

User Interfaces

7th Week, 2021



UNREAL
ENGINE



User Interfaces

- › The game UI is one of the main ways to show information to the player.
 - This is shown to the player mostly in the form of images and text.
- › UIs usually added on top of the rendering of the game.
 - They are in front of everything else you see in the game and behave as layers.
- › *Diegetic UI*
 - This type of UI isn't layered onto the game's screen, but rather exists inside of the game itself.



Game UI (1)

- › Two different types of game UI: **menus** and **HUDs**
- › Menus
 - Main menus
 - Level select menus
 - And many other options
- › HUDs (Heads-Up Displays)
 - UI panels that are present during game playing that give the player information that they should always know
 - Ex) how many lives they have left, which special abilities they can use, and so on



Game UI (2)

› **Unreal Motion Graphics (UMG)**

- The main way to create a game UI
- The tool to make a game UI (also called Widgets in UE4 terms) featuring menus and HUDs, and add them to screen

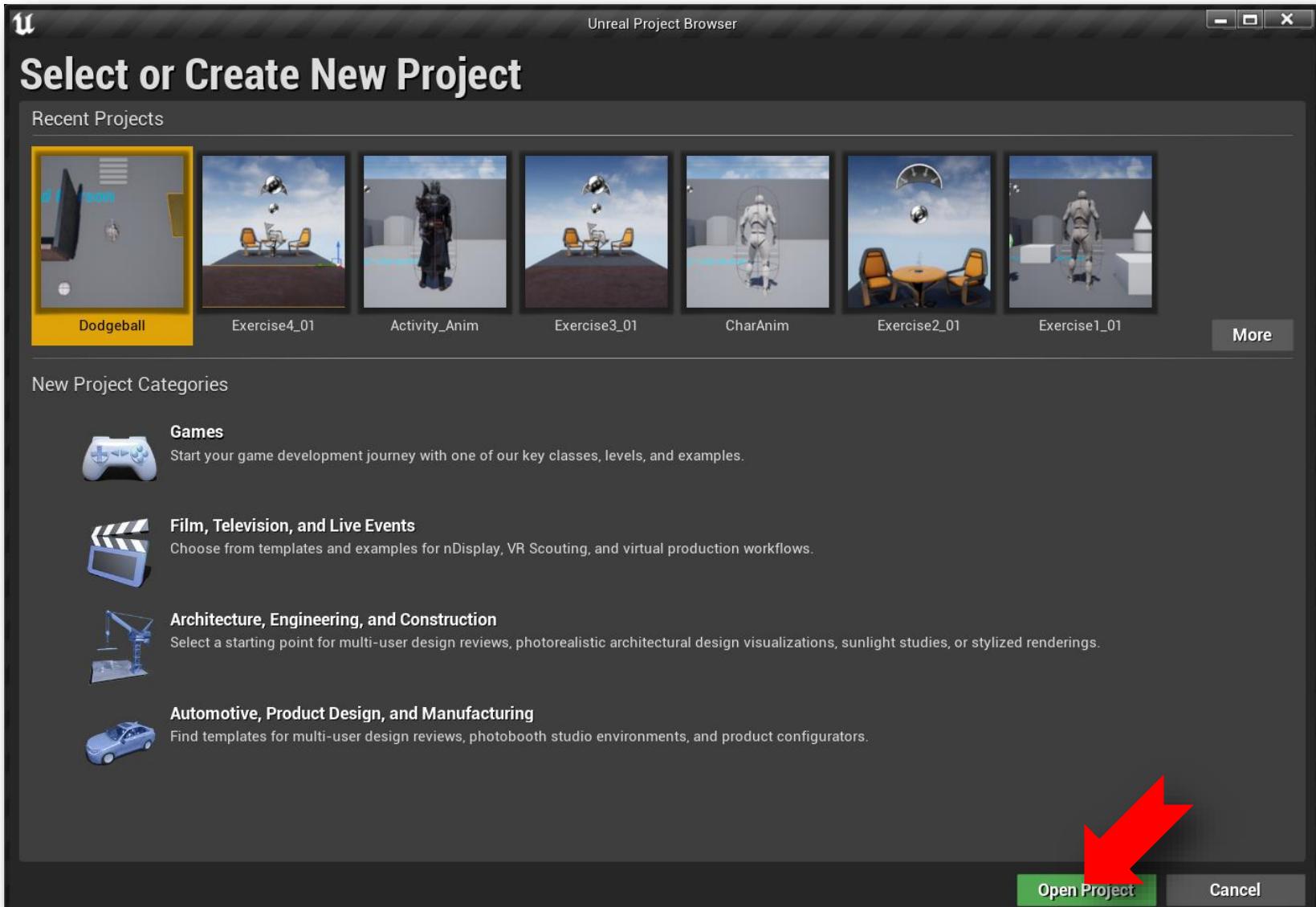


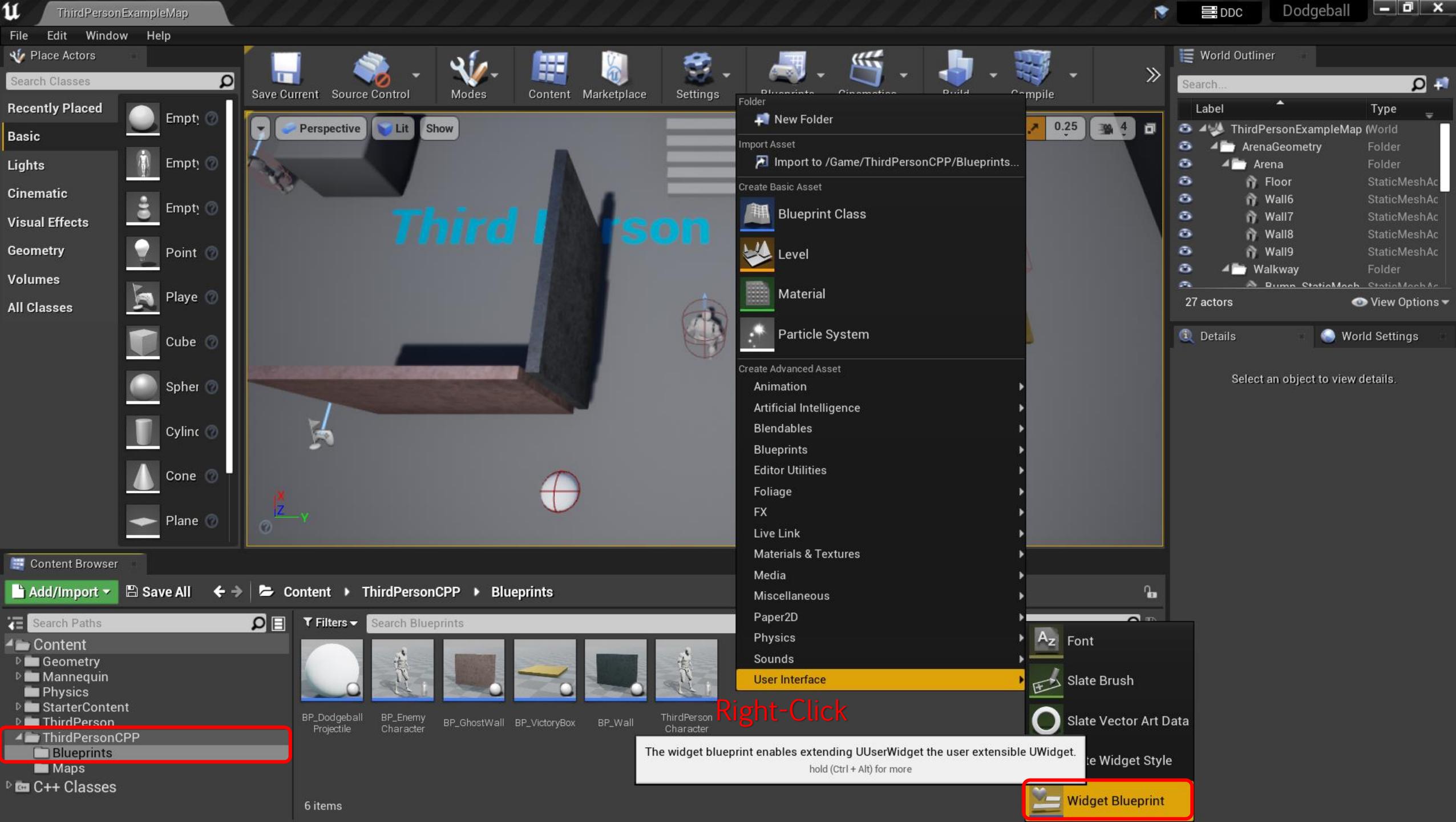
UMG Basics

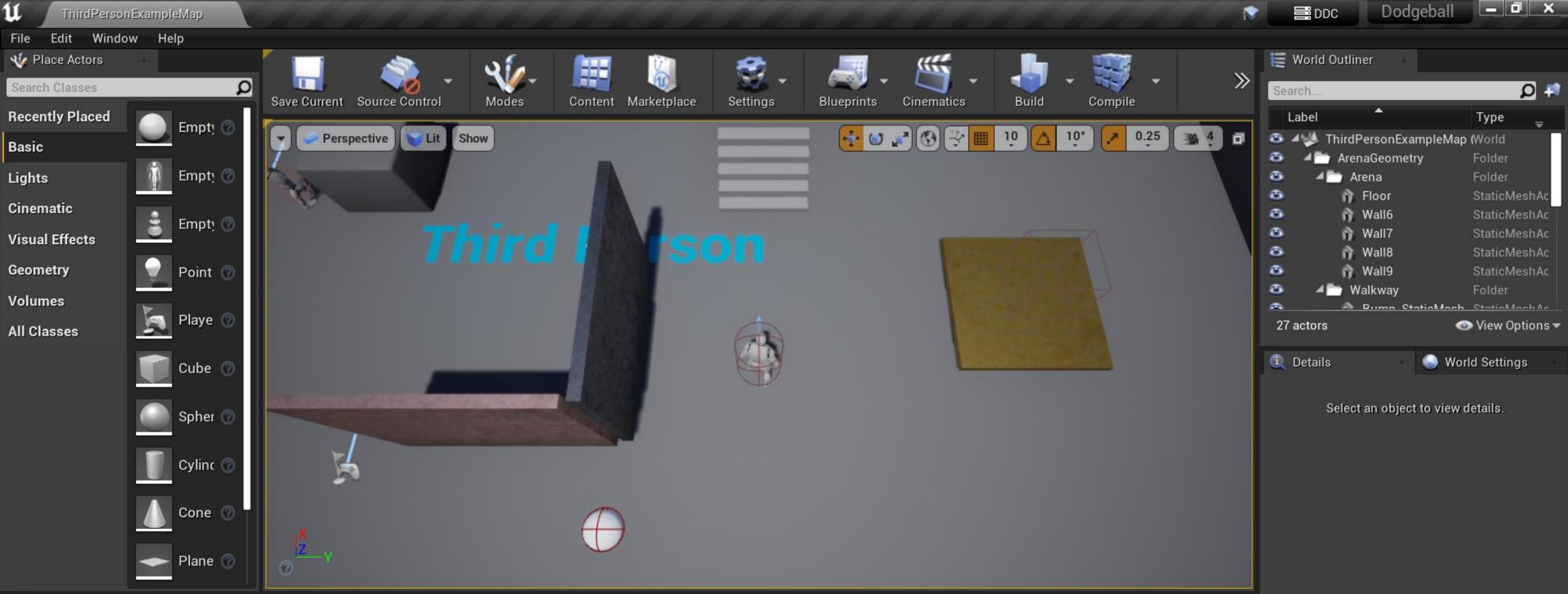
- › The main way to create a game UI is by using the UMG tool.
 - The UMG tool will allow you to make game UI in the form of **Widgets** and to easily edit your game UI in a visual manner, through **Designer** tab.
 - It will allow you to add functionality to your game UI through UMG's **Graph** tab.
- › **Widgets**
 - Basic UI elements such as Buttons, Text, and Images
 - Combined to create more complex and complete Widgets, such as menus and HUDs



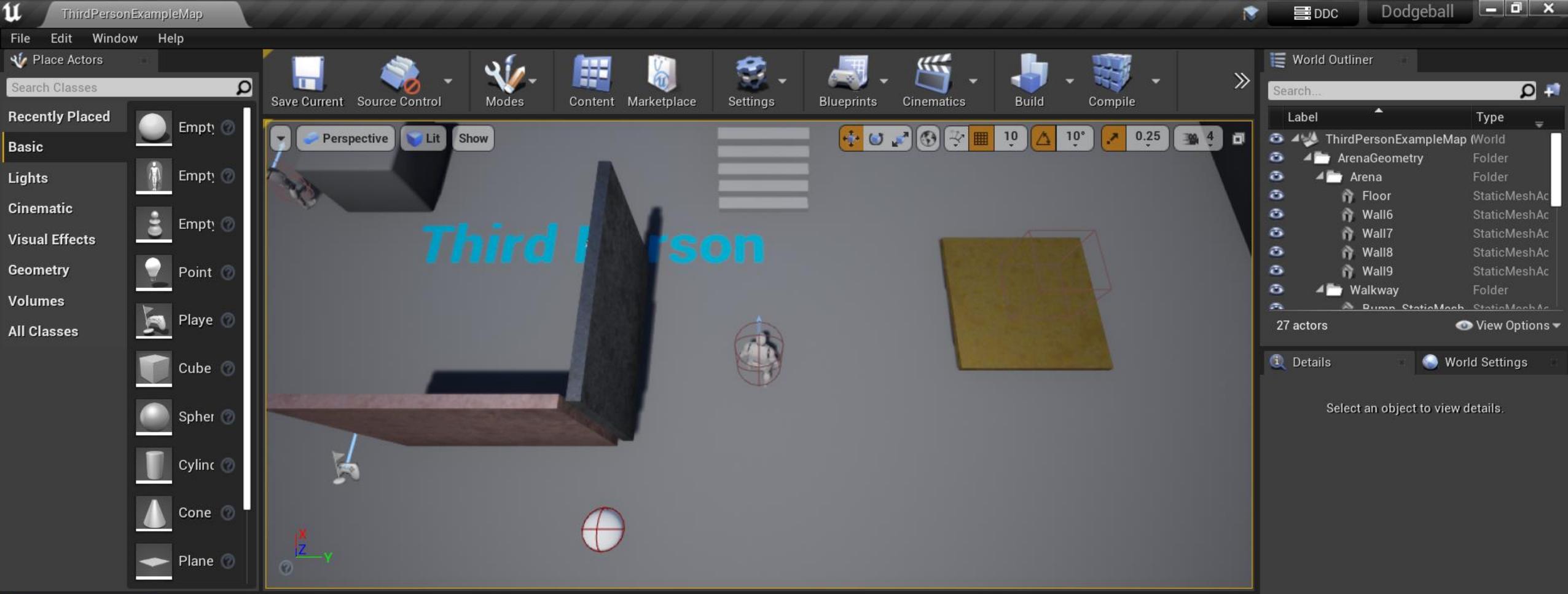
Exercise 8.01: Creating a Widget Blueprint







The Content Browser is open, showing the "Blueprints" category under "ThirdPersonCPP". The list includes "BP_Dodgeball Projectile", "BP_Enemy Character", "BP_GhostWall", "BP_VictoryBox", "BP_Wall", "idgetBlueprint" (selected and highlighted with a red dashed box), and "ThirdPerson Character". The left sidebar shows the project structure with "Content", "Geometry", "Mannequin", "Physics", "StarterContent", "ThirdPerson", and "ThirdPersonCPP" expanded. The bottom status bar indicates "7 items (1 selected)" and "View Options".

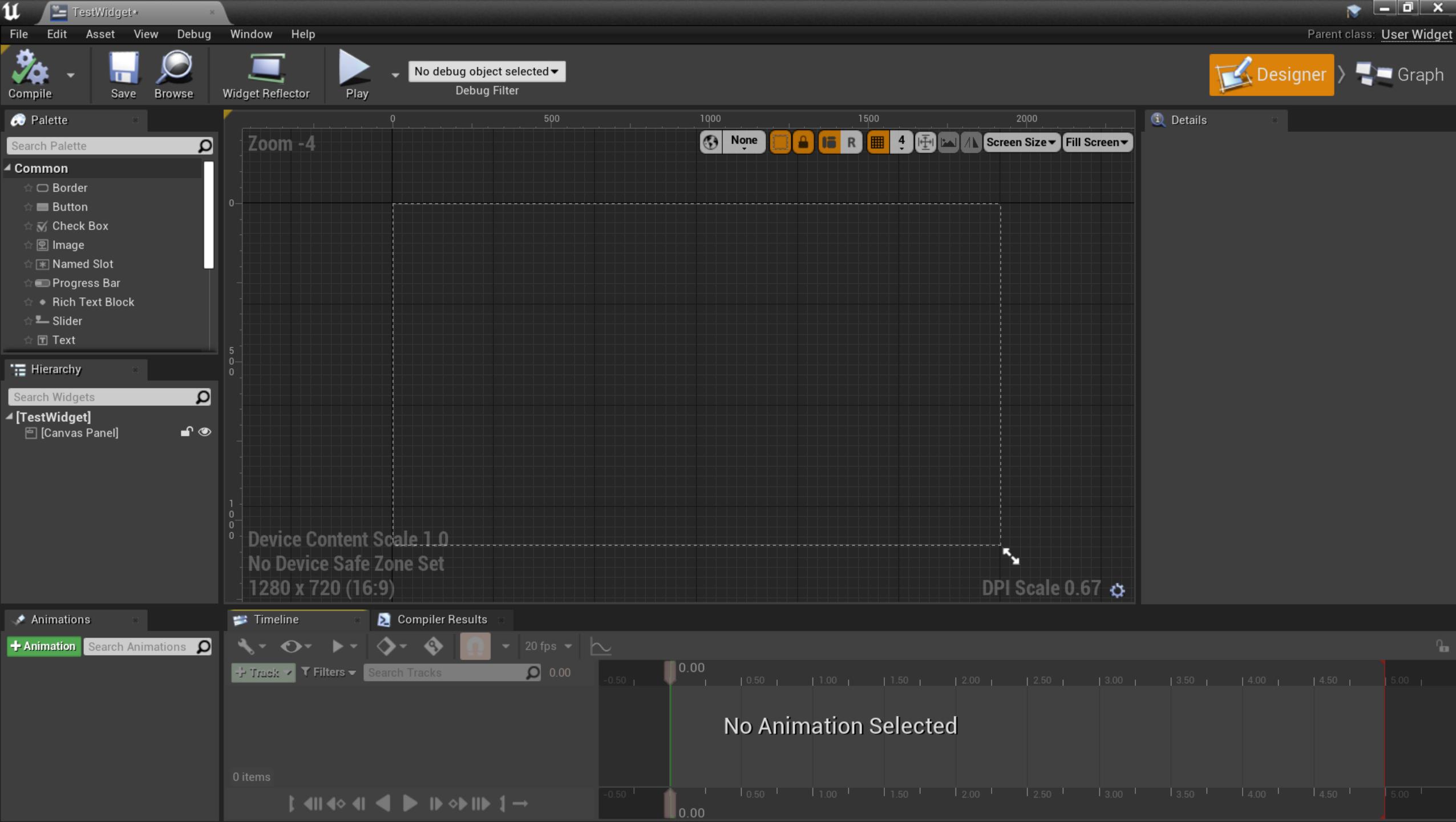


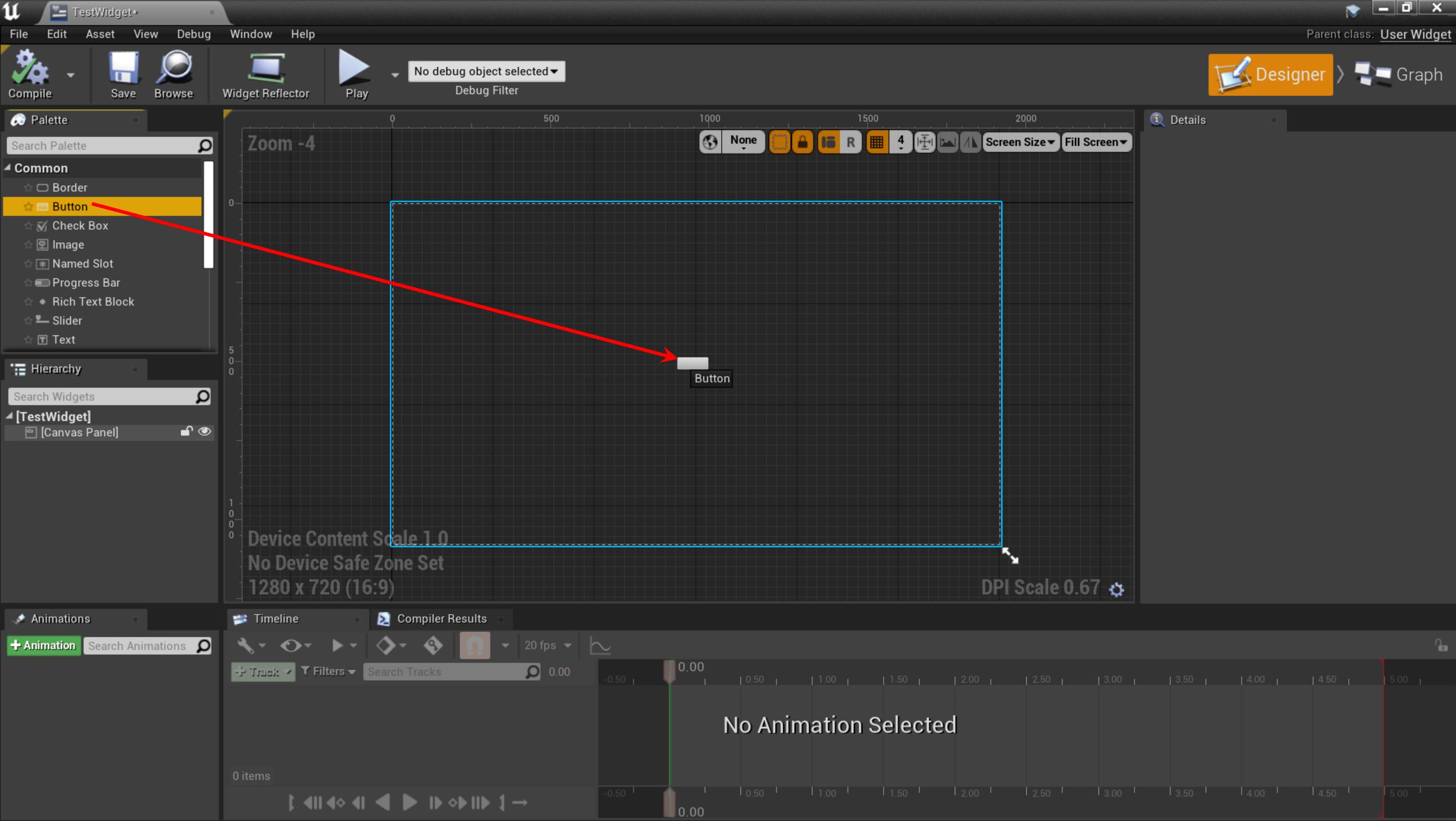
The screenshot shows the Content Browser interface. The top bar includes buttons for Add/Import, Save All, and navigation. The main area is titled "Content" and shows the "Blueprints" category. A search bar labeled "Search Blueprint" is present.

The Content Browser lists the following items:

- BP_DodgeballProjectile
- BP_EnemyCharacter
- BP_GhostWall
- BP_VictoryBox
- BP_Wall
- TestWidget (highlighted with a red box)
- ThirdPersonCharacter

The bottom left corner shows the total count: "7 items (1 selected)".





TestWidget* Parent class: User Widget

File Edit Asset View Debug Window Help

Compile Save Browse Widget Reflector Play No debug object selected Debug Filter

Designer Graph

Palette Search Palette

Common
Border
Button
Check Box
Image
Named Slot
Progress Bar
Rich Text Block
Slider
Text

Hierarchy Search Widgets

[TestWidget]
[Canvas Panel]
Button_0

Device Content Scale 1.0
No Device Safe Zone Set
1280 x 720 (16:9)

DPI Scale 0.67

Animations + Animation Search Animations

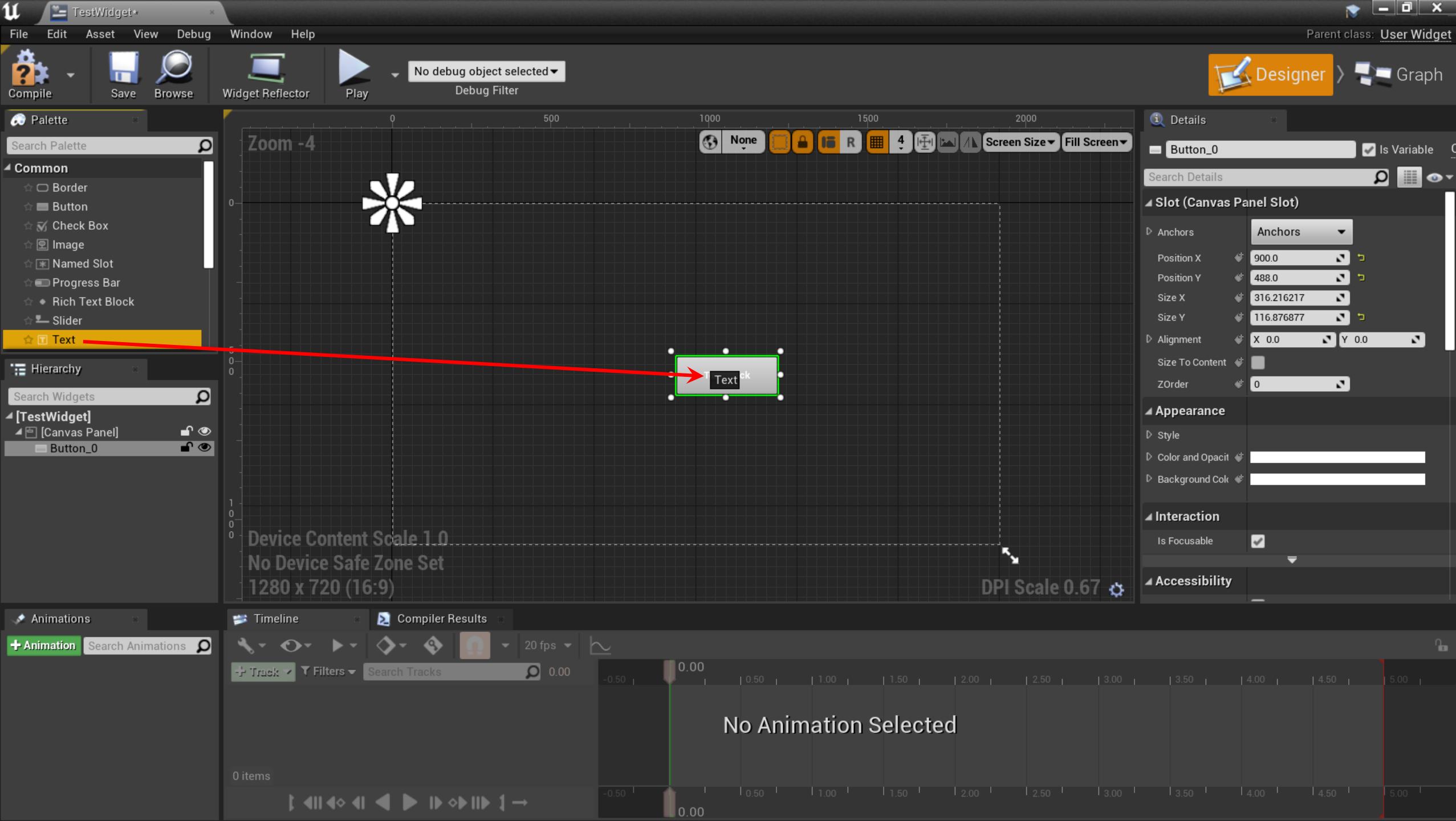
Timeline Compiler Results
Track Filters Search Tracks 0.00 20 fps

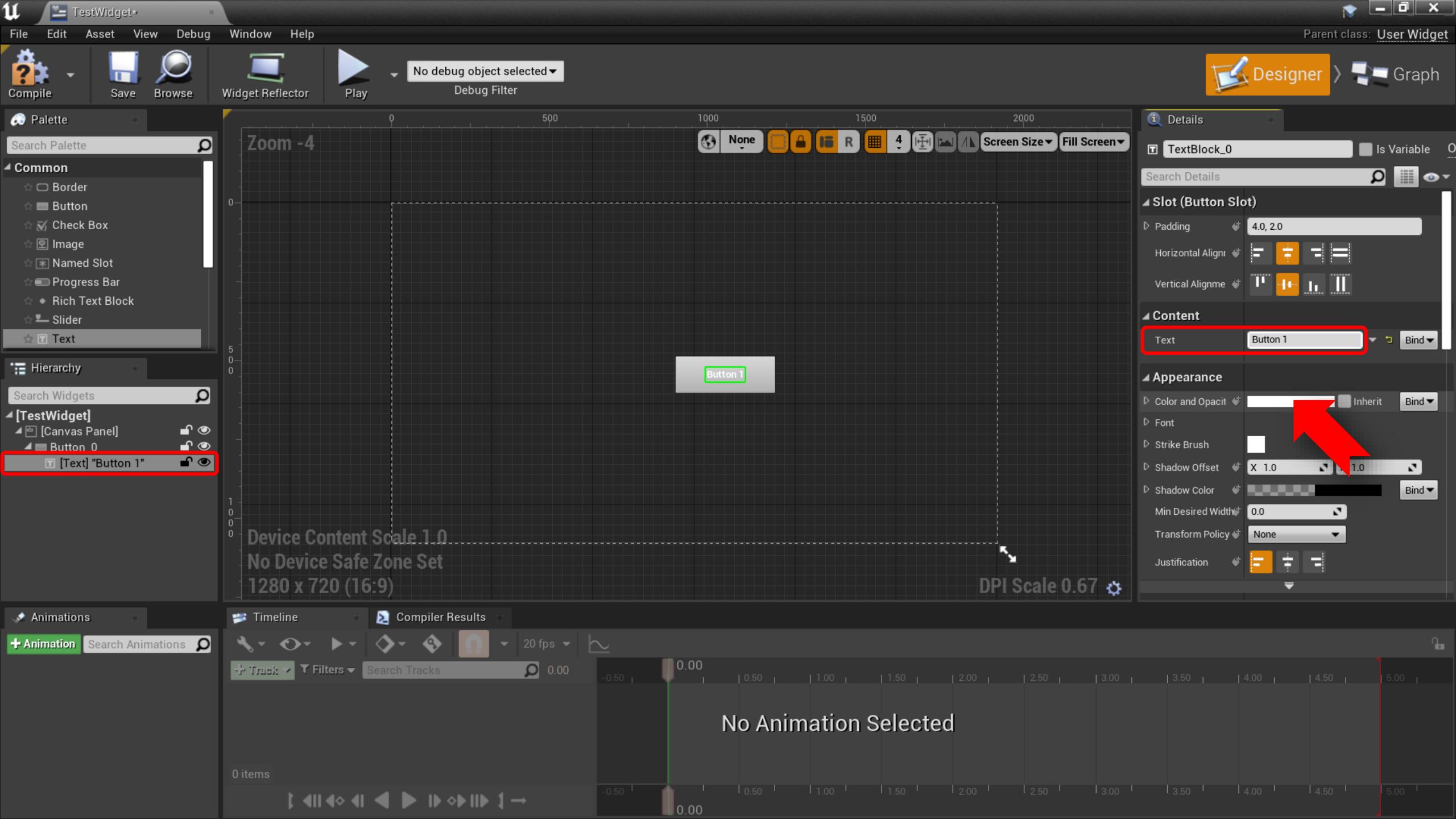
No Animation Selected

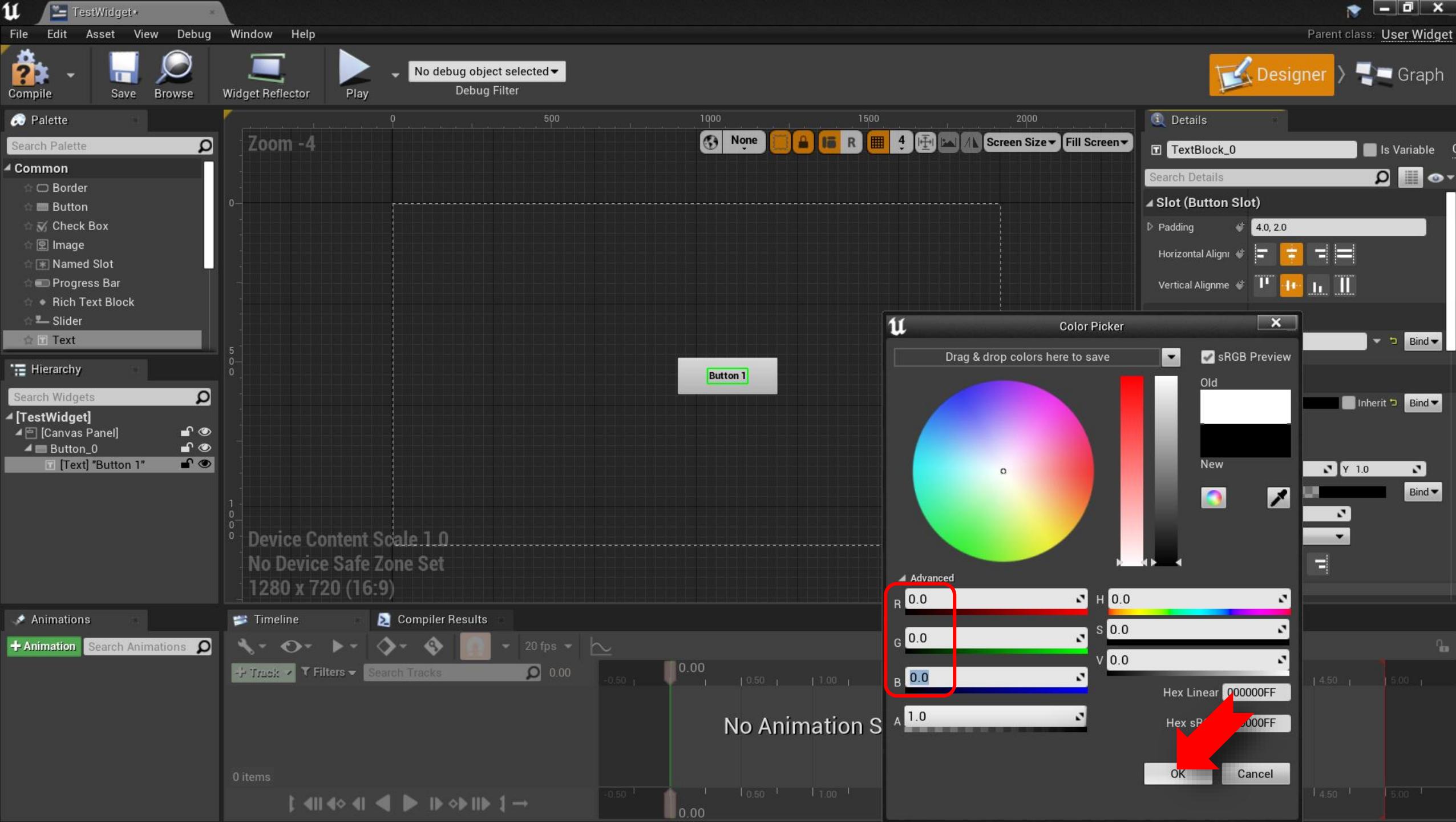
0 items

0.00 0.00

The screenshot shows the Unreal Engine Designer interface. The main workspace displays a canvas panel containing a button. The button is highlighted with a green selection border. A red arrow points from the center of the button towards the bottom right corner of the canvas panel. The canvas panel has a blue selection border and is positioned within a larger dashed blue frame representing the device content scale. The interface includes various toolbars and panels on the left and right sides, such as the Palette, Hierarchy, and Details panel, which shows the properties of the selected button.







Compile Save Browse Widget Reflector Play No debug object selected Debug Filter

Zoom -4 0 500 1000 1500 2000 Screen Size Fill Screen

Device Content Scale 1.0
No Device Safe Zone Set
1280 x 720 (16:9)

DPI Scale 0.67

Details TextBlock_0 Is Variable

Slot (Button Slot)
Padding: 4.0, 2.0
Horizontal Align: Center
Vertical Align: Top
Content Text: Button 1
Bind

Appearance
Color and Opacity: Inherit Bind
Font
Strike Brush: White
Shadow Offset: X: 1.0, Y: 1.0
Shadow Color: Bind
Min Desired Width: 0.0
Transform Policy: None
Justification: Center

Animations Animation Search Animations

Timeline Compiler Results 20 fps 0.00 0 items

No Animation Selected

The screenshot displays the Unreal Engine 4 Designer interface. The main workspace shows a button component named "Button 1" on a canvas grid. The properties panel on the right provides detailed configuration for the button's slot and appearance. The timeline at the bottom indicates no animations are currently selected. The left sidebar includes the Hierarchy and Animations panels.



Anchors

- › The menus can adapt to all different screen sizes with many different resolutions.
- › **Anchors**
 - To specify how you want a UI element's size to adapt as the screen resolution changes by specifying the proportion of the screen
 - As the size of the screen or resolution changes, your Widget will scale and move relative to its Anchor.
- › Only elements that are direct children of a **Canvas Panel** can have an **Anchor**, which you can visualize through the **Anchor Medallion**.



Exercise 8.02: Editing UMG Anchors

Screenshot of the Unreal Engine Designer mode interface.

Toolbar: File, Edit, Asset, View, Debug, Window, Help, Compile, Save, Browse, Widget Reflector, Play, No debug object selected, Debug Filter.

Designer View: Shows a grid with a red dashed circle centered at (0,0) and a green-bordered button labeled "Button 1" located at approximately (900, 488).

Details Panel: Shows properties for "Button_0". A red arrow points to the "Anchors" dropdown menu.

Property	Value
Position X	900.0
Position Y	488.0
Size X	316.216217
Size Y	116.876877
Alignment	X: 0.0, Y: 0.0
Size To Content	False
ZOrder	0

Hierarchy Panel: Shows the widget hierarchy: [TestWidget] > [Canvas Panel] > Button_0. The "Button_0" node is highlighted with a red border.

Device Content Scale: 1.0
No Device Safe Zone Set
1280 x 720 (16:9)

DPI Scale: 0.67

Bottom Navigation: Animations, Timeline, Compiler Results, Animation (green bar), Search Animations, 20 fps, 2000 ms, 200 ms.



Zoom -4

No debug object selected ▾

0 500 1000 1500 2000

None Screen Size Fill Screen

Common

- Border
- Button
- Check Box
- Image
- Named Slot
- Progress Bar
- Rich Text Block
- Slider
- Text**

Hierarchy

[TestWidget]
└ [Canvas Panel]
 └ Button_0
 └ [Text] "Button 1"

Device Content Scale 1.0
No Device Safe Zone Set
1280 x 720 (16:9)

DPI Scale 0.67

Details

Button_0 Is Variable

Slot (Canvas Panel Slot)

Anchors Anchors

Position	Position	Position	Position
Size X	Size Y	Size X	Size Y
Alignment	Size To	Alignment	Size To
ZOrder		ZOrder	

Appears

Style Color and Background

Interactions

Hold Shift to update the alignment to match.
Hold Ctrl to update the position to match.

Accessibility

Animations

Animation Search Animations

Timeline Compiler Results

20 fps

0.00

0.00

No Animation Selected

0 items

Animations

+ Animation Search Animations

Timeline Compiler Results

20 fps

0.00

0.00

No Animation Selected

0 items

TestWidget

File Edit Asset View Debug Window Help Parent class: User Widget

Compile Save Browse Widget Reflector Play No debug object selected Debug Filter

Zoom -4 2,093 x 823

Device Content Scale 1.0
No Device Safe Zone Set
1280 x 720 (16:9)

DPI Scale 0.67

Animations + Animation Search Animations

Timeline Compiler Results Track Filters Search Tracks 0.00 20 fps

No Animation Selected

0 items

0.00 0.00

Common

- Border
- Button
- Check Box
- Image
- Named Slot
- Progress Bar
- Rich Text Block
- Slider
- Text**

Hierarchy

- [TestWidget]
- [Canvas Panel]
- Button_0**
- [Text] "Button 1"

Details

Slot (Canvas Panel Slot)

Anchors	Anchors
Position X	-60.960938
Position Y	-52.540527
Size X	316.216217
Size Y	116.876877
Alignment	X 0.0 Y 0.0
Size To Content	
ZOrder	0

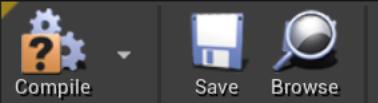
Appearance

Interaction

Accessibility

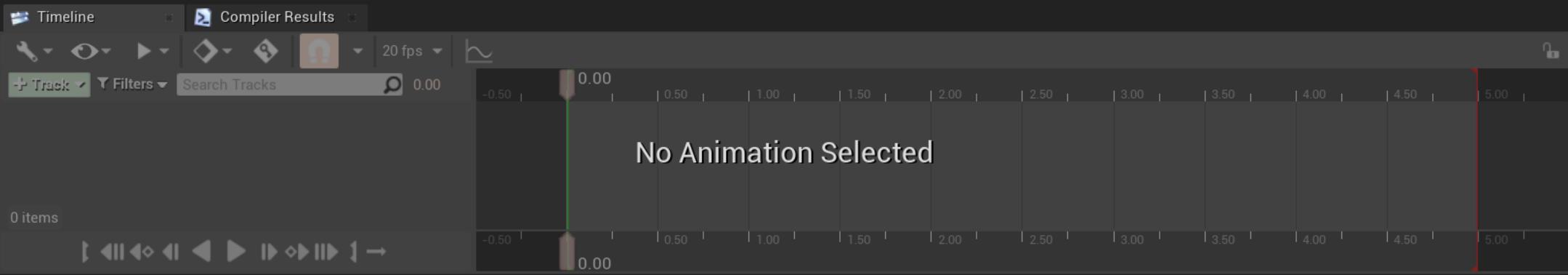
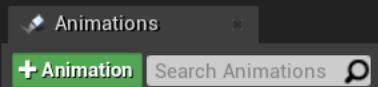
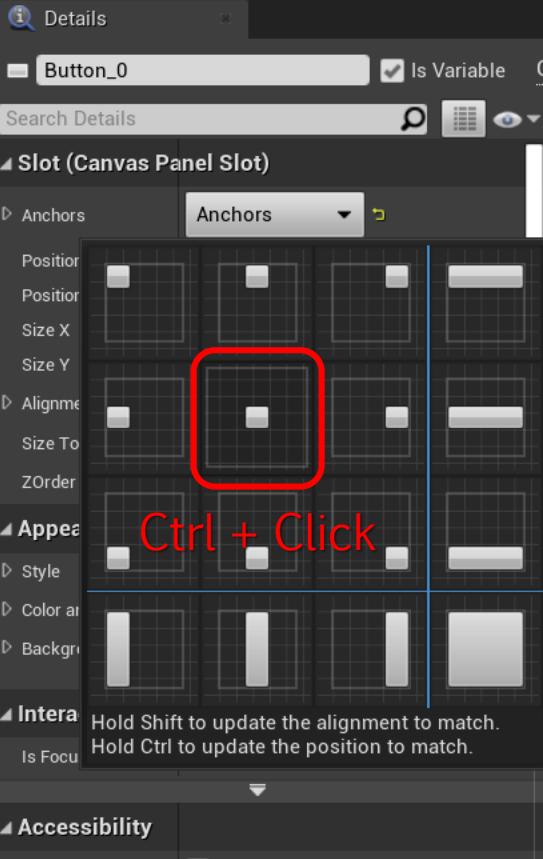
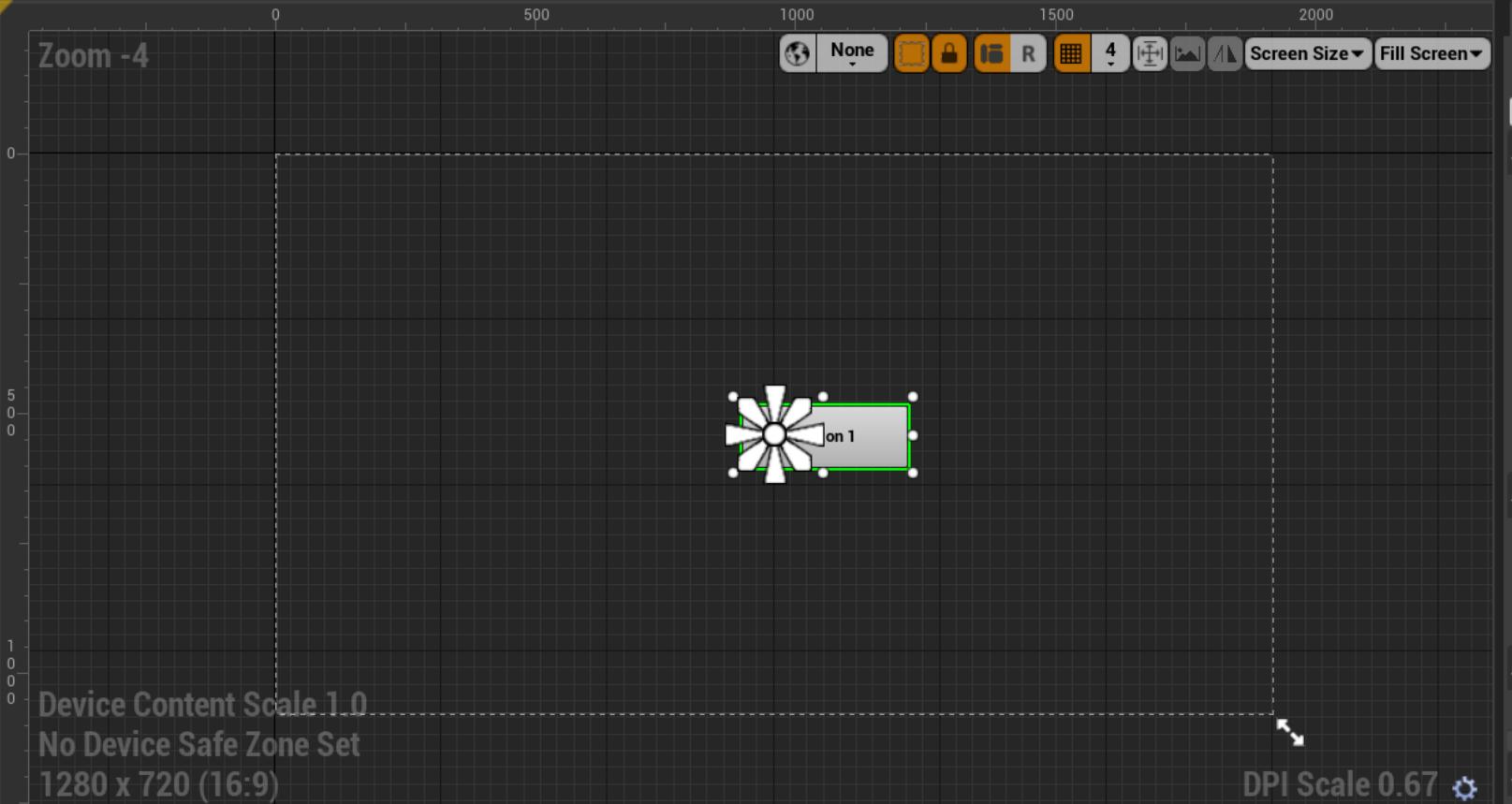
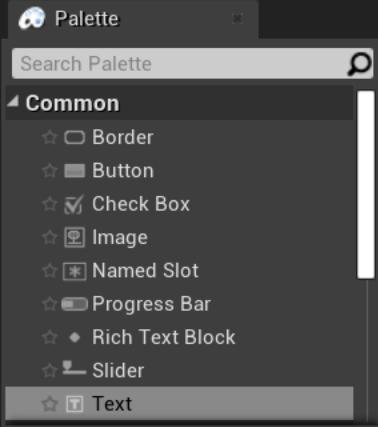
0 items

0.00 0.00



No debug object selected ▾ Debug Filter

Designer Graph



TestWidget

File Edit Asset View Debug Window Help Parent class: User Widget

Compile Save Browse Widget Reflector Play No debug object selected Debug Filter

Zoom -4 2,107 x 539

Device Content Scale 1.0
No Device Safe Zone Set
1280 x 720 (16:9)

DPI Scale 0.67

Animations + Animation Search Animations

Timeline Compiler Results Track Filters Search Tracks 0.00 20 fps

No Animation Selected

0 items

0.00 0.00

0 500 1000 1500 2000

0 5 10 15 20 25 30 35 40 45 50

0.00 0.50 1.00 1.50 2.00 2.50 3.00 3.50 4.00 4.50 5.00

0.00 0.50 1.00 1.50 2.00 2.50 3.00 3.50 4.00 4.50 5.00

Common Border Button Check Box Image Named Slot Progress Bar Rich Text Block Slider Text

Search Palette

[TestWidget] [Canvas Panel] [Button_0] [Text "Button 1"]

Details

Slot (Canvas Panel Slot)

- Anchors
 - Position X: 0.0
 - Position Y: 0.0
 - Size X: 316.216217
 - Size Y: 116.876877
 - Alignment: X 0.0 Y 0.0
- Size To Content
- ZOrder: 0

Appearance

- Style
- Color and Opacity
- Background Color

Interaction

- Is Focused:

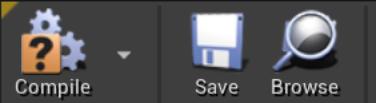
Accessibility

0 500 1000 1500 2000

0 5 10 15 20 25 30 35 40 45 50

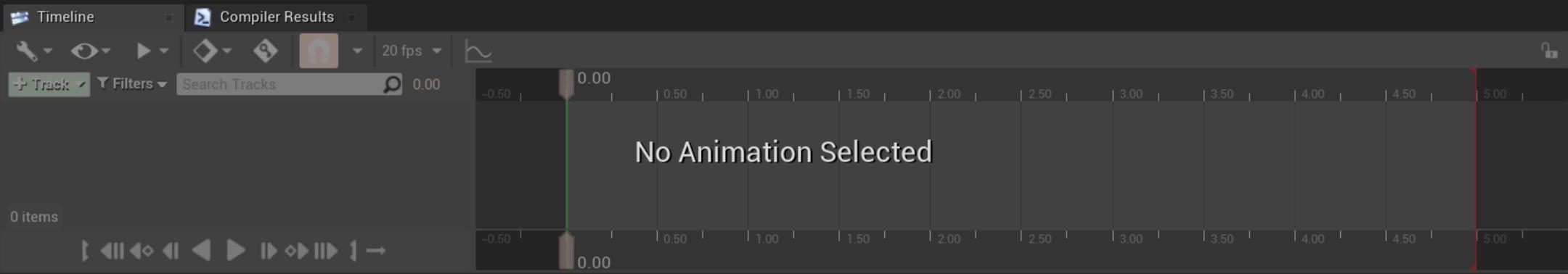
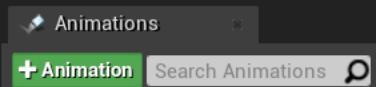
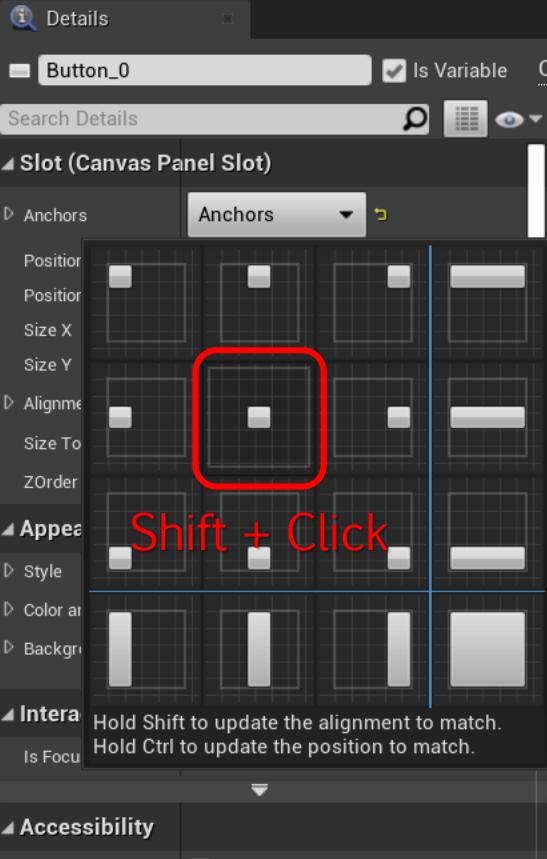
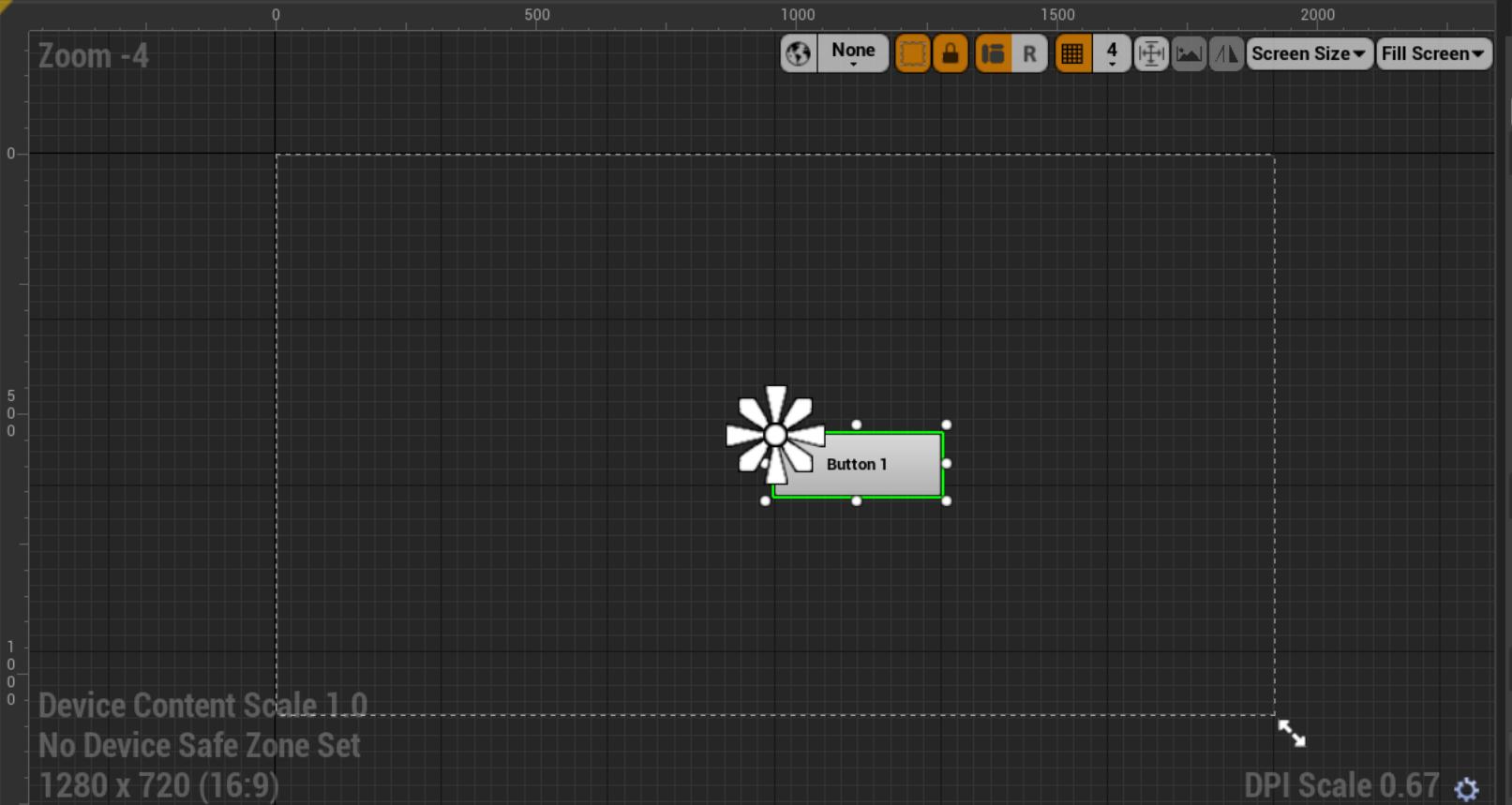
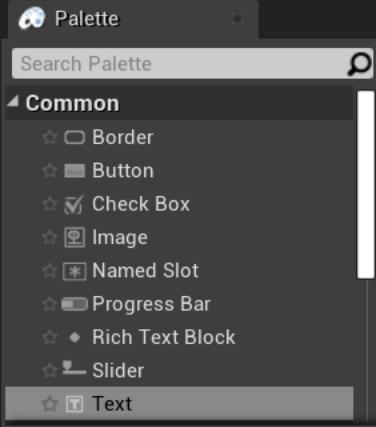
0.00 0.50 1.00 1.50 2.00 2.50 3.00 3.50 4.00 4.50 5.00

0.00 0.50 1.00 1.50 2.00 2.50 3.00 3.50 4.00 4.50 5.00



No debug object selected
Debug Filter

Designer Graph



TestWidget

File Edit Asset View Debug Window Help Parent class: User Widget

Compile Save Browse Widget Reflector Play No debug object selected Debug Filter

Zoom -4 2,112 x 551

0 500 1000 1500 2000

None Screen Size Fill Screen

Search Palette

Common Border Button Check Box Image Named Slot Progress Bar Rich Text Block Slider Text

Hierarchy [TestWidget] [Canvas Panel] Button_0 [Text] "Button 1"

Device Content Scale 1.0
No Device Safe Zone Set
1280 x 720 (16:9)

DPI Scale 0.67

Animations + Animation Search Animations

Timeline Compiler Results

20 fps

0.00

No Animation Selected

0 items

0.00 0.50 1.00 1.50 2.00 2.50 3.00 3.50 4.00 4.50 5.00

0.00 0.50 1.00 1.50 2.00 2.50 3.00 3.50 4.00 4.50 5.00

Designer Graph

Details

Button_0 Is Variable

Slot (Canvas Panel Slot)

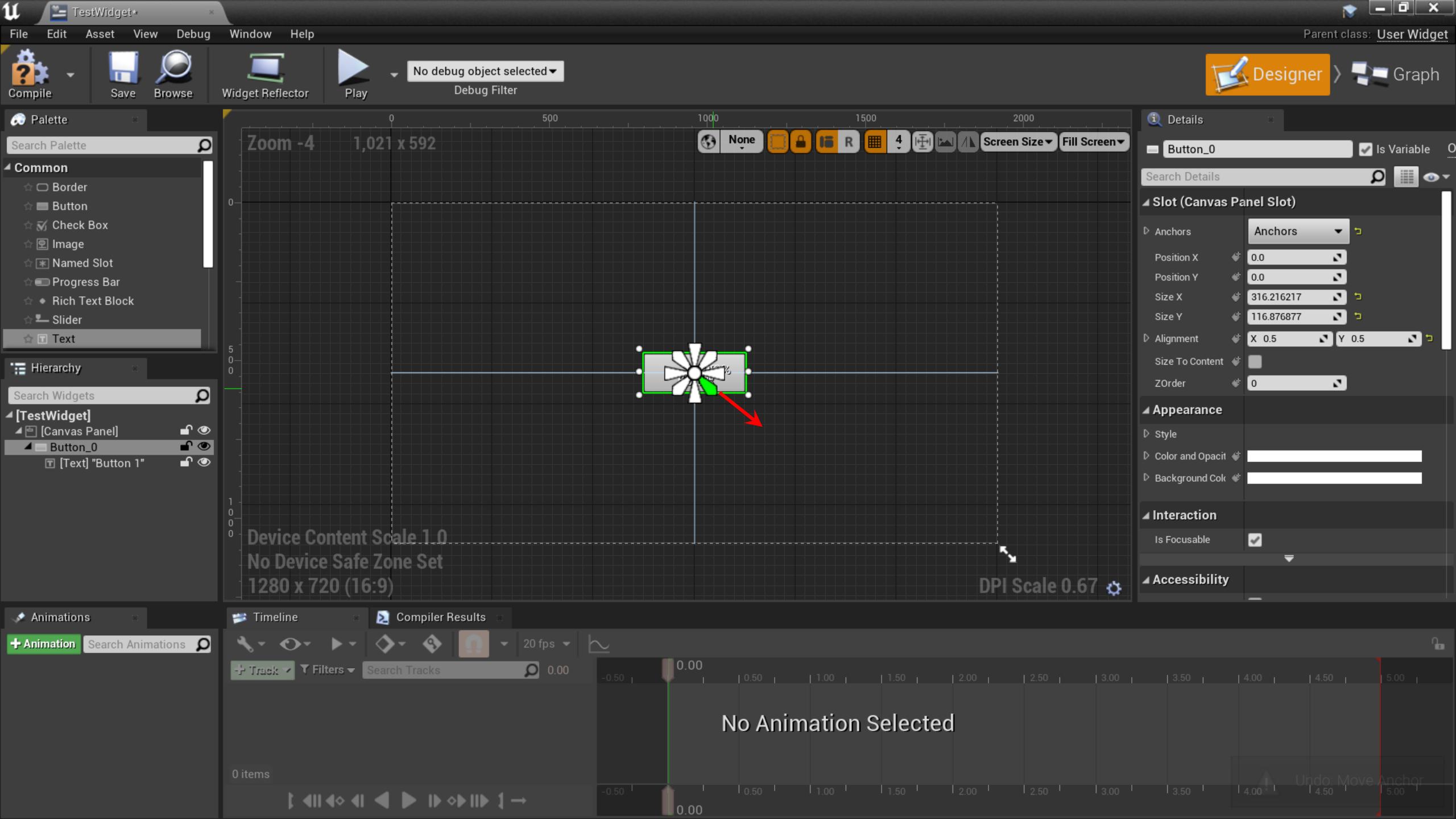
Anchors Position X: 0.0 Position Y: 0.0 Size X: 316.216217 Size Y: 116.876877 Alignment: X: 0.5 Y: 0.5

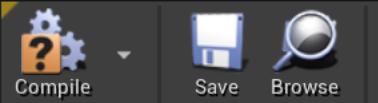
Appearance Style Color and Opacity Background Color

Interaction Is Focused

Accessibility

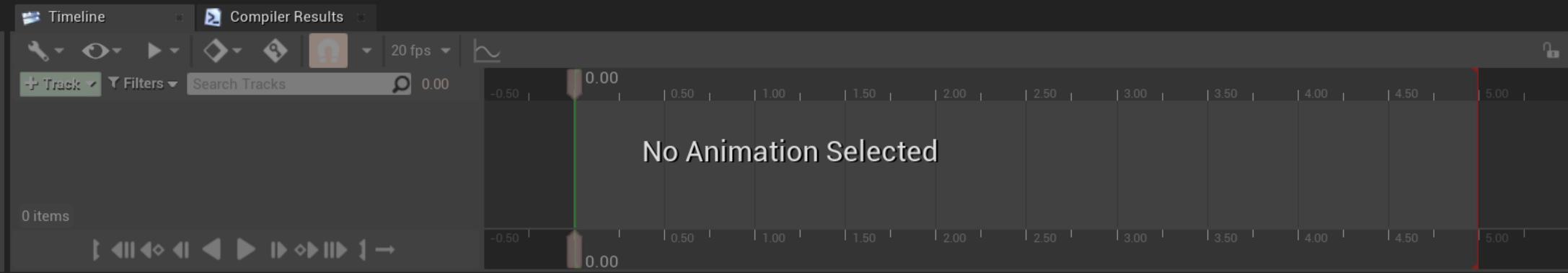
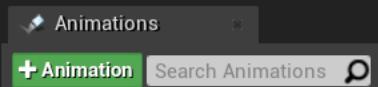
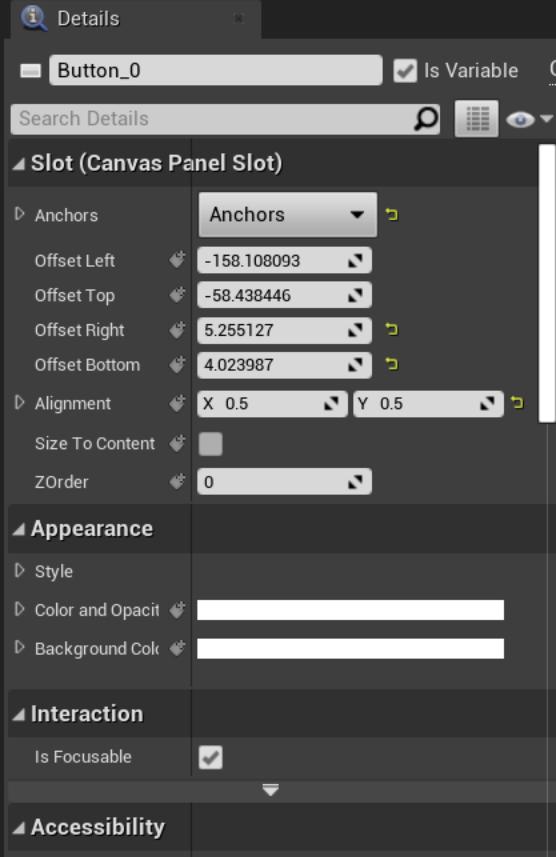
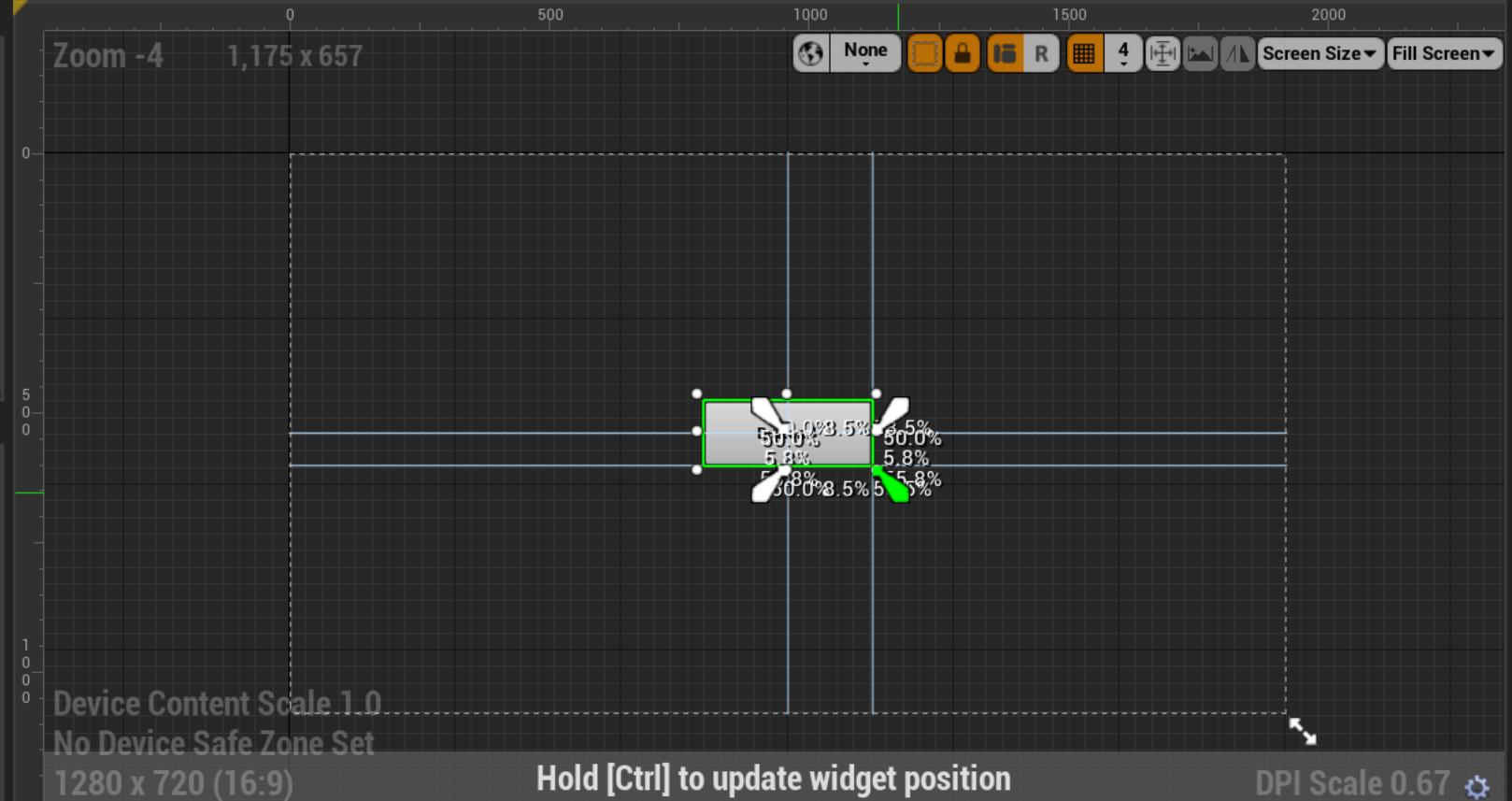
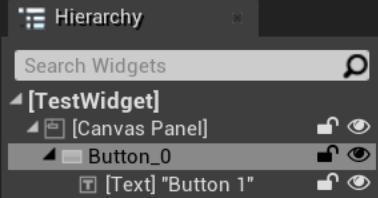
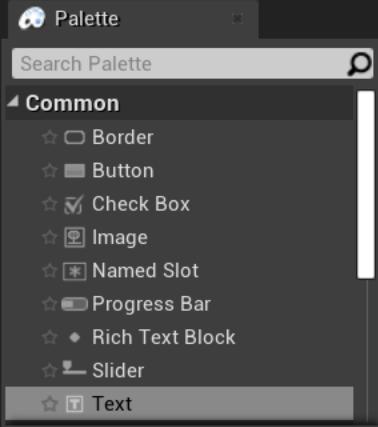
The screenshot shows the Unreal Engine 4 Designer interface. In the center, there is a canvas with a grid. A button labeled "Button_0" is placed on the canvas, containing a flower icon. The "Details" panel on the right shows the properties for "Button_0". The "Anchors" section is currently selected, with "Position X" and "Position Y" both set to 0.0. The "Size X" is 316.216217 and "Size Y" is 116.876877. The "Alignment" section shows "X: 0.5" and "Y: 0.5". Other sections like "Appearance", "Interaction", and "Accessibility" are also visible but not selected. At the bottom, the Timeline and Compiler Results panels are shown, with the Timeline indicating "No Animation Selected".

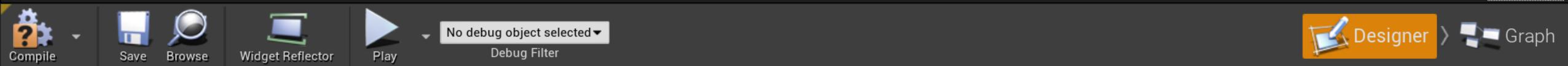




No debug object selected ▾
Debug Filter

Designer Graph





Zoom -4 925 x 491

0 500 1000 1500 2000

None Screen Size Fill Screen

Slot (Canvas Panel Slot)

Common

- Border
- Button
- Check Box
- Image
- Named Slot
- Progress Bar
- Rich Text Block
- Slider
- Text**

Hierarchy

[TestWidget]
 [Canvas Panel]
 Button_0
 [Text] "Button 1"

Device Content Scale 1.0
No Device Safe Zone Set
1280 x 720 (16:9)

DPI Scale 0.67

Details

Button_0 Is Variable

Search Details

Slot (Canvas Panel Slot)

- Anchors Anchors -158.108093 -58.438446 5.255127 4.023987
- Offset Left Offset Top Offset Right Offset Bottom
- Alignment X 0.5 Y 0.5
- Size To Content
- ZOrder 0

Appearance

- Style
- Color and Opacity
- Background Color

Interaction

- Is Focused

Accessibility

Animations

+ Animation Search Animations

Timeline Compiler Results

20 fps

0.00

No Animation Selected

0 items

0.00 0.00



No debug object selected
Debug Filter

Designer Graph

Palette

Search Palette

Common

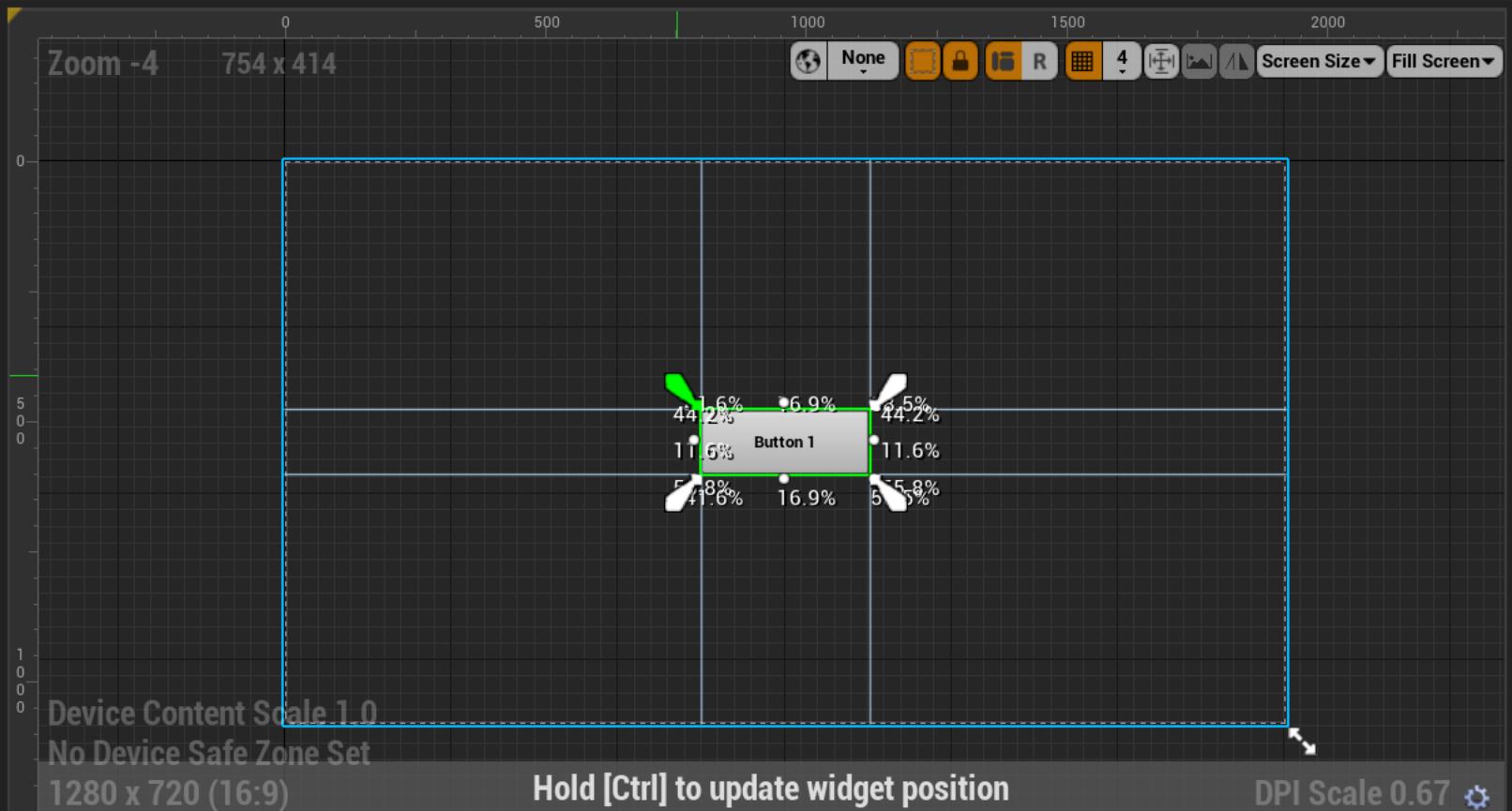
- Border
- Button
- Check Box
- Image
- Named Slot
- Progress Bar
- Rich Text Block
- Slider
- Text**

Hierarchy

Search Widgets

[TestWidget]

- [Canvas Panel]
 - Button_0
 - [Text] "Button 1"



Details

Button_0 Is Variable

Search Details

Slot (Canvas Panel Slot)

Anchors

- Offset Left: 2.852722
- Offset Top: 4.023956
- Offset Right: 5.255127
- Offset Bottom: 4.023987

Alignment: X 0.5 Y 0.5

Size To Content

ZOrder: 0

Appearance

Style

Color and Opacity

Background Color

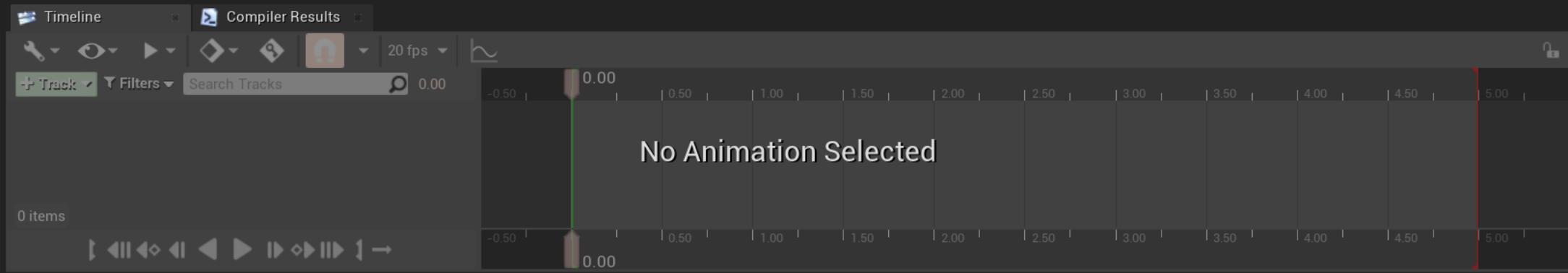
Interaction

Is Focused

Accessibility

Animations

+ Animation Search Animations



TestWidget* Parent class: User Widget

File Edit Asset View Debug Window Help

Compile Save Browse Widget Reflector Play No debug object selected Debug Filter

Zoom -4 1,953 x 1,111

0 500 1000 1500 2000

None R Screen Size Fill Screen

Device Content Scale 1.0
No Device Safe Zone Set
1301 x 729 (729:1301)

1136 x 640 1280 x 720

DPI Scale 0.67

Animations + Animation Search Animations

Timeline Compiler Results 20 fps

Track Filters Search Tracks 0.00

No Animation Selected

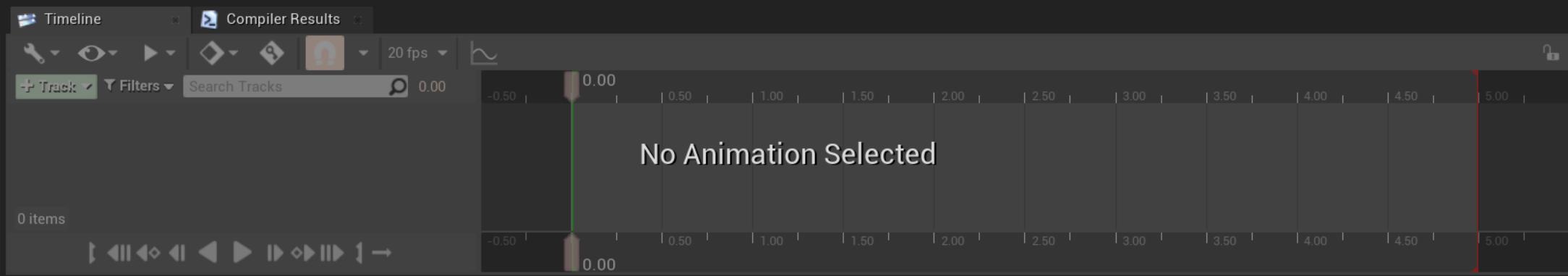
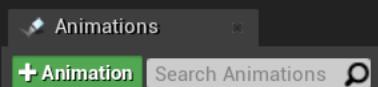
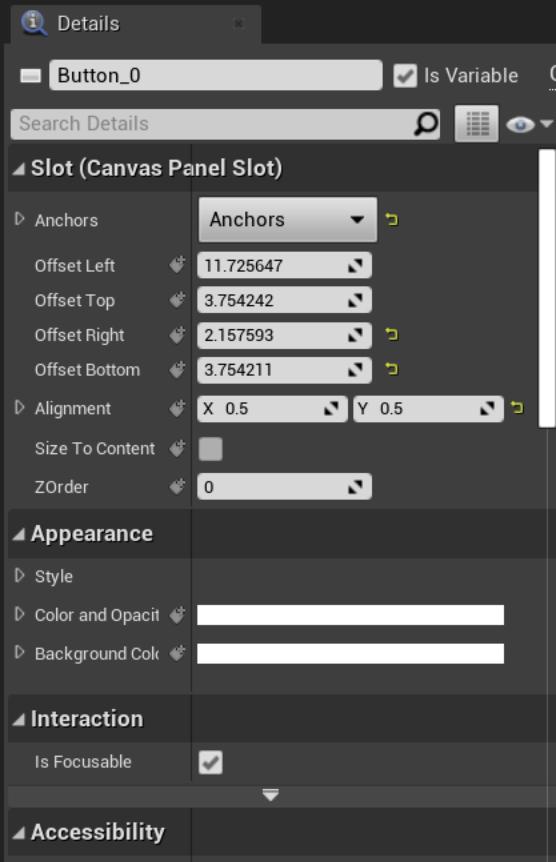
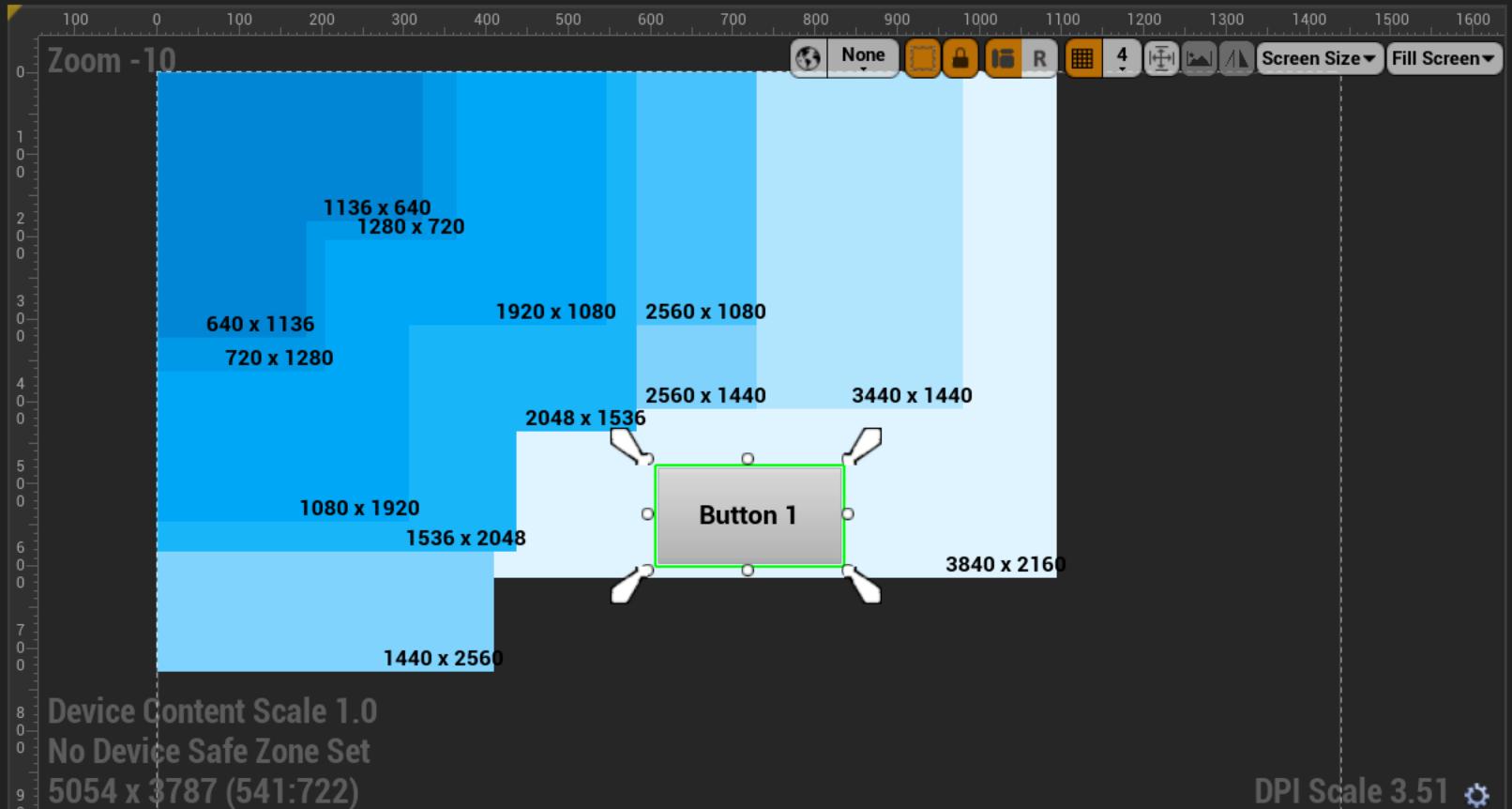
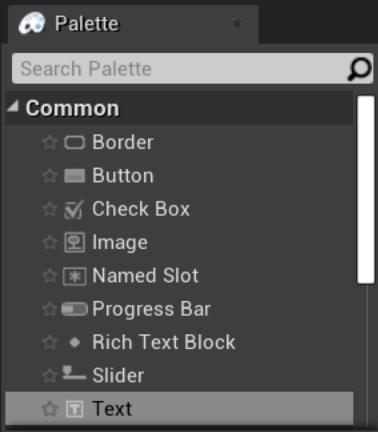
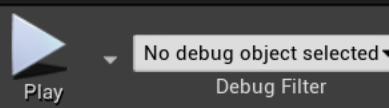
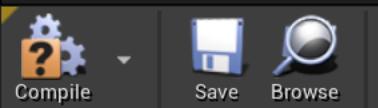
0 items

Common

- Border
- Button
- Check Box
- Image
- Named Slot
- Progress Bar
- Rich Text Block
- Slider
- Text

Hierarchy

- [TestWidget]
 - [Canvas Panel]
 - Button_0
 - [Text] "Button 1"



Exercise 8.03 Creating The RestartWidget C++ Class

The screenshot shows the Unreal Engine Editor interface. The top bar includes the Unreal Engine logo, menu items (파일(F), 편집(E), 보기(V), Git(G), 프로젝트(P), 빌드(B), 디버그(D), 테스트(S), 분석(N), 도구(I), 확장(X), 창(W), 도움말(H)), a search bar (검색 (Ctrl+Q)), and project name (Dodgeball). The toolbar below has various icons for file operations. The main area shows a code editor with several files open: Dodgeball.Build.cs, LookAtActorComponent.cpp, LookAtActorComponent.h, HealthInterface.h, and EnemyCharacter.cpp. The left sidebar shows project navigation with tabs for 서버(서버), 탐색기(탐색기), and 도구(도구). The right sidebar shows the Solution Explorer with a tree view of the project structure under the Dodgeball target. A red box highlights the Dodgeball.Build.cs file in both the code editor and the Solution Explorer.

```
// Copyright Epic Games, Inc. All Rights Reserved.  
using UnrealBuildTool;  
  
public class Dodgeball : ModuleRules  
{  
    public Dodgeball(ReadOnlyTargetRules Target) : base(Target)  
    {  
        PCHUsage = PCHUsageMode.UseExplicitOrSharedPCHs;  
  
        PublicDependencyModuleNames.AddRange(new string[] { "Core", "CoreUObject", "Engine", "InputCore", "HeadMountedDisplay" });  
    }  
}
```

솔루션 탐색기

솔루션 탐색기 검색(Ctrl+F)

- 솔루션 'Dodgeball' (2/2개 프로젝트)
 - Engine
 - UE4
 - Games
 - Dodgeball
 - 참조
 - 외부 종속성
 - Config
 - Source
 - Dodgeball

Dodgeball.Build.cs

- Dodgeball.cpp
- Dodgeball.h
- DodgeballCharacter.cpp
- DodgeballCharacter.h
- DodgeballFunctionLibrary.h
- DodgeballFunctionLibrary.cpp
- DodgeballGameMode.cpp
- DodgeballGameMode.h
- DodgeballProjectile.cpp
- DodgeballProjectile.h
- EnemyCharacter.cpp
- EnemyCharacter.h

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(I) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+Q) Dodgeball — □ ×

Develop Win64 로컬 Windows 디버거 Live Share

Dodgeball.Build.cs* LookAtActorComponent.cpp LookAtActorComponent.h HealthInterface.h EnemyCharacter.cpp

// Copyright Epic Games, Inc. All Rights Reserved.

```
1 using UnrealBuildTool;
2
3 public class Dodgeball : ModuleRules
4 {
5     public Dodgeball(ReadOnlyTargetRules Target) : base(Target)
6     {
7         PCHUsage = PCHUsageMode.UseExplicitOrSharedPCHs;
8
9         PublicDependencyModuleNames.AddRange(new string[] { "Core", "CoreUObject", "Engine", "InputCore", "HeadMountedDisplay",
10                                         "UMG", "Slate", "SlateCore" });
11    }
12 }
13
14 }
15 }
```

Ctrl+S

슬루션 탐색기

슬루션 탐색기 검색(Ctrl+.)

슬루션 탐색기 'Dodgeball' (2/2개 프로젝트)

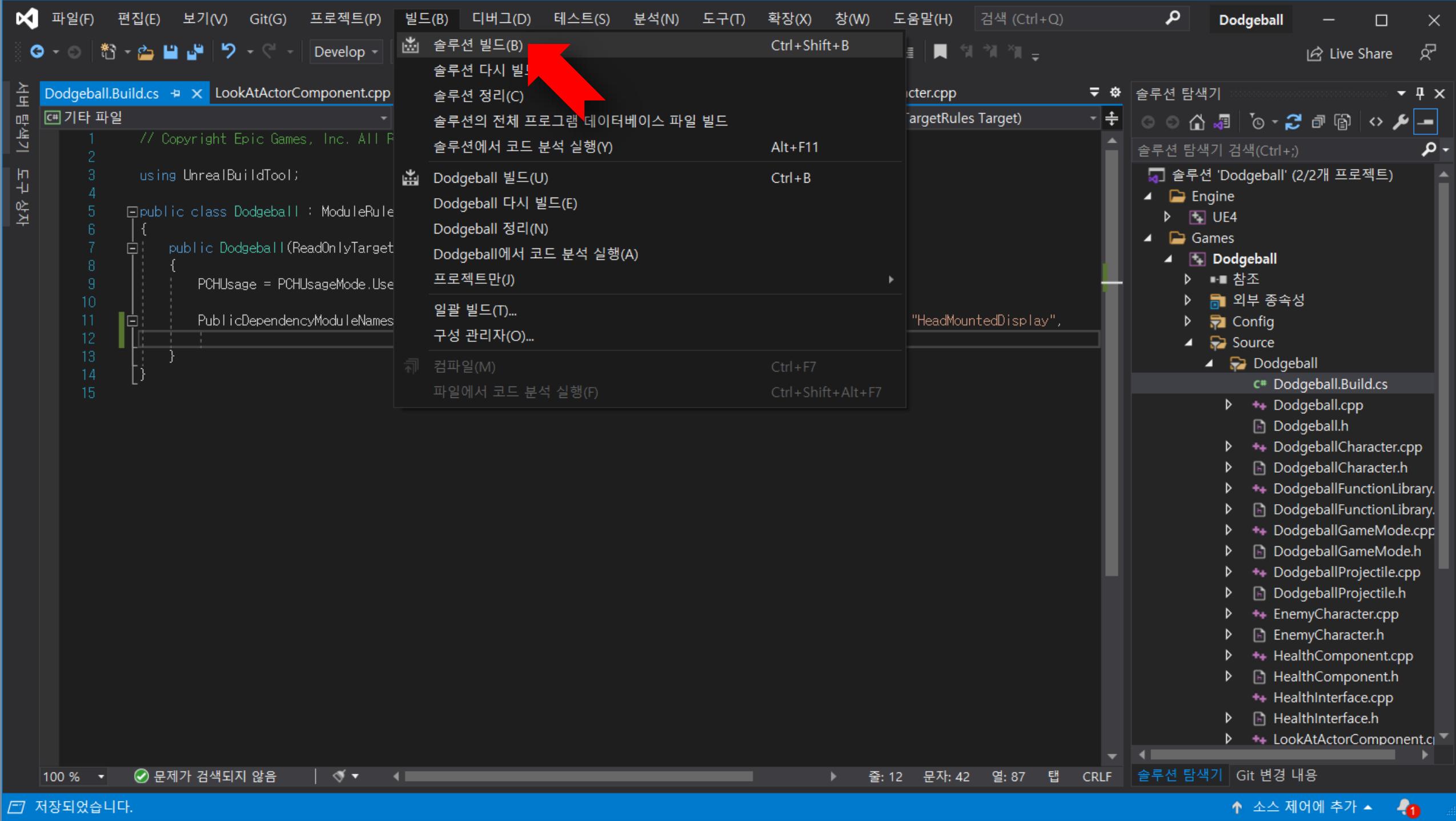
- Engine
- UE4
- Games
- Dodgeball
 - 참조
 - 외부 종속성
 - Config
 - Source
 - Dodgeball
 - Dodgeball.Build.cs
 - Dodgeball.cpp
 - Dodgeball.h
 - DodgeballCharacter.cpp
 - DodgeballCharacter.h
 - DodgeballFunctionLibrary.cpp
 - DodgeballFunctionLibrary.h
 - DodgeballGameMode.cpp
 - DodgeballGameMode.h
 - DodgeballProjectile.cpp
 - DodgeballProjectile.h
 - EnemyCharacter.cpp
 - EnemyCharacter.h
 - HealthComponent.cpp
 - HealthComponent.h
 - HealthInterface.cpp
 - HealthInterface.h
 - LookAtActorComponent.cpp

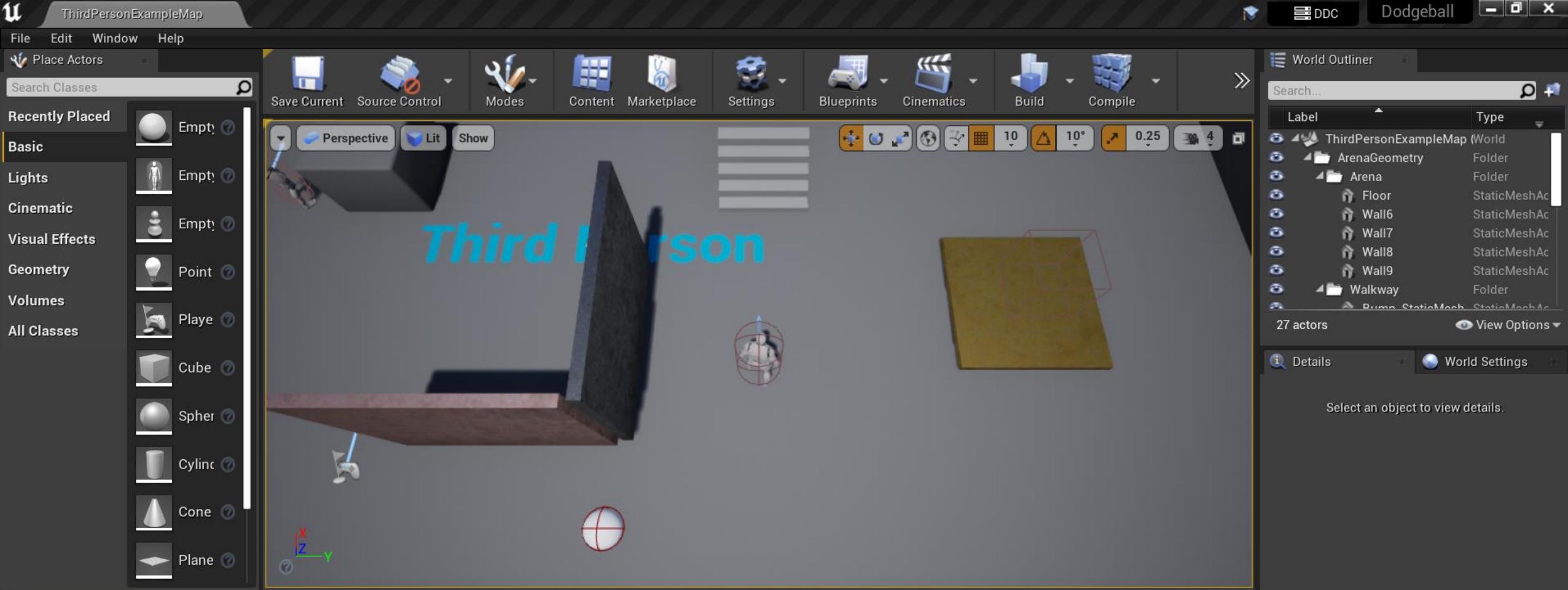
준비

문제가 검색되지 않음

줄: 12 문자: 42 열: 87 템 CRLF

슬루션 탐색기 Git 변경 내용 ↑ 소스 제어에 추가 ↻





The screenshot shows the Content Browser with a green header bar for "Add/Import" and "Save All". The main area displays a grid of assets under the "Dodgeball" folder. The assets include: DodgeballCharacter, DodgeballFunctionLibrary, DodgeballGameMode, DodgeballProjectile, EnemyCharacter, HealthInterface, LookAtActorComponent, VictoryBox, and Wall. A context menu is open over the "Dodgeball" asset, with options for "New Folder", "New C++ Class...", and "Create a new class in /Classes_Game/Dodgeball.". The "C++ Classes" section is highlighted with a red box, and the "Dodgeball" item is selected. The bottom status bar shows "10 items" and "View Options".

Right-Click



Choose Parent Class

This will add a C++ header and source code file to your game project.

Show All Classes

userw

Object

Visual

Widget

UserWidget

4 items (1 selected)

View Options ▾

The screenshot shows a search bar at the top left containing 'userw'. Below it is a tree view of class categories: Object, Visual, and Widget. Under Widget, the class 'UserWidget' is listed and highlighted with a red rectangle. At the bottom left, there are two status lines: 'Selected Class' followed by 'User Widget' and 'Selected Class Source' followed by 'UserWidget.h'. On the right side, there are four buttons: a large green 'Next >' button with a red arrow pointing to it, a smaller green 'Create Class' button, and two grey 'Cancel' buttons.

Selected Class

User Widget

Selected Class Source

UserWidget.h

Next >

Create Class

Cancel



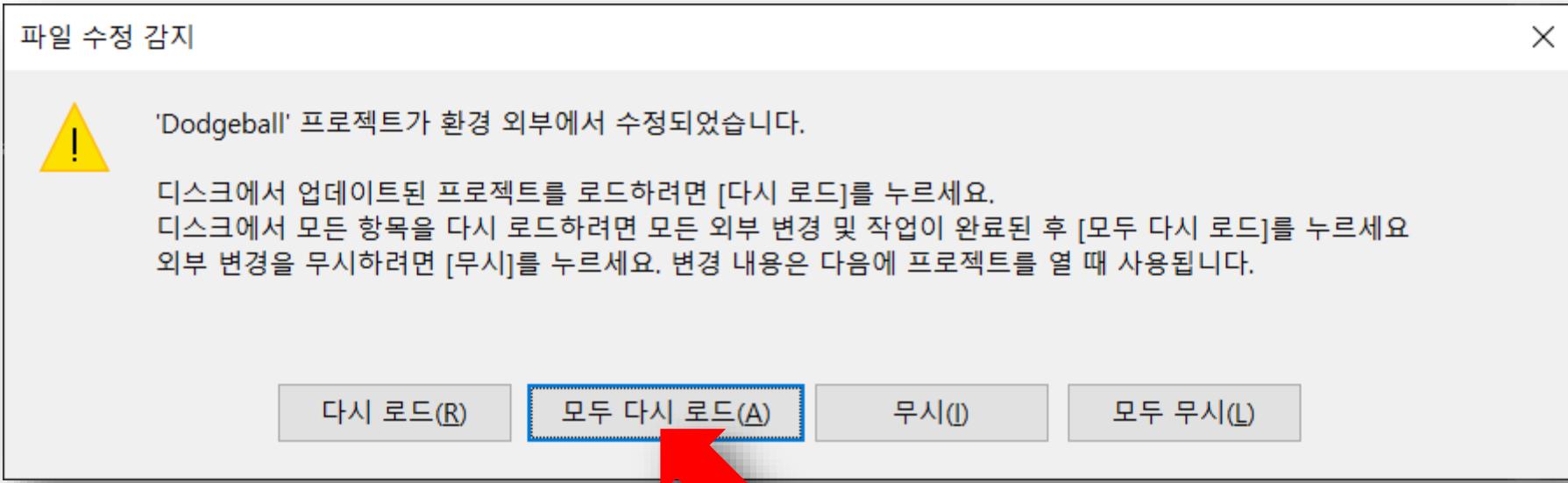
Name Your New User Widget

Enter a name for your new class. Class names may only contain alphanumeric characters, and may not contain a space.

When you click the "Create" button below, a header (.h) file and a source (.cpp) file will be made using this name.

Name	<input type="text" value="RestartWidget"/>	Dodgeball (Runtime)	Public	Private
Path	<input type="text" value="C:/Users/sunje/Desktop/Unreal/Dodgeball/Source/Dodgeball/"/>			
Header File	C:/Users/sunje/Desktop/Unreal/Dodgeball/Source/Dodgeball/RestartWidget.h			
Source File	C:/Users/sunje/Desktop/Unreal/Dodgeball/Source/Dodgeball/RestartWidget.cpp			


[Back](#)[Create Class](#)[Cancel](#)



파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(I) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+Q) Dodgeball — □ ×

Live Share

RestartWidget.h* ✘ X Dodgeball.Build.cs LookAtActorComponent.cpp LookAtActorComponent.h HealthInterface.h

Dodgeball

```
1 // Fill out your copyright notice in the Description page of Project Settings.
2
3 #pragma once
4
5 #include "CoreMinimal.h"
6 #include "Blueprint/UserWidget.h"
7 #include "RestartWidget.generated.h"
8
9 /**
10 * 
11 */
12 UCLASS()
13 class DODGEBALL_API URestartWidget : public UUserWidget
14 {
15     GENERATED_BODY()
16
17 protected:
18     UFUNCTION()
19     void OnRestartClicked();
20
21 public:
22     UPROPERTY(meta = (BindWidget))
23     class UButton* RestartButton;
24 };
25
26
27
28 }
```

Ctrl+S

슬루션 탐색기

슬루션 탐색기 검색(Ctrl+Shift+F)

- 슬루션 'Dodgeball' (2/2개 프로젝트)
 - Engine
 - UE4
 - Games
 - Dodgeball
 - 참조
 - 외부 종속성
 - Config
 - Source
 - Dodgeball
 - Dodgeball.Build.cs
 - Dodgeball.cpp
 - Dodgeball.h
 - DodgeballCharacter.cpp
 - DodgeballCharacter.h
 - DodgeballFunctionLibrary.cpp
 - DodgeballFunctionLibrary.h
 - DodgeballGameMode.cpp
 - DodgeballGameMode.h
 - DodgeballProjectile.cpp
 - DodgeballProjectile.h
 - EnemyCharacter.cpp
 - EnemyCharacter.h
 - HealthComponent.cpp
 - HealthComponent.h
 - HealthInterface.cpp
 - HealthInterface.h
 - LookAtActorComponent.cpp

준비

소스 제어에 추가

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(I) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+Q) Dodgeball — □ ×

Develop Win64 로컬 Windows 디버거 Live Share

RestartWidget.cpp* ✘ RestartWidget.h Dodgeball.Build.cs LookAtActorComponent.cpp LookAtActorComponent.h

Dodgeball

```
1 // Fill out your copyright notice in the Description page of Project Settings.  
2  
3  
4 #include "RestartWidget.h"  
5 #include "Kismet/GameplayStatics.h"  
6  
7 void URestartWidget::OnRestartClicked()  
8 {  
9     UGameplayStatics::OpenLevel(this, FName(*UGameplayStatics::GetCurrentLevelName(this)));  
10 }  
11
```

OnRestartClicked()

솔루션 탐색기

솔루션 탐색기 검색(Ctrl+Shift+F)

솔루션 탐색기 'Dodgeball' (2/2개 프로젝트)

- Engine
- UE4
- Games
- Dodgeball
 - 참조
 - 외부 종속성
 - Config
 - Source
 - Dodgeball
 - Dodgeball.Build.cs
 - Dodgeball.cpp
 - Dodgeball.h
 - DodgeballCharacter.cpp
 - DodgeballCharacter.h
 - DodgeballFunctionLibrary.cpp
 - DodgeballFunctionLibrary.h
 - DodgeballGameMode.cpp
 - DodgeballGameMode.h
 - DodgeballProjectile.cpp
 - DodgeballProjectile.h
 - EnemyCharacter.cpp
 - EnemyCharacter.h
 - HealthComponent.cpp
 - HealthComponent.h
 - HealthInterface.cpp
 - HealthInterface.h

Ctrl+S

100% 문제가 검색되지 않음 줄: 9 문자: 89 열: 92 템 CRLF

솔루션 탐색기 Git 변경 내용

준비 ↑ 소스 제어에 추가 ↻ 1

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(I) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+Q) Dodgeball — □ ×

Develop Win64 로컬 Windows 디버거 Live Share

RestartWidget.cpp RestartWidget.h* Dodgeball.Build.cs LookAtActorComponent.cpp LookAtActorComponent.h

Dodgeball

```
1 // Fill out your copyright notice in the Description page of Project Settings.
2
3 #pragma once
4
5 #include "CoreMinimal.h"
6 #include "Blueprint/UserWidget.h"
7 #include "RestartWidget.generated.h"
8
9 /**
10 * 
11 */
12 UCLASS()
13 class DODGEBALL_API URestartWidget : public UUserWidget
14 {
15     GENERATED_BODY()
16
17 protected:
18     UFUNCTION()
19     void OnRestartClicked();
20
21 public:
22
23     UPROPERTY(meta = (BindWidget))
24     class UButton* RestartButton;
25
26     virtual void NativeOnInitialized() override;
27 }
28
29
30 }
```

Ctrl+S

100 % 문제가 검색되지 않음 줄: 27 문자: 46 열: 49 탭 CRLF

저장되었습니다. ↑ 소스 제어에 추가 ↗

솔루션 탐색기

솔루션 탐색기 검색(Ctrl+.)

솔루션 'Dodgeball' (2/2개 프로젝트)

- Engine
- UE4
- Games
- Dodgeball
- 참조
- 외부 종속성
- Config
- Source
- Dodgeball
- Dodgeball.Build.cs
- Dodgeball.cpp
- Dodgeball.h
- DodgeballCharacter.cpp
- DodgeballCharacter.h
- DodgeballFunctionLibrary.cpp
- DodgeballFunctionLibrary.h
- DodgeballGameMode.cpp
- DodgeballGameMode.h
- DodgeballProjectile.cpp
- DodgeballProjectile.h
- EnemyCharacter.cpp
- EnemyCharacter.h
- HealthComponent.cpp
- HealthComponent.h
- HealthInterface.cpp
- HealthInterface.h
- LookAtActorComponent.cpp

솔루션 탐색기 Git 변경 내용

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(I) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+Q) Dodgeball — □ ×

Develop Win64 로컬 Windows 디버거 Live Share

RestartWidget.cpp* ✘ RestartWidget.h Dodgeball.Build.cs LookAtActorComponent.cpp LookAtActorComponent.h

```
// Fill out your copyright notice in the Description page of Project Settings.

#include "RestartWidget.h"
#include "Kismet/GameplayStatics.h"
#include "Components/Button.h"

void URestartWidget::OnRestartClicked()
{
    UGameplayStatics::OpenLevel(this, FName(*UGameplayStatics::GetCurrentLevelName(this)));
}

void URestartWidget::NativeOnInitialized()
{
    Super::NativeOnInitialized();

    if (RestartButton != nullptr)
    {
        RestartButton->OnClicked.AddDynamic(this, &URestartWidget::OnRestartClicked);
    }
}
```

Ctrl+S

100% 문제가 검색되지 않음 줄: 19 문자: 80 열: 86 탭 CRLF

저장되었습니다. ↑ 소스 제어에 추가 ↗

서버 템플릿 최근 속자

솔루션 탐색기

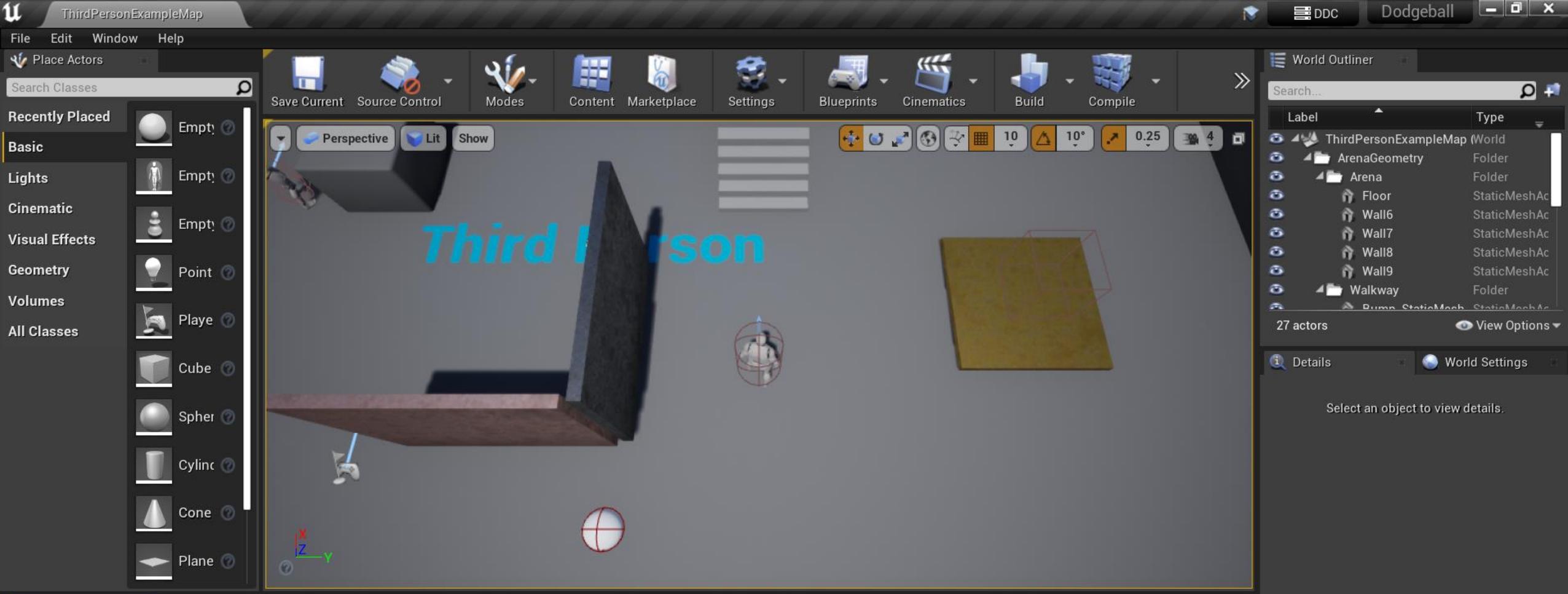
솔루션 탐색기 검색(Ctrl+Shift+F)

솔루션 'Dodgeball' (2/2개 프로젝트)

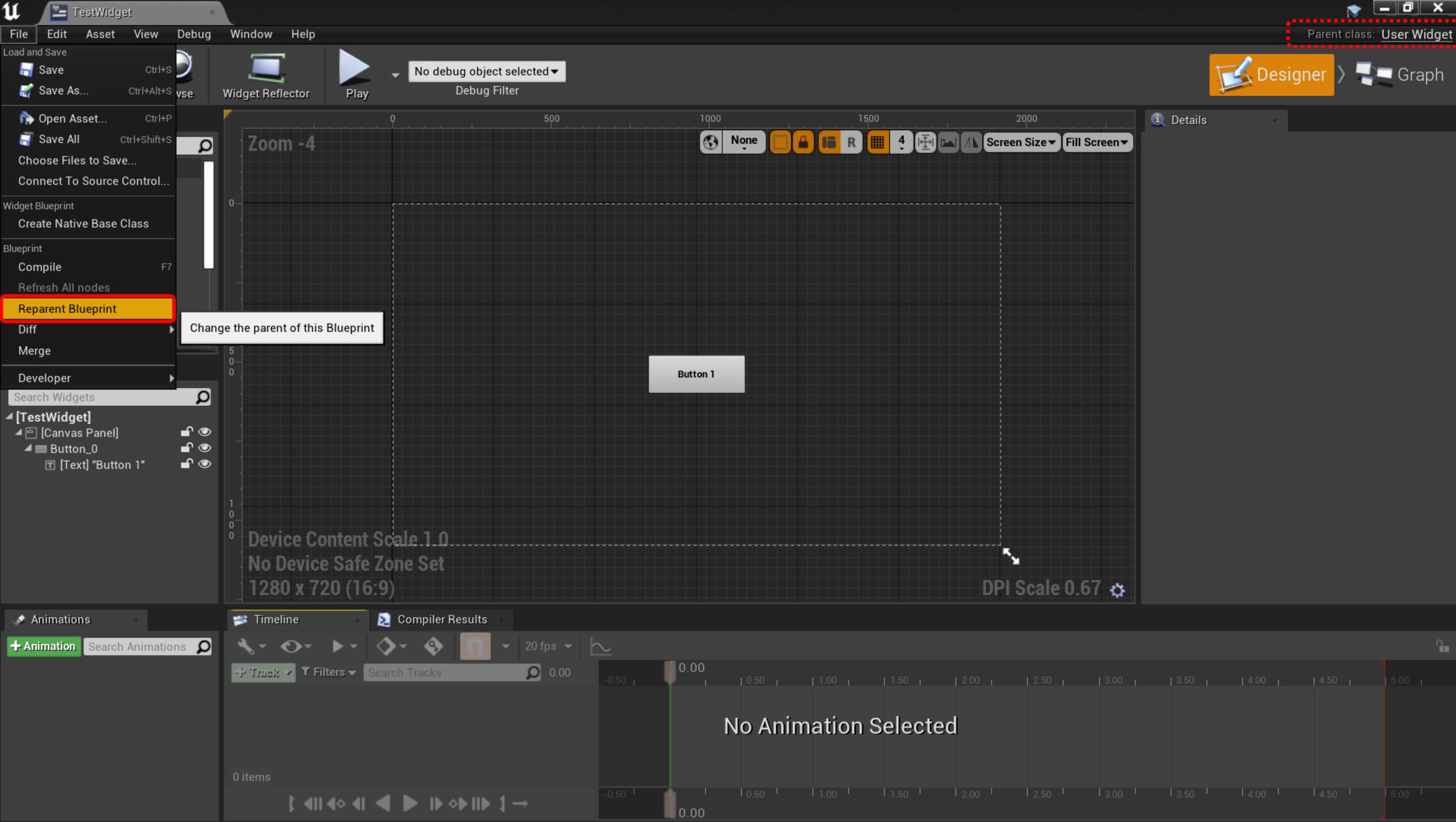
- Engine
- UE4
- Games
- Dodgeball
- 참조
- 외부 종속성
- Config
- Source
- Dodgeball
- Dodgeball.Build.cs
- Dodgeball.cpp
- Dodgeball.h
- DodgeballCharacter.cpp
- DodgeballCharacter.h
- DodgeballFunctionLibrary.cpp
- DodgeballFunctionLibrary.h
- DodgeballGameMode.cpp
- DodgeballGameMode.h
- DodgeballProjectile.cpp
- DodgeballProjectile.h
- EnemyCharacter.cpp
- EnemyCharacter.h
- HealthComponent.cpp
- HealthComponent.h
- HealthInterface.cpp
- HealthInterface.h
- LookAtActorComponent.cpp

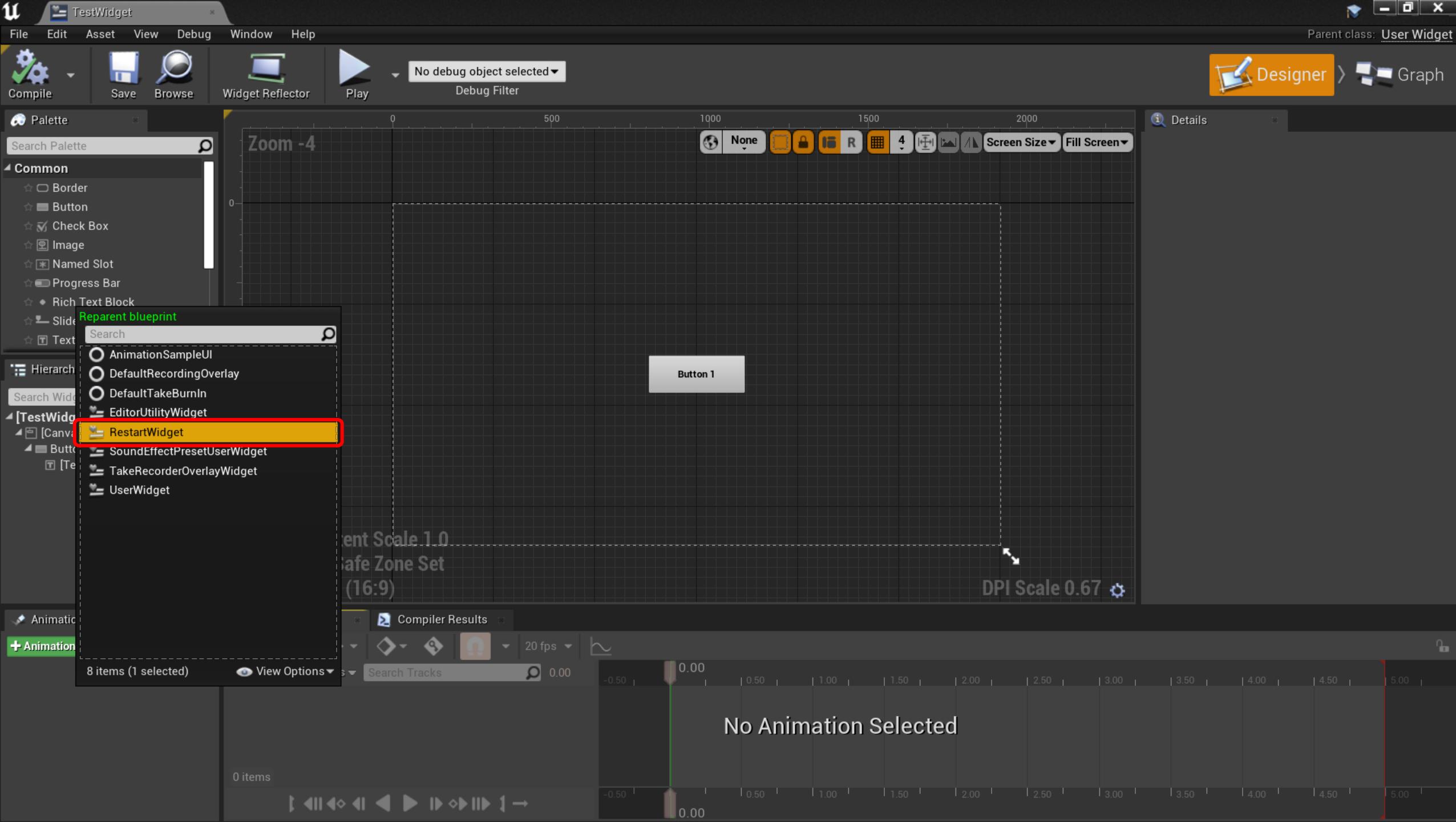
The screenshot shows the Microsoft Visual Studio interface. The top menu bar is visible with various options like 파일(F), 편집(E), 보기(V), Git(G), 프로젝트(P), 빌드(B), 디버그(D), 테스트(S), 분석(N), 도구(T), 확장(X), 창(W), 도움말(H), and 검색 (Ctrl+Q). A red arrow points to the 'Solution Build' option in the 'Build' menu, which is the second item in the list. The main code editor window on the left contains C++ code for a class named URestartWidget. To the right of the editor is a Solution Explorer window showing project files for 'Dodgeball' and its subfolders like Engine, Games, and Dodgeball. The bottom status bar displays file paths, character count, and line numbers.

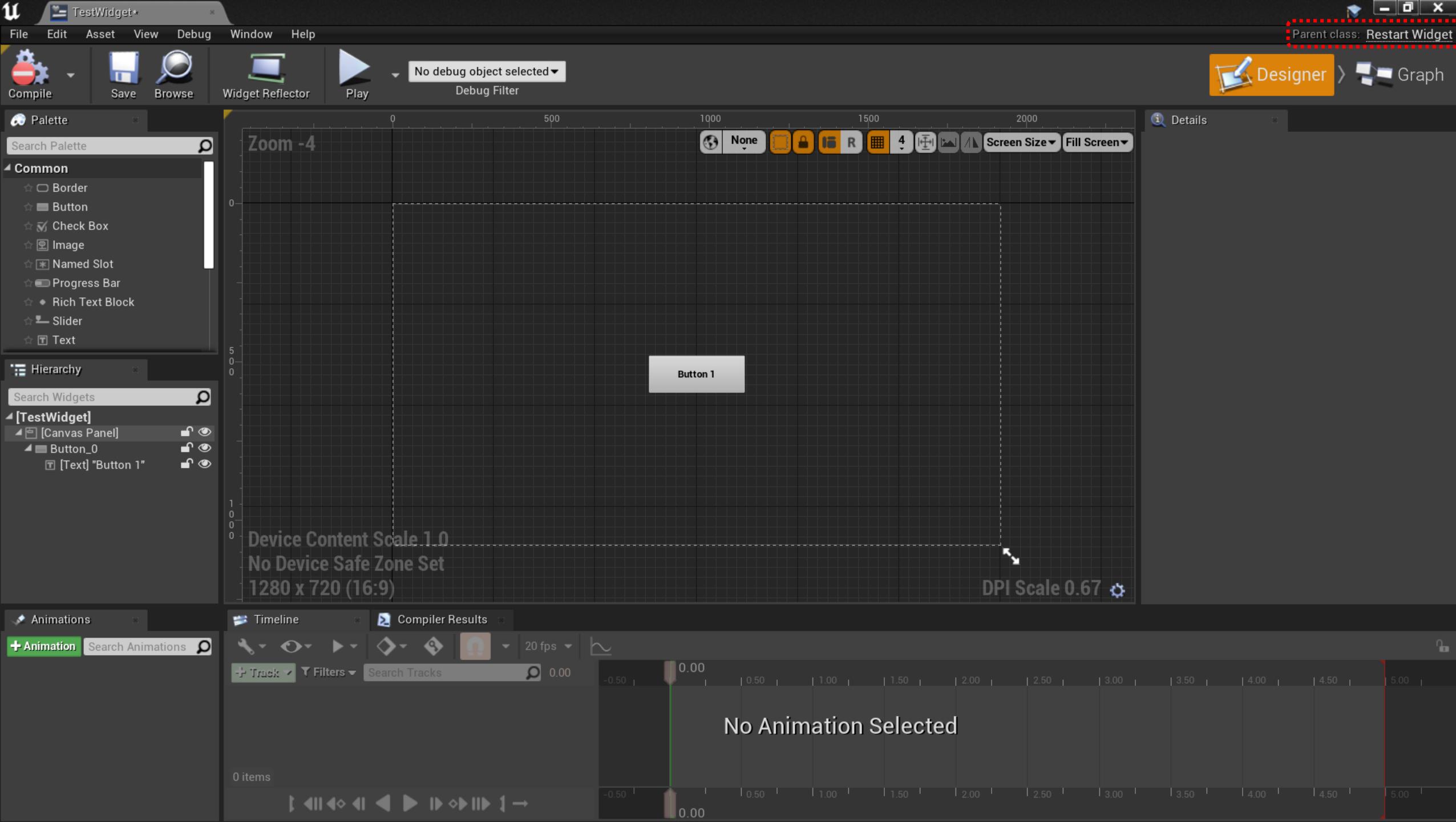
```
// Fill out your copyright notice in  
// the License header above.  
  
#include "RestartWidget.h"  
#include "Kismet/GameplayStatics.h"  
#include "Components/Button.h"  
  
void URestartWidget::OnRestartClick()  
{  
    UGameplayStatics::OpenLevel(this, 0);  
}  
  
void URestartWidget::NativeOnInitialized()  
{  
    Super::NativeOnInitialized();  
  
    if (RestartButton != nullptr)  
    {  
        RestartButton->OnClicked.AddDynamic(this, &URestartWidget::OnRestartClicked);  
    }  
}
```

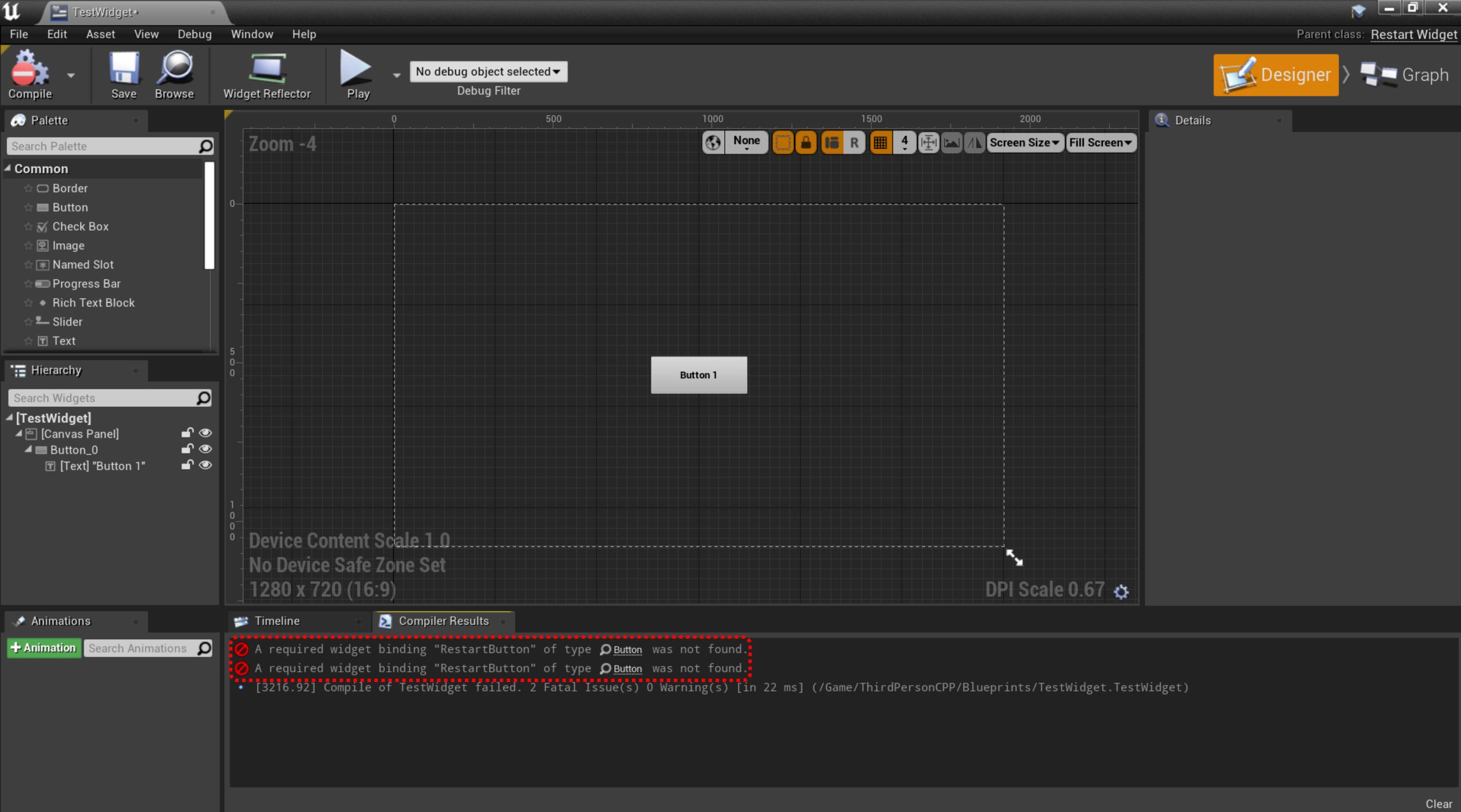


The Content Browser interface is shown at the bottom of the screen. The left sidebar displays the file structure: Content > Geometry, Mannequin, Physics, StarterContent, ThirdPerson, ThirdPersonCPP (highlighted with a red box), Blueprints, Maps, and C++ Classes. The Dodgeball class is listed under C++ Classes. The main area shows a grid of blueprint icons. The "TestWidget" blueprint is selected and highlighted with a yellow border. A red arrow points to this selected item. The status bar at the bottom indicates "7 items (1 selected)" and "View Options".









TestWidget

File Edit Asset View Debug Window Help

Parent class: Restart Widget

Compile Save Browse Widget Reflector Play No debug object selected Debug Filter

Designer Graph

Palette Search Palette

Common Border Button Check Box Image Named Slot Progress Bar Rich Text Block Slider Text

Hierarchy [TestWidget] [Canvas Panel] Button_0 [Text]

Right-Click

Cut Ctrl+X Copy Ctrl+C Paste Ctrl+V Duplicate Ctrl+W Delete Delete Rename F2 Rename current selection Wrap With... Replace With...

Device Content Scale 1.0
No Device Safe Zone Set

Animations

Animation Search Animations Compiler Results

A required widget binding "RestartButton" of type `PButton` was not found.
A required widget binding "RestartButton" of type `PButton` was not found.
[3216.92] Compile of TestWidget failed. 2 Fatal Issue(s) 0 Warning(s) [in 22 ms] (/Game/ThirdPersonCPP/Blueprints/TestWidget.TestWidget)

Details

Button_0 Is Variable

Search Details

Slot (Canvas Panel Slot)

Anchors Anchors
Offset Left 11.725647
Offset Top 3.754242
Offset Right 2.157593
Offset Bottom 3.754211
Alignment X 0.5 Y 0.5
Size To Content
ZOrder 0

Appearance Style
Color and Opacity
Background Color

Interaction Is Focused

Accessibility

DPI Scale 0.67

Clear

No debug object selected
Debug Filter

Palette

Search Palette

Zoom -4

0 500 1000 1500 2000

None Screen Size Fill Screen

Common

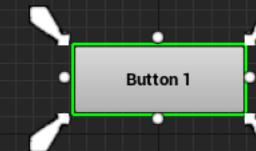
- Border
- Button
- Check Box
- Image
- Named Slot
- Progress Bar
- Rich Text Block
- Slider
- Text

Hierarchy

Search Widgets

[TestWidget]

- [Canvas Panel]
 - RestartButton
 - [Text] Button 1



Device Content Scale 1.0
No Device Safe Zone Set
1280 x 720 (16:9)

DPI Scale 0.67

Animations

+ Animation Search Animations

Timeline

Compiler Results

- ✗ A required widget binding "RestartButton" of type was not found.
- ✗ A required widget binding "RestartButton" of type was not found.
- [3216.92] Compile of TestWidget failed. 2 Fatal Issue(s) 0 Warning(s) [in 22 ms] (/Game/ThirdPersonCPP/Blueprints/TestWidget.TestWidget)

Clear

TestWidget

File Edit Asset View Debug Window Help Parent class: Restart Widget

Compiler Save Browse Widget Reflector Play No debug object selected Debug Filter

Zoom -4 0 500 1000 1500 2000 Screen Size Fill Screen

Details Search Details

RestartButton Is Variable

Slot (Canvas Panel Slot)

- Anchors Anchors
- Offset Left 11.725647
- Offset Top 3.754242
- Offset Right 2.157593
- Offset Bottom 3.754211
- Alignment X 0.5 Y 0.5
- Size To Content
- ZOrder 0

Appearance

- Style
- Color and Opacity
- Background Color

Interaction

- Is Focused

Accessibility

Hierarchy

[TestWidget] [Canvas Panel] RestartButton [Text] "Button 1"

Device Content Scale 1.0
No Device Safe Zone Set
1280 x 720 (16:9)

DPI Scale 0.67

Animations

Timeline Compiler Results

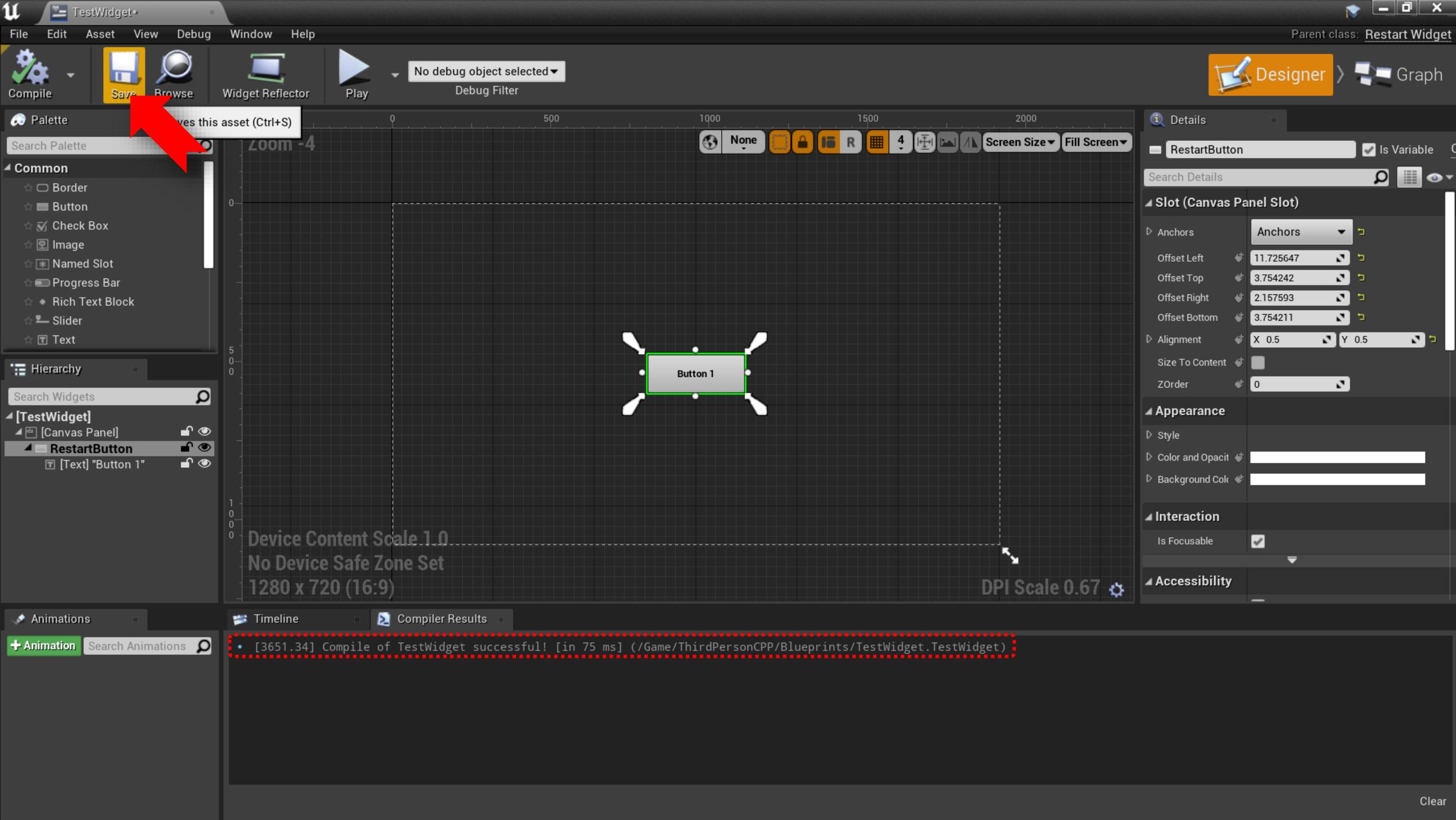
+ Animation Search Animations

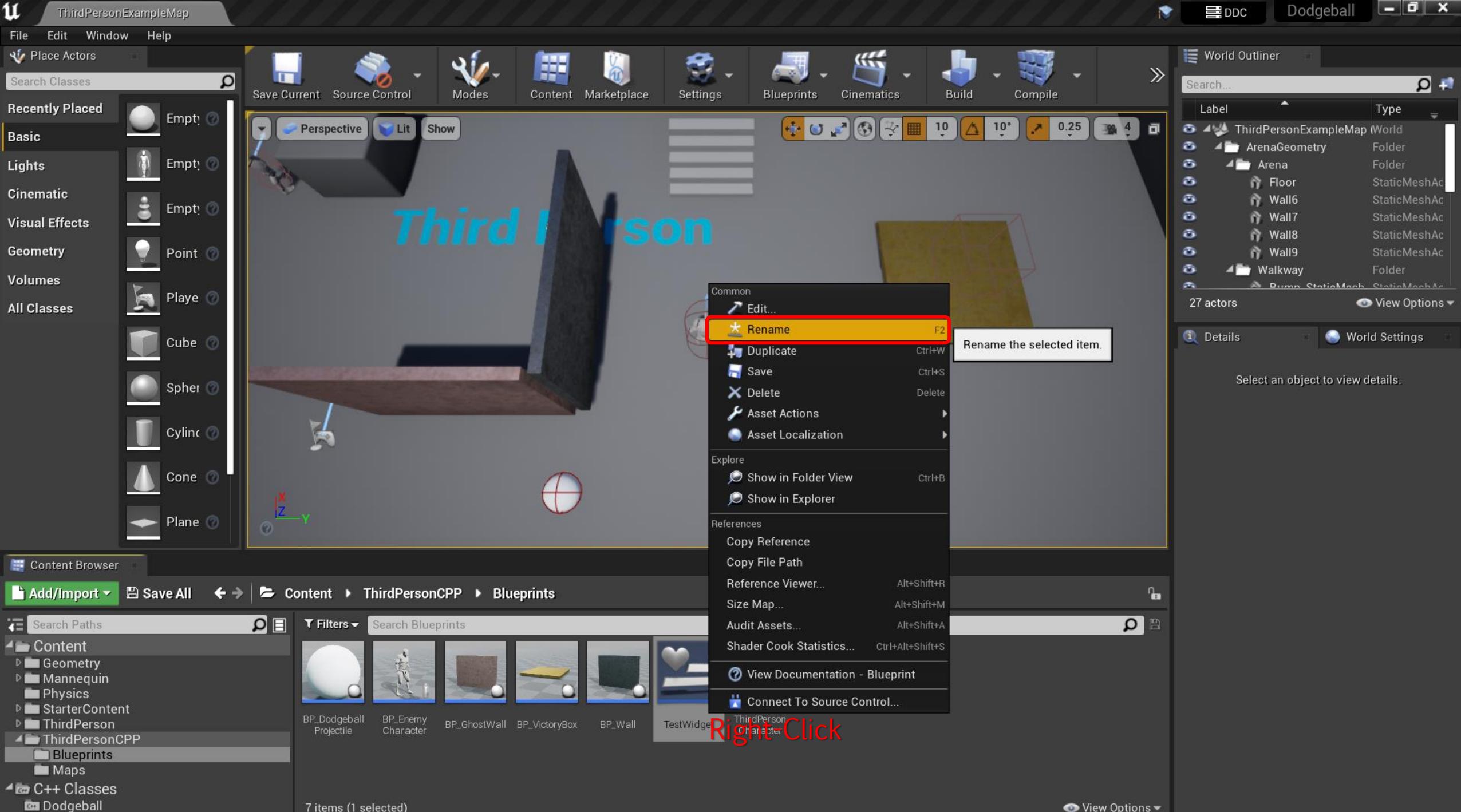
Compiler Results

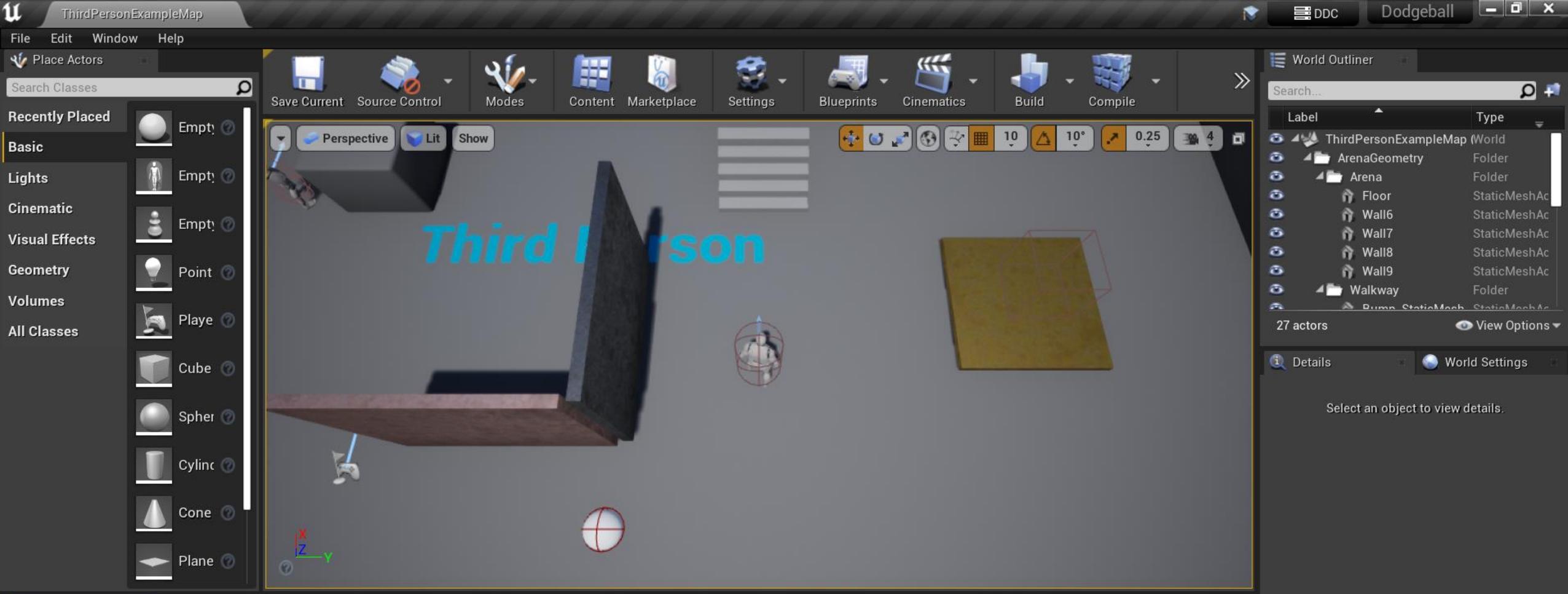
- A required widget binding "RestartButton" of type `PButton` was not found.
- A required widget binding "RestartButton" of type `PButton` was not found.
- [3216.92] Compile of TestWidget failed. 2 Fatal Issue(s) 0 Warning(s) [in 22 ms] (/Game/ThirdPersonCPP/Blueprints/TestWidget.TestWidget)

Clear

The screenshot shows the Unreal Engine Designer interface. At the top, there's a toolbar with icons for File, Edit, Asset, View, Debug, Window, Help, and a 'Parent class: Restart Widget'. Below the toolbar is a menu bar with 'TestWidget' as the active window. The main workspace is a grid-based canvas where a button component is placed. The button has a green selection border and is labeled 'Button 1'. To the left of the canvas is a 'Hierarchy' panel showing the structure of the widget, with 'RestartButton' currently selected. On the right side, there are several panels for 'Details', 'Slot (Canvas Panel Slot)', 'Appearance', 'Interaction', and 'Accessibility'. At the bottom, there are tabs for 'Animations' and 'Timeline', and a 'Compiler Results' panel which displays errors related to missing widget bindings. A red arrow points to the 'Compile' icon in the top left corner of the interface.







The Content Browser is open, showing the "Blueprints" category under "ThirdPersonCPP". The browser includes search paths, filters, and a preview area. Several blueprints are listed: BP_DodgeballProjectile, BP_EnemyCharacter, BP_GhostWall, BP_RestartWidget (which is highlighted with a red box), BP_VictoryBox, BP_Wall, and ThirdPersonCharacter. The preview area shows small images of each blueprint. The bottom status bar indicates "7 items (1 selected)".



Exercise 8.04: Creating The Logic For Adding The RestartWidget To The Screen

The screenshot shows the Unreal Engine Editor interface. The Content Browser at the bottom left displays a file tree under 'Content' and highlights 'C++ Classes' with a red box. A sub-tree for 'Dodgeball' contains 'Dodgeball' (selected), 'Blueprints', and 'Maps'. The main tab 'C++ Classes' is also highlighted with a red box. A yellow button labeled 'New C++ Class...' is visible. The Level Editor window in the center shows a 3D scene with a character in a blue capsule, a large yellow cube, and a blue sphere. The right panel shows the 'World Settings' and a list of 27 actors, including various wall and floor static meshes. A tooltip at the bottom says 'Create a new class in /Classes_Game/Dodgeball.'

Cinematic
Visual Effects
Geometry
Volumes
All Classes

Empty
Point
Plane
Play
Cube
Sphere
Cylinder
Cone
Plane

Floor
Wall6
Wall7
Wall8
Wall9
Walkway
Runn Static Mesh
Static Mesh Actor

27 actors

View Options

Details

World Settings

Select an object to view details.

Add/Import Save All C++ Classes Dodgeball

Filters Search Dodgeball

Content Geometry Mannequin Physics StarterContent ThirdPerson ThirdPersonCPP Blueprints Maps

Dodgeball Dodgeball Dodgeball Dodgeball Dodgeball Dodgeball Dodgeball Dodgeball Dodgeball

New Folder New C++ Class... Create a new class in /Classes_Game/Dodgeball.

View Options

Right-Click

11 items



Choose Parent Class

This will add a C++ header and source code file to your game project.

Show All Classes

Actor Component

An ActorComponent is a reusable component that can be added to any actor.

Scene Component

A Scene Component is a component that has a scene transform and can be attached to other scene components.

Player Camera Manager

A PlayerCameraManager is responsible for managing the camera for a particular player.

Player Controller

A Player Controller is an actor responsible for controlling a Pawn used by the player.

Game Mode Base

Game Mode Base defines the game being played, its rules, scoring, and other facets of the game type.

Selected Class

Player Controller

Selected Class Source

PlayerController.h



Next >

Create Class

Cancel



Name Your New Player Controller

Enter a name for your new class. Class names may only contain alphanumeric characters, and may not contain a space.

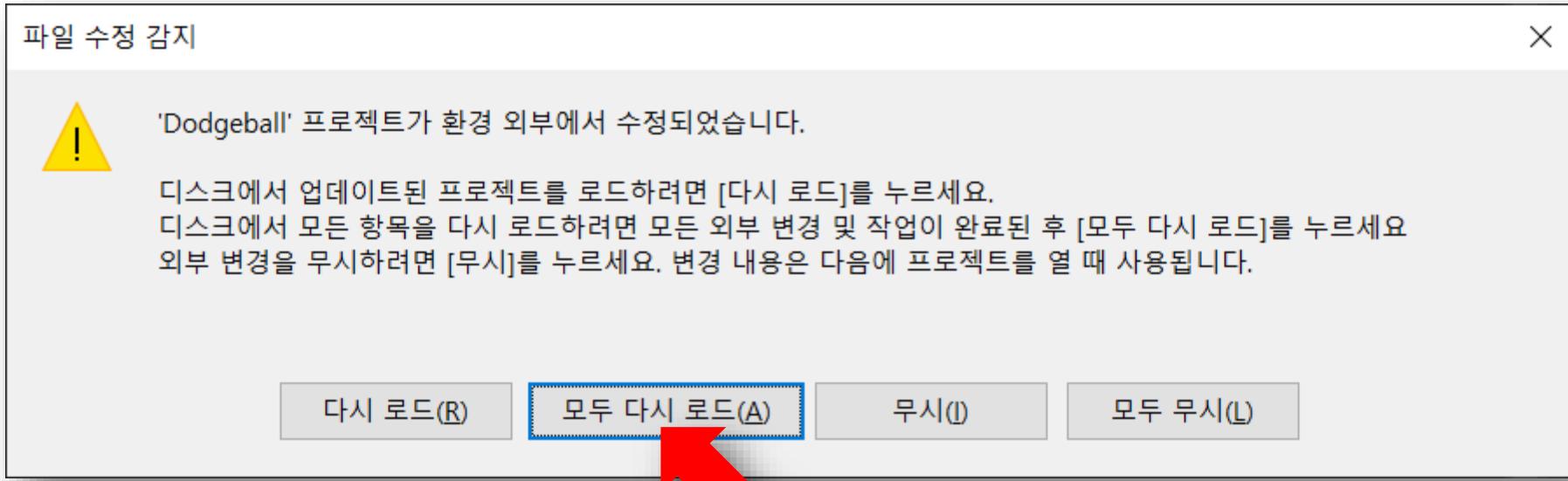
When you click the "Create" button below, a header (.h) file and a source (.cpp) file will be made using this name.

Name	DodgeballPlayerController	Dodgeball (Runtime) ▼	Public	Private
Path	C:/Users/sunje/Desktop/Unreal/Dodgeball/Source/Dodgeball/	Choose Folder		
Header File	C:/Users/sunje/Desktop/Unreal/Dodgeball/Source/Dodgeball/DodgeballPlayerController.h			
Source File	C:/Users/sunje/Desktop/Unreal/Dodgeball/Source/Dodgeball/DodgeballPlayerController.cpp			

< Back

Create Class

Cancel



파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(I) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+Q) Dodgeball — □ ×

Develop Win64 로컬 Windows 디버거 Live Share

DodgeballPlayerController.cpp DodgeballPlayerController.h* × RestartWidget.cpp RestartWidget.h Dodgeball.Build.cs

Dodgeball // Fill out your copyright notice in the Description page of Project Settings.

```
1 // Fill out your copyright notice in the Description page of Project Settings.
2
3 #pragma once
4
5 #include "CoreMinimal.h"
6 #include "GameFramework/PlayerController.h"
7 #include "DodgeballPlayerController.generated.h"
8
9 /**
10 * 
11 */
12 UCLASS()
13 class DODGEBALL_API ADodgeballPlayerController : public APlayerController
14 {
15     GENERATED_BODY()
16
17 public:
18     UPROPERTY(EditDefaultsOnly)
19     TSubclassOf<class URestartWidget> BP_RestartWidget;
20
21     void ShowRestartWidget();
22
23 private:
24     UPROPERTY()
25     class URestartWidget* RestartWidget;
26 };
27
28
29 }
30 
```

Ctrl+S

슬루션 탐색기

슬루션 탐색기 검색(Ctrl+.)

솔루션 'Dodgeball' (2/2개 프로젝트)

- Engine
- UE4
- Games
- Dodgeball
 - 참조
 - 외부 종속성
 - Config
 - Source
 - Dodgeball
 - Dodgeball.Build.cs
 - Dodgeball.cpp
 - Dodgeball.h
 - DodgeballCharacter.cpp
 - DodgeballCharacter.h
 - DodgeballFunctionLibrary.cpp
 - DodgeballFunctionLibrary.h
 - DodgeballGameMode.cpp
 - DodgeballGameMode.h
 - DodgeballPlayerController.cpp
 - DodgeballPlayerController.h
 - DodgeballProjectile.cpp
 - DodgeballProjectile.h
 - EnemyCharacter.cpp
 - EnemyCharacter.h
 - HealthComponent.cpp
 - HealthComponent.h
 - HealthInterface.cpp

준비

소스 제어에 추가

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(I) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+Q) Dodgeball — □ ×

Develop Win64 로컬 Windows 디버거 Live Share

DodgeballPlayerController.cpp* DodgeballPlayerController.h RestartWidget.cpp RestartWidget.h Dodgeball.Build.cs

Dodgeball // Fill out your copyright notice in the Description page of Project Settings.

```
#include "DodgeballIPPlayerController.h"
#include "RestartWidget.h"

void ADodgeballIPPlayerController::ShowRestartWidget()
{
    if (BP_RestartWidget != nullptr)
    {
        SetPause(true);
        SetInputMode(FInputModeUIOnly());
        bShowMouseCursor = true;
        RestartWidget = CreateWidget<URestartWidget>(this, BP_RestartWidget);
        RestartWidget->AddToViewport();
    }
}
```

← three input modes: Game Only, Game and UI, and UI Only

Ctrl+S

100% 문제가 검색되지 않음 줄: 15 문자: 34 열: 40 탭 CRLF

준비

솔루션 탐색기

솔루션 탐색기 검색(Ctrl+.)

솔루션 'Dodgeball' (2/2개 프로젝트)

- Engine
- UE4
- Games
- Dodgeball
 - 참조
 - 외부 종속성
 - Config
 - Source
 - Dodgeball
 - Dodgeball.Build.cs
 - Dodgeball.cpp
 - Dodgeball.h
 - DodgeballCharacter.cpp
 - DodgeballCharacter.h
 - DodgeballFunctionLibrary.
 - DodgeballFunctionLibrary.
 - DodgeballGameMode.cpp
 - DodgeballGameMode.h
 - DodgeballPlayerController
 - DodgeballProjectile.cpp
 - DodgeballProjectile.h
 - EnemyCharacter.cpp
 - EnemyCharacter.h
 - HealthComponent.cpp
 - HealthComponent.h
 - HealthInterface.cpp

솔루션 탐색기 Git 변경 내용 ↑ 소스 제어에 추가 ↗ 1

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(I) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+Q) Dodgeball — □ ×

Develop Win64 로컬 Windows 디버거 Live Share

DodgeballPlayerController.cpp DodgeballPlayerController.h* × RestartWidget.cpp RestartWidget.h Dodgeball.Build.cs

Dodgeball // Fill out your copyright notice in the Description page of Project Settings.

```
1 // Fill out your copyright notice in the Description page of Project Settings.
2
3 #pragma once
4
5 #include "CoreMinimal.h"
6 #include "GameFramework/PlayerController.h"
7 #include "DodgeballPlayerController.generated.h"
8
9 /**
10  */
11 UCLASS()
12 class DODGEBALL_API ADodgeballPlayerController : public APlayerController
13 {
14     GENERATED_BODY()
15
16 public:
17
18     UPROPERTY(EditDefaultsOnly)
19     TSubclassOf<class URestartWidget> BP_RestartWidget;
20
21     void ShowRestartWidget();
22     void HideRestartWidget();
23
24 private:
25
26     UPROPERTY()
27     class URestartWidget* RestartWidget;
28 };
29
30
31
```

Ctrl+S

솔루션 탐색기

솔루션 탐색기 검색(Ctrl+.)

솔루션 탐색기 'Dodgeball' (2/2개 프로젝트)

- Engine
- UE4
- Games
- Dodgeball
 - 참조
 - 외부 종속성
 - Config
 - Source
 - Dodgeball
 - Dodgeball.Build.cs
 - Dodgeball.cpp
 - Dodgeball.h
 - DodgeballCharacter.cpp
 - DodgeballCharacter.h
 - DodgeballFunctionLibrary.cpp
 - DodgeballFunctionLibrary.h
 - DodgeballGameMode.cpp
 - DodgeballGameMode.h
 - DodgeballPlayerController.cpp
 - DodgeballPlayerController.h
 - DodgeballProjectile.cpp
 - DodgeballProjectile.h
 - EnemyCharacter.cpp
 - EnemyCharacter.h
 - HealthComponent.cpp
 - HealthComponent.h
 - HealthInterface.cpp

100 % 문제가 검색되지 않음 줄: 23 문자: 27 열: 30 템 CRLF

저장되었습니다. ↑ 소스 제어에 추가 ↗

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(I) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+Q) Dodgeball — □ ×

Develop Win64 로컬 Windows 디버거 Live Share

DodgeballPlayerController.cpp* DodgeballPlayerController.h RestartWidget.cpp RestartWidget.h Dodgeball.Build.cs

Dodgeball // Fill out your copyright notice in the Description page of Project Settings.

```
1 // Fill out your copyright notice in the Description page of Project Settings.
2
3
4 #include "DodgeballIPPlayerController.h"
5 #include "RestartWidget.h"
6
7 void ADodgeballIPPlayerController::ShowRestartWidget()
8 {
9     if (BP_RestartWidget != nullptr)
10    {
11        SetPause(true);
12        SetInputMode(FInputModeUIOnly());
13        bShowMouseCursor = true;
14        RestartWidget = CreateWidget<URestartWidget>(this, BP_RestartWidget);
15        RestartWidget->AddToViewport();
16    }
17 }
18
19 void ADodgeballIPPlayerController::HideRestartWidget()
20 {
21     RestartWidget->RemoveFromParent();
22     RestartWidget->Destruct();
23     SetPause(false);
24     SetInputMode(FInputModeGameOnly());
25     bShowMouseCursor = false;
26 }
```

Ctrl+S

슬루션 탐색기

슬루션 탐색기 검색(Ctrl+.)

슬루션 탐색기 'Dodgeball' (2/2개 프로젝트)

- Engine
- UE4
- Games
- Dodgeball
 - 참조
 - 외부 종속성
 - Config
 - Source
 - Dodgeball
 - Dodgeball.Build.cs
 - Dodgeball.cpp
 - Dodgeball.h
 - DodgeballCharacter.cpp
 - DodgeballCharacter.h
 - DodgeballFunctionLibrary.cpp
 - DodgeballFunctionLibrary.h
 - DodgeballGameMode.cpp
 - DodgeballGameMode.h
 - DodgeballPlayerController.cpp
 - DodgeballProjectile.cpp
 - DodgeballProjectile.h
 - EnemyCharacter.cpp
 - EnemyCharacter.h
 - HealthComponent.cpp
 - HealthComponent.h
 - HealthInterface.cpp

100 % 문제가 검색되지 않음 줄: 25 문자: 27 열: 30 템 CRLF

저장되었습니다. ↑ 소스 제어에 추가 ▲ 1

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(I) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+Q) Dodgeball — □ ×

Develop Win64 로컬 Windows 디버거 Live Share

DodgeballPlayerController.cpp* DodgeballPlayerController.h RestartWidget.cpp RestartWidget.h DodgeballCharacter.cpp* ✖

Copyright Epic Games, Inc. All Rights Reserved.

```
1 // Copyright Epic Games, Inc. All Rights Reserved.
2
3 #include "DodgeballCharacter.h"
4 #include "HeadMountedDisplayFunctionLibrary.h"
5 #include "Camera/CameraComponent.h"
6 #include "Components/CapsuleComponent.h"
7 #include "Components/InputComponent.h"
8 #include "HealthComponent.h"
9 #include "GameFramework/CharacterMovementComponent.h"
10 #include "GameFramework/Controller.h"
11 #include "GameFramework/SpringArmComponent.h"
12 #include "Kismet/KismetSystemLibrary.h"
13 #include "DodgeballPlayerController.h"
14
15 // ADodgeballCharacter
16
17 ADodgeballCharacter::ADodgeballCharacter()
18 {
19     // Set size for collision capsule
20     GetCapsuleComponent()->InitCapsuleSize(42.f, 96.0f);
21
22     // set our turn rates for input
23     BaseTurnRate = 45.f;
24     BaseLookUpRate = 45.f;
25
26     // Don't rotate when the controller rotates. Let that just affect the camera.
27     bUseControllerRotationPitch = false;
28     bUseControllerRotationYaw = false;
29     bUseControllerRotationRoll = false;
30
31     // Configure character movement
32     GetCharacterMovement()->b0rrientRotationToMovement = true; // Character moves in the direction of input...
33     GetCharacterMovement()->RotationRate = FRotator(0.0f, 540.0f, 0.0f); // ...at this rotation rate
34     GetCharacterMovement()->JumpZVelocity = 600.f;
35 }
```

100 % 문제가 검색되지 않음 출: 65 문자: 41 열: 47 탭 CRLF

솔루션 탐색기

솔루션 탐색기 검색(Ctrl+.)

솔루션 'Dodgeball' (2/2개 프로젝트)

- Engine
- UE4
- Games
- Dodgeball
 - 참조
 - 외부 종속성
 - Config
 - Source
 - Dodgeball
 - Dodgeball.Build.cs
 - Dodgeball.cpp
 - Dodgeball.h
 - DodgeballCharacter.cpp
 - DodgeballCharacter.h
 - DodgeballFunctionLibrary.h
 - DodgeballFunctionLibrary.cpp
 - DodgeballGameMode.cpp
 - DodgeballGameMode.h
 - DodgeballPlayerController.cpp
 - DodgeballProjectile.cpp
 - DodgeballProjectile.h
 - EnemyCharacter.cpp
 - EnemyCharacter.h
 - HealthComponent.cpp
 - HealthComponent.h
 - HealthInterface.cpp

솔루션 탐색기 Git 변경 내용

준비 ↑ 소스 제어에 추가 ↗

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(I) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+Q) Dodgeball — □ ×

Develop Win64 로컬 Windows 디버거 Live Share

DodgeballPlayerController.cpp* DodgeballPlayerController.h RestartWidget.cpp RestartWidget.h DodgeballCharacter.cpp* DodgeballCharacter.h

47 // Create a follow camera
48 FollowCamera = CreateDefaultSubobject<UCameraComponent>(TEXT("FollowCamera"));
49 FollowCamera->SetupAttachment(CameraBoom, USpringArmComponent::SocketName); // Attach the camera to the end of the boom and let
50 FollowCamera->bUsePawnControlRotation = false; // Camera does not rotate relative to arm
51
52 HealthComponent = CreateDefaultSubobject<UHealthComponent>(TEXT("Health Component"));
53
54 // Note: The skeletal mesh and anim blueprint references on the Mesh component (inherited from Character)
55 // are set in the derived blueprint asset named MyCharacter (to avoid direct content references in C++)
56
57 }
58
59 void ADodgeballCharacter::OnDeath_Implementation()
60 {
61 //UKismetSystemLibrary::QuitGame(this, nullptr, EQuitPreference::Quit, true);
62 ADodgeballPlayerController* PlayerController = Cast<ADodgeballPlayerController>(GetController());
63 if (PlayerController != nullptr)
64 {
65 PlayerController->ShowRestartWidget();
66 }
67 }
68
69 //
70 // Input
71
72 void ADodgeballCharacter::SetupPlayerInputComponent(class UInputComponent* PlayerInputComponent)
73 {
74 // Set up gameplay key bindings
75 check(PlayerInputComponent);
76 //PlayerInputComponent->BindAction("Jump", IE_Pressed, this, &ACharacter::Jump);
77 //PlayerInputComponent->BindAction("Jump", IE_Released, this, &ACharacter::StopJumping);
78
79 PlayerInputComponent->BindAxis("MoveForward", this, &ADodgeballCharacter::MoveForward);
80 PlayerInputComponent->BindAxis("MoveRight", this, &ADodgeballCharacter::MoveRight);
81 }

Ctrl+S

100% 문제가 검색되지 않음 출: 65 문자: 41 열: 47 탭 CRLF

슬루션 탐색기

슬루션 탐색기 검색(Ctrl+Shift+F)

슬루션 탐색기 'Dodgeball' (2/2개 프로젝트)

- Engine
- UE4
- Games
- Dodgeball
 - 참조
 - 외부 종속성
 - Config
 - Source
 - Dodgeball
 - Dodgeball.Build.cs
 - Dodgeball.cpp
 - Dodgeball.h
 - DodgeballCharacter.cpp
 - DodgeballCharacter.h
 - DodgeballFunctionLibrary.h
 - DodgeballFunctionLibrary.cpp
 - DodgeballGameMode.cpp
 - DodgeballGameMode.h
 - DodgeballPlayerController.h
 - DodgeballPlayerController.cpp
 - DodgeballProjectile.cpp
 - DodgeballProjectile.h
 - EnemyCharacter.cpp
 - EnemyCharacter.h
 - HealthComponent.cpp
 - HealthComponent.h
 - HealthInterface.cpp

슬루션 탐색기 Git 변경 내용

준비 ↑ 소스 제어에 추가 ▲

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(I) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+Q) Dodgeball — □ ×

Develop Win64 로컬 Windows 디버거 Live Share

DodgeballPlayerController.cpp* DodgeballPlayerController.h RestartWidget.cpp* X RestartWidget.h DodgeballCharacter.cpp

Dodgeball // Fill out your copyright notice in the Description page of Project Settings.

```
#include "RestartWidget.h"
#include "Kismet/GameplayStatics.h"
#include "Components/Button.h"
#include "DodgeballIPController.h"

void URestartWidget::OnRestartClicked()
{
    ADodgeballIPController* PlayerController = Cast<ADodgeballIPController>(GetOwningPlayer());
    if (PlayerController != nullptr)
    {
        PlayerController->HideRestartWidget();
    }

    UGameplayStatics::OpenLevel(this, FName(*UGameplayStatics::GetCurrentLevelName(this)));
}

void URestartWidget::NativeOnInitialized()
{
    Super::NativeOnInitialized();

    if (RestartButton != nullptr)
    {
        RestartButton->OnClicked.AddDynamic(this, &URestartWidget::OnRestartClicked);
    }
}
```

Ctrl+S

100% 문제가 검색되지 않음 줄: 14 문자: 41 열: 47 탭 CRLF

저장되었습니다. ↑ 소스 제어에 추가 ↗

슬루션 탐색기

슬루션 탐색기 검색(Ctrl+.)

슬루션 탐색기 'Dodgeball' (2/2개 프로젝트)

- Engine
- UE4
- Games
- Dodgeball
- 참조
- 외부 종속성
- Config
- Source
- Dodgeball
- Dodgeball.Build.cs
- Dodgeball.cpp
- Dodgeball.h
- DodgeballCharacter.cpp
- DodgeballCharacter.h
- DodgeballFunctionLibrary.
- DodgeballFunctionLibrary.
- DodgeballGameMode.cpp
- DodgeballGameMode.h
- DodgeballPlayerController
- DodgeballProjectile.cpp
- DodgeballProjectile.h
- EnemyCharacter.cpp
- EnemyCharacter.h
- HealthComponent.cpp
- HealthComponent.h
- HealthInterface.cpp

슬루션 탐색기 Git 변경 내용

The screenshot shows the Unreal Engine 4 Editor interface with the following details:

- Top Bar:** 파일(F), 편집(E), 보기(V), Git(G), 프로젝트(P), 빌드(B), 디버그(D), 테스트(S), 분석(N), 도구(T), 확장(X), 창(W), 도움말(H), 검색 (Ctrl+Q), Dodgeball.
- Left Sidebar:** 서버, 팀생기, 도구상자.
- Central Area:** Two code editors are visible:
 - DodgeballPlayerController.cpp***: Contains C++ code for a player controller, including methods like `OnRestartClick` and `NativeOnInitialized`.
 - DodgeballPlaye...**: Partially visible code editor.
- Build Menu (Open):** 솔루션 빌드(B) (highlighted with a red arrow), 솔루션 다시 빌드, 솔루션 정리(C), 솔루션의 전체 프로그램 데이터베이스 파일 빌드, 솔루션에서 코드 분석 실행(Y), Dodgeball 빌드(U), Dodgeball 다시 빌드(E), Dodgeball 정리(N), Dodgeball에서 코드 분석 실행(A), 프로젝트만(), 일괄 빌드(T)..., 구성 관리자(O)..., 컴파일(M), 파일에서 코드 분석 실행(F).
- Right Sidebar:** 솔루션 탐색기, 솔루션 탐색기 검색(Ctrl+Shift+F). The solution browser shows the project structure:
 - Engine
 - UE4
 - Games
 - Dodgeball
 - 참조
 - 외부 종속성
 - Config
 - Source
 - Dodgeball
 - Dodgeball.Build.cs
 - Dodgeball.cpp
 - Dodgeball.h
 - DodgeballCharacter.cpp
 - DodgeballCharacter.h
 - DodgeballFunctionLibrary...
 - DodgeballFunctionLibrary...
 - DodgeballGameMode.cpp
 - DodgeballGameMode.h
 - DodgeballPlayerController...
 - DodgeballProjectile.cpp
 - DodgeballProjectile.h
 - EnemyCharacter.cpp
 - EnemyCharacter.h
 - HealthComponent.cpp
 - HealthComponent.h
 - HealthInterface.cpp
- Bottom Status Bar:** 100%, 문제가 검색되지 않음, 줄: 14, 문자: 41, 열: 47, 탭, CRLF, 솔루션 탐색기, Git 변경 내용.
- Bottom Right Corner:** 저장되었습니다., 소스 제어에 추가, 1.



Exercise 8.05: Setting Up The DodgeballPlayerController Blueprint Class

The screenshot shows the Unreal Engine Content Browser interface. On the left, a sidebar lists various asset types: Cinematic, Visual Effects, Geometry, Volumes, and All Classes. Below this is the Content Browser toolbar with buttons for Add/Import, Save All, and navigation. The Content Browser tree view shows a folder structure: Content > ThirdPersonCPP > Blueprints. A red box highlights the 'Blueprints' folder under 'ThirdPersonCPP'. The main Content Browser area displays several blueprints: BP_DodgeballProjectile, BP_EnemyCharacter, BP_GhostWall, BP_RestartWidget, BP_VictoryBox, BP_Wall, and ThirdPersonCharacter. A red box also highlights the 'Blueprints' folder in the Content Browser tree. In the center, a 3D preview window shows a character in a room with large text 'Third Person' overlaid. To the right, a context menu is open, with the 'Blueprint Class' option highlighted by a yellow box and a red border. The menu also includes options for Folder, Import Asset, Create Basic Asset (Level, Material, Particle System), and Create Advanced Asset (Animation, Artificial Intelligence, Blendables, Blueprints, Editor Utilities, Foliage, FX, Live Link, Materials & Textures, Media, Miscellaneous, Paper2D, Physics, Sounds, User Interface). The top right corner of the screen shows a list of actors: Floor, Wall6, Wall7, Wall8, Wall9, Walkway, and a folder named 'Walkway'. There are 27 actors in total. A 'View Options' dropdown is also visible.

Cinematic
Visual Effects
Geometry
Volumes
All Classes

Add/Import Save All Content > ThirdPersonCPP > Blueprints

Content

- Geometry
- Mannequin
- Physics
- StarterContent
- ThirdPerson
- ThirdPersonCPP
 - Blueprints
 - Maps
- C++ Classes
- Dodgeball

Filters Search Blueprints

BP_DodgeballProjectile BP_EnemyCharacter BP_GhostWall BP_RestartWidget BP_VictoryBox BP_Wall ThirdPersonCharacter

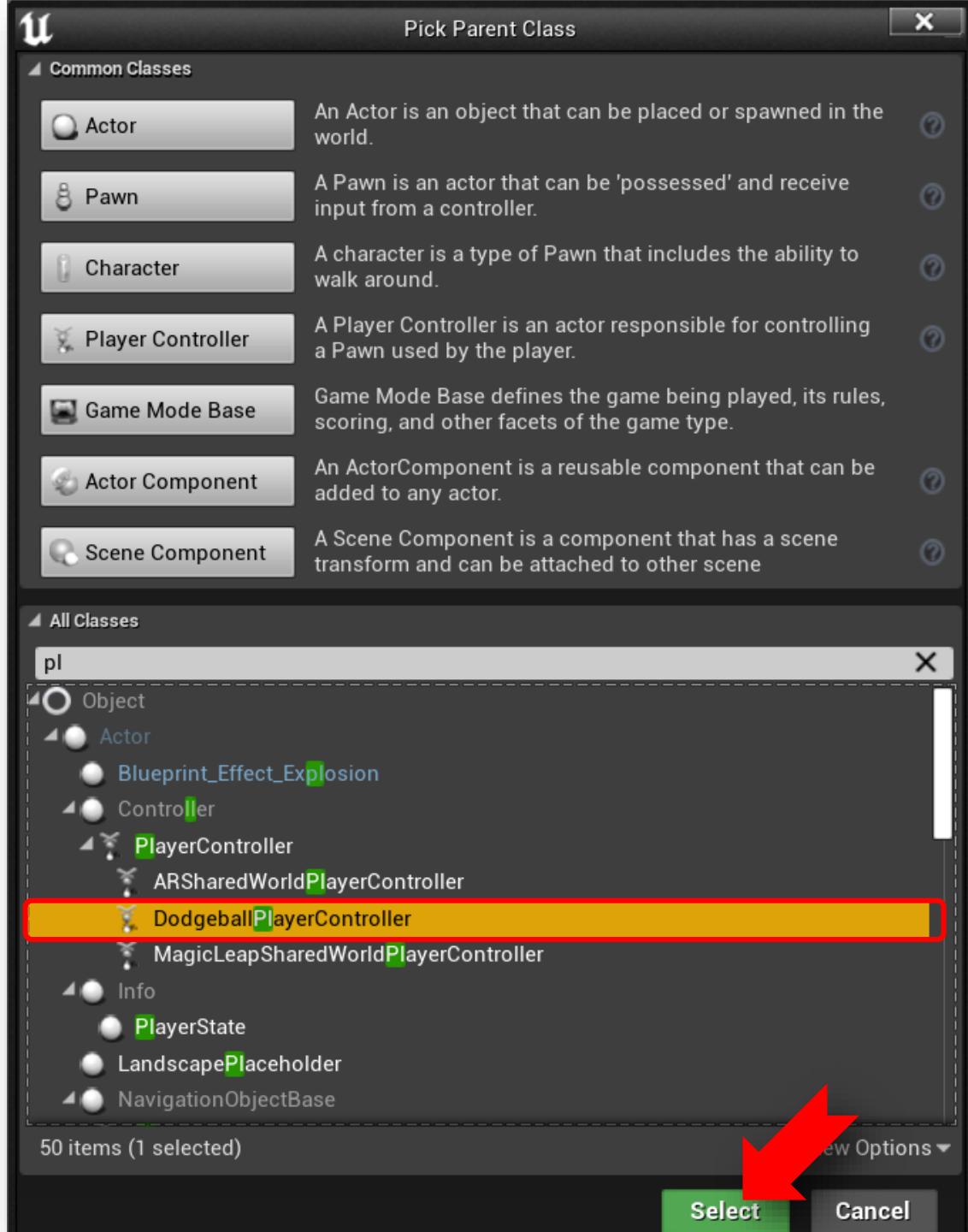
Folder New Folder Import Asset Import to /Game/ThirdPersonCPP/Blueprints... Create Basic Asset Blueprint Class Level Material Particle System Create Advanced Asset Animation Artificial Intelligence Blendables Blueprints Editor Utilities Foliage FX Live Link Materials & Textures Media Miscellaneous Paper2D Physics Sounds User Interface

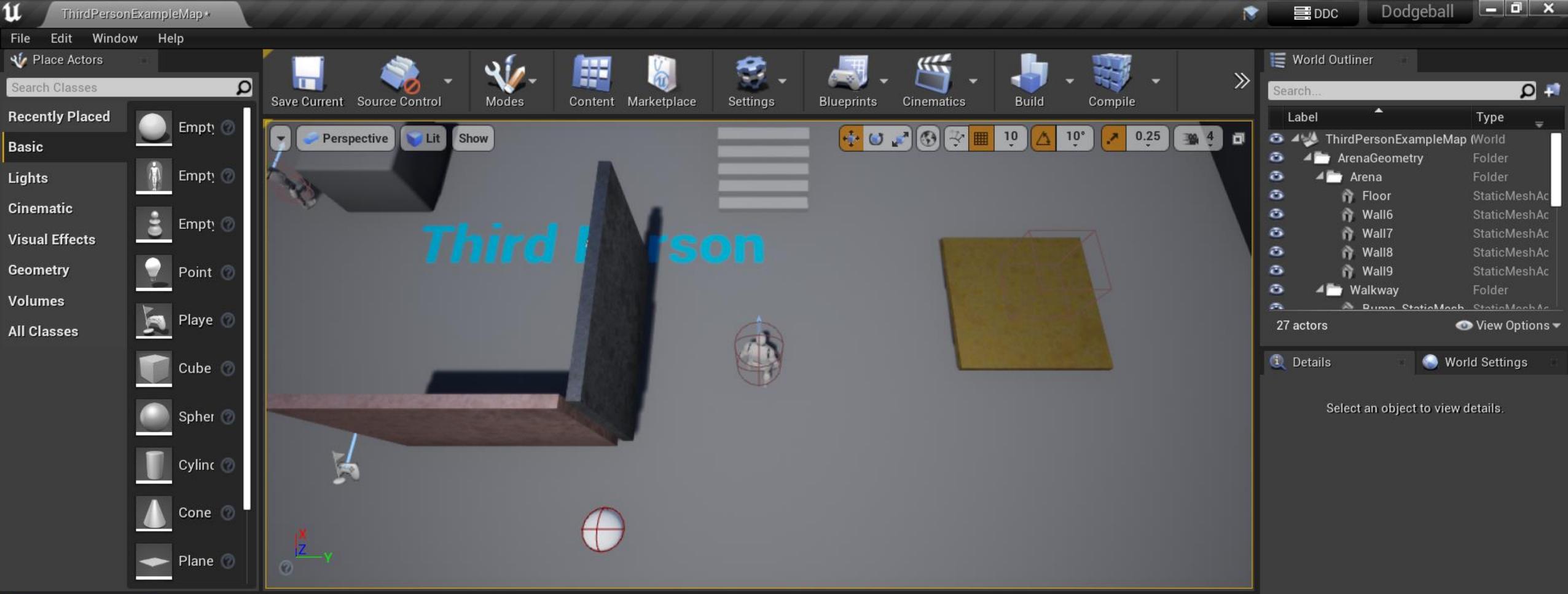
Floor Wall6 Wall7 Wall8 Wall9 Walkway

27 actors View Options

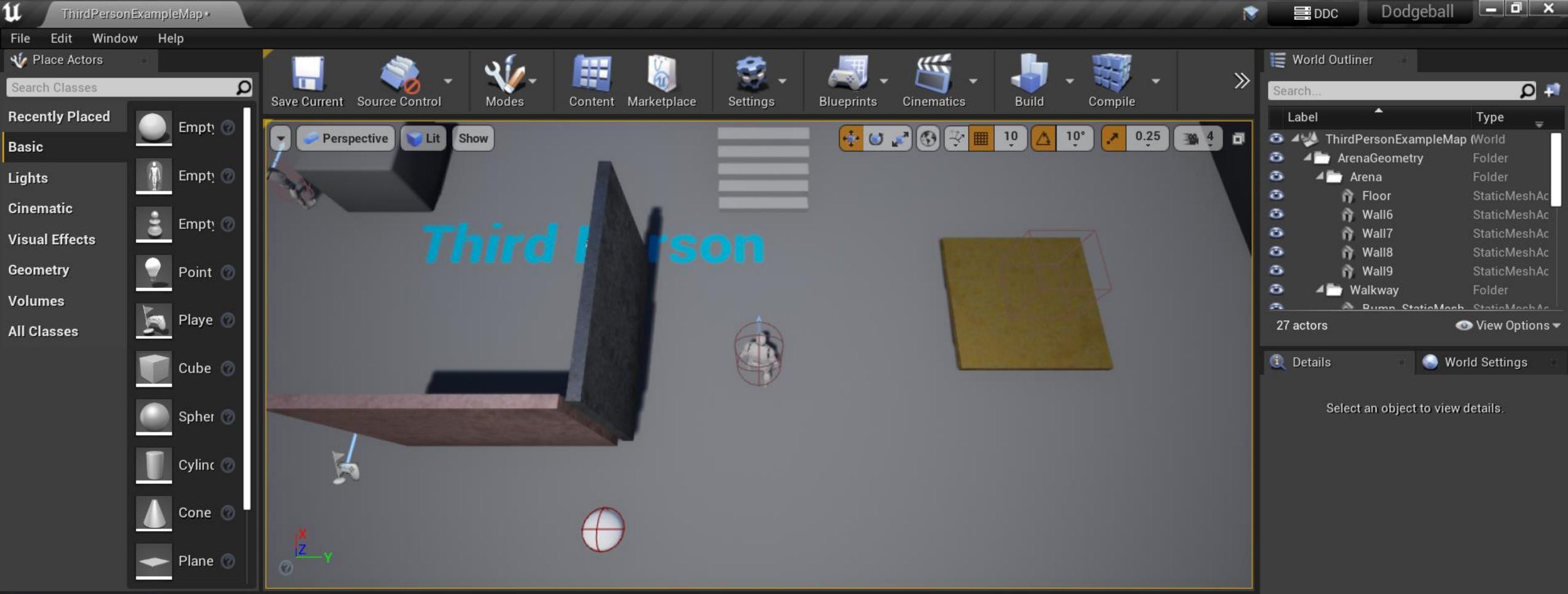
Details World Settings Select an object to view details.

Right-Click View Options

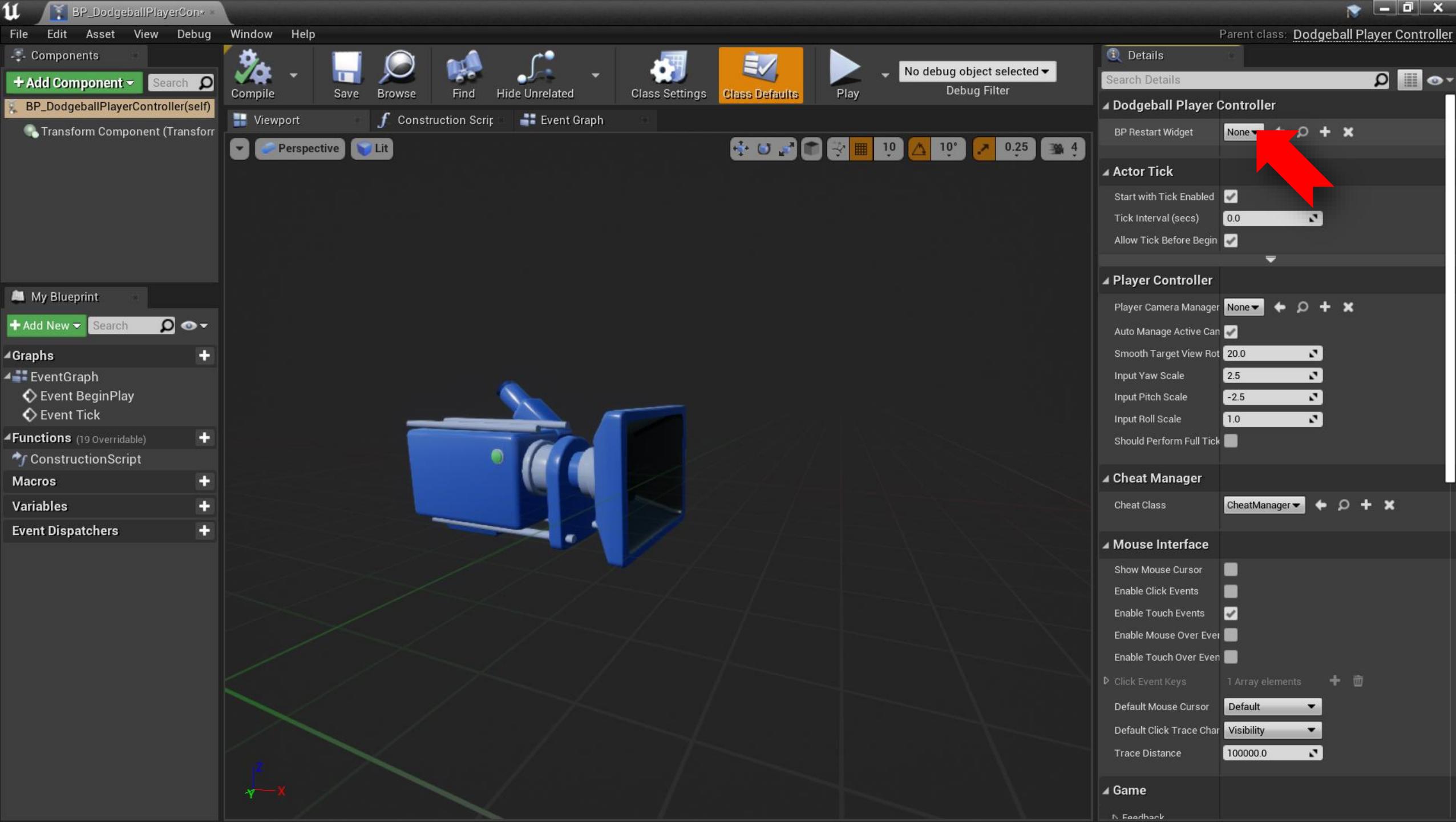


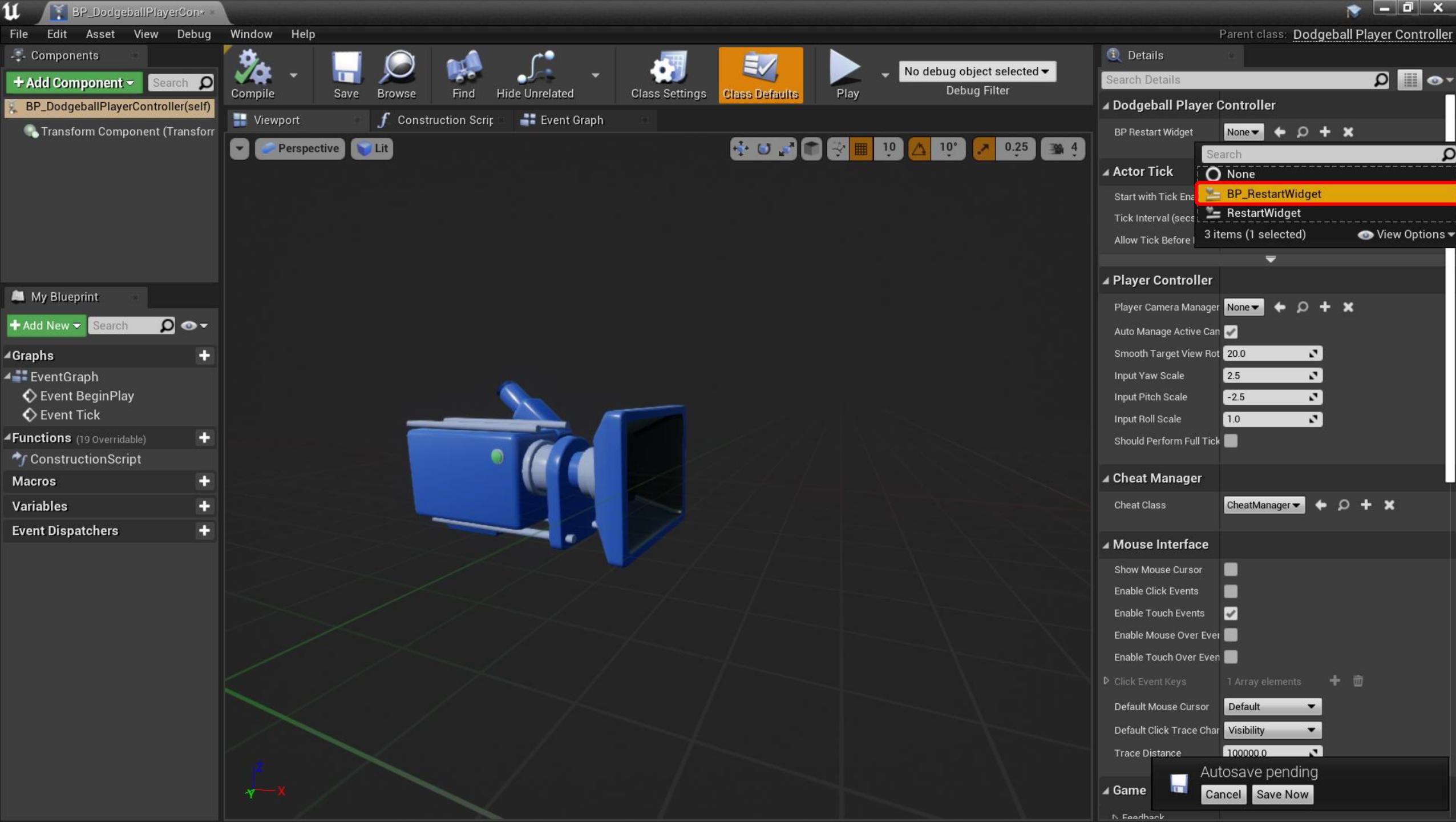


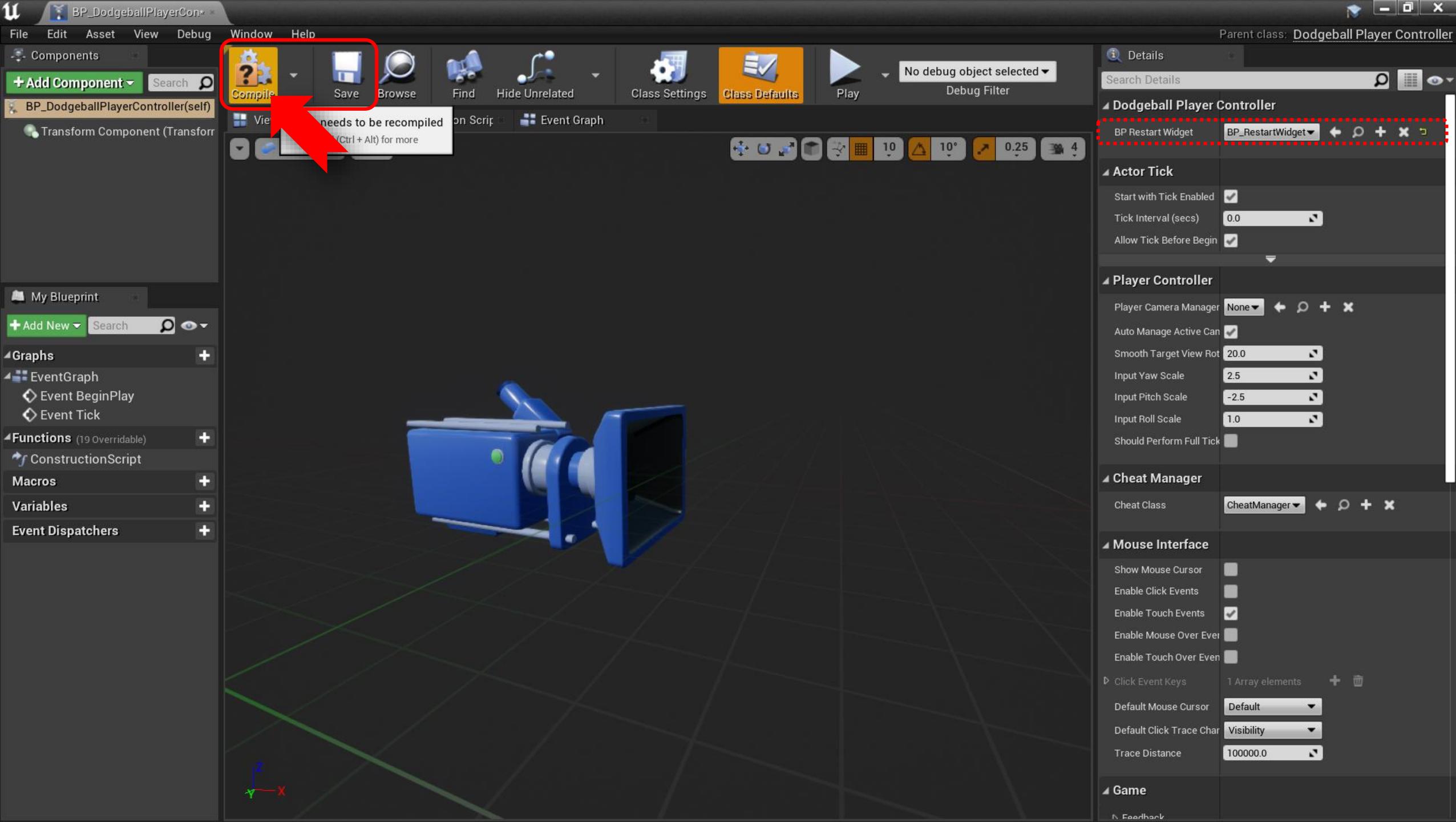
The screenshot shows the "Content Browser" interface. The top bar includes "Add/Import" and "Save All" buttons, and a search bar. The left sidebar shows the project structure: Content, Geometry, Mannequin, Physics, StarterContent, ThirdPerson, and ThirdPersonCPP (selected). The "Blueprints" folder under ThirdPersonCPP is also selected. The main area displays a grid of asset thumbnails: BP_DodgeballProjectile, BP_EnemyCharacter, BP_GhostWall, BP_RestartWidget, BP_VictoryBox, BP_Wall, NewBlueprint (selected and highlighted with a red dashed box), and ThirdPersonCharacter. A status bar at the bottom indicates "8 items (1 selected)".

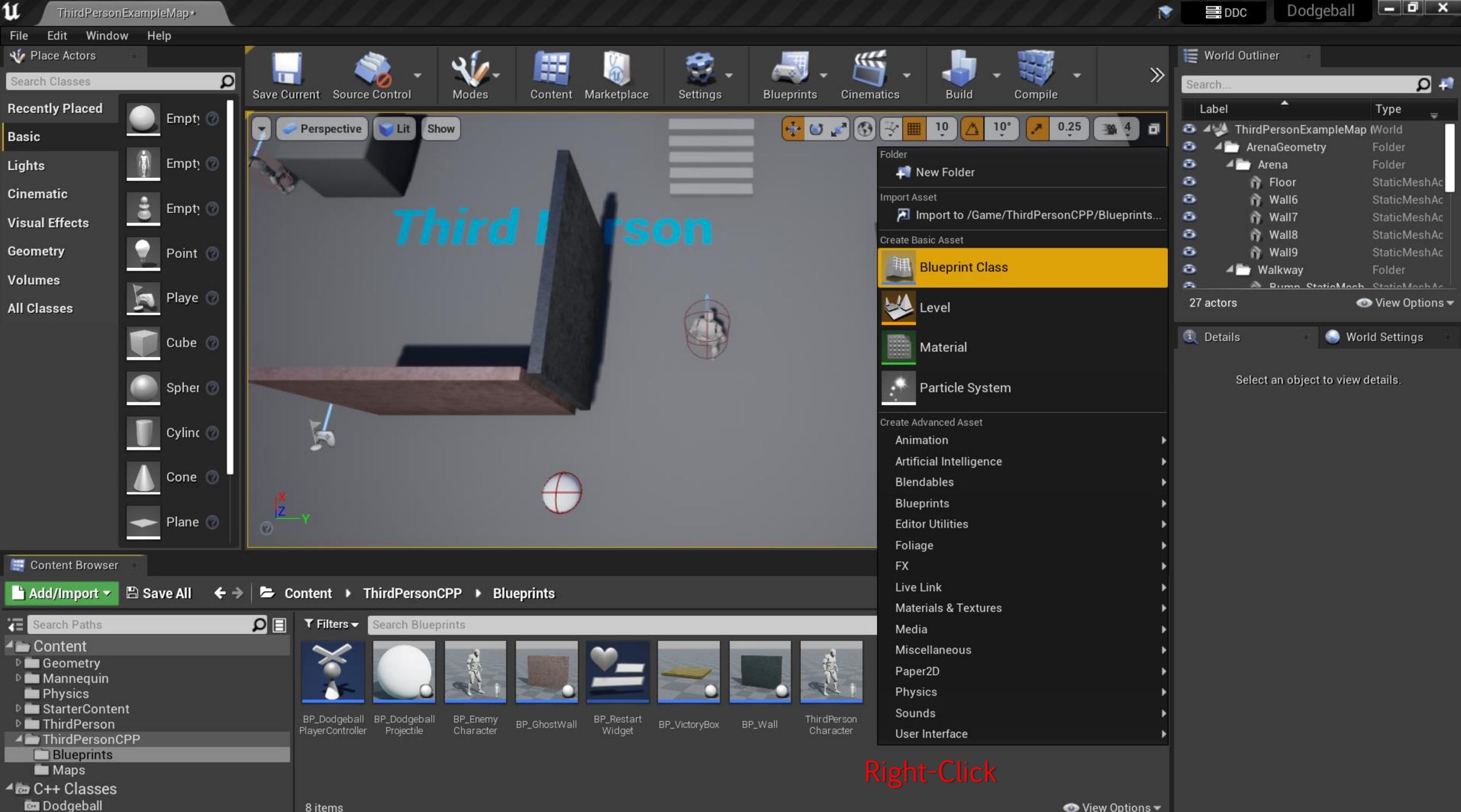


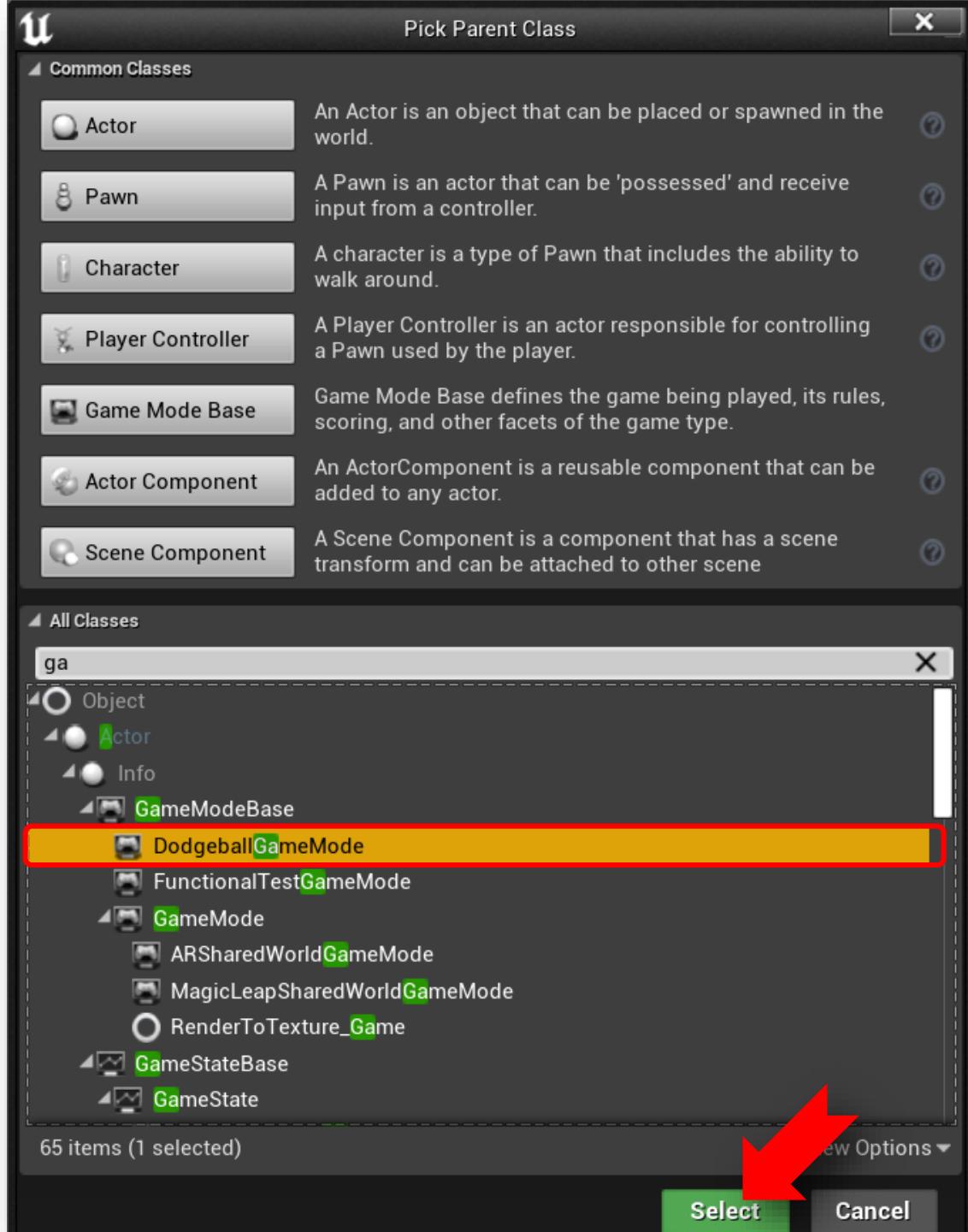
The screenshot shows the Content Browser interface. The top bar includes Add/Import, Save All, and Content navigation. The left sidebar shows the file structure under "Content" with "ThirdPersonCPP" selected. The main area displays various blueprint assets: BP_DodgeballPlayerController (selected and highlighted with a red box), BP_DodgeballProjectile, BP_EnemyCharacter, BP_GhostWall, BP_RestartWidget, BP_VictoryBox, BP_Wall, and ThirdPersonCharacter. The bottom status bar indicates 8 items (1 selected) and View Options.

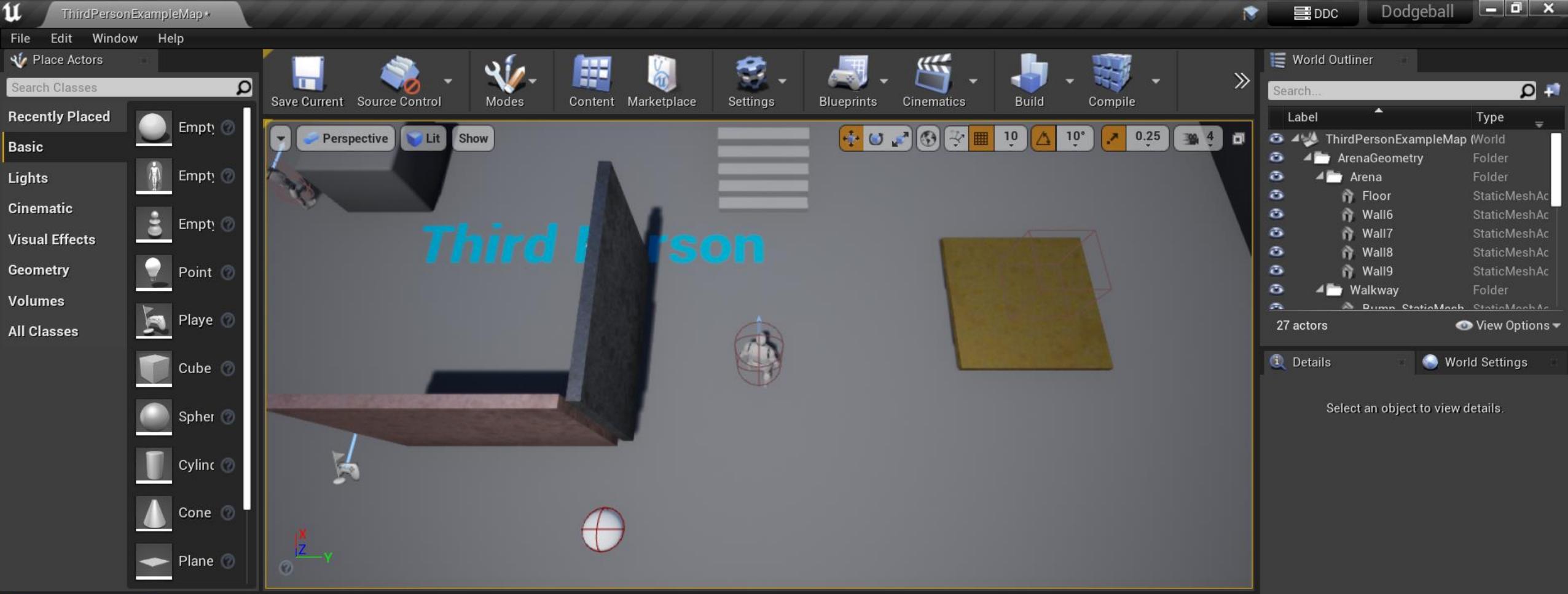








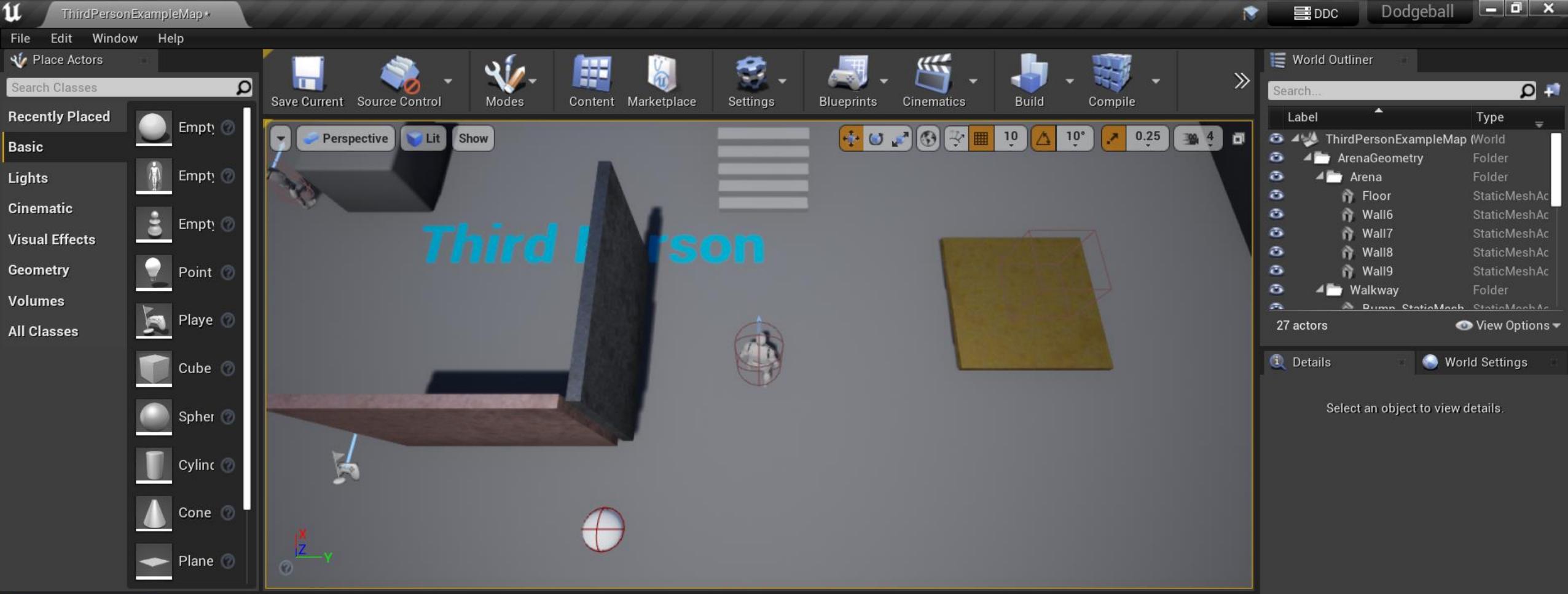




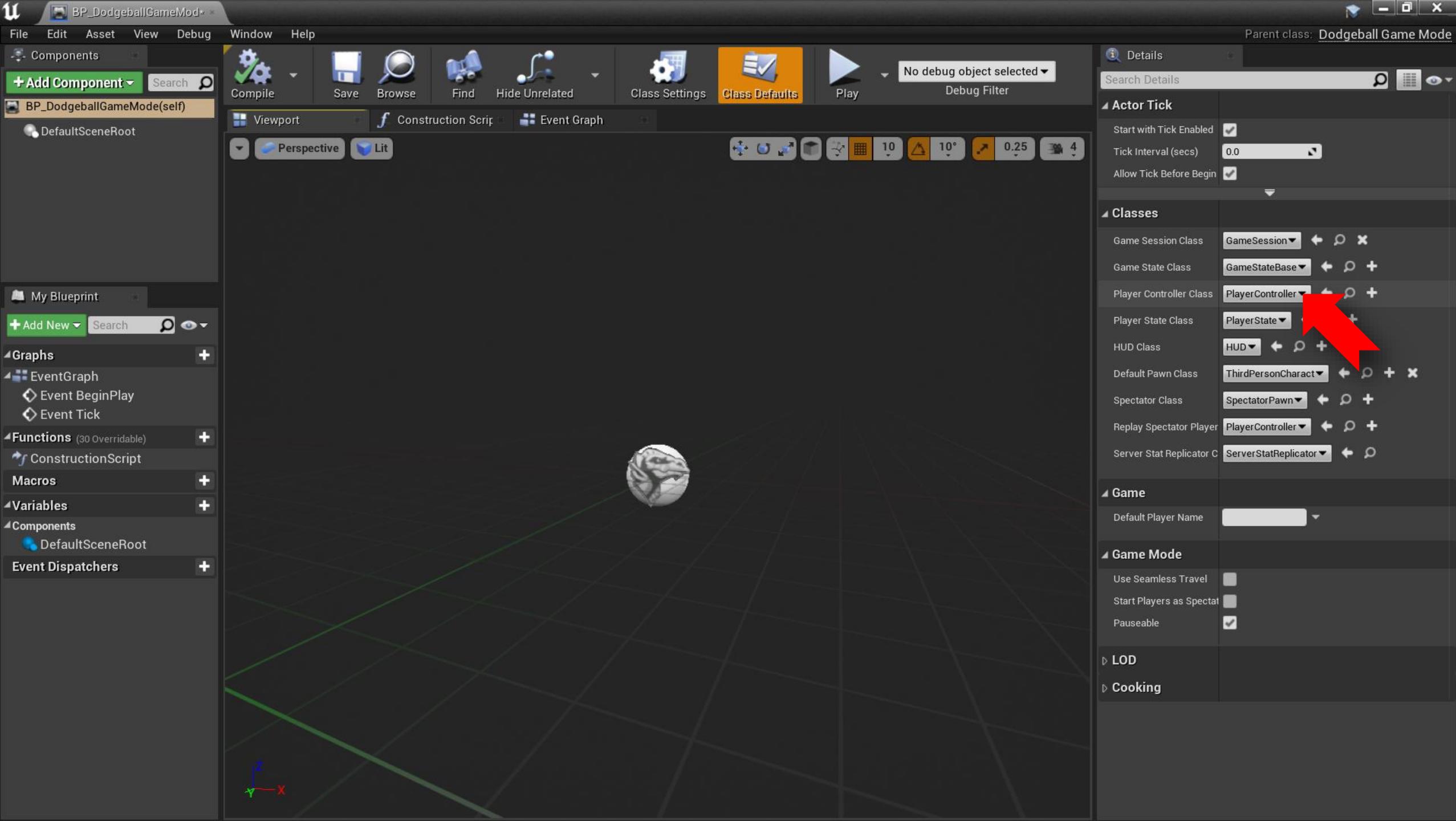
The Content Browser is open, showing the "Blueprints" section under "ThirdPersonCPP". The left sidebar shows the project structure with "Content" expanded, revealing "Geometry", "Mannequin", "Physics", "StarterContent", "ThirdPerson", and "ThirdPersonCPP" (which is selected). The "Blueprints" folder under "ThirdPersonCPP" is also selected.

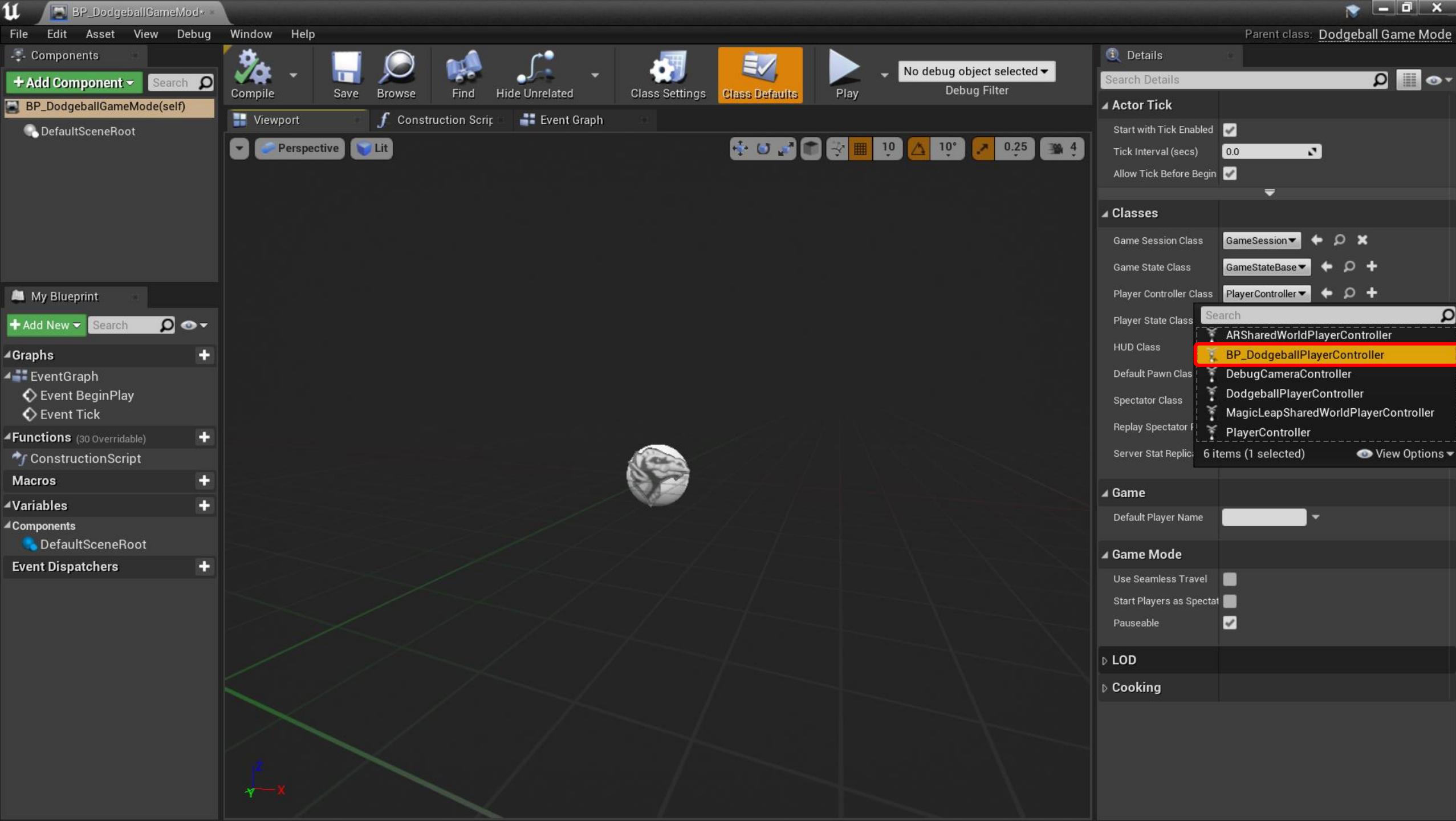
The main area displays a grid of Blueprint icons. One icon, labeled "NewBlueprint", is highlighted with a red dashed box. Other icons include BP_DodgeballPlayerController, BP_DodgeballProjectile, BP_EnemyCharacter, BP_GhostWall, BP_RestartWidget, BP_VictoryBox, and BP_Wall.

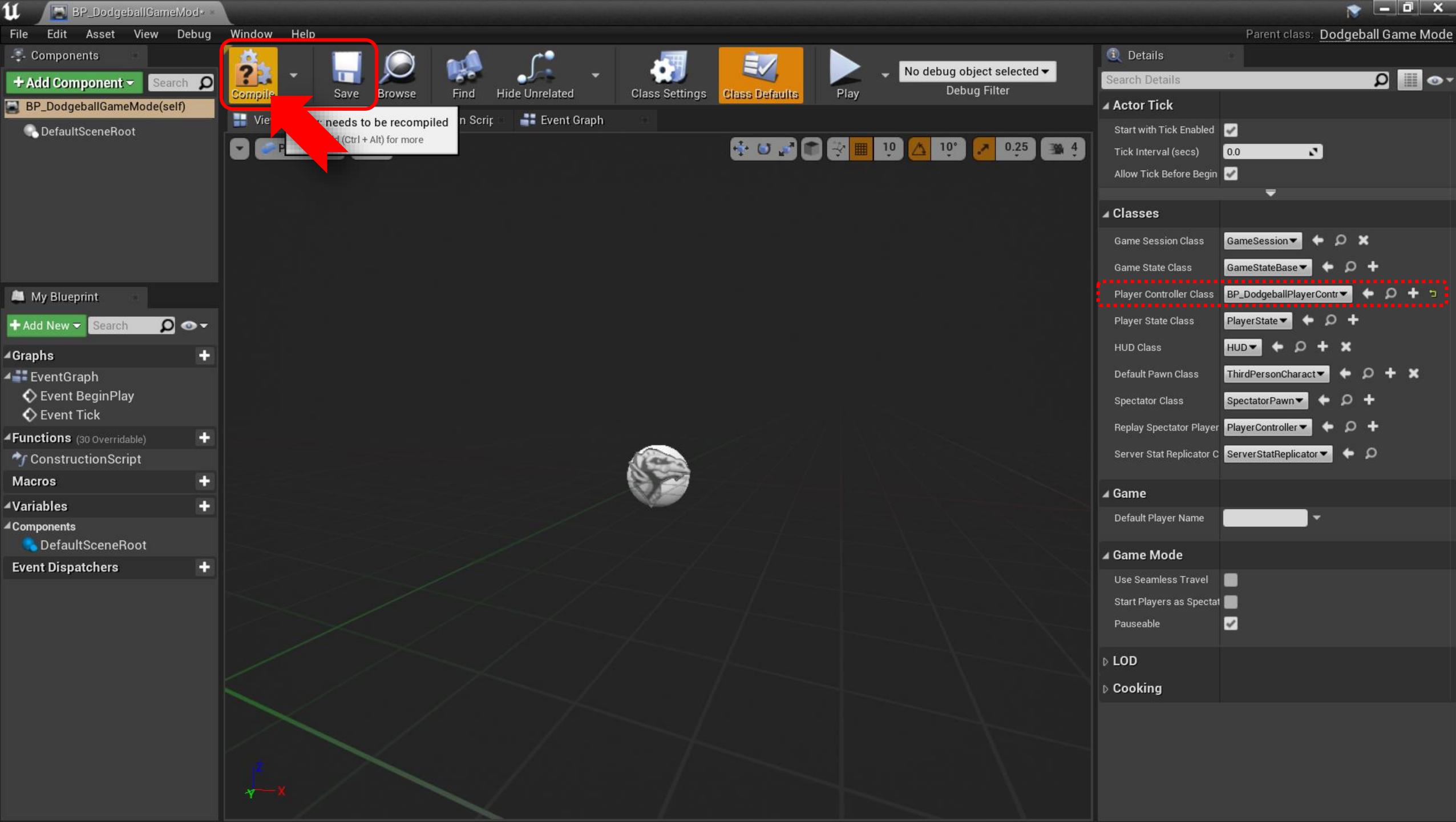
At the bottom of the Content Browser, it says "9 items (1 selected)" and "View Options".

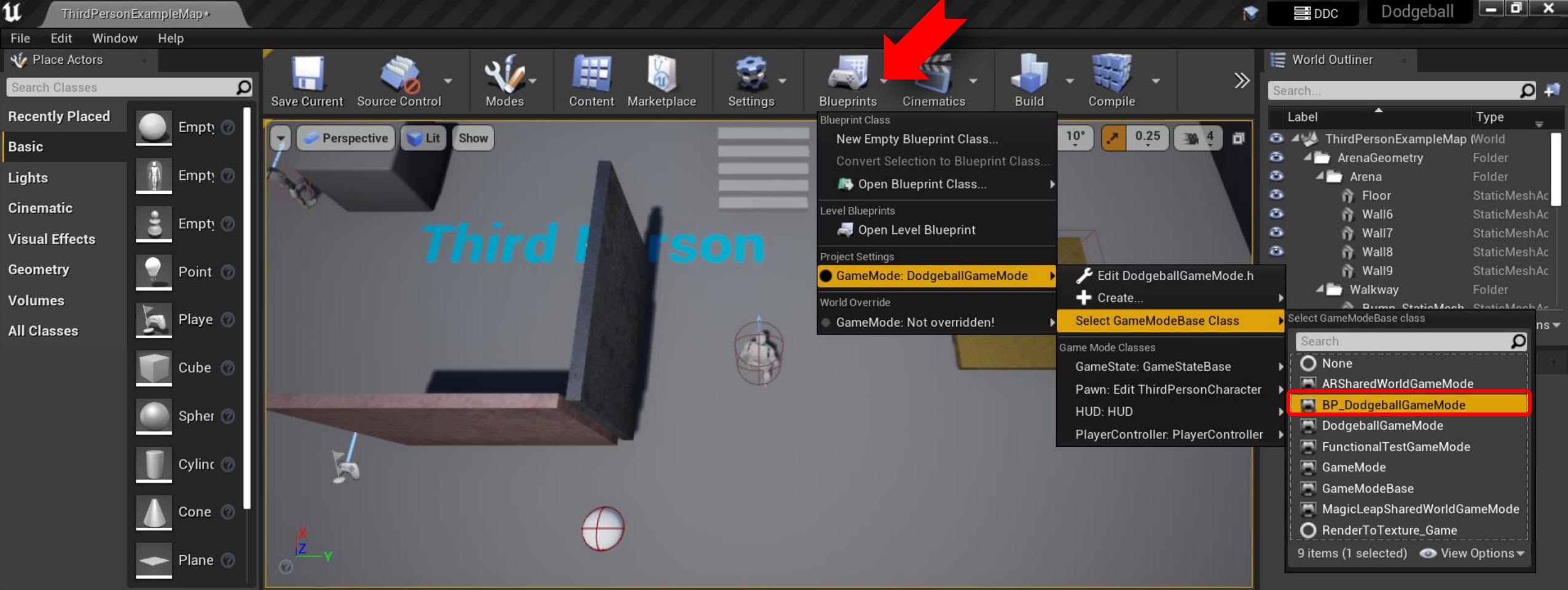


The screenshot shows the Content Browser panel. The top bar has buttons for "Add/Import", "Save All", and navigation. The left sidebar shows the file tree: "Content", "Geometry", "Mannequin", "Physics", "StarterContent", "ThirdPerson", "ThirdPersonCPP" (selected), "Blueprints", and "Maps". The main area displays various blueprint assets: "BP_Dodgeball GameMode" (highlighted with a red box), "BP_Dodgeball PlayerController", "BP_Dodgeball Projectile", "BP_EnemyCharacter", "BP_GhostWall", "BP_RestartWidget", "BP_VictoryBox", "BP_Wall", and "ThirdPersonCharacter". A search bar at the top says "Search Blueprints". The bottom left shows "9 items (1 selected)" and the bottom right has a "View Options" dropdown.









Add/Import Save All Content > ThirdPersonCPP > Blueprints

Filters Search Blueprints

Content

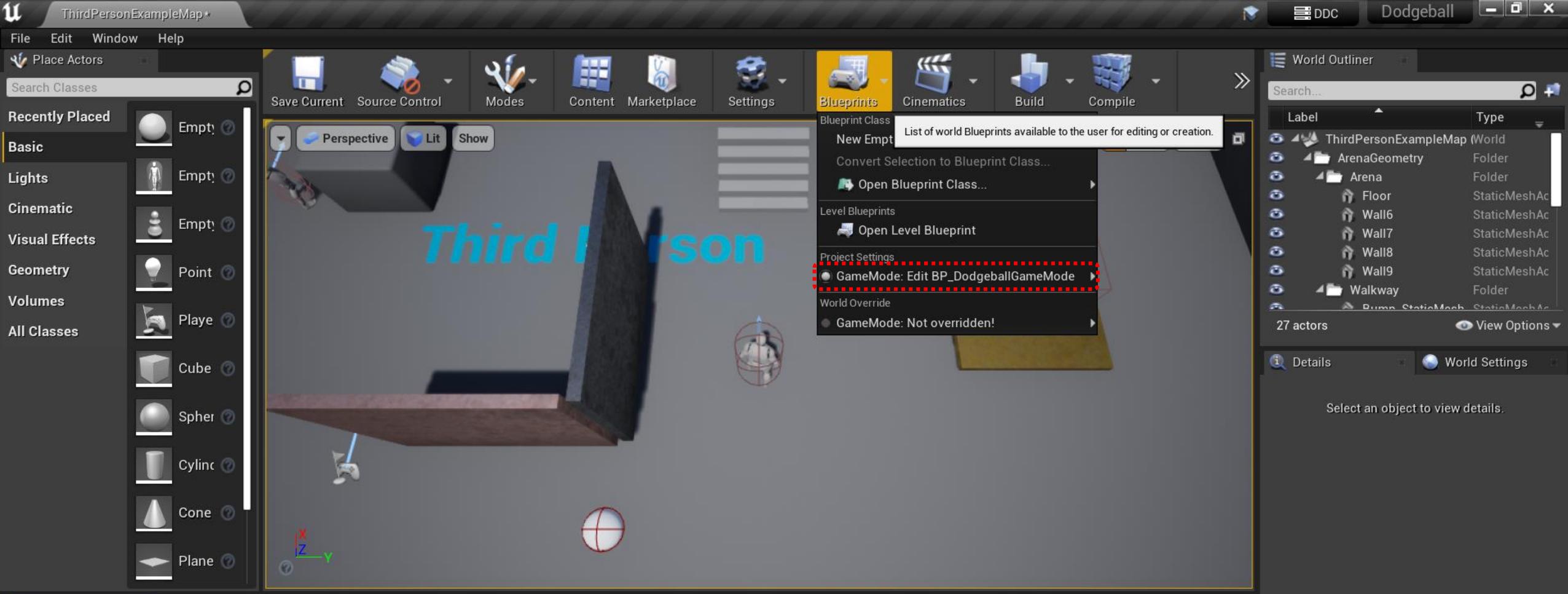
- Geometry
- Mannequin
- Physics
- StarterContent
- ThirdPerson
- ThirdPersonCPP
- Blueprints
- Maps

C++ Classes

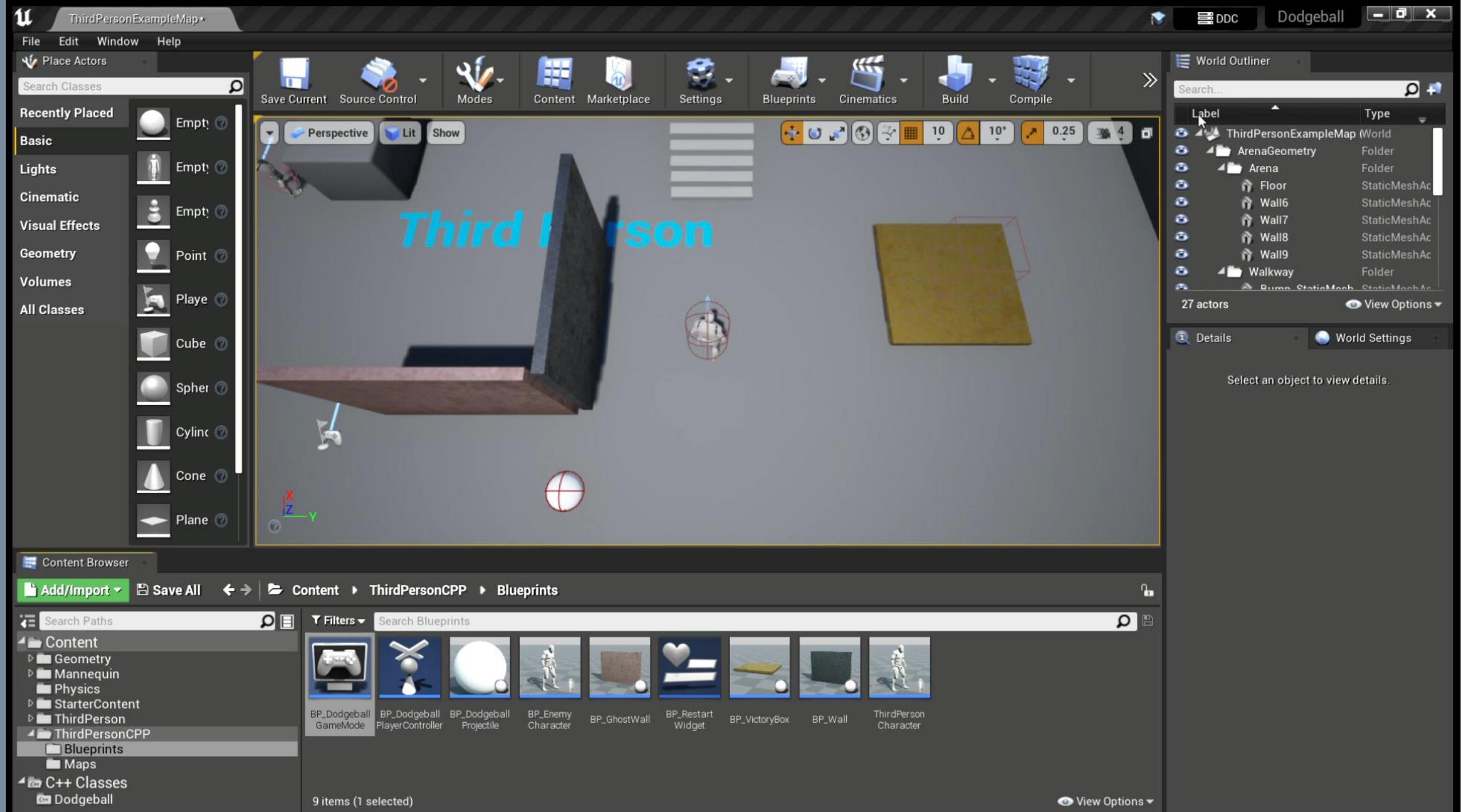
Dodgeball

9 items (1 selected) View Options

BP_DodgeballGameMode BP_DodgeballPlayerController BP_DodgeballProjectile BP_EnemyCharacter BP_GhostWall BP_RestartWidget BP_VictoryBox BP_Wall ThirdPersonCharacter



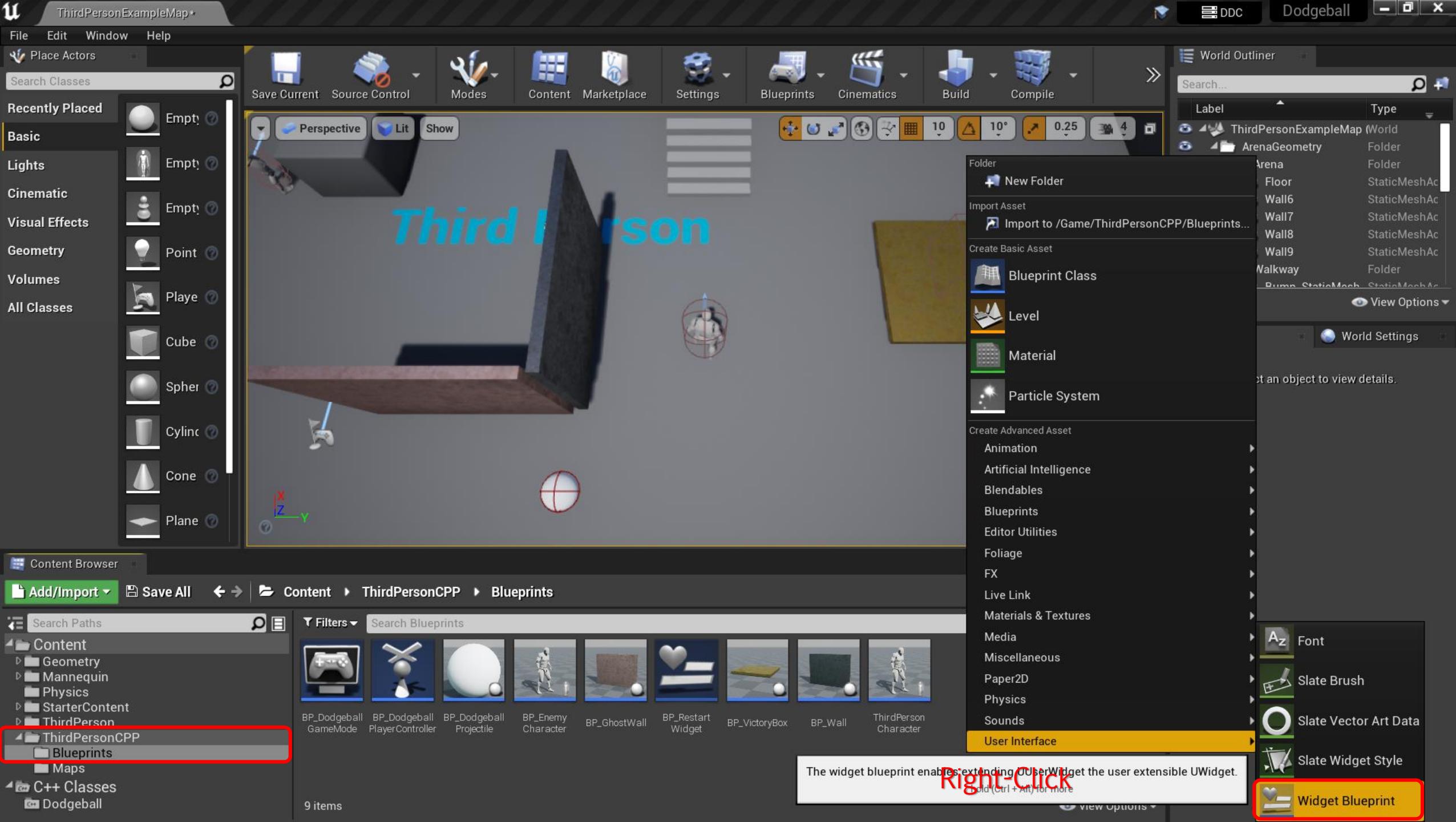
The screenshot shows the Content Browser interface. The top bar includes Add/Import, Save All, and Content navigation. The left sidebar shows the Content tree with folders for Content, Geometry, Mannequin, Physics, StarterContent, ThirdPerson, and ThirdPersonCPP (selected). The right sidebar shows a grid of blueprint assets: BP_Dodgeball GameMode, BP_Dodgeball PlayerController, BP_Dodgeball Projectile, BP_Enemy Character, BP_GhostWall, BP_Restart Widget, BP_VictoryBox, BP_Wall, and ThirdPerson Character. The bottom status bar indicates 9 items (1 selected) and View Options.

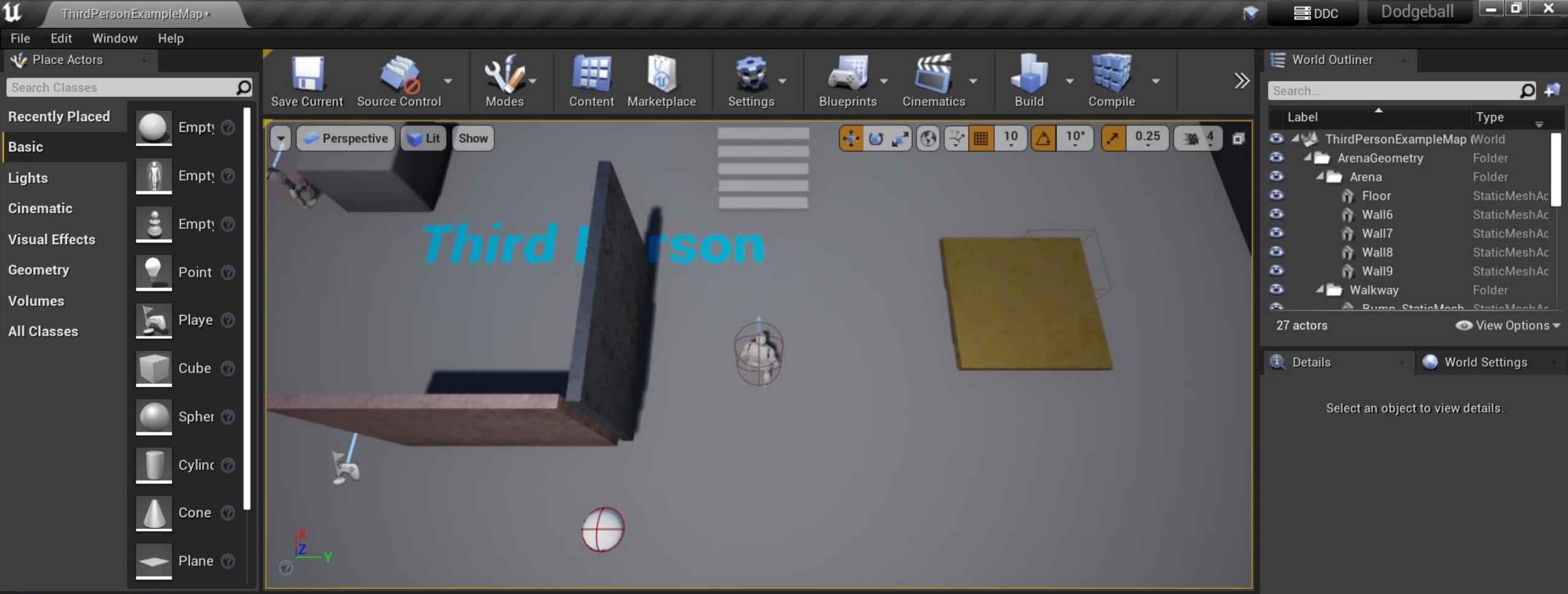




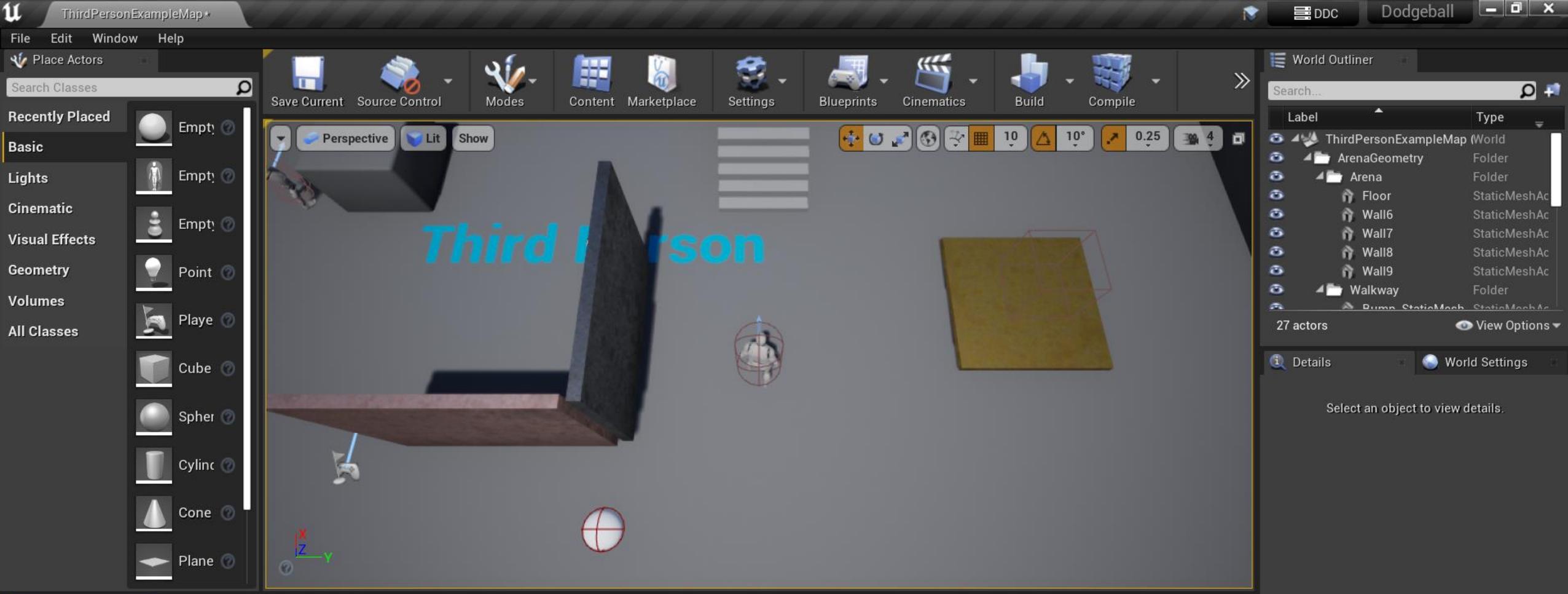
Progress Bars

- › One of the ways that video games can represent character stats such as health, stamina, and so on
- › Essentially a shape, usually rectangular, that can be filled and emptied in order to show the player how a specific stat is progressing.
- › This Progress Bar will be the only element in our Dodgeball game's HUD.

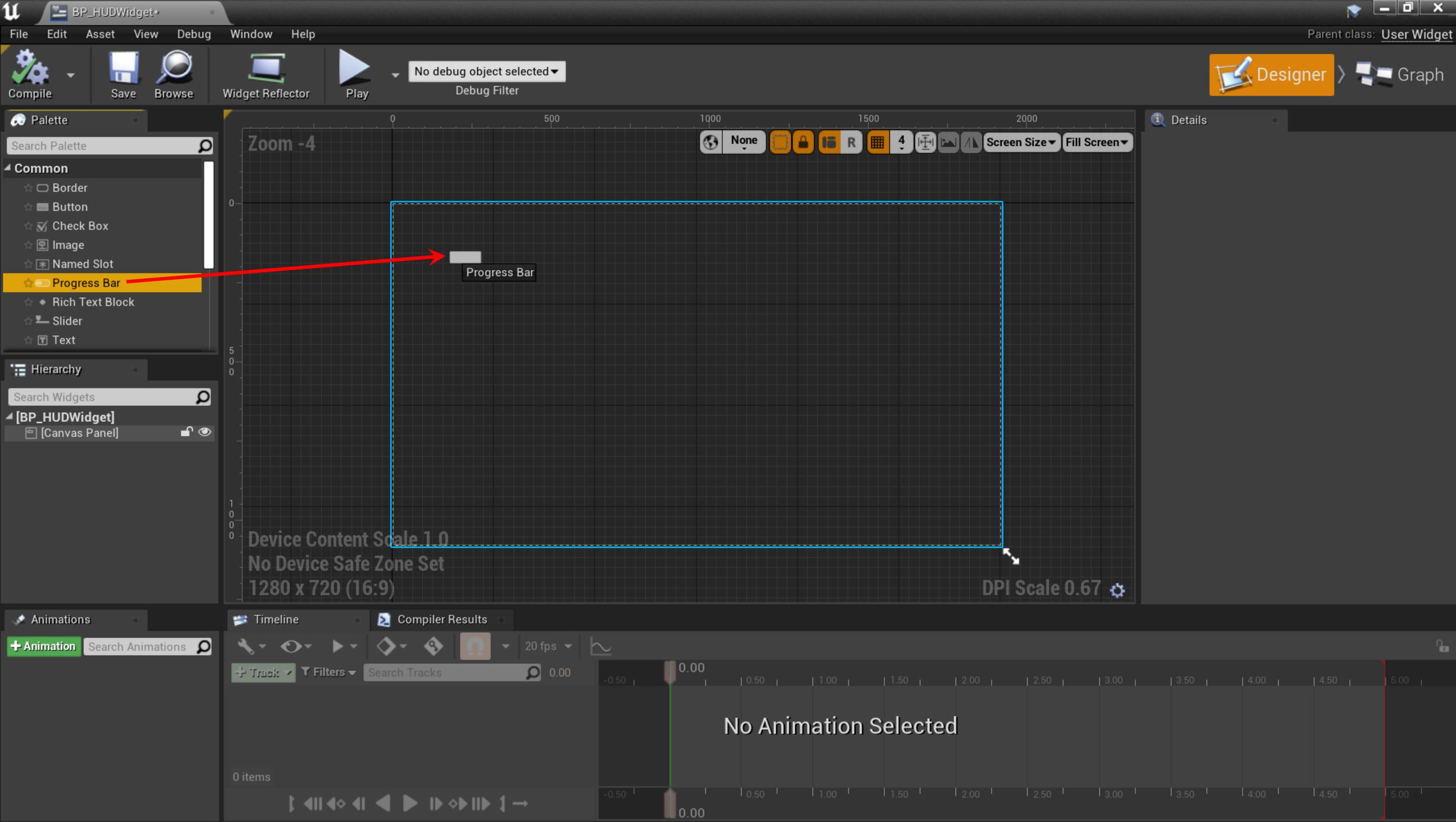


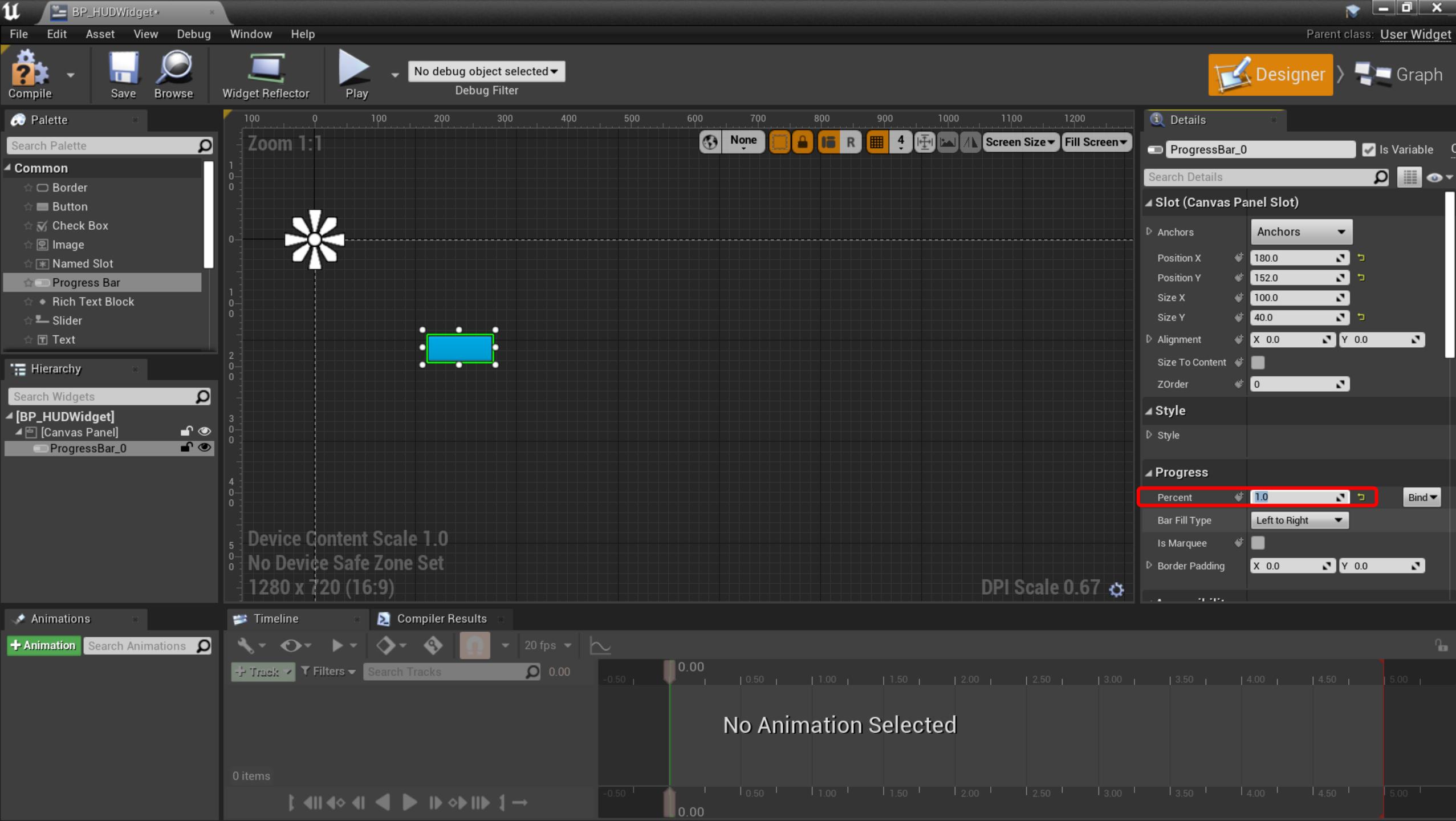


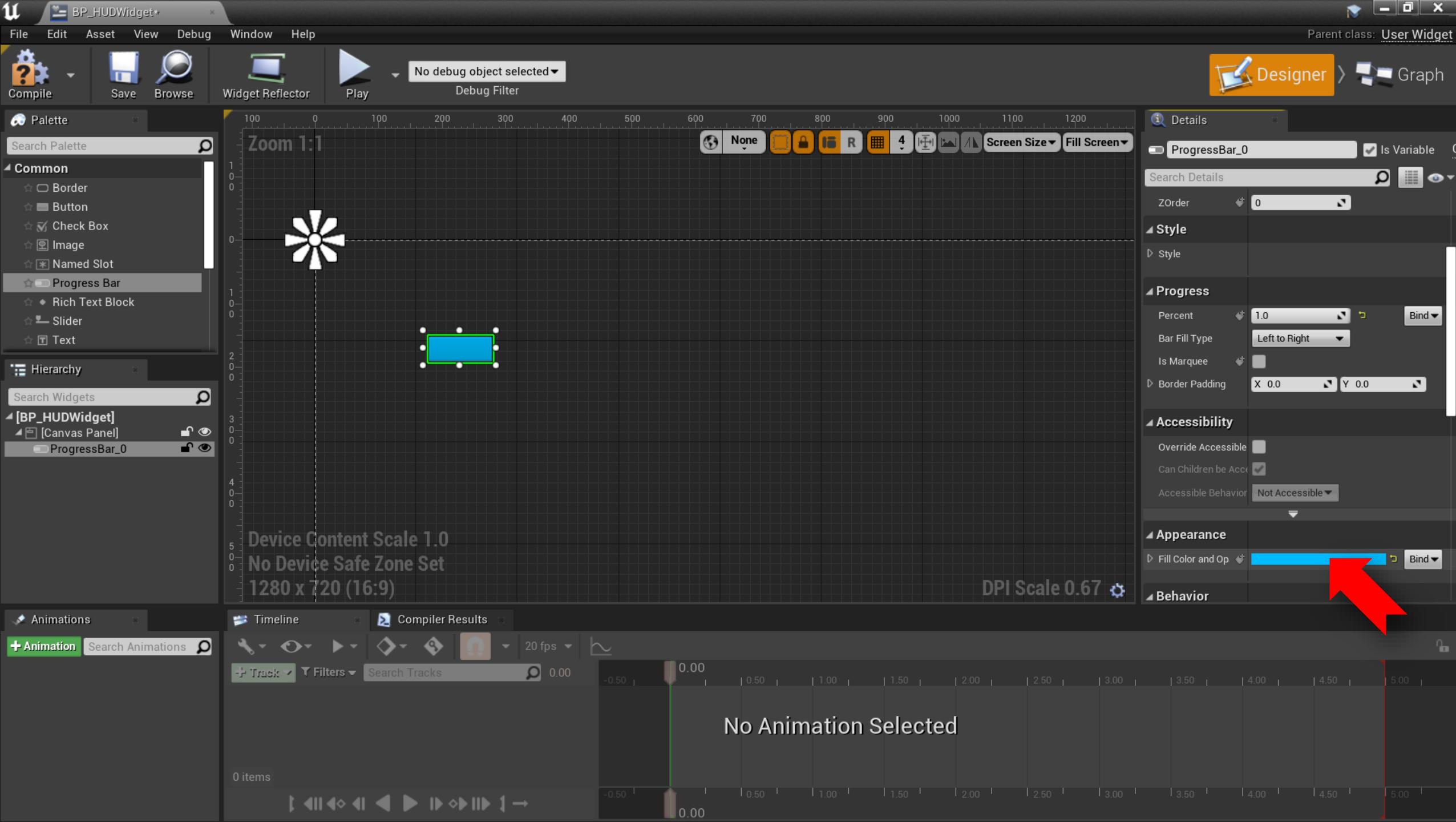
The screenshot shows the Content Browser interface. The top bar includes Add/Import, Save All, and Content navigation. The left sidebar shows the Content tree with categories like Content, Geometry, Mannequin, Physics, StarterContent, ThirdPerson, and ThirdPersonCPP. The Blueprints folder under ThirdPersonCPP is selected. The main area displays a grid of Blueprint assets: BP_DodgeballGameMode, BP_DodgeballPlayerController, BP_DodgeballProjectile, BP_EnemyCharacter, BP_GhostWall, BP_RestartWidget, BP_VictoryBox, BP_Wall, idgetBlueprint (highlighted with a red dashed box), and ThirdPersonCharacter. The bottom status bar indicates 10 items (1 selected) and View Options.

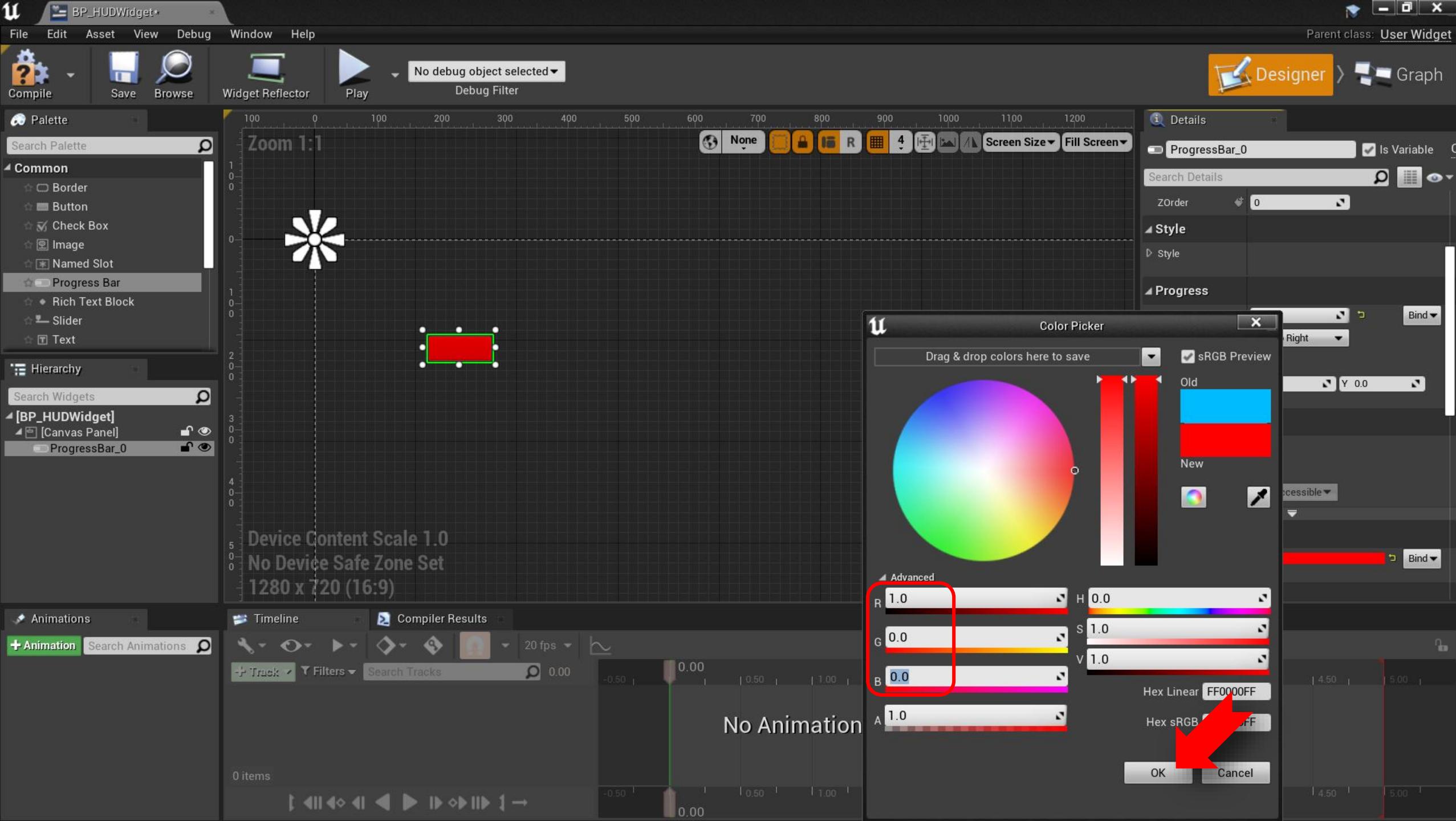


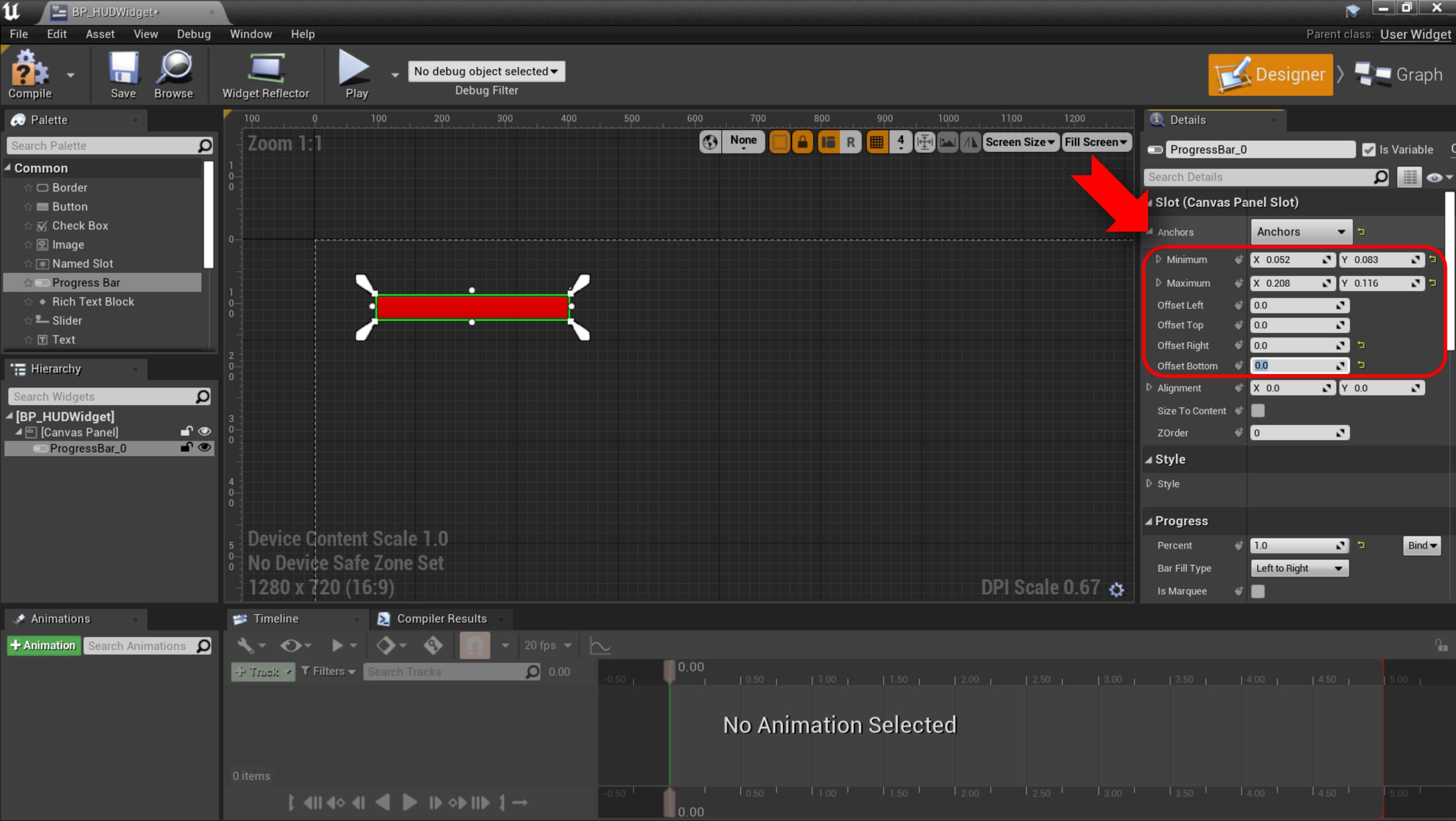
The Content Browser is open in the bottom left, showing the "Blueprints" section under "ThirdPersonCPP". The list includes "BP_Dodgeball GameMode", "BP_Dodgeball PlayerController", "BP_Dodgeball Projectile", "BP_EnemyCharacter", "BP_GhostWall", "BP_HUDWidget" (which is highlighted with a red box), "BP_RestartWidget", "BP_VictoryBox", "BP_Wall", and "ThirdPersonCharacter". The sidebar shows the project structure: Content, Geometry, Mannequin, Physics, StarterContent, ThirdPerson, and ThirdPersonCPP (which is selected). The bottom status bar indicates "10 items (1 selected)" and "View Options".

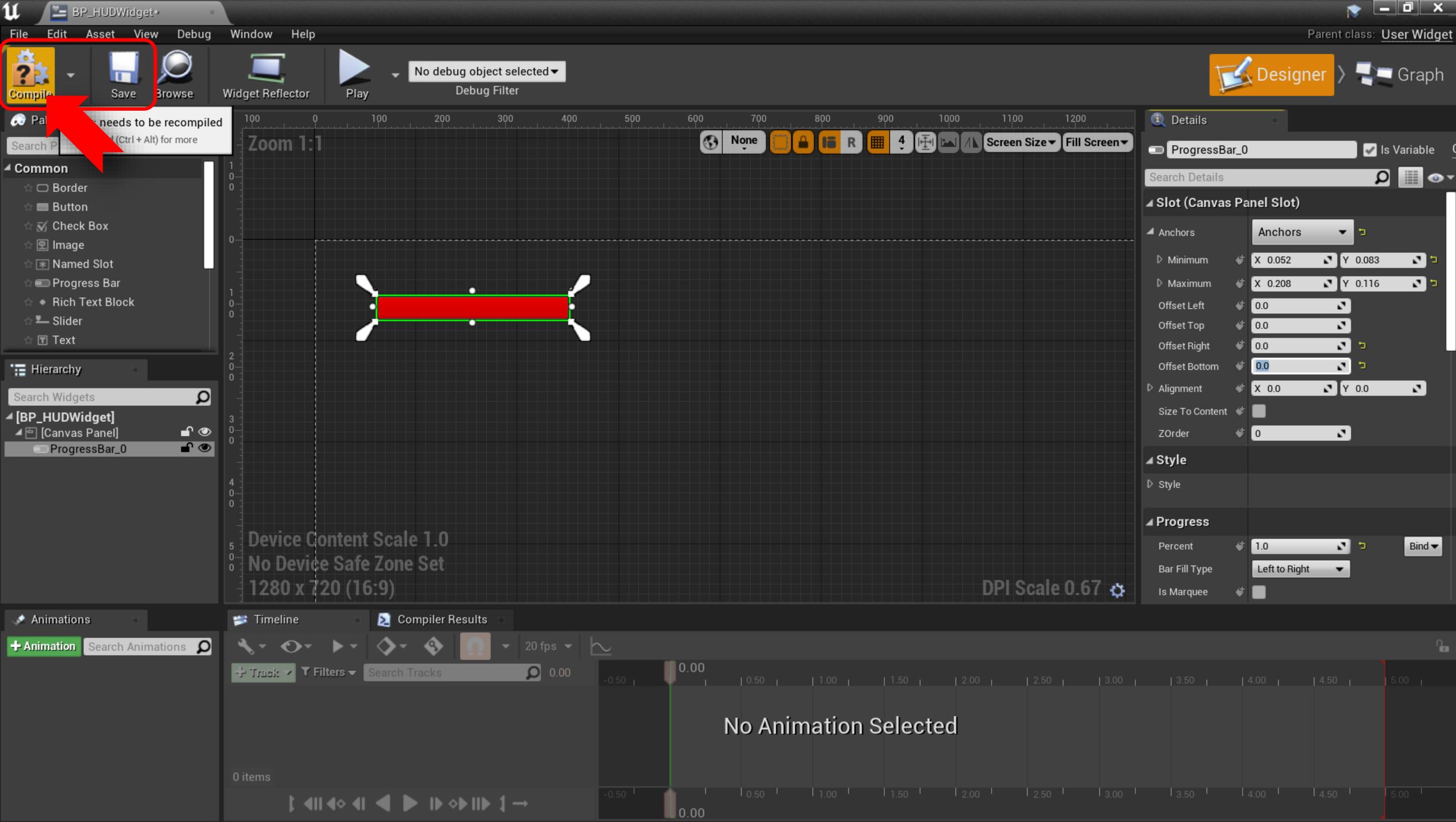




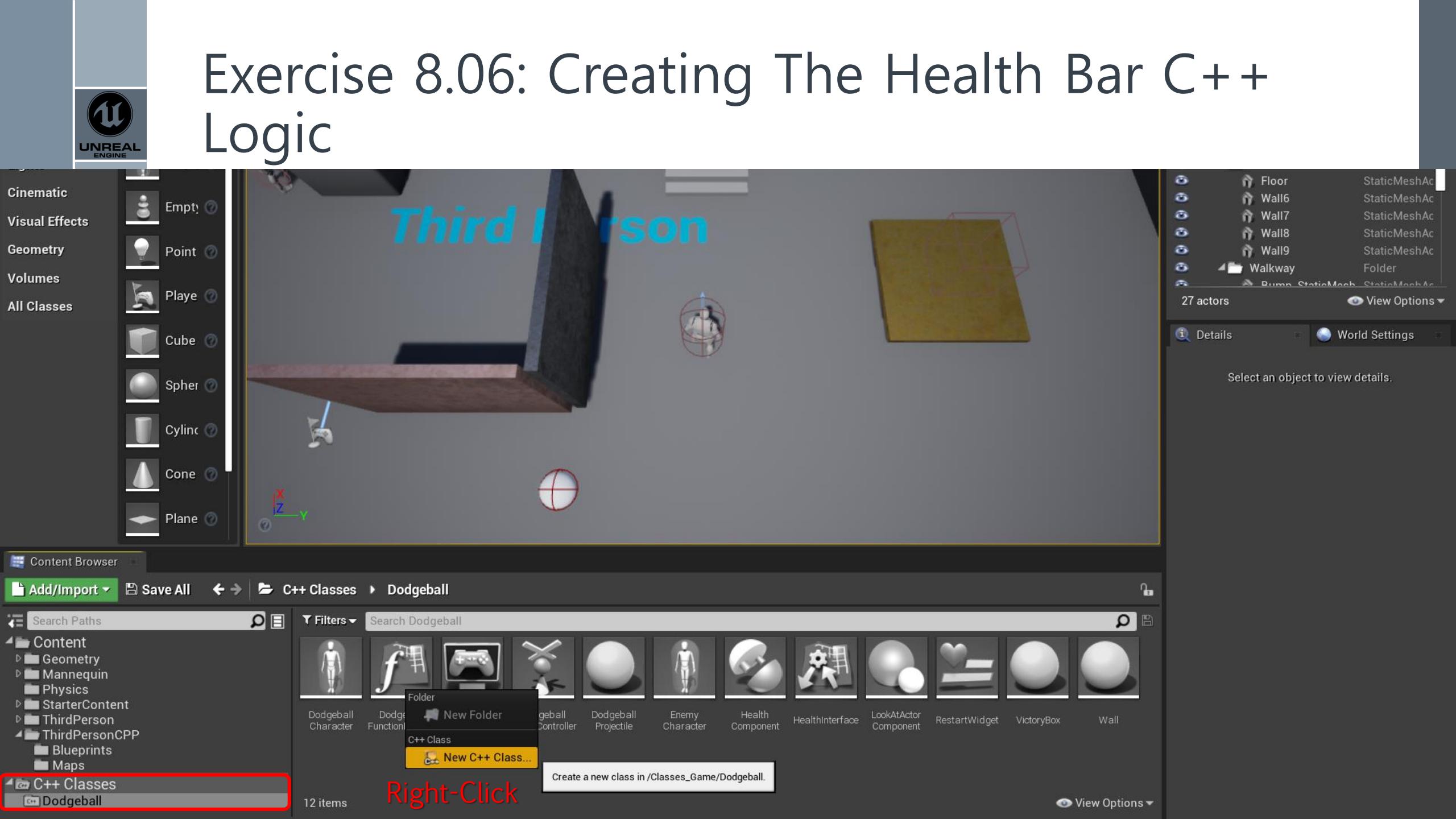








Exercise 8.06: Creating The Health Bar C++ Logic





Choose Parent Class

This will add a C++ header and source code file to your game project.

Show All Classes

userw

Object

Visual

Widget

UserWidget

4 items (1 selected)

View Options ▾

The screenshot shows the 'Choose Parent Class' dialog from the Unreal Engine. At the top, there's a message about adding a C++ header and source code file to the project. A checkbox for 'Show All Classes' is present. Below is a tree view of classes under 'userw': Object, Visual, Widget, and UserWidget. The 'UserWidget' node is highlighted with a red rectangle. At the bottom left, it says 'Selected Class' is 'User Widget' and 'Selected Class Source' is 'UserWidget.h'. On the right, there are 'Next >', 'Create Class', and 'Cancel' buttons. A large red arrow points to the 'Next >' button.

Selected Class

User Widget

Selected Class Source

UserWidget.h

Next >

Create Class

Cancel



Name Your New User Widget

Enter a name for your new class. Class names may only contain alphanumeric characters, and may not contain a space.

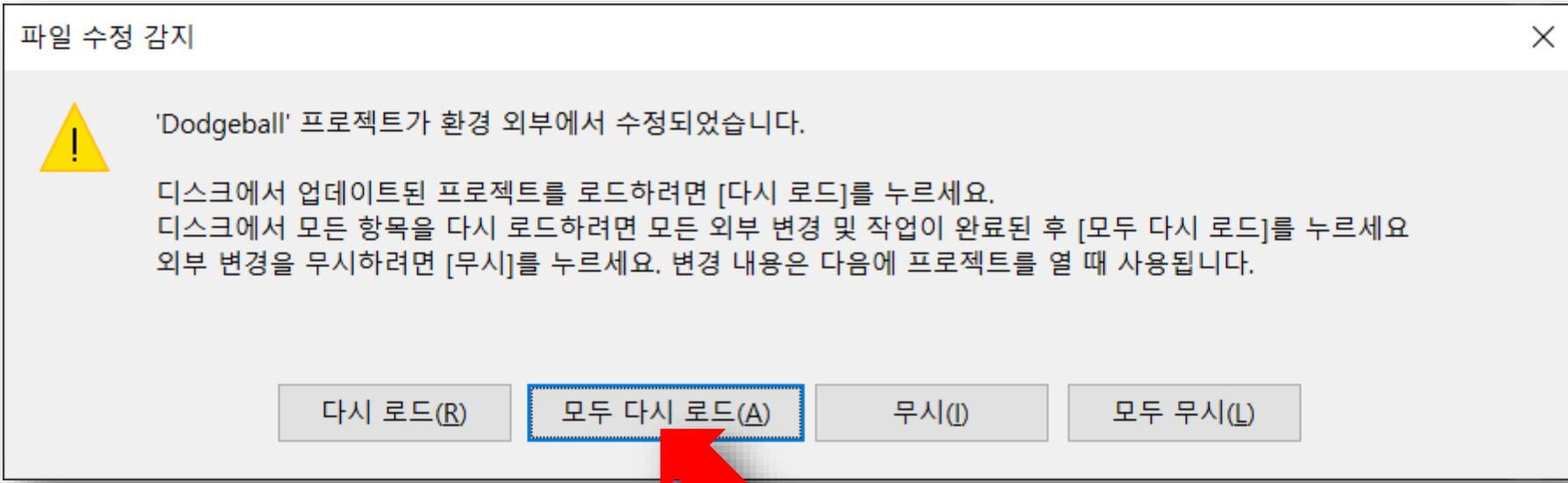
When you click the "Create" button below, a header (.h) file and a source (.cpp) file will be made using this name.

Name	<input type="text" value="HUDWidget"/>	Dodgeball (Runtime) ▼	Public	Private
Path	<input type="text" value="C:/Users/sunje/Desktop/Unreal/Dodgeball/Source/Dodgeball/"/> Choose Folder			
Header File	C:/Users/sunje/Desktop/Unreal/Dodgeball/Source/Dodgeball/HUDWidget.h			
Source File	C:/Users/sunje/Desktop/Unreal/Dodgeball/Source/Dodgeball/HUDWidget.cpp			

< Back

Create Class

Cancel



파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(I) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+Q) Dodgeball — □ ×

Develop Win64 로컬 Windows 디버거 Live Share

HUDWidget.h* ✘ X DodgeballPlayerController.cpp DodgeballPlayerController.h RestartWidget.cpp RestartWidget.h

Dodgeball UHUDWidget UpdateHealthPercent(float HealthPercent)

```
1 // Fill out your copyright notice in the Description page of Project Settings.
2
3 #pragma once
4
5 #include "CoreMinimal.h"
6 #include "Blueprint/UserWidget.h"
7 #include "HUDWidget.generated.h"
8
9 /**
10 * 
11 */
12 UCLASS()
13 class DODGEBALL_API UHUDWidget : public UUserWidget
14 {
15     GENERATED_BODY()
16
17 public:
18     UPROPERTY(meta = (BindWidget))
19     class UProgressBar* HealthBar;
20
21     void UpdateHealthPercent(float HealthPercent);
22 };
23
24 }
25
```

Ctrl+S

서버 템플릿 툴상자

솔루션 탐색기

솔루션 탐색기 검색(Ctrl+.)

솔루션 'Dodgeball' (2/2개 프로젝트)

- Engine
- UE4
- Games
- Dodgeball
- 참조
- 외부 종속성
- Config
- Source
- Dodgeball
- Dodgeball.Build.cs
- Dodgeball.cpp
- Dodgeball.h
- DodgeballCharacter.cpp
- DodgeballCharacter.h
- DodgeballFunctionLibrary.cpp
- DodgeballFunctionLibrary.h
- DodgeballGameMode.cpp
- DodgeballGameMode.h
- DodgeballPlayerController.cpp
- DodgeballPlayerController.h
- DodgeballProjectile.cpp
- DodgeballProjectile.h
- EnemyCharacter.cpp
- EnemyCharacter.h
- HealthComponent.cpp
- HealthComponent.h
- HealthInterface.cpp

100% 문제가 검색되지 않음 줄: 22 문자: 48 열: 51 템 CRLF 솔루션 탐색기 Git 변경 내용

준비 ↑ 소스 제어에 추가 ▲ 1

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(I) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+Q) Dodgeball — □ ×

Live Share

HUDWidget.cpp* ✘ X HUDWidget.h DodgeballPlayerController.cpp DodgeballPlayerController.h RestartWidget.cpp

Dodgeball

```
// Fill out your copyright notice in the Description page of Project Settings.

#include "HUDWidget.h"
#include "Components/ProgressBar.h"

void UHUDWidget::UpdateHealthPercent(float HealthPercent)
{
    HealthBar->SetPercent(HealthPercent);
}
```

Ctrl+S

100 % 문제가 검색되지 않음 줄: 9 문자: 39 열: 42 탭 CRLF

슬루션 탐색기

슬루션 탐색기 검색(Ctrl+.)

- 슬루션 'Dodgeball' (2/2개 프로젝트)
 - Engine
 - UE4
 - Games
 - Dodgeball
 - 참조
 - 외부 종속성
 - Config
 - Source
 - Dodgeball
 - Dodgeball.Build.cs
 - Dodgeball.cpp
 - Dodgeball.h
 - DodgeballCharacter.cpp
 - DodgeballCharacter.h
 - DodgeballFunctionLibrary.cpp
 - DodgeballFunctionLibrary.h
 - DodgeballGameMode.cpp
 - DodgeballGameMode.h
 - DodgeballPlayerController.cpp
 - DodgeballPlayerController.h
 - DodgeballProjectile.cpp
 - DodgeballProjectile.h
 - EnemyCharacter.cpp
 - EnemyCharacter.h
 - HealthComponent.cpp
 - HealthComponent.h
 - HealthInterface.cpp

준비 ↑ 소스 제어에 추가 ▲ 1

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(I) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+Q) Dodgeball — □ ×

Live Share

HUDWidget.cpp HUDWidget.h DodgeballPlayerController.cpp DodgeballPlayerController.h* RestartWidget.cpp

Dodgeball

```
11 */  
12 UCLASS()  
13 class DODGEBALL_API ADodgeballPlayerController : public APlayerController  
14 {  
15     GENERATED_BODY()  
16  
17     public:  
18  
19         UPROPERTY(EditDefaultsOnly)  
20         TSubclassOf<class URestartWidget> BP_RestartWidget;  
21  
22         void ShowRestartWidget();  
23         void HideRestartWidget();  
24  
25         UPROPERTY(EditDefaultsOnly)  
26         TSubclassOf<class UHUDWidget> BP_HUDWidget;  
27  
28         void UpdateHealthPercent(float HealthPercent);  
29  
30     protected:  
31         virtual void BeginPlay() override;  
32  
33     private:  
34  
35         UPROPERTY()  
36         class URestartWidget* RestartWidget;  
37  
38         UPROPERTY()  
39         class UHUDWidget* HUDWidget;  
40     };  
41  
42 }
```

Ctrl+S

슬루션 탐색기

슬루션 탐색기 검색(Ctrl+Shift+F)

솔루션 'Dodgeball' (2/2개 프로젝트)

- Engine
- UE4
- Games
- Dodgeball
 - 참조
 - 외부 종속성
 - Config
 - Source
 - Dodgeball
 - Dodgeball.Build.cs
 - Dodgeball.cpp
 - Dodgeball.h
 - DodgeballCharacter.cpp
 - DodgeballCharacter.h
 - DodgeballFunctionLibrary.cpp
 - DodgeballFunctionLibrary.h
 - DodgeballGameMode.cpp
 - DodgeballGameMode.h
 - DodgeballPlayerController.cpp
 - DodgeballPlayerController.h
 - DodgeballProjectile.cpp
 - DodgeballProjectile.h
 - EnemyCharacter.cpp
 - EnemyCharacter.h
 - HealthComponent.cpp
 - HealthComponent.h
 - HealthInterface.cpp

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(I) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+Q) Dodgeball — □ ×

Develop Win64 로컬 Windows 디버거 Live Share

HUDWidget.cpp HUDWidget.h DodgeballPlayerController.cpp* DodgeballPlayerController.h RestartWidget.cpp

Dodgeball // Fill out your copyright notice in the Description page of Project Settings.

#include "DodgeballIPPlayerController.h"
#include "RestartWidget.h"
#include "HUDWidget.h"

void ADodgeballIPPlayerController::BeginPlay()
{
 Super::BeginPlay();

 if (BP_HUDWidget != nullptr)
 {
 HUDWidget = CreateWidget<UHUDWidget>(this, BP_HUDWidget);
 HUDWidget->AddToViewport();
 }
}

void ADodgeballIPPlayerController::UpdateHealthPercent(float HealthPercent)
{
 if (HUDWidget != nullptr)
 {
 HUDWidget->UpdateHealthPercent(HealthPercent);
 }
}

void ADodgeballIPPlayerController::ShowRestartWidget()
{
 if (BP_RestartWidget != nullptr)
 {
 SetPause(true);
 SetInputMode(FInputModeUIOnly());
 bShowMouseCursor = true;
 RestartWidget = CreateWidget<URestartWidget>(this, BP_RestartWidget);
 RestartWidget->AddToViewport();
 }
}

Ctrl+S

슬루션 탐색기

슬루션 탐색기 검색(Ctrl+.)

슬루션 탐색기 'Dodgeball' (2/2개 프로젝트)

- Engine
- UE4
- Games
- Dodgeball
- 참조
- 외부 종속성
- Config
- Source
- Dodgeball
- Dodgeball.Build.cs
- Dodgeball.cpp
- Dodgeball.h
- DodgeballCharacter.cpp
- DodgeballCharacter.h
- DodgeballFunctionLibrary.cpp
- DodgeballFunctionLibrary.h
- DodgeballGameMode.cpp
- DodgeballGameMode.h
- DodgeballPlayerController.cpp
- DodgeballPlayerController.h
- DodgeballProjectile.cpp
- DodgeballProjectile.h
- EnemyCharacter.cpp
- EnemyCharacter.h
- HealthComponent.cpp
- HealthComponent.h
- HealthInterface.cpp

슬루션 탐색기 Git 변경 내용

준비 ↑ 소스 제어에 추가 ▲

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(I) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+Q) Dodgeball — □ ×

Develop Win64 로컬 Windows 디버거 Live Share

DodgeballPlayerController.cpp DodgeballPlayerController.h RestartWidget.cpp HealthInterface.cpp HealthInterface.h*

// Fill out your copyright notice in the Description page of Project Settings.

```
#pragma once

#include "CoreMinimal.h"
#include "UObject/Interface.h"
#include "HealthInterface.generated.h"

// This class does not need to be modified.
UINTERFACE(MinimalAPI)
class UHealthInterface : public UInterface
{
    GENERATED_BODY()

    /**
     */
class DODGEBALL_API IHealthInterface
{
    GENERATED_BODY()

    // Add interface functions to this class. This is the class that will be inherited to implement this interface.
public:

    UFUNCTION(BlueprintNativeEvent, Category = Health)
    void OnDeath();

    virtual void OnDeath_Implementation() = 0;

    UFUNCTION(BlueprintNativeEvent, Category = Health)
    void OnTakeDamage();
    virtual void OnTakeDamage_Implementation() = 0;
};
```

Ctrl+S

슬루션 탐색기

슬루션 탐색기 검색(Ctrl+I)

솔루션 'Dodgeball' (2/2개 프로젝트)

- Engine
- UE4
- Games
- Dodgeball
 - 참조
 - 외부 종속성
 - Config
 - Source
 - Dodgeball
 - Dodgeball.Build.cs
 - Dodgeball.cpp
 - Dodgeball.h
 - DodgeballCharacter.cpp
 - DodgeballCharacter.h
 - DodgeballFunctionLibrary.cpp
 - DodgeballFunctionLibrary.h
 - DodgeballGameMode.cpp
 - DodgeballGameMode.h
 - DodgeballPlayerController.cpp
 - DodgeballPlayerController.h
 - DodgeballProjectile.cpp
 - DodgeballProjectile.h
 - EnemyCharacter.cpp
 - EnemyCharacter.h
 - HealthComponent.cpp
 - HealthComponent.h
 - HealthInterface.cpp

준비

문제가 검색되지 않음

줄: 32 문자: 49 열: 52 템 CRLF

슬루션 탐색기 Git 변경 내용

↑ 소스 제어에 추가 ↑

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(I) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+Q) Dodgeball — □ ×

Develop Win64 로컬 Windows 디버거 Live Share

DodgeballPlayerController.cpp DodgeballPlayerController.h HealthInterface.cpp HealthInterface.h HealthComponent.h* (Current)

4
5 #include "CoreMinimal.h"
6 #include "Components/ActorComponent.h"
7 #include "HealthComponent.generated.h"
8
9 UCLASS(ClassGroup=(Custom), meta=(BlueprintSpawnableComponent))
10 class DODGEBALL_API UHealthComponent : public UActorComponent
11 {
12 GENERATED_BODY()
13
14 public:
15 // Sets default values for this component's properties
16 UHealthComponent();
17
18 protected:
19 // Called when the game starts
20 virtual void BeginPlay() override;
21
22 // The owner's initial and current amount health points
23 UPROPERTY(EditDefaultsOnly, Category = Health)
24 float Health = 100.f;
25
26 public:
27 // Called every frame
28 virtual void TickComponent(float DeltaTime, ELevelTick TickType, FActorComponentTickFunction* ThisTickFunction) override;
29
30 // Take health points from its owner
31 void LoseHealth(float Amount);
32
33 FORCEINLINE float GetHealthPercent() const { return Health / 100.f; }
34 };

Ctrl+S

슬루션 탐색기

슬루션 탐색기 검색(Ctrl+Shift+F)

슬루션 탐색기 'Dodgeball' (2/2개 프로젝트)

- Engine
- UE4
- Games
- Dodgeball
- 참조
- 외부 종속성
- Config
- Source
- Dodgeball
- Dodgeball.Build.cs
- Dodgeball.cpp
- Dodgeball.h
- DodgeballCharacter.cpp
- DodgeballCharacter.h
- DodgeballFunctionLibrary.cpp
- DodgeballFunctionLibrary.h
- DodgeballGameMode.cpp
- DodgeballGameMode.h
- DodgeballPlayerController.cpp
- DodgeballPlayerController.h
- DodgeballProjectile.cpp
- DodgeballProjectile.h
- EnemyCharacter.cpp
- EnemyCharacter.h
- HealthComponent.cpp
- HealthComponent.h
- HealthInterface.cpp

준비

문제가 검색되지 않음

줄: 34 문자: 71 열: 74 템 CRLF

↑ 소스 제어에 추가 ▲

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(I) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+Q) Dodgeball — □ ×

Develop Win64 로컬 Windows 디버거 Live Share

DodgeballPlayerController.cpp HealthInterface.cpp HealthInterface.h HealthComponent.h HealthComponent.cpp*

19 // Called when the game starts
20 void UHealthComponent::BeginPlay()
21 {
22 Super::BeginPlay();
23 // ...
24 }
25
26 // Called every frame
27 void UHealthComponent::TickComponent(float DeltaTime, ELevelTick TickType, FActorComponentTickFunction* ThisTickFunction)
28 {
29 Super::TickComponent(DeltaTime, TickType, ThisTickFunction);
30 // ...
31 }
32
33 // Called when health is lost
34 void UHealthComponent::LoseHealth(float Amount)
35 {
36 Health -= Amount;
37 if (GetOwner()>Implements<UHealthInterface>())
38 {
39 IHealthInterface::Execute_OnTakeDamage(GetOwner());
40 }
41 if (Health <= 0.f)
42 {
43 Health = 0.f;
44 //UKismetSystemLibrary::QuitGame(this, nullptr, EQuitPreference::Quit, true);
45 if (GetOwner()>Implements<UHealthInterface>())
46 {
47 IHealthInterface::Execute_OnDeath(GetOwner());
48 }
49 }
50 }
51 }

100 % 1 ! 0 ← → 줄: 43 문자: 54 열: 60 탭 CRLF

Ctrl+S

슬루션 탐색기

슬루션 탐색기 검색(Ctrl+I)

- 슬루션 'Dodgeball' (2/2개 프로젝트)
 - Engine
 - UE4
 - Games
 - Dodgeball
 - 참조
 - 외부 종속성
 - Config
 - Source
 - Dodgeball
 - Dodgeball.Build.cs
 - Dodgeball.cpp
 - Dodgeball.h
 - DodgeballCharacter.cpp
 - DodgeballCharacter.h
 - DodgeballFunctionLibrary.cpp
 - DodgeballFunctionLibrary.h
 - DodgeballGameMode.cpp
 - DodgeballGameMode.h
 - DodgeballPlayerController.cpp
 - DodgeballProjectile.cpp
 - DodgeballProjectile.h
 - EnemyCharacter.cpp
 - EnemyCharacter.h
 - HealthComponent.cpp
 - HealthComponent.h
 - HealthInterface.cpp

준비 ↑ 소스 제어에 추가 ▲ 1

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(I) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+Q) Dodgeball — □ ×

Live Share

HealthInterface.cpp HealthInterface.h HealthComponent.h HealthComponent.cpp DodgeballCharacter.h*

// Copyright Epic Games, Inc. All Rights Reserved.

```
#pragma once

#include "CoreMinimal.h"
#include "GameFramework/Character.h"
#include "HealthInterface.h"
#include "DodgeballCharacter.generated.h"

UCLASS(config=Game)
class ADodgeballCharacter : public ACharacter, public IHealthInterface
{
    GENERATED_BODY()

    /** Camera boom positioning the camera behind the character */
    UPROPERTY(VisibleAnywhere, BlueprintReadOnly, Category = Camera, meta = (AllowPrivateAccess = "true"))
    class USpringArmComponent* CameraBoom;

    /** Follow camera */
    UPROPERTY(VisibleAnywhere, BlueprintReadOnly, Category = Camera, meta = (AllowPrivateAccess = "true"))
    class UCameraComponent* FollowCamera;

    class UHealthComponent* HealthComponent;

public:
    ADodgeballCharacter();

    virtual void OnDeath_Implementation() override;
    virtual void OnTakeDamage_Implementation() override;

    /** Base turn rate, in deg/sec. Other scaling may affect final turn rate. */
    UPROPERTY(VisibleAnywhere, BlueprintReadOnly, Category=Camera)
    float BaseTurnRate;

    /** Base look up/down rate, in deg/sec. Other scaling may affect final rate. */
}
```

DodgeballCharacter.h*

OnTakeDamage_Implementation()

솔루션 탐색기

솔루션 탐색기 검색(Ctrl+Shift+F)

솔루션 탐색기 'Dodgeball' (2/2개 프로젝트)

- Engine
- UE4
- Games
- Dodgeball
 - 참조
 - 외부 종속성
 - Config
 - Source
 - Dodgeball
 - Dodgeball.Build.cs
 - Dodgeball.cpp
 - Dodgeball.h
 - DodgeballCharacter.cpp
 - DodgeballCharacter.h
 - DodgeballFunctionLibrary.cpp
 - DodgeballFunctionLibrary.h
 - DodgeballGameMode.cpp
 - DodgeballGameMode.h
 - DodgeballPlayerController.cpp
 - DodgeballPlayerController.h
 - DodgeballProjectile.cpp
 - DodgeballProjectile.h
 - EnemyCharacter.cpp
 - EnemyCharacter.h
 - HealthComponent.cpp
 - HealthComponent.h
 - HealthInterface.cpp

Ctrl+S

준비

소스 제어에 추가

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(I) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+Q) Dodgeball — □ ×

Live Share

HealthInterface.h HealthComponent.h HealthComponent.cpp DodgeballCharacter.h DodgeballCharacter.cpp* ✎ X

Dodgeball CameraBoom->bInheritRoll = false;

// Create a follow camera
FollowCamera = CreateDefaultSubobject<UCameraComponent>(TEXT("FollowCamera"));
FollowCamera->SetupAttachment(CameraBoom, USpringArmComponent::SocketName); // Attach the camera to the end of the boom and let
FollowCamera->bUsePawnControlRotation = false; // Camera does not rotate relative to arm

HealthComponent = CreateDefaultSubobject<UHealthComponent>(TEXT("Health Component"));

// Note: The skeletal mesh and anim blueprint references on the Mesh component (inherited from Character)
// are set in the derived blueprint asset named MyCharacter (to avoid direct content references in C++)

void ADodgeballCharacter::OnDeath_Implementation()

{
 //UKismetSystemLibrary::QuitGame(this, nullptr, EQuitPreference::Quit, true);
 ADodgeballPlayerController* PlayerController = Cast<ADodgeballPlayerController>(GetController());
 if (PlayerController != nullptr)
 {
 PlayerController->ShowRestartWidget();
 }
}

void ADodgeballCharacter::OnTakeDamage_Implementation()

{
 ADodgeballPlayerController* PlayerController = Cast<ADodgeballPlayerController>(GetController());
 if (PlayerController != nullptr)
 {
 PlayerController->UpdateHealthPercent(HealthComponent->GetHealthPercent());
 }
}

//////////
// Input

100 % ✅ 문제가 검색되지 않음 줄: 74 문자: 78 열: 84 탭 CRLF

솔루션 탐색기

솔루션 탐색기 검색(Ctrl+Shift+F)

솔루션 'Dodgeball' (2/2개 프로젝트)

- Engine
- UE4
- Games
- Dodgeball
 - 참조
 - 외부 종속성
 - Config
 - Source
 - Dodgeball
 - Dodgeball.Build.cs
 - Dodgeball.cpp
 - Dodgeball.h
 - DodgeballCharacter.cpp
 - DodgeballCharacter.h
 - DodgeballFunctionLibrary.h
 - DodgeballFunctionLibrary.cpp
 - DodgeballGameMode.h
 - DodgeballGameMode.cpp
 - DodgeballPlayerController.h
 - DodgeballPlayerController.cpp
 - DodgeballProjectile.h
 - DodgeballProjectile.cpp
 - EnemyCharacter.cpp
 - EnemyCharacter.h
 - HealthComponent.cpp
 - HealthComponent.h
 - HealthInterface.cpp

Ctrl+S

준비 ↑ 소스 제어에 추가 ▲

The screenshot shows the Microsoft Visual Studio interface with the 'Dodgeball' project open. The 'Build' menu is currently active, displaying various build-related options. A large red arrow points to the first item in the list, '솔루션 빌드(B)' (Solution Build). The menu also includes other options like '솔루션 다시 빌드' (Rebuild Solution), '솔루션 정리(C)' (Solution Clean), and '솔루션의 전체 프로그램 데이터베이스 파일 빌드' (Build All Program Database Files).

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+Q) Dodgeball Live Share

HealthInterface.h HealthComponent.h

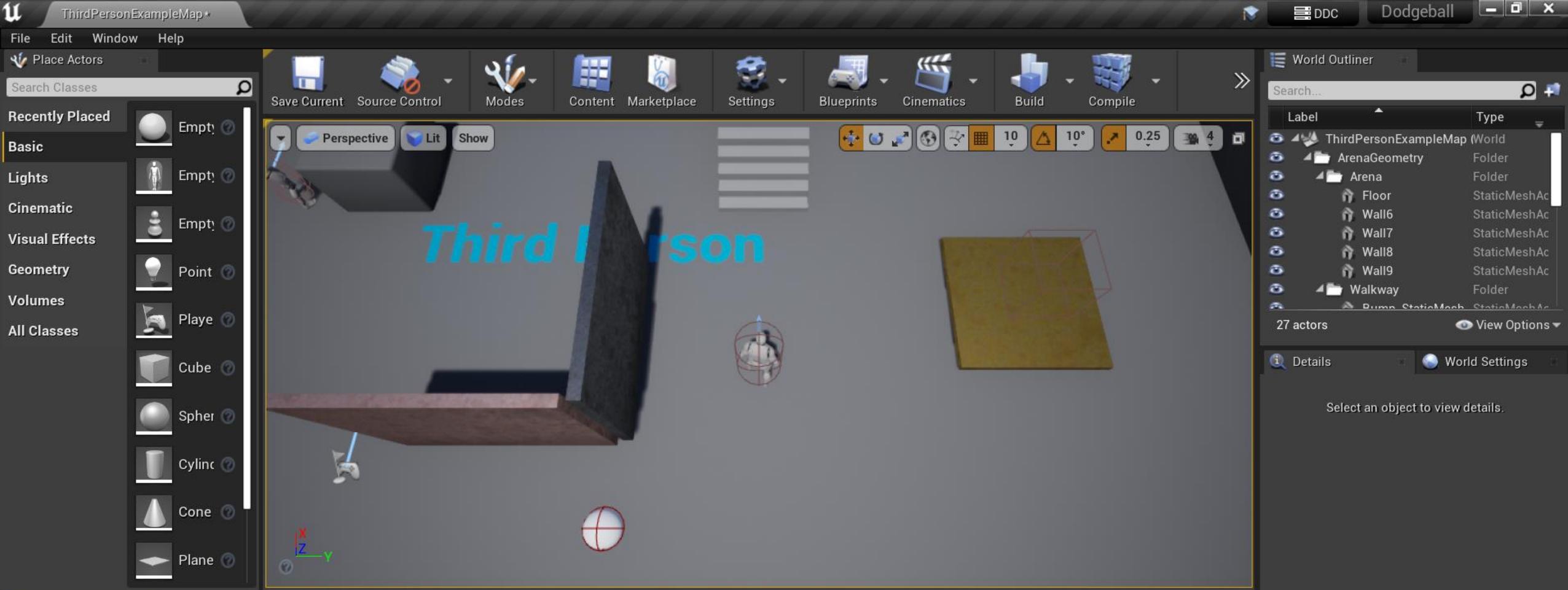
Dodgeball

```
46 CameraBoom->bInheritRoll = false;
47
48 // Create a follow camera
49 FollowCamera = CreateDefaultSubObject<UFollowCamera>(this);
50 FollowCamera->SetupAttachment(Camera->GetRootComponent());
51 FollowCamera->bUsePawnControlRole = true;
52
53 HealthComponent = CreateDefaultSubObject<UHealthComponent>(this);
54
55 // Note: The skeletal mesh and materials are set in the derived blueprint class
56 }
57
58 void ADodgeballCharacter::OnDeath_Implementation()
59 {
60     //UKismetSystemLibrary::QuitGame(this, nullptr, EQuitPreference::Quit, true);
61     ADodgeballPlayerController* PlayerController = Cast<ADodgeballPlayerController>(GetController());
62     if (PlayerController != nullptr)
63     {
64         PlayerController->ShowRestartWidget();
65     }
66 }
67
68 void ADodgeballCharacter::OnTakeDamage_Implementation()
69 {
70     ADodgeballPlayerController* PlayerController = Cast<ADodgeballPlayerController>(GetController());
71     if (PlayerController != nullptr)
72     {
73         PlayerController->UpdateHealthPercent(HealthComponent->GetHealthPercent());
74     }
75 }
76
77 // Input
```

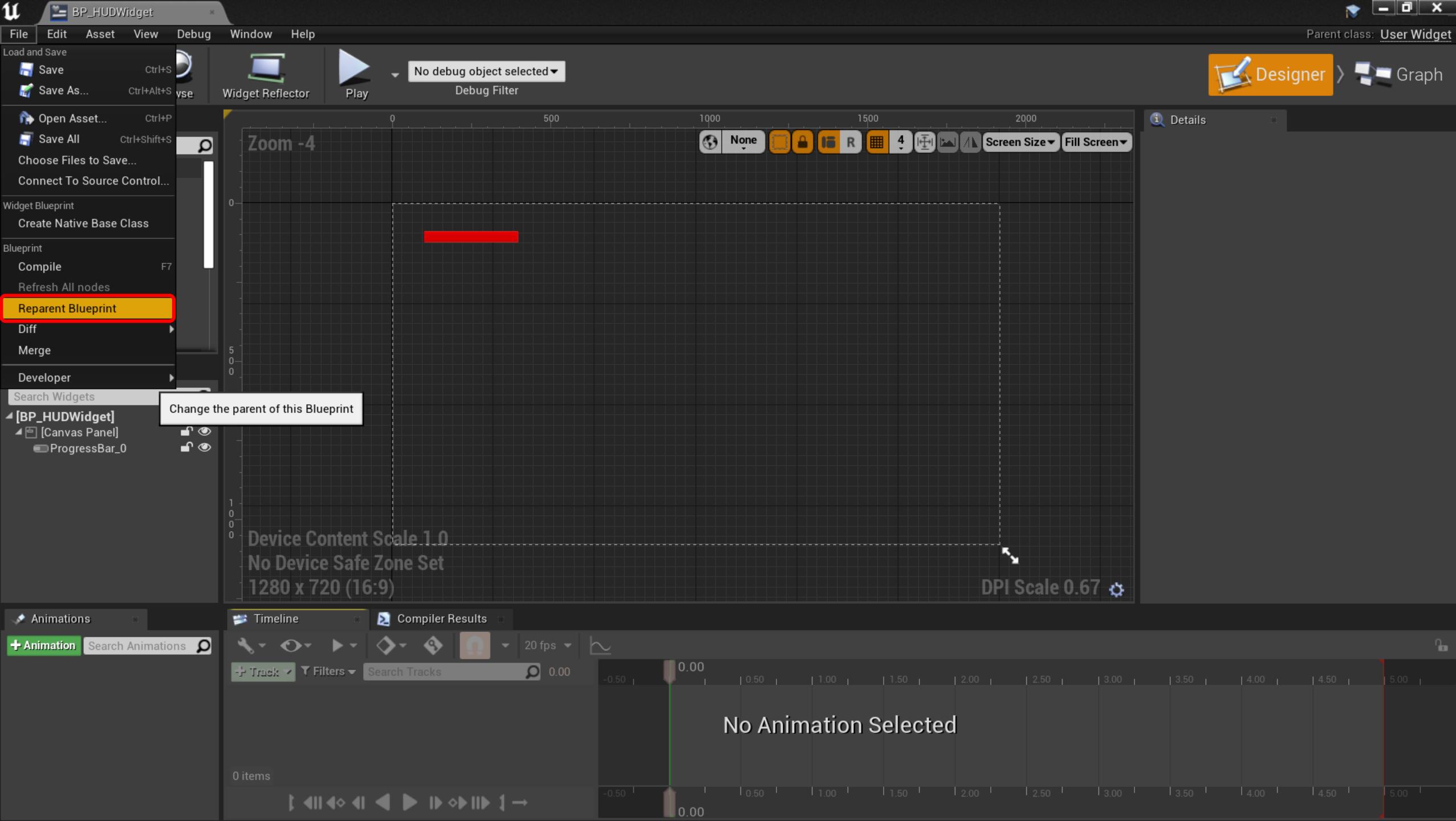
100 % 문제가 검색되지 않음 줄: 74 문자: 78 열: 84 탭 CRLF

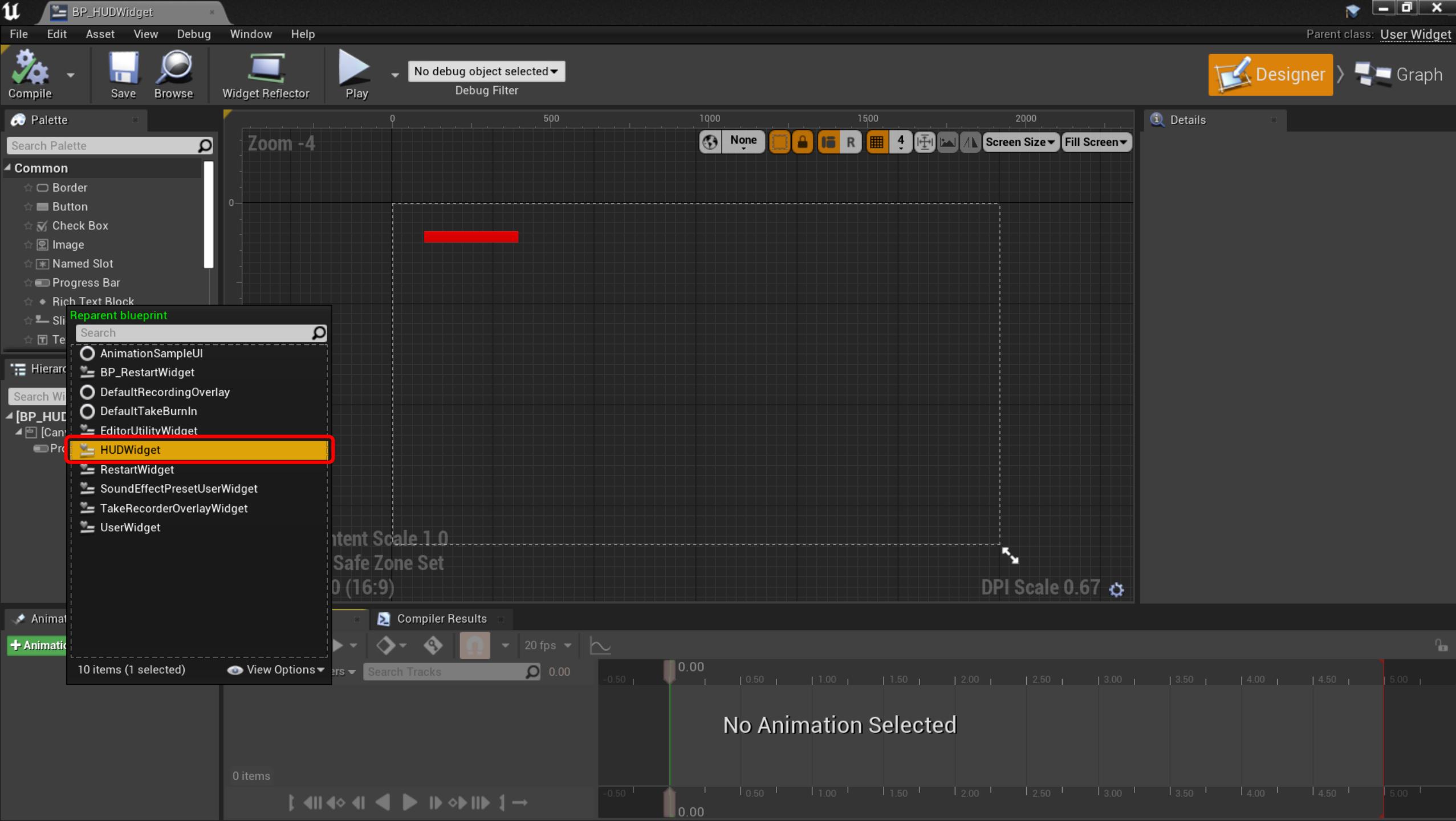
솔루션 탐색기 Git 변경 내용

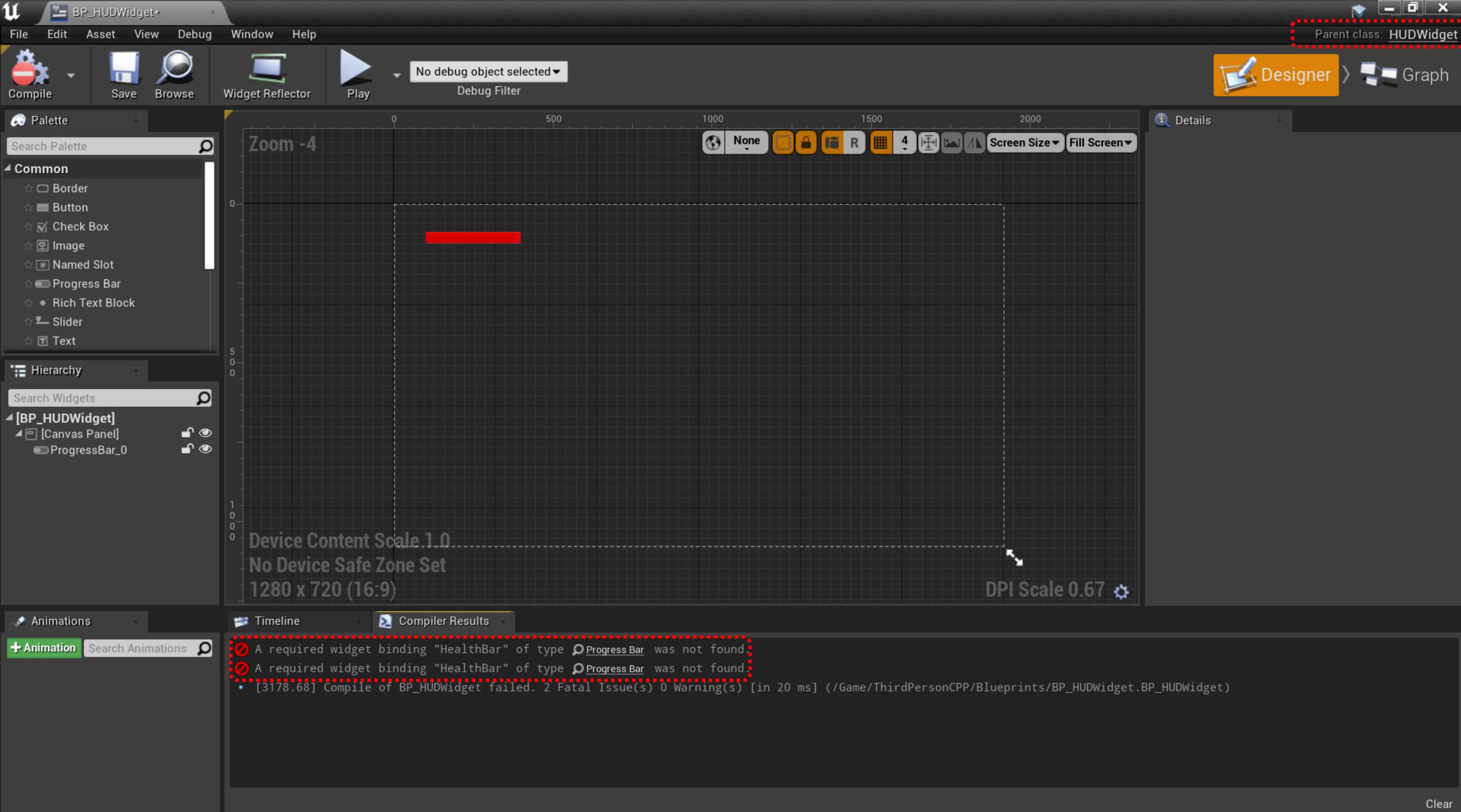
저장되었습니다. ↑ 소스 제어에 추가 ▲ 1

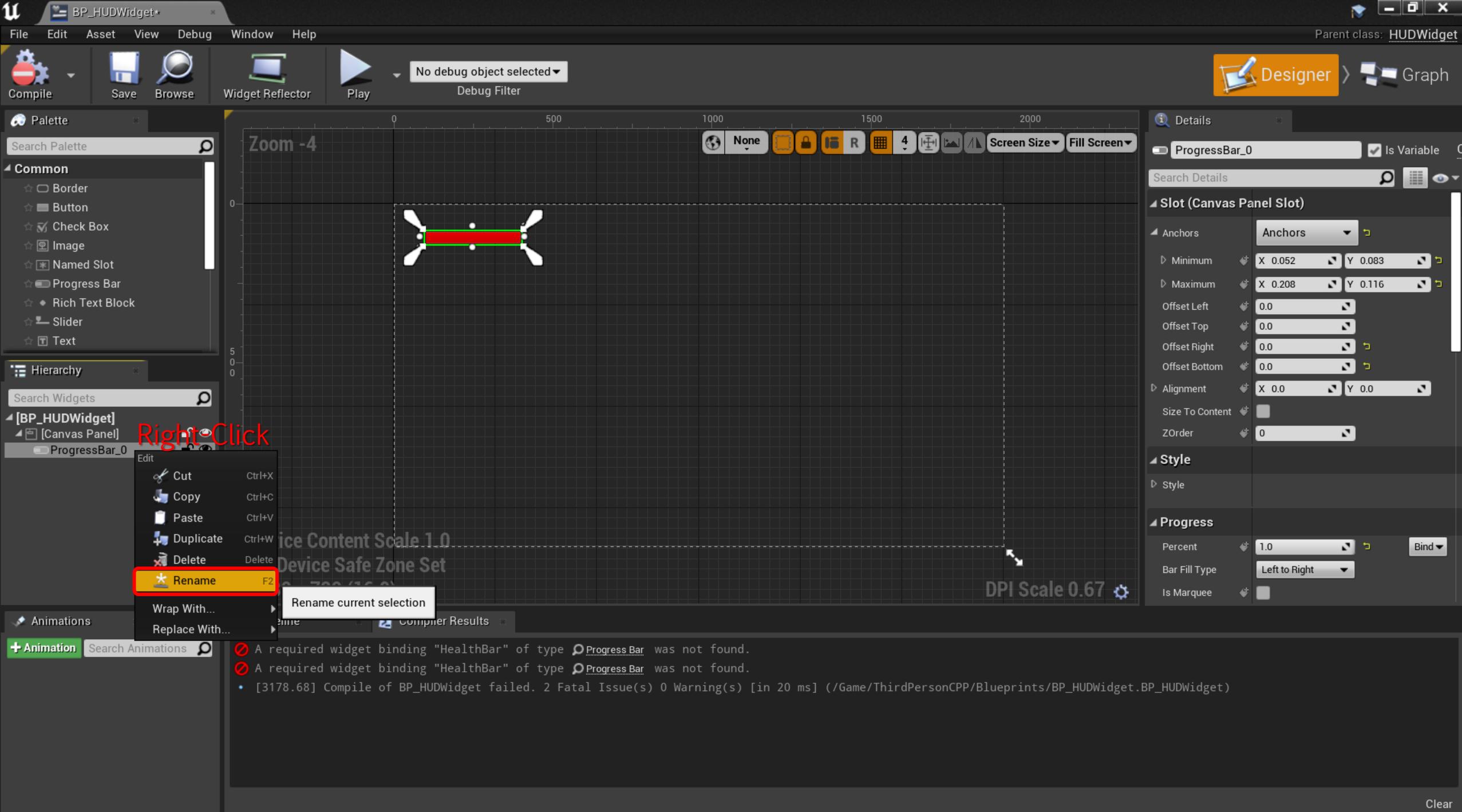


The screenshot shows the Content Browser interface. The top bar has buttons for Add/Import, Save All, and Content navigation. The left sidebar shows the project structure: Content, Geometry, Mannequin, Physics, StarterContent, ThirdPerson, and ThirdPersonCPP (selected). The Content Browser lists items under the "Blueprints" category: BP_DodgeballGameMode, BP_DodgeballPlayerController, BP_DodgeballProjectile, BP_EnemyCharacter, BP_GhostWall, BP_HUDWidget (selected and highlighted with a yellow box), BP_RestartWidget, BP_VictoryBox, BP_Wall, and ThirdPersonCharacter. A red arrow points to the "BP_HUDWidget" icon. The bottom status bar indicates "10 items (1 selected)" and "View Options".









BP_HUDWidget*

File Edit Asset View Debug Window Help Parent class: HUDWidget

Compile Save Browse Widget Reflector Play No debug object selected Debug Filter

Zoom -4 -125 x 1,101 Screen Size Fill Screen

0 500 1000 1500 2000

Common Border Button Check Box Image Named Slot Progress Bar Rich Text Block Slider Text

Hierarchy [BP_HUDWidget] [Canvas Panel] HealthBar

Device Content Scale 1.0
No Device Safe Zone Set
1280 x 720 (16:9)

DPI Scale 0.67

Animations + Animation Search Animations

Timeline Compiler Results

No A required widget binding "HealthBar" of type Progress Bar was not found.
A required widget binding "HealthBar" of type Progress Bar was not found.
[3178.68] Compile of BP_HUDWidget failed. 2 Fatal Issue(s) 0 Warning(s) [in 20 ms] (/Game/ThirdPersonCPP/Blueprints/BP_HUDWidget.BP_HUDWidget)

Designer Graph

Details

ProgressBar_0 Is Variable

Search Details

Slot (Canvas Panel Slot)

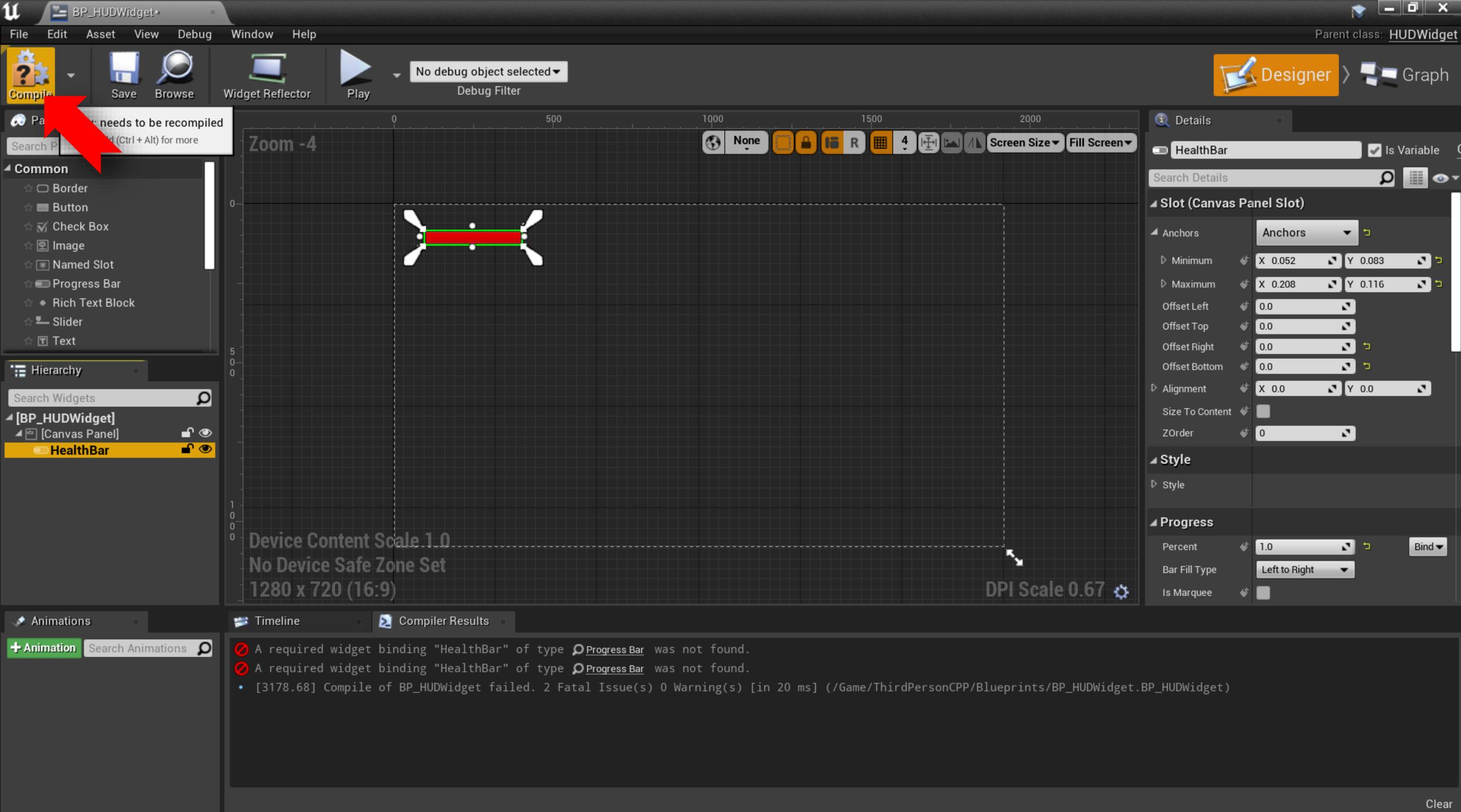
Anchors Anchors Minimum X: 0.052 Y: 0.083 Maximum X: 0.208 Y: 0.116 Offset Left 0.0 Offset Top 0.0 Offset Right 0.0 Offset Bottom 0.0 Alignment X: 0.0 Y: 0.0 Size To Content ZOrder 0

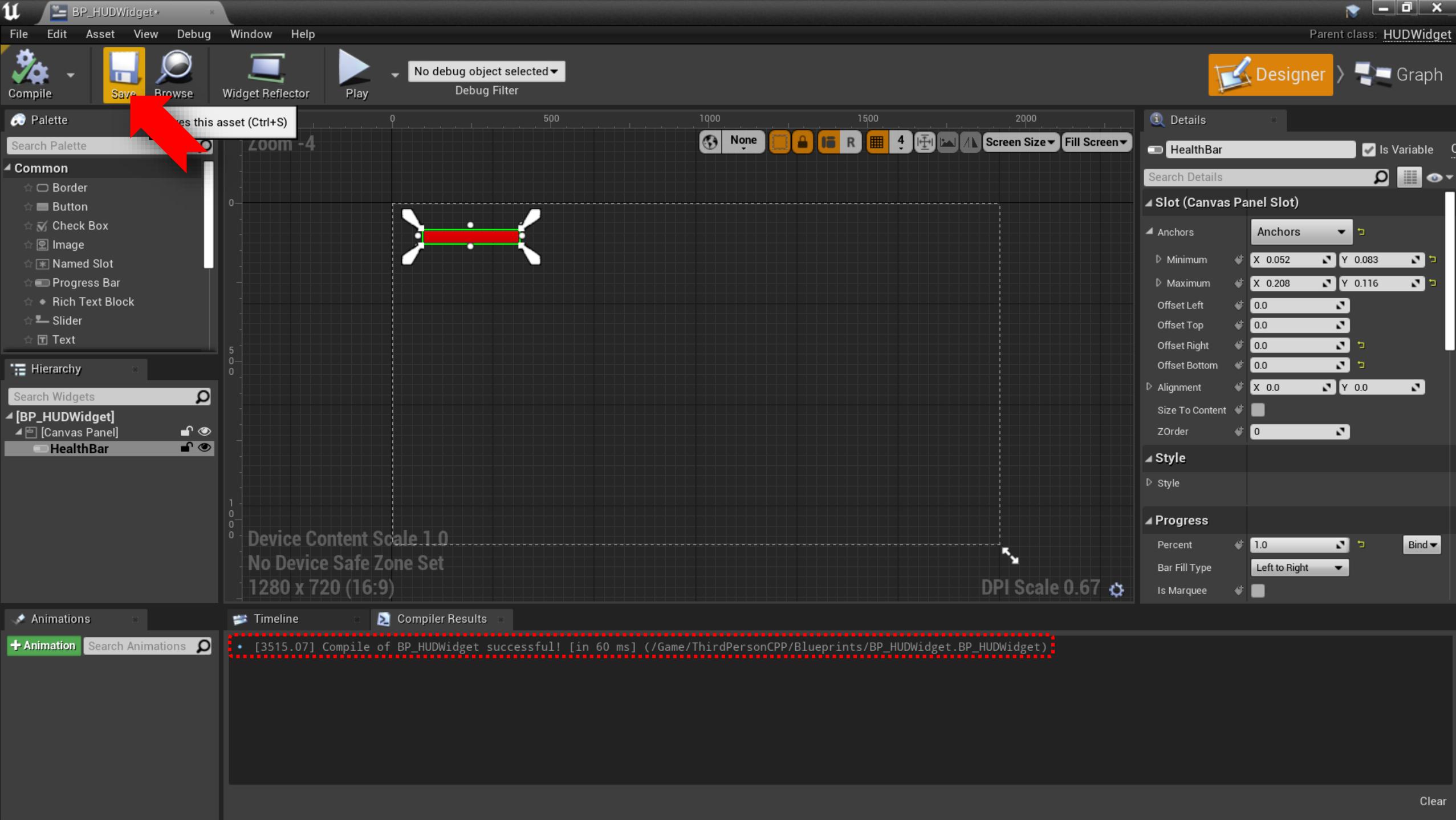
Style

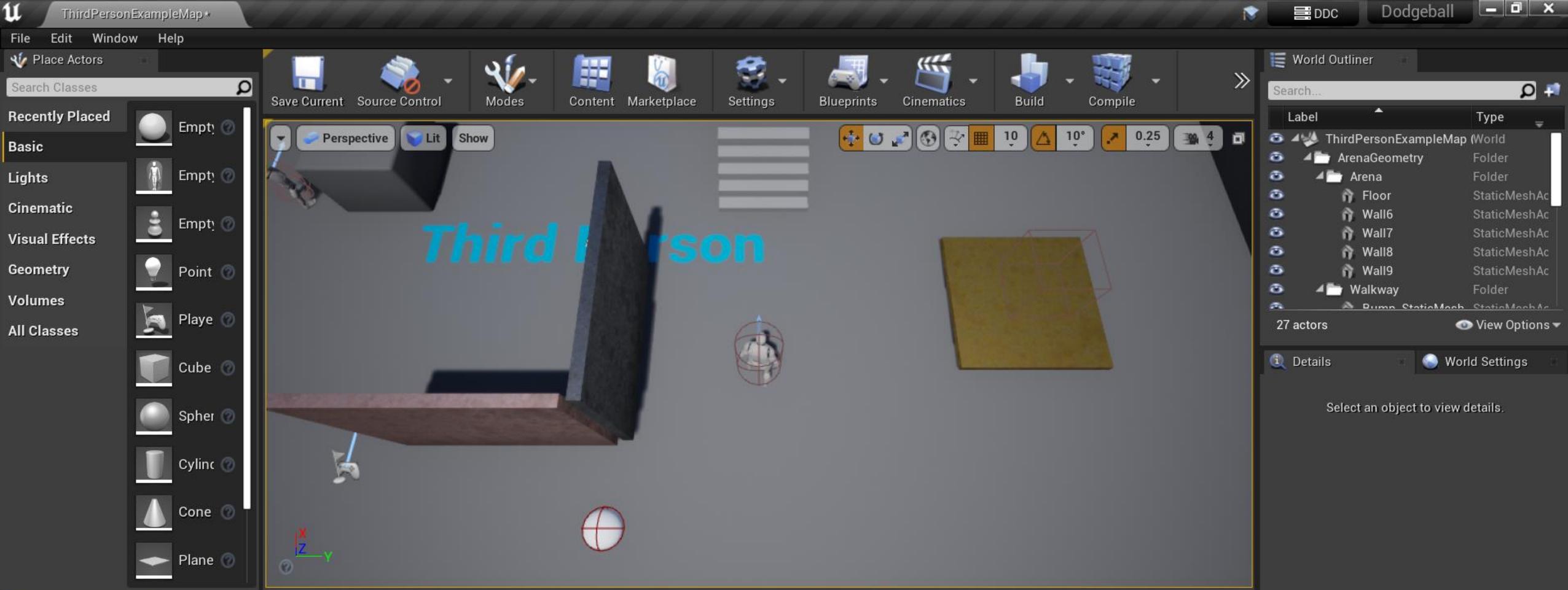
Progress

Percent 1.0 Bar Fill Type Left to Right Is Marquee

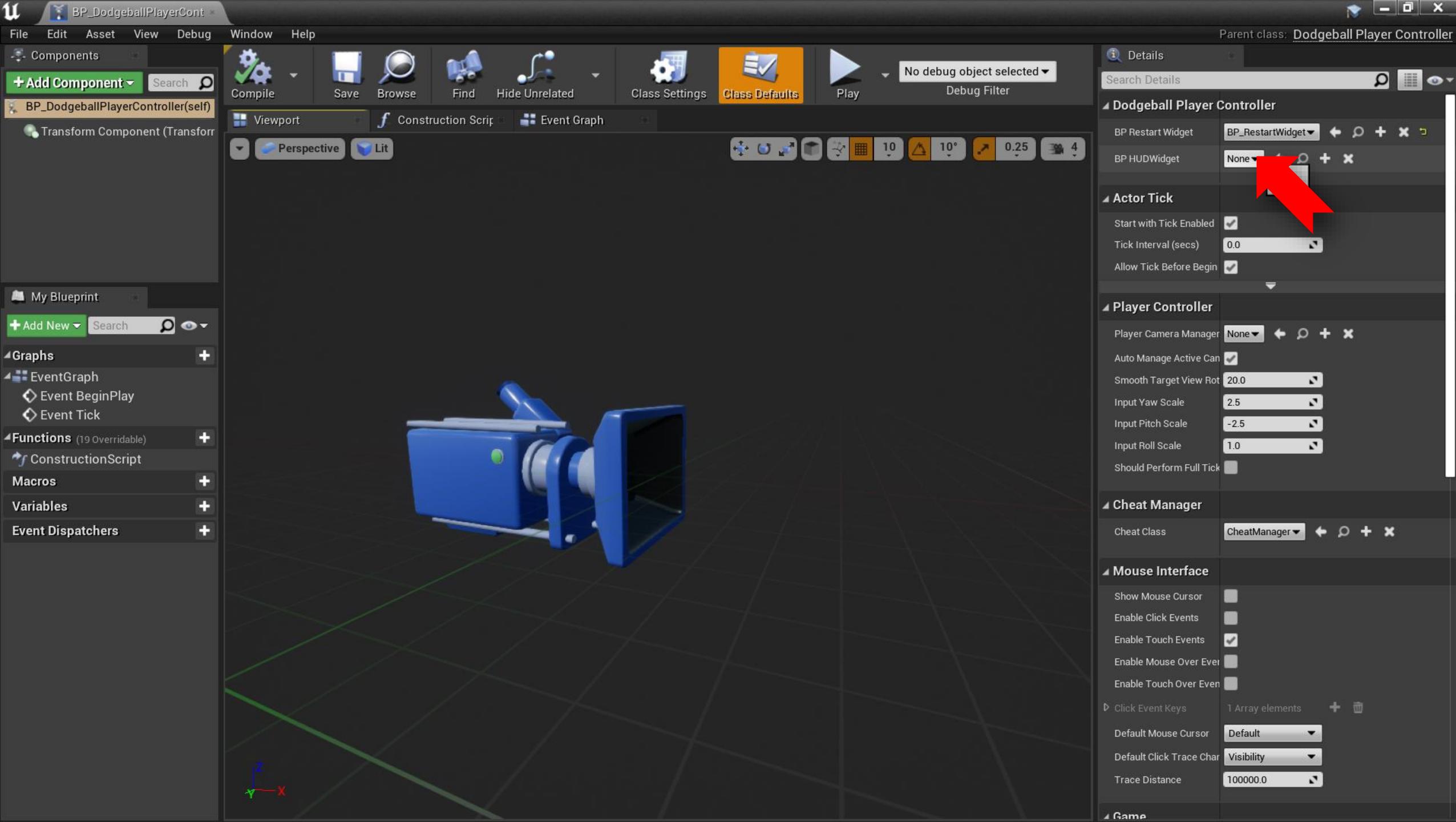
Clear

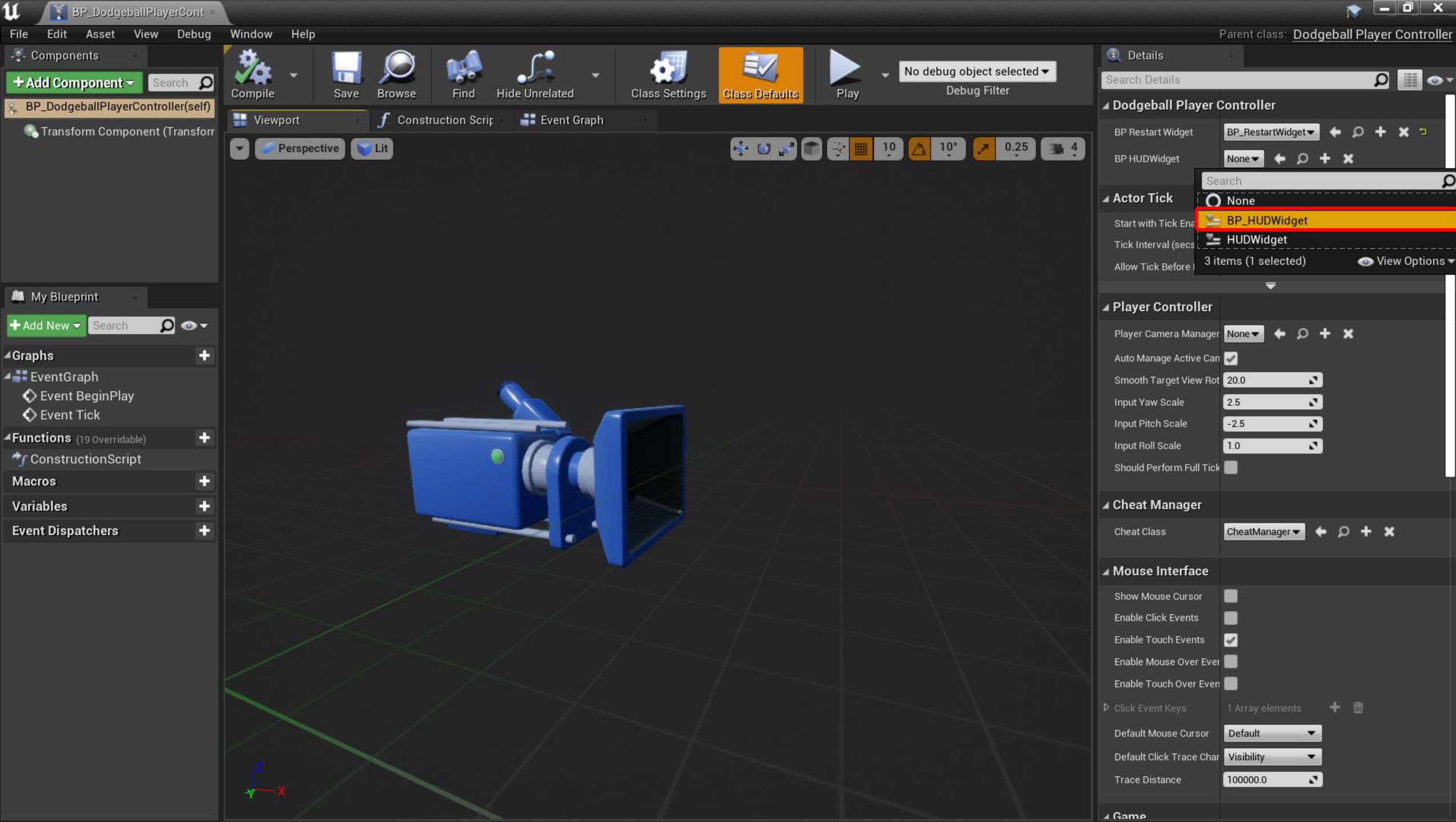


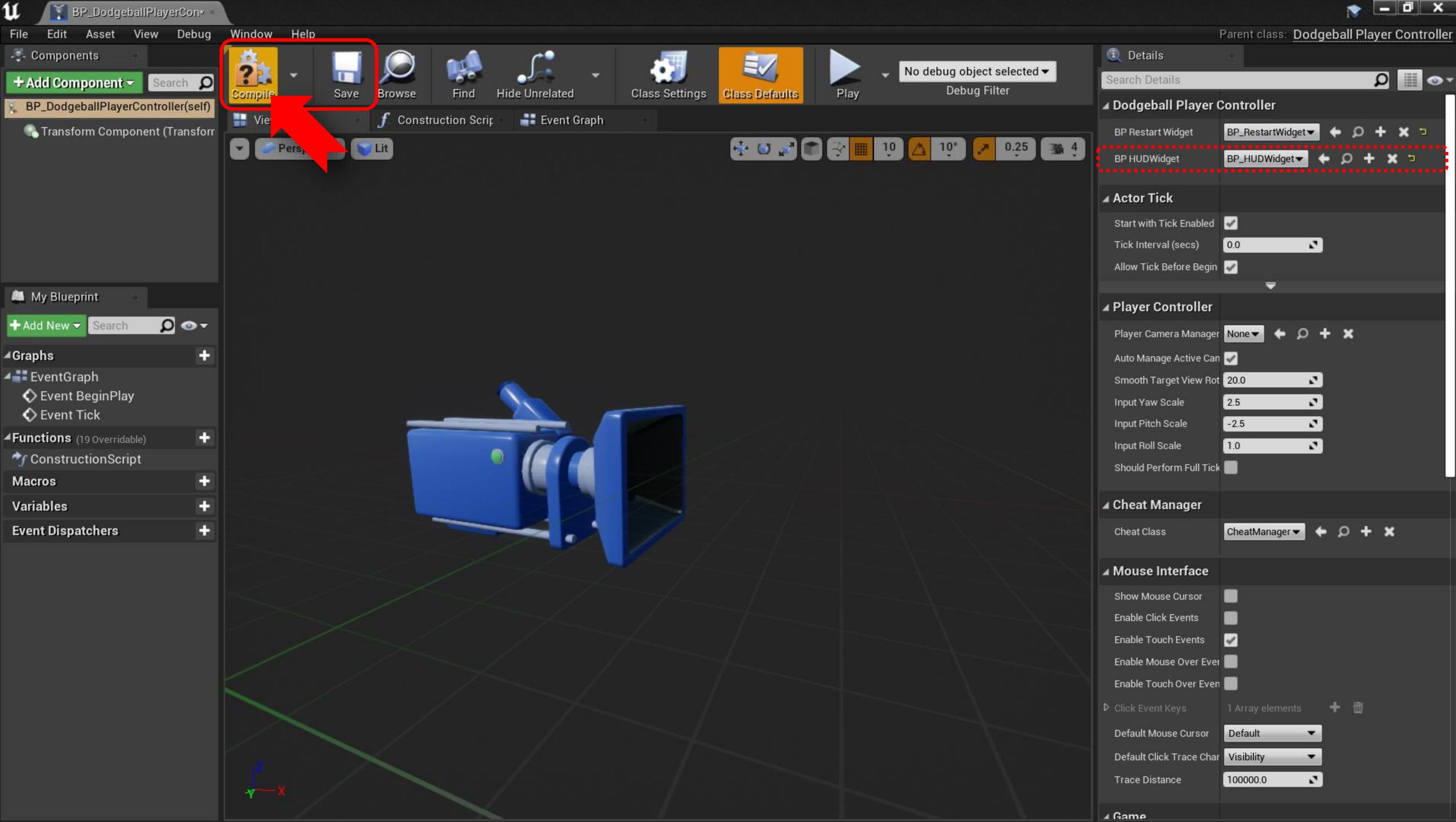


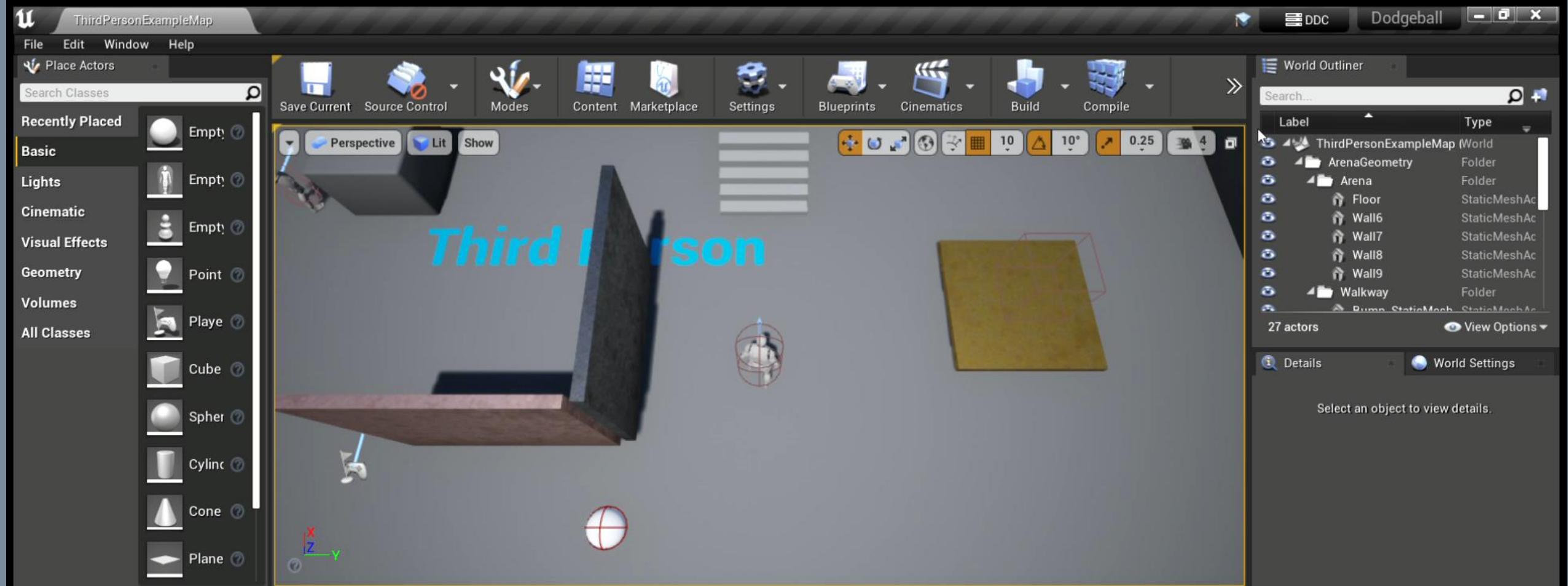


The screenshot shows the Content Browser interface. The top bar includes Add/Import, Save All, and Content navigation. The left sidebar shows a tree structure with Content, Geometry, Mannequin, Physics, StarterContent, ThirdPerson, and ThirdPersonCPP (selected). The Content Browser lists items under the ThirdPersonCPP/Blueprints folder: BP_DodgeballGameMode, BP_DodgeballPlayerController (selected and highlighted with a red arrow), BP_DodgeballProjectile, BP_EnemyCharacter, BP_GhostWall, BP_HUDWidget, BP_RestartWidget, BP_VictoryBox, BP_Wall, and ThirdPersonCharacter. The bottom status bar indicates "10 items (1 selected)" and "View Options".









The Content Browser panel displays the "Blueprints" section under "ThirdPersonCPP". It shows 10 items: BP_Dodgeball, BP_DodgeballGameMode, BP_DodgeballPlayerController, BP_DodgeballProjectile, BP_EnemyCharacter, BP_GhostWall, BP_HUDWidget, BP_RestartWidget, BP_VictoryBox, and BP_Wall. Each item has a preview icon and a "ThirdPersonCharacter" entry. The bottom of the panel indicates "10 items" and "View Options".



Activity 8.01: Improving The RestartWidget

The screenshot shows the Unreal Engine Content Browser interface. The top half displays a 3D preview of a game scene titled "Third Person". In the center, there's a large blue text "Third Person". To the right, there's a yellow cube with a red wireframe cube inside it. A small character model is positioned near the bottom left. The bottom half of the screen shows the Content Browser's file tree and preview area. The file tree on the left lists categories like Cinematic, Visual Effects, Geometry, Volumes, and All Classes. The main preview area shows various game assets: a controller icon, a mannequin, a sphere, a cylinder, a cone, a plane, a ghost wall, a HUD widget, a victory box, and a wall character. A red arrow points to the "BP_Restart Widget" icon, which is highlighted with a yellow box. The status bar at the bottom indicates "10 items (1 selected)".

Cinematic
Visual Effects
Geometry
Volumes
All Classes

Content Browser

Add/Import Save All Content > ThirdPersonCPP > Blueprints

Content

- Geometry
- Mannequin
- Physics
- StarterContent
- ThirdPerson
- ThirdPersonCPP
- Blueprints
- Maps
- C++ Classes

Search Paths Filters Search Blueprints

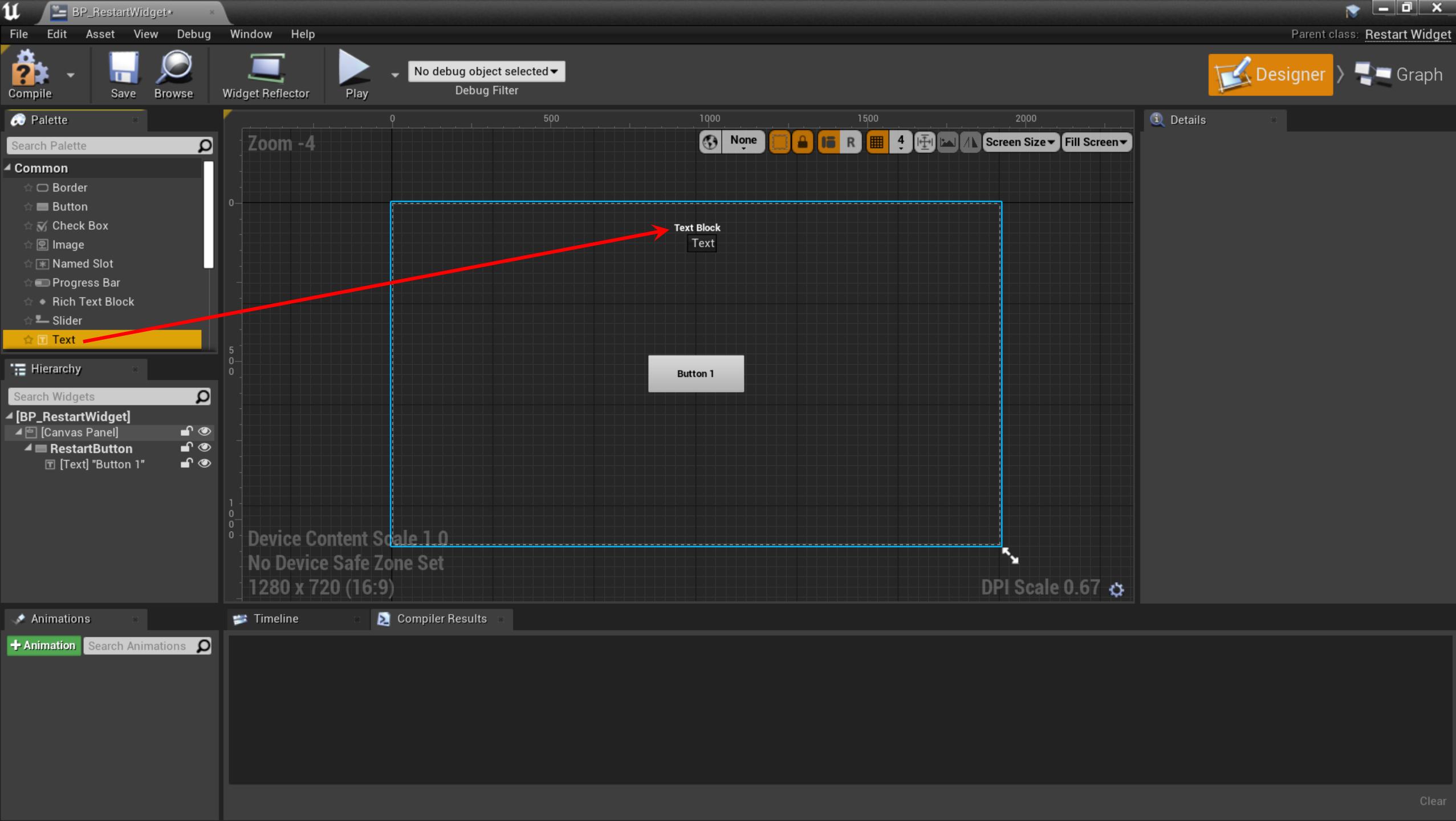
BP_Dodgeball GameMode
BP_Dodgeball PlayerController
BP_Dodgeball Projectile
BP_EnemyCharacter
BP_GhostWall
BP_HUDWidget
BP_RestartWidget
BP_VictoryBox
BP_Wall
ThirdPersonCharacter

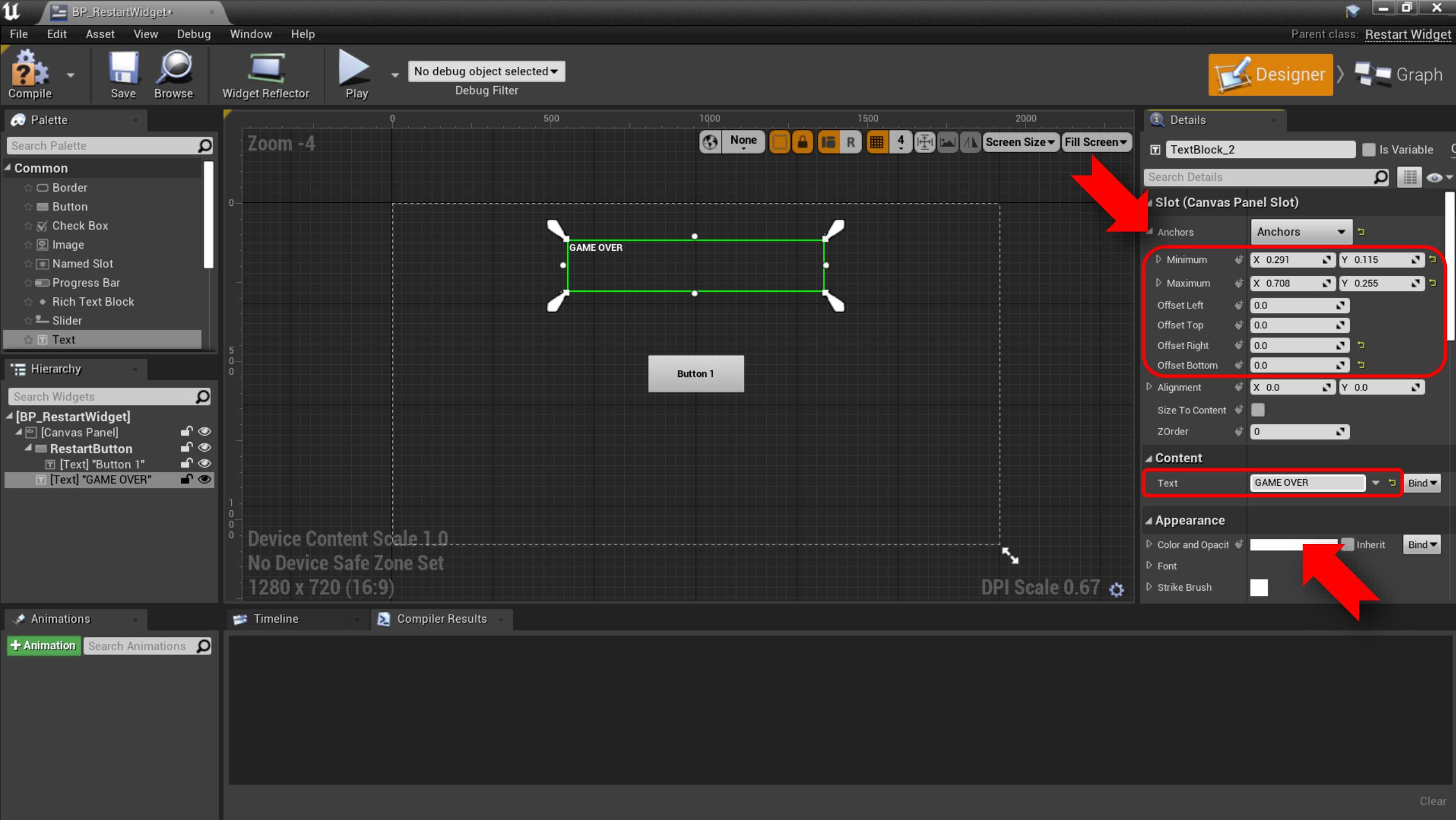
10 items (1 selected)

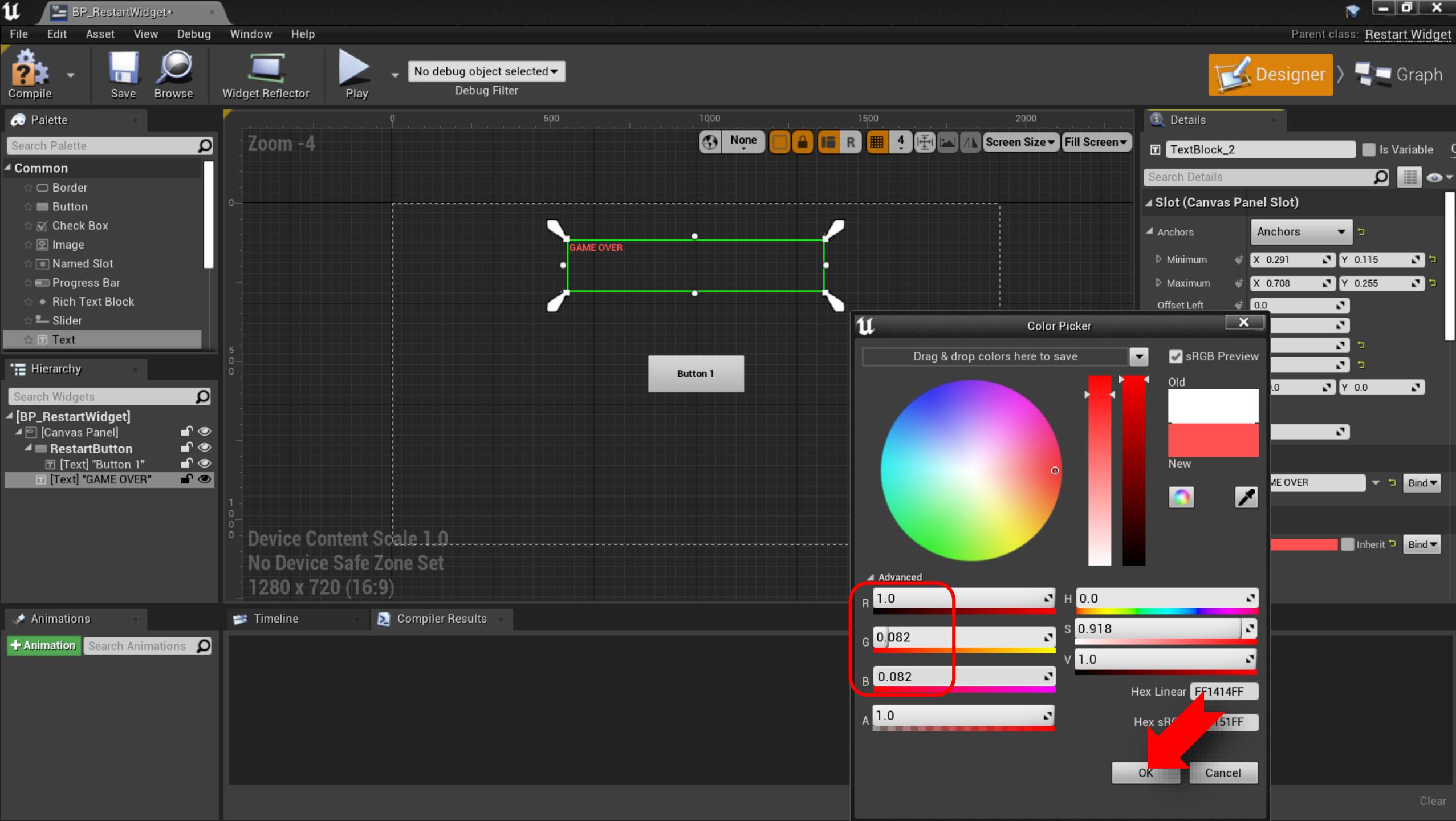
Floor
Wall6
Wall7
Wall8
Wall9
Walkway
Runn StaticMesh StaticMeshActor
27 actors

View Options Details World Settings

Select an object to view details.







BP_RestartWidget*

File Edit Asset View Debug Window Help

Parent class: Restart Widget

Compile Save Browse Widget Reflector Play No debug object selected Debug Filter

Zoom -4

0 500 1000 1500 2000

None Screen Size Fill Screen

TextBlock_2 Is Variable

Search Details

Common

- Border
- Button
- Check Box
- Image
- Named Slot
- Progress Bar
- Rich Text Block
- Slider
- Text

Hierarchy

[BP_RestartWidget]

- [Canvas Panel]
- RestartButton
 - [Text] "Button 1"
 - [Text] "GAME OVER"

Device Content Scale 1.0
No Device Safe Zone Set
1280 x 720 (16:9)

DPI Scale 0.67

Animations

+ Animation Search Animations

Timeline Compiler Results

Designer Graph

Details

TextBlock_2 Is Variable

Appearance

Color and Opacity Inherit Bind

Font

Font Family Roboto

Typeface Bold

Size 100 Bind

Letter Spacing 0

Font Material None

Outline Settings

Strike Brush

Shadow Offset X 1.0 Y 1.0

Shadow Color Bind

Min Desired Width 0.0

Transform Policy None

Justification Bind

Clear

The screenshot shows the Unreal Engine Designer interface. A 'GAME OVER' text element is selected in the center canvas, which has a bounding box and handles. To the right, the 'Details' panel displays the text's properties under the 'Appearance' section. The 'Size' field is set to 100, and the 'Justification' dropdown is highlighted with a red box. The 'Designer' tab is active at the top right.

BP_RestartWidget*

File Edit Asset View Debug Window Help

Parent class: Restart Widget

Compile Save Browse Widget Reflector Play No debug object selected Debug Filter

Zoom -4 0 500 1000 1500 2000 Screen Size Fill Screen

GAME OVER

Restart

Device Content Scale 1.0
No Device Safe Zone Set
1280 x 720 (16:9)

DPI Scale 0.67

Details TextBlock_0 Is Variable

Slot (Button Slot) Padding 4.0, 2.0
Horizontal Align Vertical Alignme

Content Text Restart Bind

Appearance Color and Opacity Inherit Bind

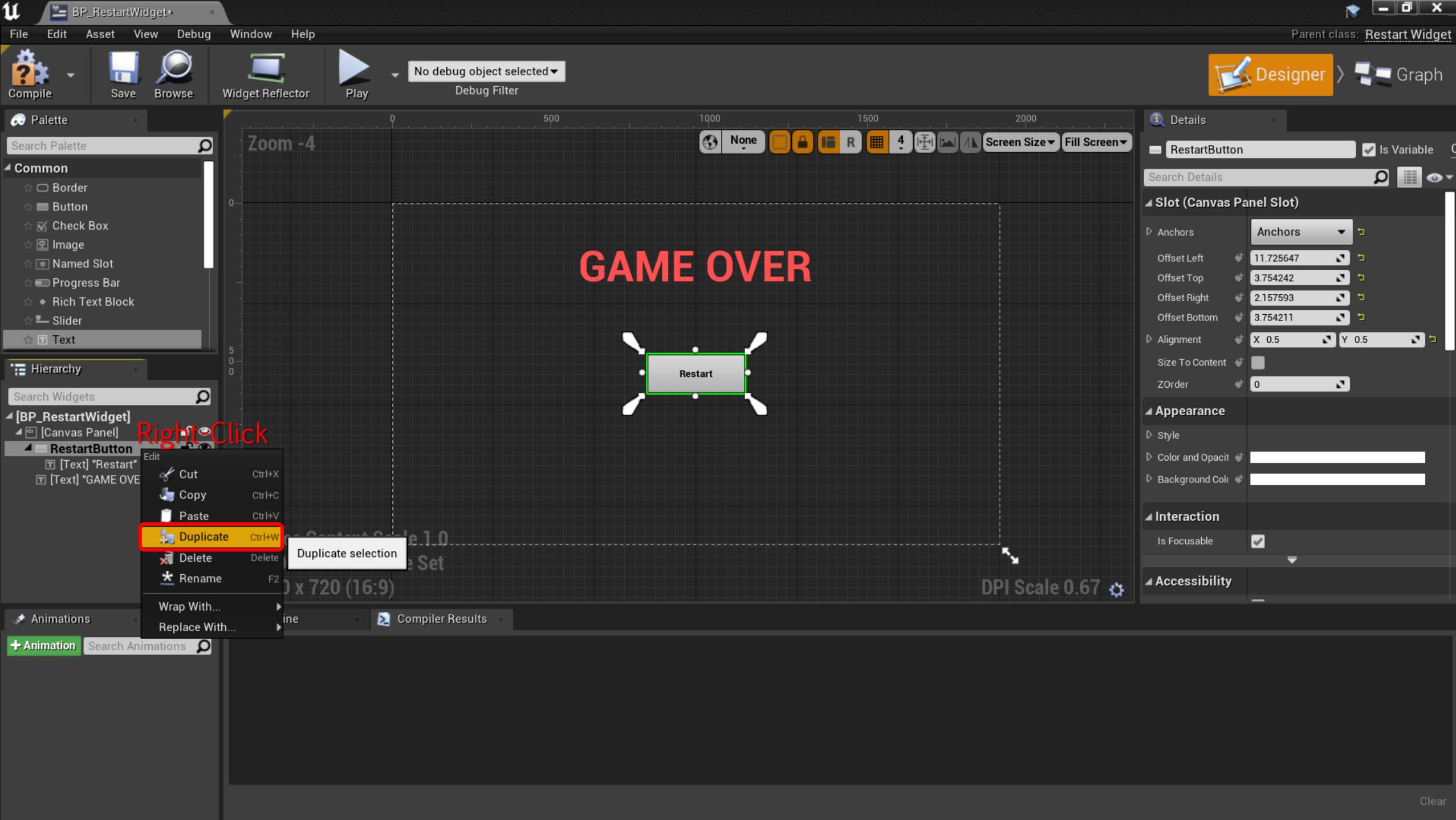
Font Family Roboto Typeface Bold Size 24 Letter Spacing 0 Font Material None

Animations Timeline Compiler Results

+ Animation Search Animations

Clear

The screenshot shows the Unreal Engine Designer interface for a widget named 'BP_RestartWidget'. The main canvas displays a large red 'GAME OVER' text and a white 'Restart' button. The 'Hierarchy' panel on the left shows the widget's structure: [BP_RestartWidget] contains a [Canvas Panel] which holds a [RestartButton] and two Text components, one labeled 'Restart' and another labeled 'GAME OVER'. The 'Content' tab in the Details panel is selected, showing the 'Text' component with its value set to 'Restart'. The 'Appearance' tab is also visible, showing font settings for 'Roboto' in bold, size 24, and letter spacing 0. The 'Timeline' and 'Compiler Results' tabs are at the bottom.



BP_RestartWidget*

File Edit Asset View Debug Window Help

Parent class: Restart Widget

Compile Save Browse Widget Reflector Play No debug object selected Debug Filter

Zoom -4 0 500 1000 1500 2000 Screen Size Fill Screen

GAME OVER

Restart

ExitButton

[BP_RestartWidget] [Canvas Panel] [RestartButton] [Text "Restart"] [Text "GAME OVER"]

Device Content Scale 1.0
No Device Safe Zone Set
1280 x 720 (16:9)

DPI Scale 0.67

Animations Timeline Compiler Results

+ Animation Search Animations

Designer Graph

Details

RestartButton_1 Is Variable

Search Details

Slot (Canvas Panel Slot)

Anchors Anchors 0.0 0.0 13.88324 7.508453 X 0.5 Y 0.5 ZOrder 0

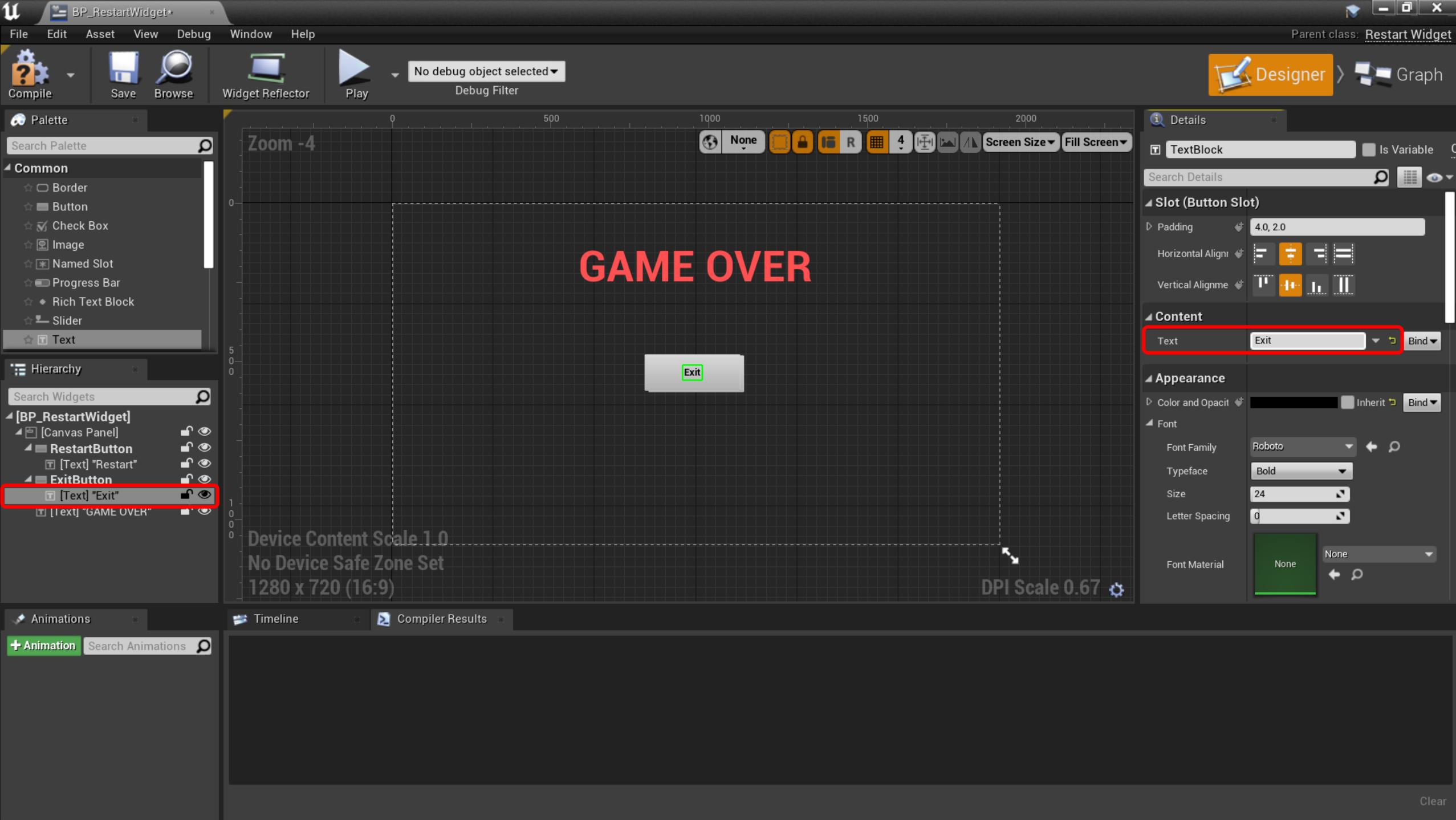
Appearance Style Color and Opacity Background Color

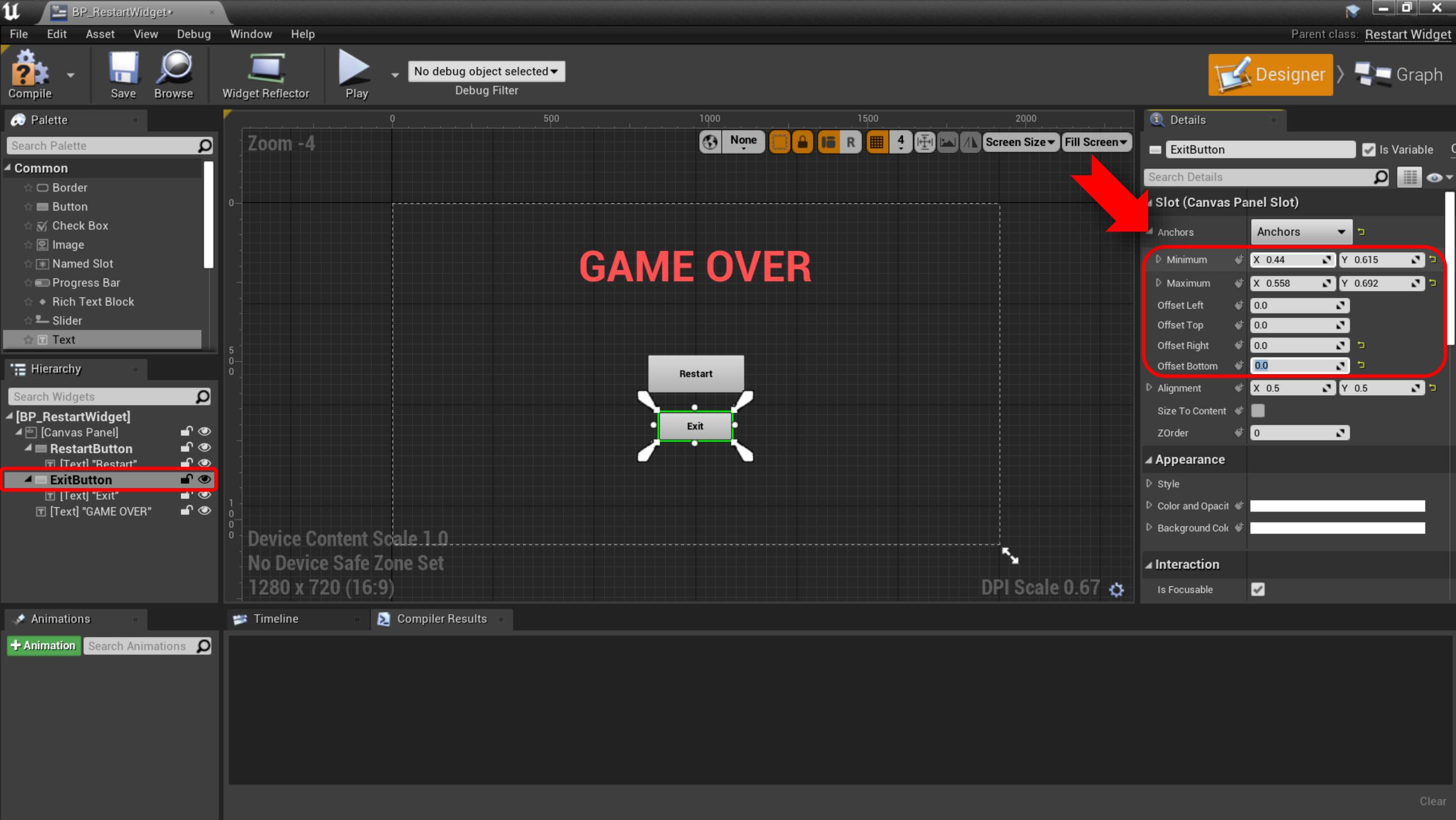
Interaction Is Focused

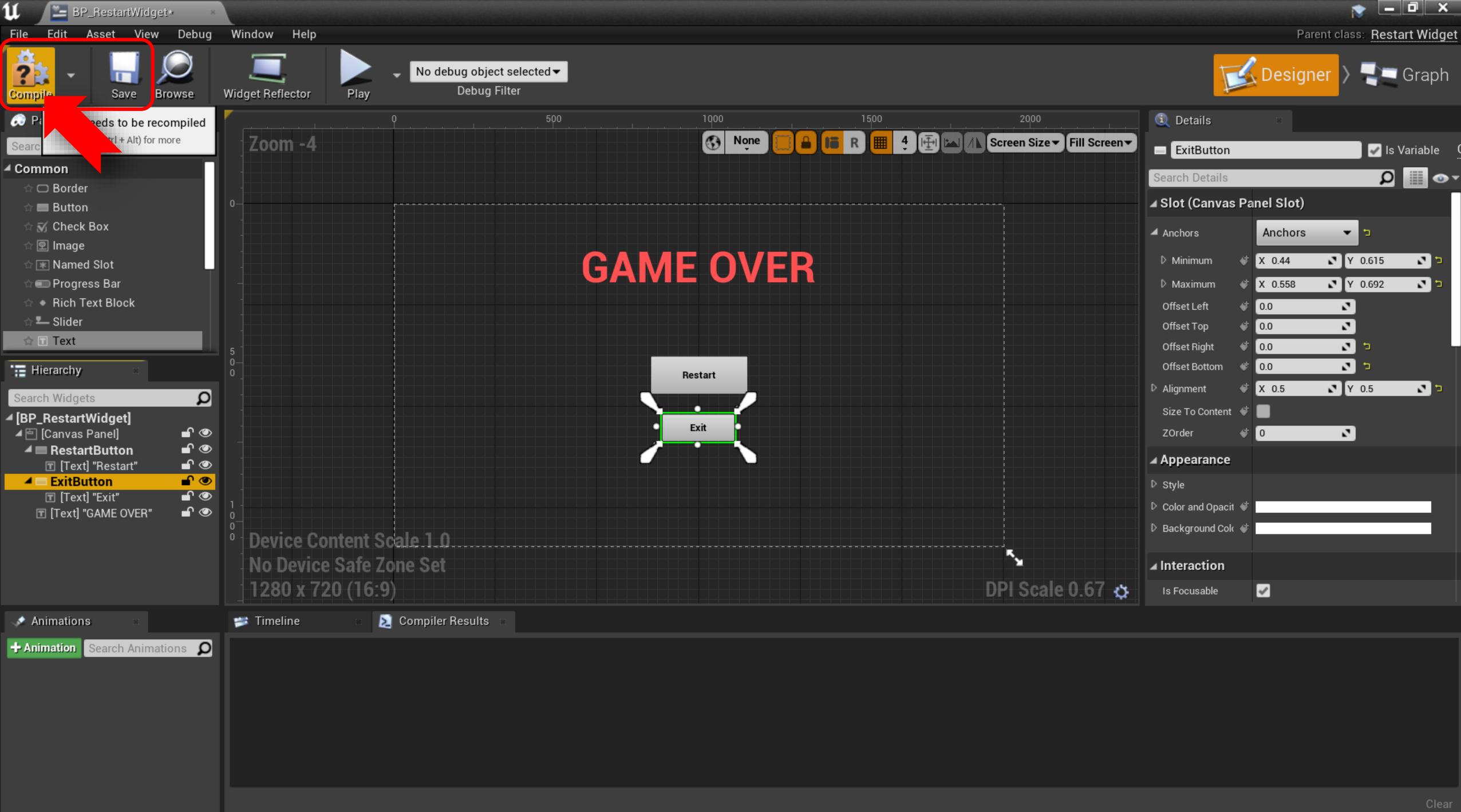
Accessibility

Clear

The screenshot shows the Unreal Engine Designer interface. The main canvas displays a large red "GAME OVER" text centered on the screen. Below it is a grey rectangular button with the word "Restart" in black text. The "ExitButton" node in the Hierarchy panel is highlighted with a red border. The Details panel on the right shows settings for the "Slot (Canvas Panel Slot)" component, including Anchors, Alignment, and Appearance options. The Hierarchy panel lists the nodes of the current widget, and the Animations panel shows an animation named "Animation". The status bar at the bottom indicates a device content scale of 1.0, a DPI scale of 0.67, and a resolution of 1280 x 720 (16:9).







파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(I) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+Q) Dodgeball — □ ×

Develop Win64 로컬 Windows 디버거 Live Share

서로다른 파일 보기

HealthComponent.h HealthComponent.cpp DodgeballCharacter.h DodgeballCharacter.cpp RestartWidget.h*

Dodgeball // Fill out your copyright notice in the Description page of Project Settings.

```
1 // Fill out your copyright notice in the Description page of Project Settings.
2
3 #pragma once
4
5 #include "CoreMinimal.h"
6 #include "Blueprint/UserWidget.h"
7 #include "RestartWidget.generated.h"
8
9 /**
10 * 
11 */
12 UCLASS()
13 class DODGEBALL_API URestartWidget : public UUserWidget
14 {
15     GENERATED_BODY()
16
17 protected:
18
19     UFUNCTION()
20     void OnRestartClicked();
21
22     UFUNCTION()
23     void OnExitClicked();
24
25 public:
26
27     UPROPERTY(meta = (BindWidget))
28     class UButton* RestartButton;
29
30     UPROPERTY(meta = (BindWidget))
31     class UButton* ExitButton;
32
33     virtual void NativeOnInitialized() override;
34
35 };

```

Ctrl+S

100 % 문제가 검색되지 않음 줄: 31 문자: 28 열: 31 탭 CRLF

솔루션 탐색기

솔루션 탐색기 검색(Ctrl+Shift+F)

솔루션 탐색기 'Dodgeball' (2/2개 프로젝트)

- Engine
- UE4
- Games
- Dodgeball
- 참조
- 외부 종속성
- Config
- Source
- Dodgeball
- Dodgeball.Build.cs
- Dodgeball.cpp
- Dodgeball.h
- DodgeballCharacter.cpp
- DodgeballCharacter.h
- DodgeballFunctionLibrary.h
- DodgeballFunctionLibrary.cpp
- DodgeballGameMode.cpp
- DodgeballGameMode.h
- DodgeballPlayerController.h
- DodgeballPlayerController.cpp
- DodgeballProjectile.cpp
- DodgeballProjectile.h
- EnemyCharacter.cpp
- EnemyCharacter.h
- HealthComponent.cpp
- HealthComponent.h
- HealthInterface.cpp

솔루션 탐색기 Git 변경 내용

준비 ↑ 소스 제어에 추가 ▲ 1

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(I) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+Q) Dodgeball — □ ×

Live Share

HealthComponent.cpp DodgeballCharacter.h DodgeballCharacter.cpp RestartWidget.h RestartWidget.cpp* NativeOnInitialized()

Dodgeball

```
4 #include "RestartWidget.h"
5 #include "Kismet/GameplayStatics.h"
6 #include "Components/Button.h"
7 #include "DodgeballIPlayerController.h"
8
9 void URestartWidget::OnRestartClicked()
10 {
11     ADodgeballIPlayerController* PlayerController = Cast<ADodgeballIPlayerController>(GetOwningPlayer());
12     if (PlayerController != nullptr)
13     {
14         PlayerController->HideRestartWidget();
15     }
16
17     UGameplayStatics::OpenLevel(this, FName(*UGameplayStatics::GetCurrentLevelName(this)));
18 }
19
20 void URestartWidget::OnExitClicked()
21 {
22     UKismetSystemLibrary::QuitGame(this, nullptr, EQuitPreference::Quit, true);
23 }
24
25 void URestartWidget::NativeOnInitialized()
26 {
27     Super::NativeOnInitialized();
28
29     if (RestartButton != nullptr)
30     {
31         RestartButton->OnClicked.AddDynamic(this, &URestartWidget::OnRestartClicked);
32     }
33     if (ExitButton != nullptr)
34     {
35         ExitButton->OnClicked.AddDynamic(this, &URestartWidget::OnExitClicked);
36     }
37 }
```

Ctrl+S

슬루션 탐색기

슬루션 탐색기 검색(Ctrl+Shift+F)

슬루션 탐색기 'Dodgeball' (2/2개 프로젝트)

- Engine
- UE4
- Games
- Dodgeball
 - 참조
 - 외부 종속성
 - Config
 - Source
 - Dodgeball
 - Dodgeball.Build.cs
 - Dodgeball.cpp
 - Dodgeball.h
 - DodgeballCharacter.cpp
 - DodgeballCharacter.h
 - DodgeballFunctionLibrary.cpp
 - DodgeballFunctionLibrary.h
 - DodgeballGameMode.cpp
 - DodgeballGameMode.h
 - DodgeballPlayerController.cpp
 - DodgeballPlayerController.h
 - DodgeballProjectile.cpp
 - DodgeballProjectile.h
 - EnemyCharacter.cpp
 - EnemyCharacter.h
 - HealthComponent.cpp
 - HealthComponent.h
 - HealthInterface.cpp

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Top Bar:** 파일(F), 편집(E), 보기(V), Git(G), 프로젝트(P), 빌드(B), 디버그(D), 테스트(S), 분석(N), 도구(T), 확장(X), 창(W), 도움말(H), 검색 (Ctrl+Q), Dodgeball.
- Solution Explorer:** Shows two files: HealthComponent.cpp and DodgeballCharacter.h, both under the Dodgeball project.
- Code Editor:** Displays the HealthComponent.cpp file with several code snippets related to restarting and exiting the game.
- Build Menu (highlighted by a red arrow):**
 - 솔루션 빌드(B) (Selected)
 - 솔루션 다시 빌드
 - 솔루션 정리(C)
 - 솔루션의 전체 프로그램 데이터베이스 파일 빌드
 - 솔루션에서 코드 분석 실행(Y) (Alt+F11)
 -
 - Dodgeball 빌드(U) (Ctrl+B)
 - Dodgeball 다시 빌드(E)
 - Dodgeball 정리(N)
 - Dodgeball에서 코드 분석 실행(A) (Alt+F11)
 - 프로젝트만()
 - 일괄 빌드(T)...
 - 구성 관리자(O)...
 -
 - 컴파일(M) (Ctrl+F7)
 - 파일에서 코드 분석 실행(F) (Ctrl+Shift+Alt+F7)
- Solution Task List:** Shows the tasks for the Dodgeball solution, including engine and game components like DodgeballCharacter.h and DodgeballProjectile.h.
- Bottom Status Bar:** 100%, 문제가 검색되지 않음, 줄: 35, 문자: 74, 열: 80, 탭, CRLF, 솔루션 탐색기, Git 변경 내용.
- Bottom Navigation Bar:** 저장되었습니다., 소스 제어에 추가, 1.



File Edit Window Help

Place Actors

Search Classes

Recently Placed



Empty



Basic



Sphere



Lights



Cylinder



Cinematic



Volume



Visual Effects



Point



Geometry



Volumetric



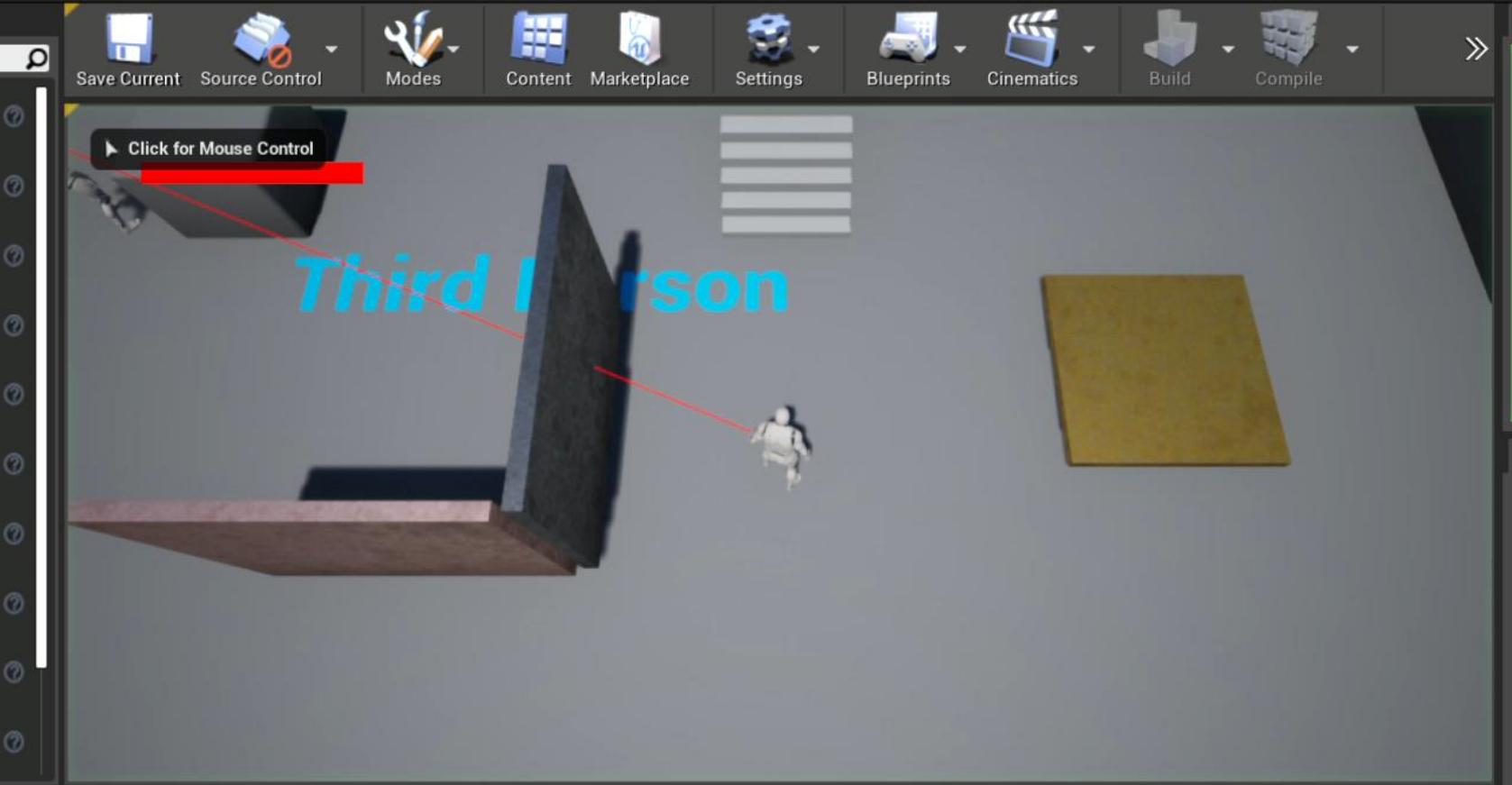
Volumes



Plane



All Classes



Content Browser

Add/Import



Content > ThirdPersonCPP > Blueprints

World Outliner

Label	Type
ThirdPersonExampleMap	World
ArenaGeometry	Folder
Arena	Folder
Floor	StaticMeshActor
Wall6	StaticMeshActor
Wall7	StaticMeshActor
Wall8	StaticMeshActor
Wall9	StaticMeshActor
Walkway	Folder
Runn_StaticMesh	StaticMeshActor

38 actors

View Options

Details

World Settings

Select an object to view details.

Content

- Geometry
- Mannequin
- Physics
- StarterContent
- ThirdPerson
- ThirdPersonCPP
 - Blueprints
 - Maps
- C++ Classes

Search Paths

Add/Import Save All Back Forward Content > ThirdPersonCPP > Blueprints

Filters Search Blueprints

10 items (1 selected)

Content

- BP_Dodgeball
- BP_Dodgeball
- BP_Dodgeball
- BP_EnemyCharacter
- BP_GhostWall
- BP_HUDWidget
- BP_RestartWidget
- BP_VictoryBox
- BP_Wall
- ThirdPersonCharacter

View Options

연습 과제

- › Activity 8.01까지 모두 완성한 **Dodgeball** 프로젝트를 제출 하시오.
- › 제출 방법: “프로젝트 폴더” 전체 압축
 - 압축 파일 내에서 다음 “5개 폴더” 삭제
 - 1) Content 폴더 안 StarterContent 폴더
 - 2) Intermediate 폴더
 - 3) Saved 폴더
 - 4) Binary 폴더
 - 5) .vs 폴더
 - 압축파일 업로드