Widget Details*

1. In the **Event Graph**, add the nodes shown in Figure 3.

A picture containing text, electronics

Description automatically generated

*Figure 3: Toggle the Widget*

1. Play the Level and press the **Enter** key.

## Outcome

When playing the game and pressing the **Enter** key, the widget that represents a status panel will appear in the Level in front of the player. If the **Enter** key is pressed again, the status panel will be hidden.

# Part 3

# Overview

In this exercise, you will learn to create a player HUD using a Widget Blueprint class and the UMG Editor Designer mode.

# Outcome

Students will demonstrate their ability to create a player HUD using a Widget Blueprint and the UMG Editor Designer mode.

# Process

## Project Setup

For this exercise, you need to complete and work from the “CharacterHealth” project associated with the exercises in the following files:

* **TeachYourselfUE4\_Hour\_16\_Lecture1\_Exercise2\_HealthPickup**
* **TeachYourselfUE4\_Hour\_16\_Lecture1\_Exercise4\_MessageActor**

You also need to have completed the exercise in **TeachYourselfUE4\_Hour\_02\_Lecture2\_Exercise2\_ImportingUIAssets**.

## Open the project.

1. Open the “CharacterHealth” project.

## Create a Widget Blueprint class asset for the player HUD.

1. In the Content Browser, create a new folder called “InterfaceWidgets” (if it has not already been added from a previous exercise).
2. Right-click on the “InterfaceWidgets” folder, and then go to *Create Advanced Asset > User Interface* and select “Widget Blueprint”.
3. Rename the Widget Blueprint asset “HUDWidget”.
4. Double-click on “HUDWidget” to open the UMG Widget Blueprint Editor.
5. In the Designer panel, go to the *Screen Size* drop-down and under “Monitors” select “21.5-24″ monitor”.

## Place the Text widget for health.

1. From the Palette panel under “Common”, drag out a Text widget and place it in the Designer panel.
2. With the placed Text widget selected, go to the UMG Details panel and in the text box at the top rename the widget “HealthText”.
3. In the Details panel under “Slot (Canvas Panel Slot)”, set the following properties:
   * Anchors: Lower right corner
   * Position X: –1884.0
   * Position Y: –1032.0
   * Size X: 108.0
   * Size Y: 40.0
   * Size To Content: True
4. In the Details panel under “Content”, type “Health:” in the *Text* property text box.
5. In the Details panel under “Appearance”, select the “Organo\_Font” asset from the *Font* property drop-down.
6. Compile and save the Widget Blueprint.

## Place the Progress Bar widget for health.

1. From the Palette panel under “Common”, drag out a Progress Bar widget and place it in the Designer panel.
2. With the placed Progress Bar widget selected, go to the UMG Details panel and in the text box at the top rename the widget “HealthBar”.
3. In the Details panel under “Slot (Canvas Panel Slot)”, set the following properties:

* Anchors: Lower right corner
* Position X: –1736.0
* Position Y: –1036.0
* Size X: 800.0
* Size Y: 40.0
* Size To Content: False

1. In the Details panel under “Progress”, set the *Percent* property to “0.75”.
2. Compile and save the Widget Blueprint.

## Place the Text widget for message.

1. From the Palette panel under “Common”, drag out a Text widget and place it in the Designer panel.
2. With the placed Text widget selected, go to the UMG Details panel and in the text box at the top rename the widget “MessageText”.
3. In the Details panel under “Slot (Canvas Panel Slot)”, set the following properties:
   * Anchor: Center
   * Position X: –400.0
   * Position Y: –100.0
   * Size X: 800.0
   * Size Y: 200.0
   * Size To Content: False
4. In the Details panel under “Content”, type “The Message” in the *Text* property text box.
5. In the Details panel under “Appearance”, select the “Organo\_Font” asset from the *Font* property drop-down.
6. Compile and save the Widget Blueprint.

## Place a Text widget for pickup count.

1. From the Palette panel under “Common”, drag out a Text widget and place it in the Designer panel.
2. With the placed Text widget selected, go to the UMG Details panel and in the text box at the top rename the widget “PickupLabelText”.
3. In the Details panel under “Slot (Canvas Panel Slot)”, set the following properties:
   * Anchor: Lower right corner
   * Position X: –1885.0
   * Position Y: –100.0
   * Size X: 151.0
   * Size Y: 40.0
   * Size To Content: True
4. In the Details panel under “Content”, type “Pickup Count” in the *Text* property text box.
5. In the Details panel under “Appearance”, select the “Organo\_Font” asset from the *Font* property drop-down.
6. Compile and save the Widget Blueprint.

## Place a second Text widget for pickup count.

1. From the Palette panel under “Common”, drag out a Text widget and place it in the Designer panel.
2. With the placed Text widget selected, go to the UMG Details panel and in the text box at the top rename the widget “PickupCount”.
3. In the Details panel under “Slot (Canvas Panel Slot)”, set the following properties:
   * Anchor: Lower right corner
   * Position X: –1780.0
   * Position Y: –85.0
   * Size X: 151.0
   * Size Y: 40.0
   * Size To Content: True
4. In the Details panel under “Content”, type “0” in the *Text* property text box.
5. In the Details panel under “Appearance”, select the “Organo\_Font” asset from the *Font* property drop-down.
6. Compile and save the Widget Blueprint.

When finished, your HUD should look something like this:A screenshot of a computer

Description automatically generated with medium confidence

# Submission Requirements

Time for completion: 10–15 minutes

* Lecturer observation and posting on the discussion board