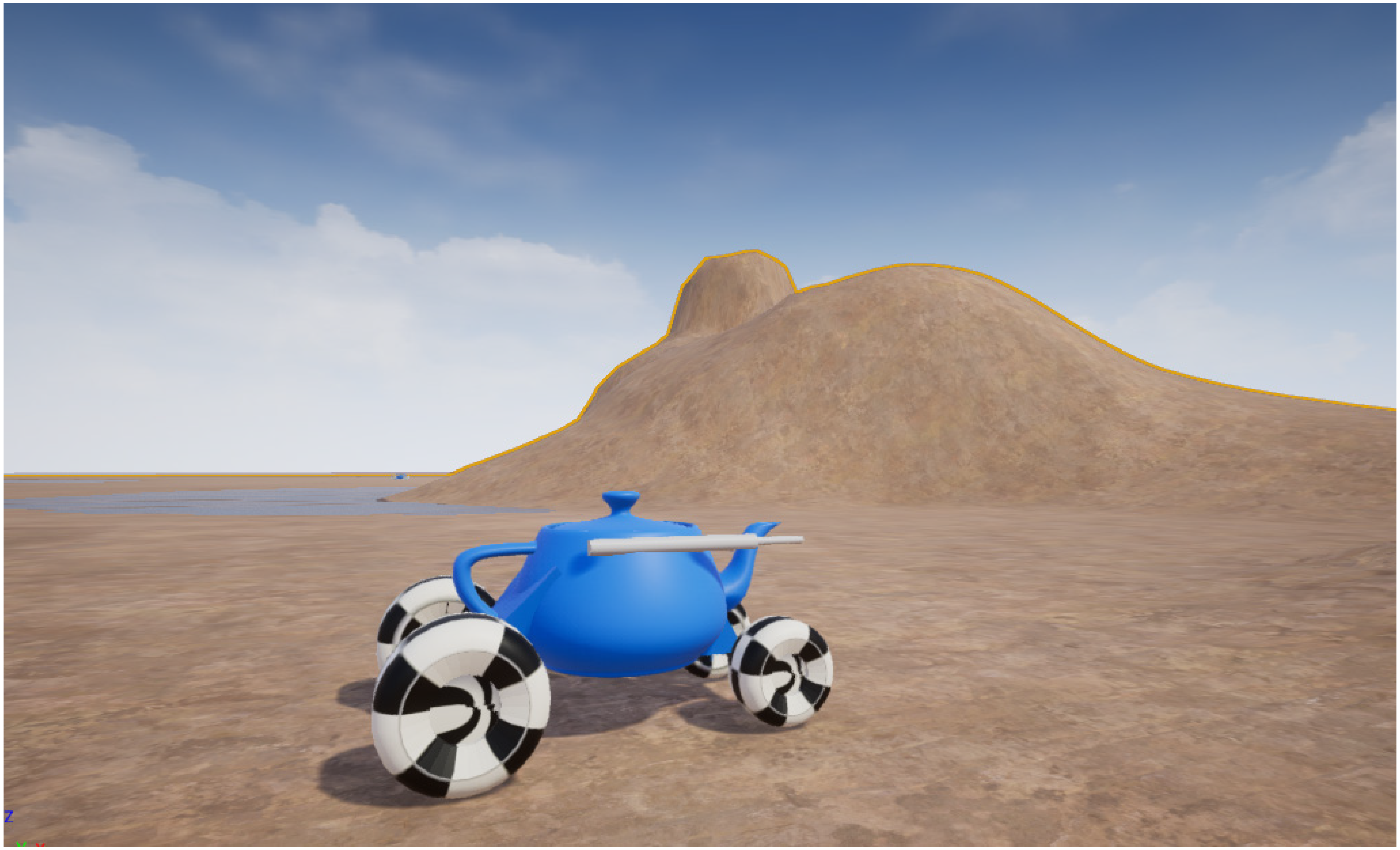
You need to get your buggy preped for exporting into Unreal.

For your first attempt, I suggest you create a very simple vehicle: a box for a body and 4 cylinders for wheels
Make the cylinders 60 centimeters in diameter and 40 centimeters high.

NOTE - Don’t expect to get this process right on the first time.There are dozens of things to go wrong
You may have to start over from the beginning!



Career Technical Education
SoCal ROC

Proper set-up is critical.

First you need to set up your model in 3D Max so it is acceptable to Unreal 4. Follow these steps:

NOTE!! - Unreal is very fussy. You must size your vehicle to Unreal's rules. Set 3D Max to CENTIMETER"S in the Units Setup.

1. Model must be in five separate meshes: the body and four separate wheels.

2. Name parts as: Body, Wheel_FR, Wheel_FL, Wheel_RR, Wheel_RL

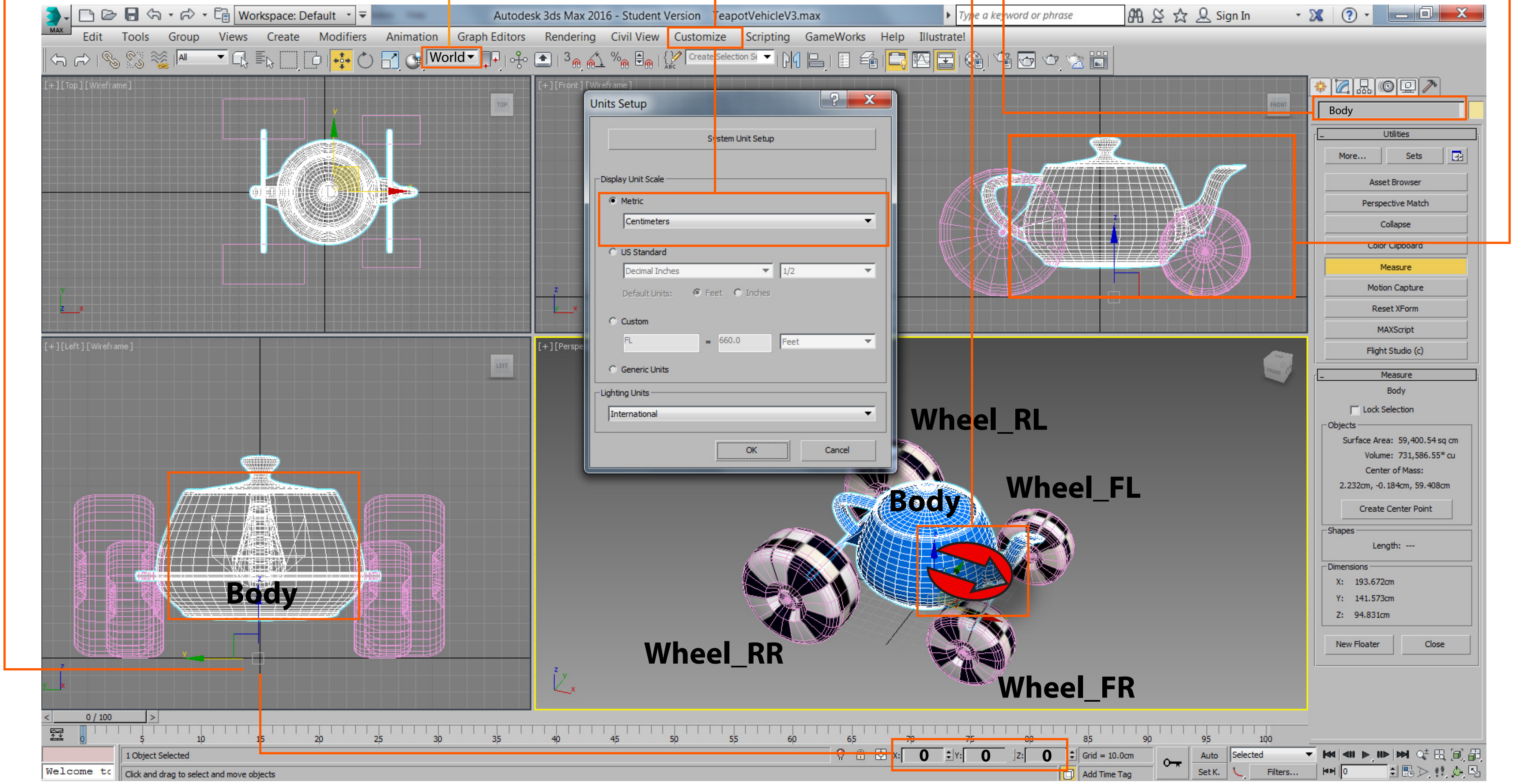
3. The Body pivot point must be at 0, 0, 0 Coordinates.

4. Model must be facing in the "X" (red) direction in the Perspective.

5. Unreal uses "cm" so set units to centimeters

6. Set View to "World"

7. Your vehicle should measure around 140 centimeters high.



Units Setup

System Unit Setup

Display Unit Scale

● Metric: Centimeters

○ US Standard: Decimal Inches, 1/2 — Default Units: ● Feet ○ Inches

○ Custom: FL = 660.0 Feet

○ Generic Units

Lighting Units: International

OK Cancel

Measure: Body

□ Lock Selection

Objects
Surface Area: 59,400.54 sq cm
Volume: 731,586.55* cu
Center of Mass: 2.232cm, -0.184cm, 59.408cm

Create Center Point

Shapes
Length: ---

Dimensions
X: 193.672cm
Y: 141.573cm
Z: 94.831cm

X: 0 Y: 0 Z: 0

8. Make your wheels 60 radius x 40 high

Make sure your wheels are evenly on the dark line (z=0)

Adding Custom Vehicles

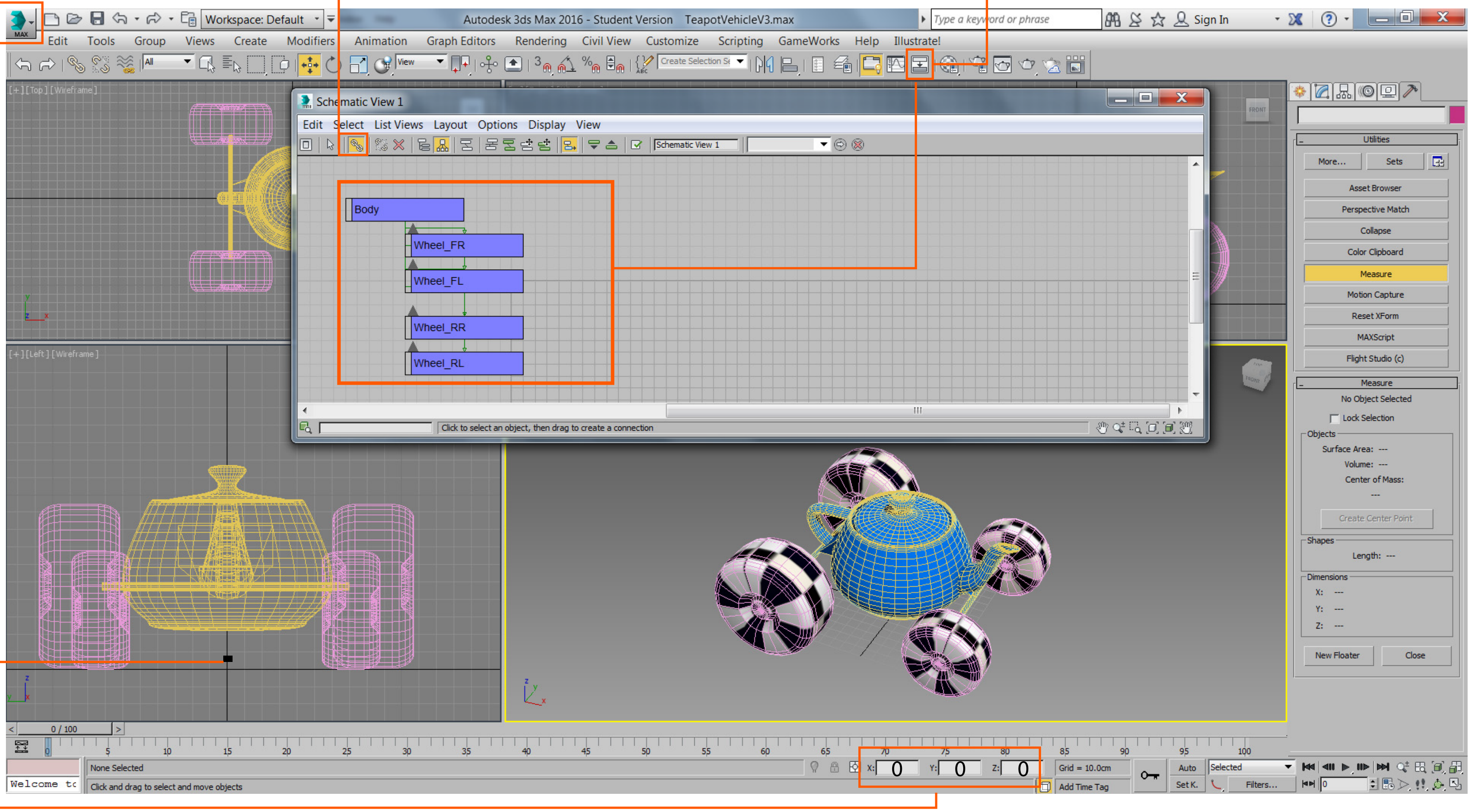
9. Right click on all four Wheels and the Body and convert them to Editible Meshes.

The last thing to do is creating a hierarchy making the wheels children of the body. This way wherever the parent (Body) goes the children (wheels) must follow.

(Make sure to link wheels to body - not body to wheels)

10. Open the "Schematic" viewport link the wheels to the body. Use the "Link" option.

The Vehicles Pivot Point must be set to 0,0,0



Click to select an object, then drag to create a connection

Measure: No Object Selected; Lock Selection; Objects: Surface Area: --- Volume: --- Center of Mass: ---; Create Center Point; Shapes: Length: ---; Dimensions: X: --- Y: --- Z: ---

None Selected

Click and drag to select and move objects

X: 0 Y: 0 Z: 0

Once prepared, export your buggy as an FBX file and save in your folder on the H drive.

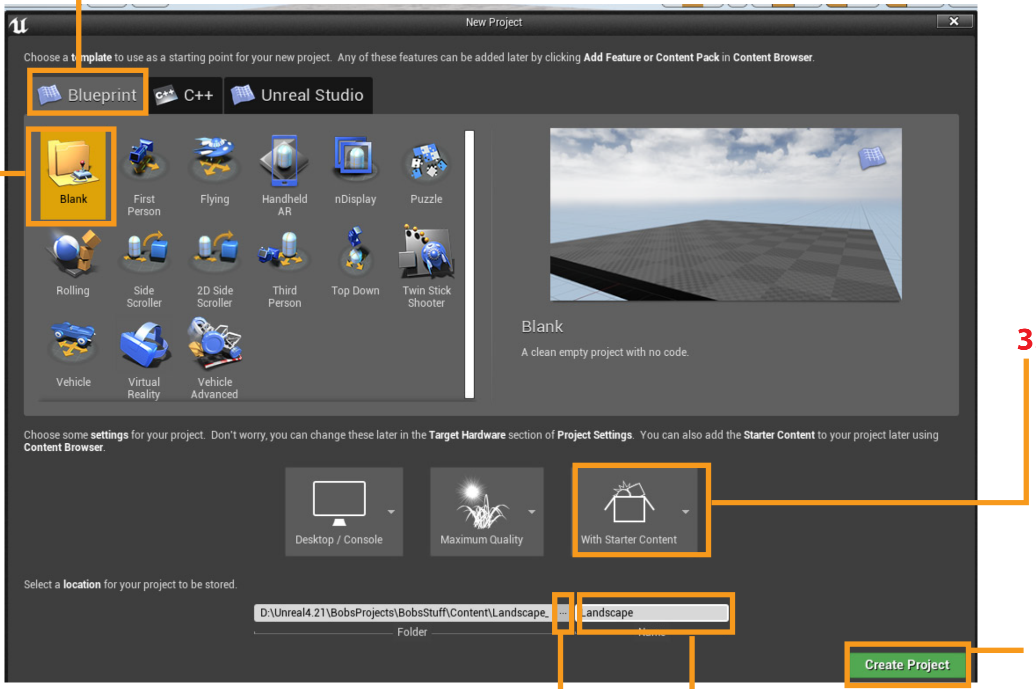
OPEN
UNREAL 4

NOTE: A most common problem with beginners is where to save your work. I recommend using the "H" drive on our system. It is secure and noone else can see your files. You may also use a flash drive as a back-up. Either way keep your files in one place.

Don't scatter them across the computer or you'll lose them.

1. Select "Blueprint"

2. Choose "Blank"



Choose a template to use as a starting point for your new project. Any of these features can be added later by clicking **Add Feature or Content Pack** in **Content Browser**.

Blank

A clean empty project with no code.

Choose some **settings** for your project. Don't worry, you can change these later in the **Target Hardware** section of **Project Settings**. You can also add the **Starter Content** to your project later using **Content Browser**.

Select a **location** for your project to be stored.

D:\Unreal4.21\BobsProjects\BobsStuff\Content\Landscape_

Landscape

[redacted]

3. Choose "With Starter Content"

4. Click on the three dots.
Create a folder on the "H" Drive.
Label the folder with your complete name.

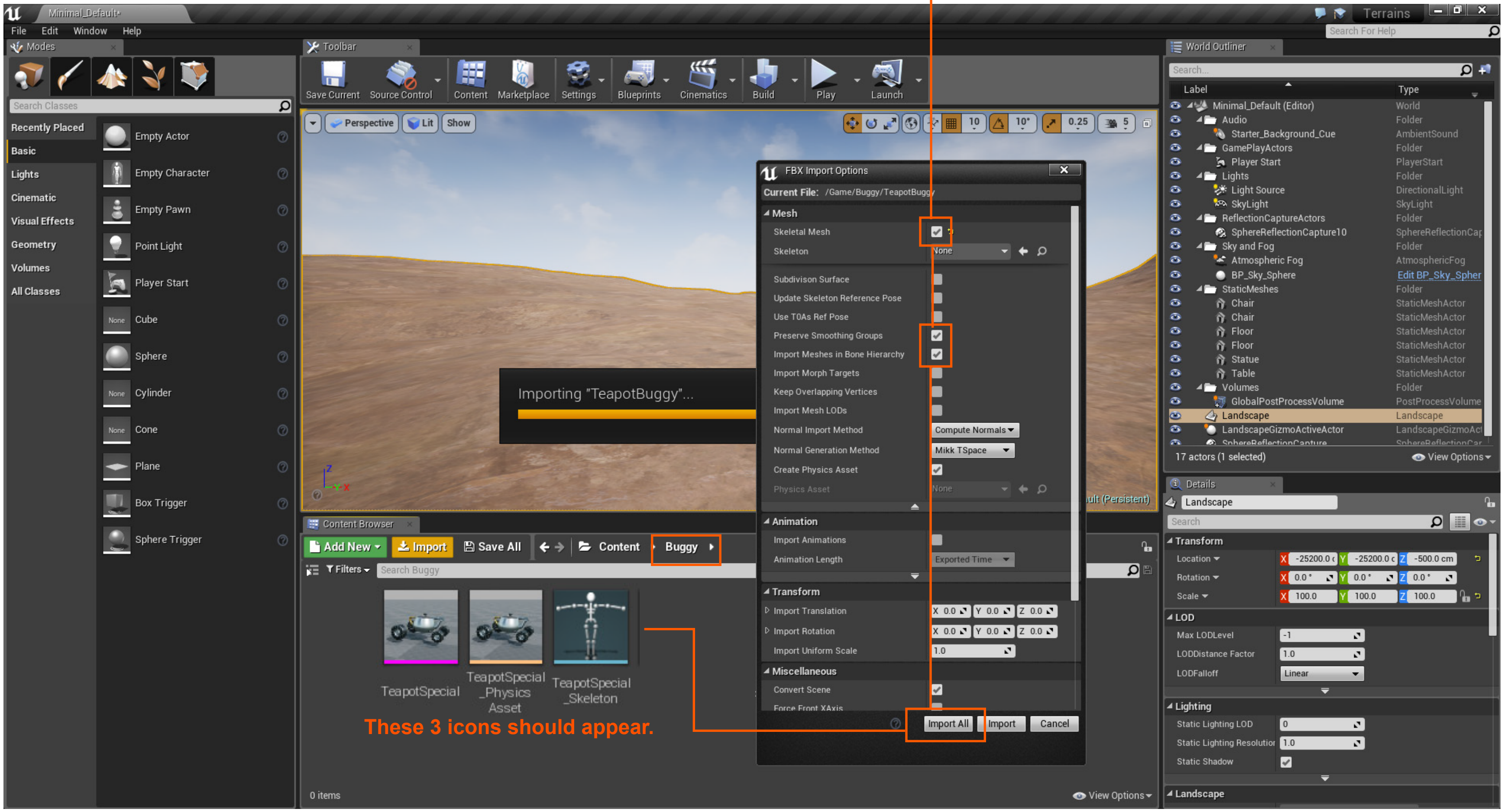
5. Name the project "Landscape".

6. Click "Create Project".

Create a new “Blank”project in Uneal 4

5. Create a new folder in the Content folder, name it after your vehicle.
6. Import the vehicle
7. Select the settings as shown

.



FBX Import Options

Current File: /Game/Buggy/TeapotBuggy

Mesh

- Skeletal Mesh ☑
- Skeleton None
- Subdivison Surface □
- Update Skeleton Reference Pose □
- Use T0As Ref Pose □
- Preserve Smoothing Groups ☑
- Import Meshes in Bone Hierarchy ☑
- Import Morph Targets □
- Keep Overlapping Vertices □
- Import Mesh LODs □
- Normal Import Method Compute Normals
- Normal Generation Method Mikk TSpace
- Create Physics Asset ☑
- Physics Asset None

Animation

- Import Animations □
- Animation Length Exported Time

Transform

- Import Translation X 0.0 Y 0.0 Z 0.0
- Import Rotation X 0.0 Y 0.0 Z 0.0
- Import Uniform Scale 1.0

Miscellaneous

- Convert Scene ☑
- Force Front XAxis □

These 3 icons should appear.

NOTE- If 4 separate wheels appear in this window, you made a mistake in 3D Max and either linked the body to the wheels or the wheels to one another. Go back and fix.

Your Vehicle Blueprint will be composed of several elements:

A Skeletal Mesh (automatically created when you import the vehicle from 3D Max)

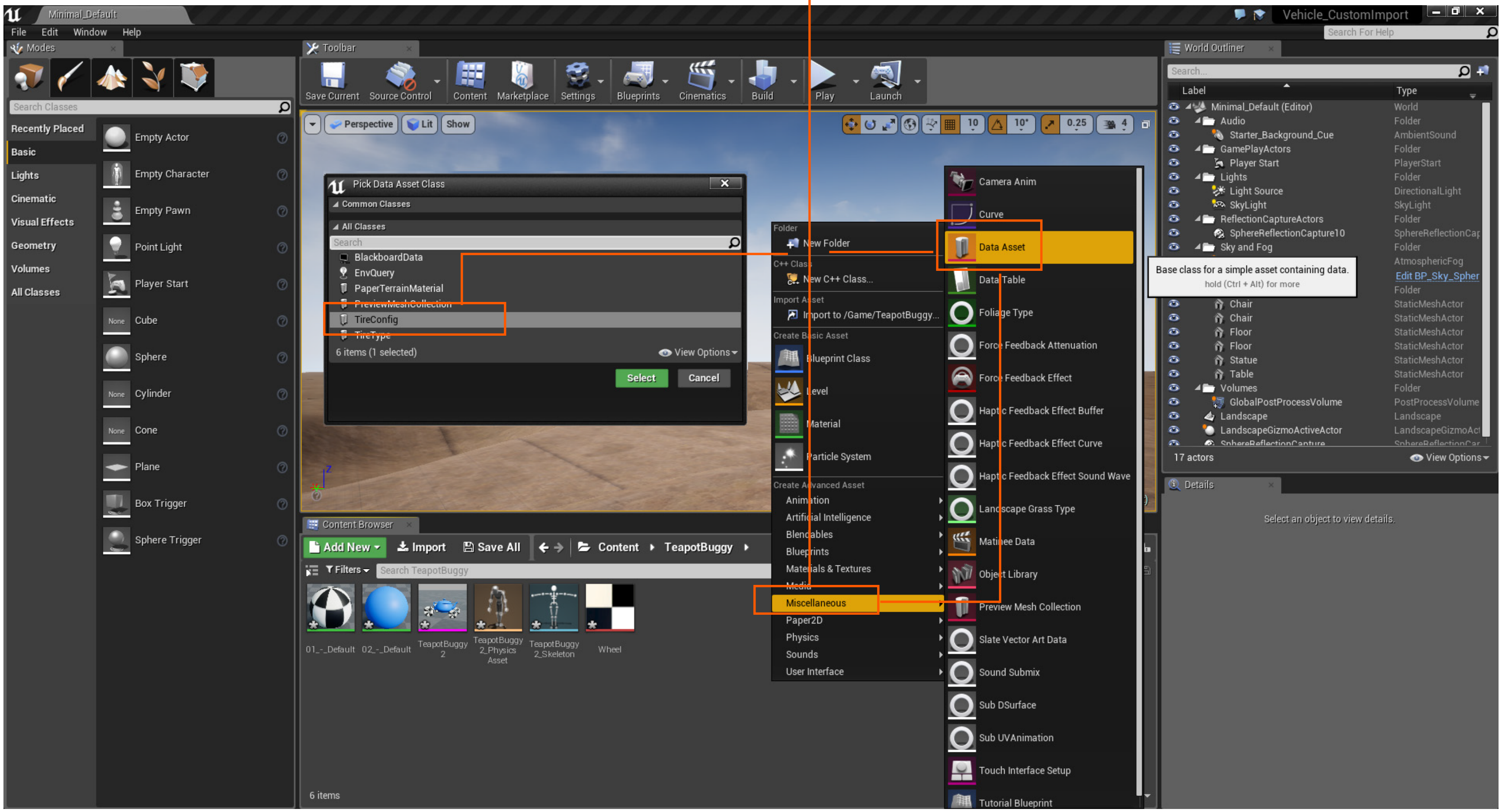
A Physics Asset (automatically created when you import the vehicle from 3D Max)

TireConfig Data Asset

Two Wheel Blueprints. (Front and Back)

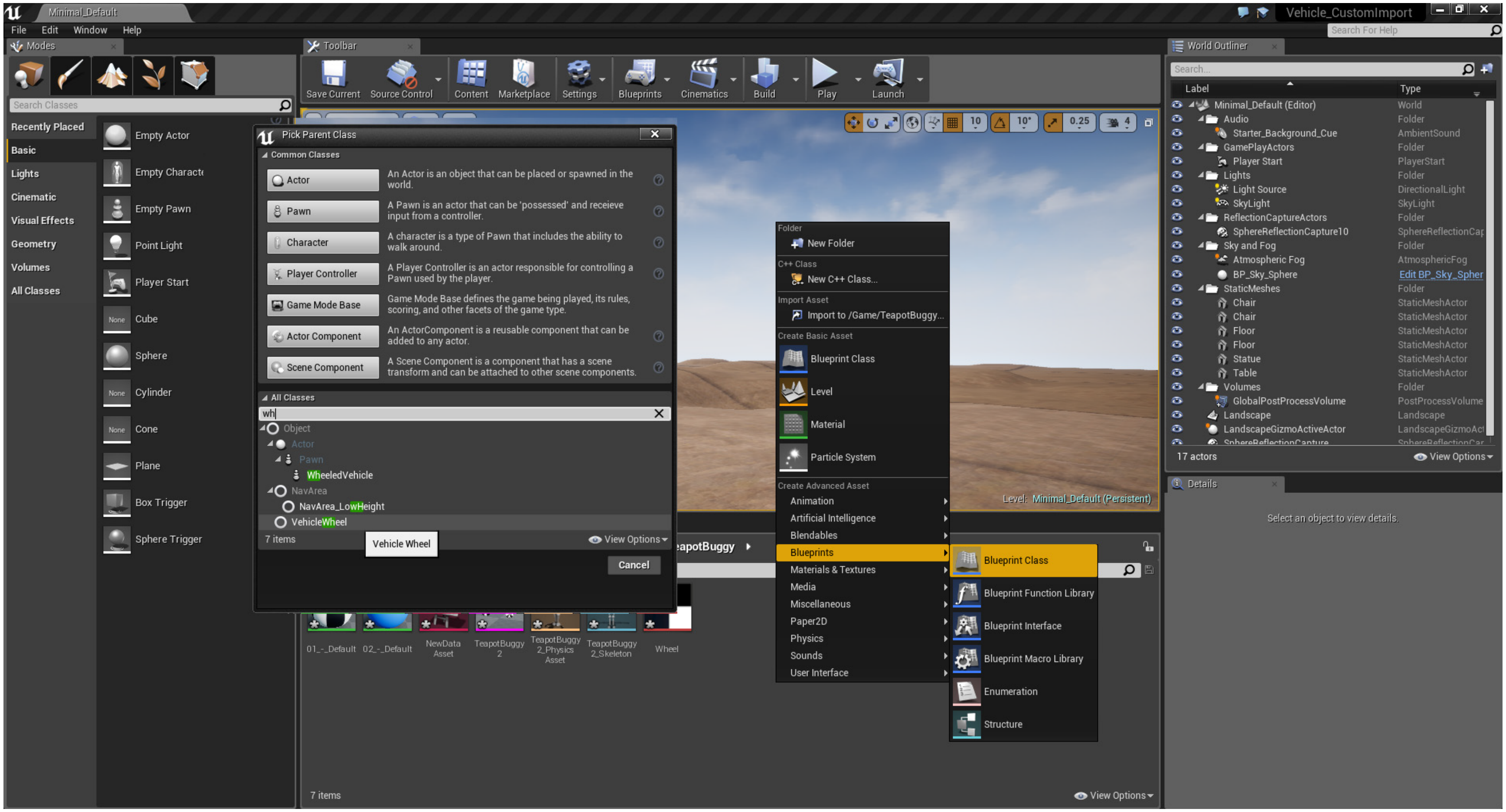
An Animation Blueprint

8. Right click in the gray Content areaand select Miscellaneous - Data Set - Tire Config.



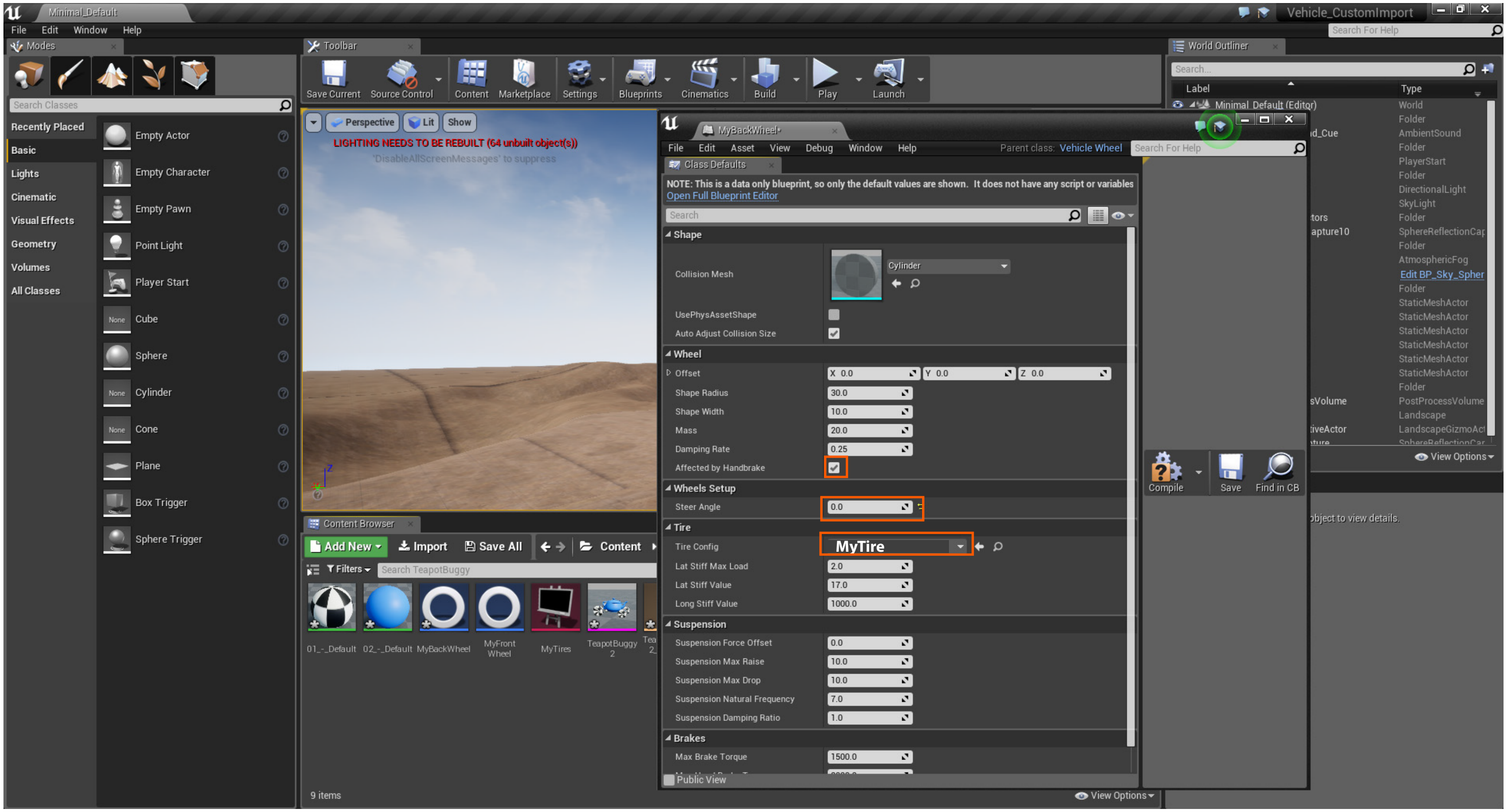
9. Right click in Content and select Blueprint - Blueprint Class - VehicleWheel. Rename this as “MyFrontWheel”.

10. Repeat step 9 and name “MyBackWheel”



11. Click on your MyBackWheel blueprint and turn the Steering Angle to “0”. (You don’t want the back wheels to turn) and place the “MyTire” config that you made in step 8 into the TireConfig slot. SAVE

12. Next, open the MyFrontWheel blueprint and turn off “Affected by Handbrake” (only the back wheels should be affected) and as in step 11, place the “MyTire” into the Tire Config slot. SAVE



LIGHTING NEEDS TO BE REBUILT (64 unbuilt object(s))
'DisableAllScreenMessages' to suppress

NOTE: This is a data only blueprint, so only the default values are shown. It does not have any script or variables
Open Full Blueprint Editor

Shape
Collision Mesh: Cylinder
UsePhysAssetShape □
Auto Adjust Collision Size ☑

Wheel
Offset: X 0.0 Y 0.0 Z 0.0
Shape Radius: 30.0
Shape Width: 10.0
Mass: 20.0
Damping Rate: 0.25
Affected by Handbrake ☑

Wheels Setup
Steer Angle: 0.0

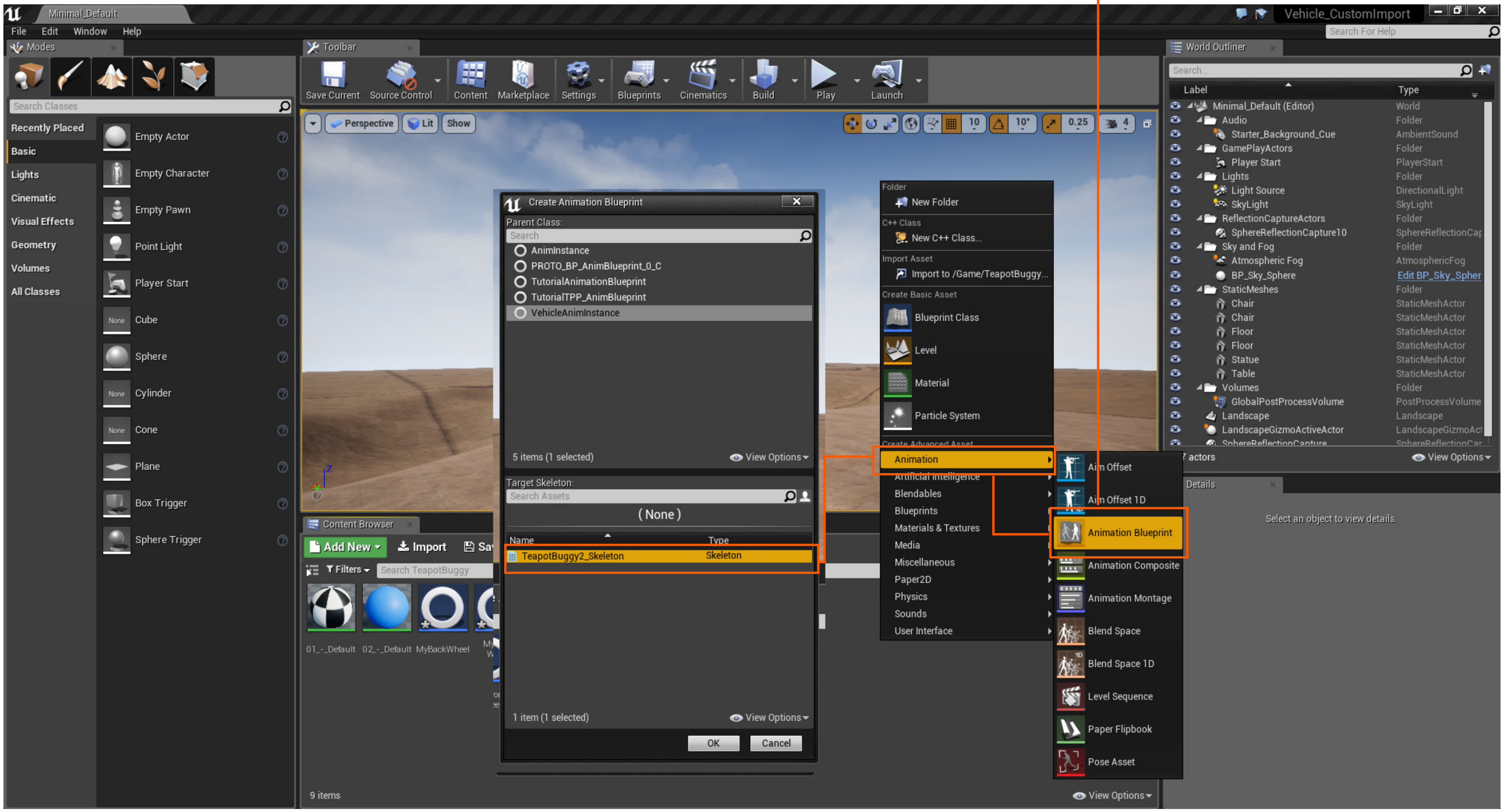
Tire
Tire Config: MyTire
Lat Stiff Max Load: 2.0
Lat Stiff Value: 17.0
Long Stiff Value: 1000.0

Suspension
Suspension Force Offset: 0.0
Suspension Max Raise: 10.0
Suspension Max Drop: 10.0
Suspension Natural Frequency: 7.0
Suspension Damping Ratio: 1.0

Brakes
Max Brake Torque: 1500.0

Next make an Animation Blueprint for controlling the vehicle.

13. Right click in the gray Content area



Create Animation Blueprint

Parent Class:

- ○ AnimInstance
- ○ PROTO_BP_AnimBlueprint_0_C
- ○ TutorialAnimationBlueprint
- ○ TutorialTPP_AnimBlueprint
- ○ VehicleAnimInstance

5 items (1 selected)

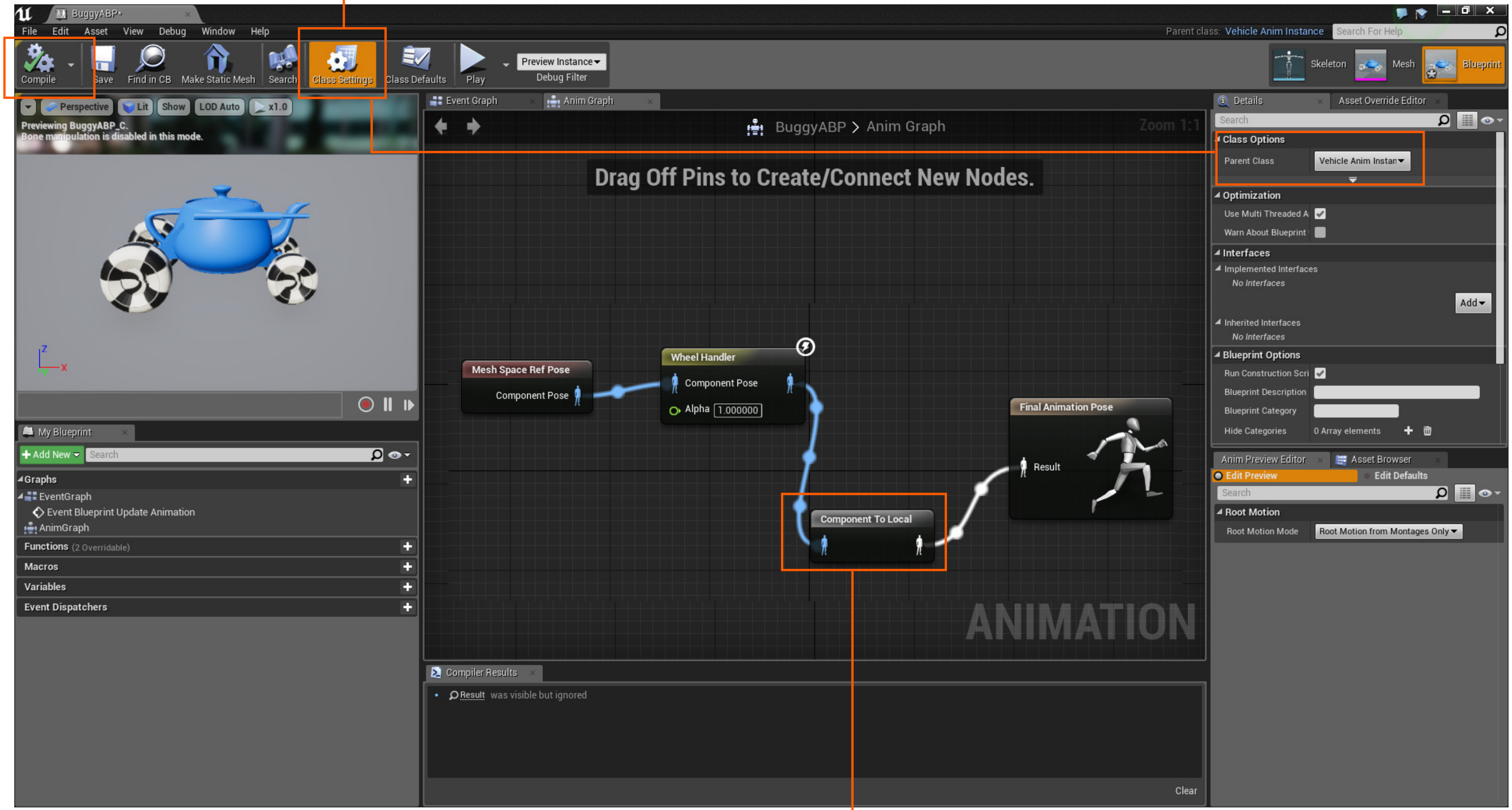
Target Skeleton:

Name	Type
TeapotBuggy2_Skeleton	Skeleton

1 item (1 selected)

Name and open the new Animation Blueprint.

14. Open the Blueprint and select



Result was visible but ignored

15. Create the following system of nodes

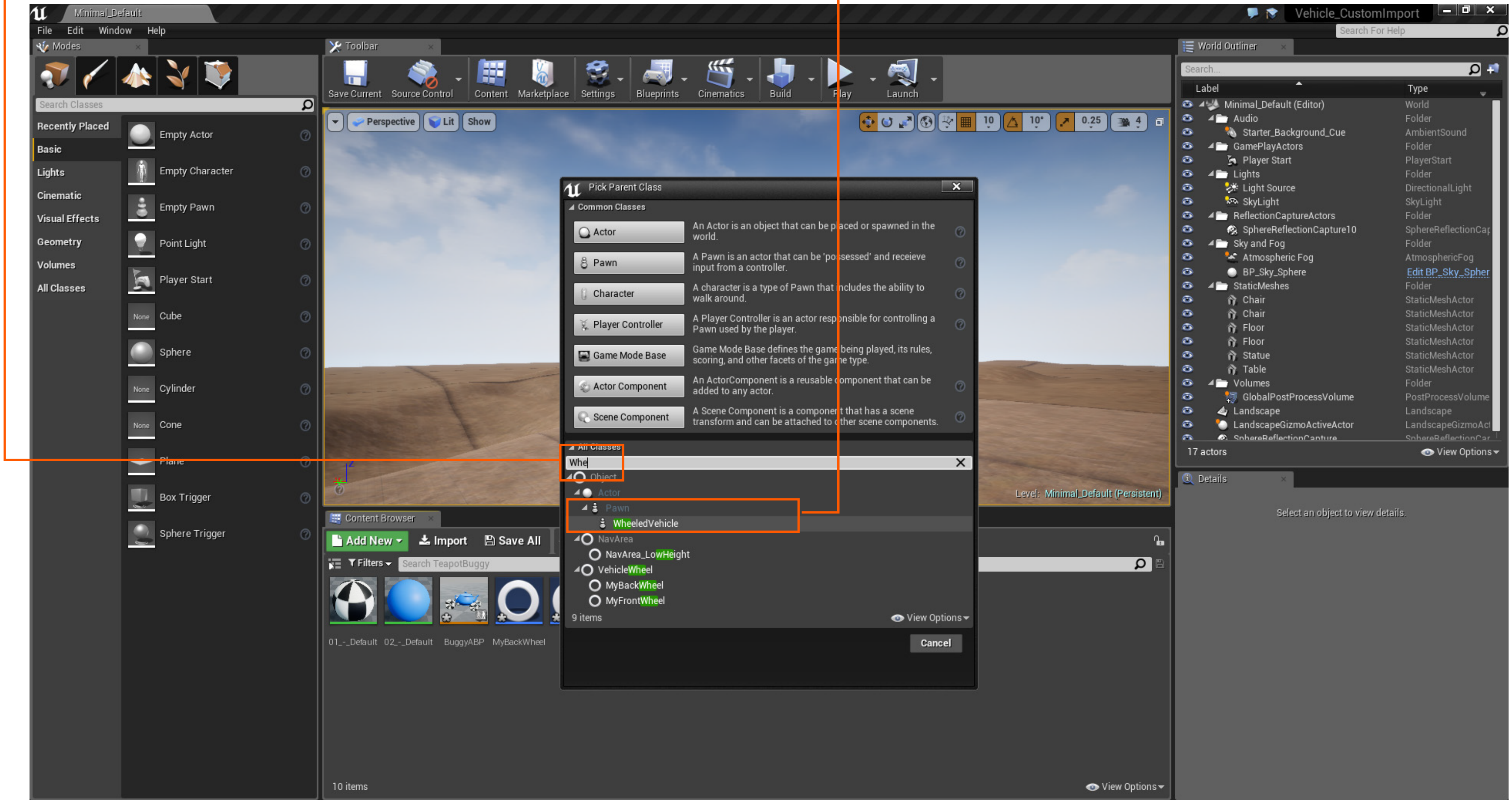
(note- the Component to Local node will appear when you connect the Handler to the Final Animation Pose

CC16. COMPILE & SAVE

Now you'll create another blueprint to bring together all the assets you just made. I refer to it as the MASTER BLUEPRINT.

NOTE: The surest way for selecting the proper 'Parent Class" is to type "Wheel" in this slot.

16. Click and select "WheeledVehicle". Name the blueprint and open.



Pick Parent Class

Common Classes

- Actor: An Actor is an object that can be placed or spawned in the world.
- Pawn: A Pawn is an actor that can be 'possessed' and receieve input from a controller.
- Character: A character is a type of Pawn that includes the ability to walk around.
- 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.
- 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.

All Classes

Whe

```
Object
  Actor
    Pawn
      WheeledVehicle
  NavArea
    NavArea_LowHeight
  VehicleWheel
    MyBackWheel
    MyFrontWheel
9 items
```

Label	Type
Minimal_Default (Editor)	World
Audio	Folder
Starter_Background_Cue	AmbientSound
GamePlayActors	Folder
Player Start	PlayerStart
Lights	Folder
Light Source	DirectionalLight
SkyLight	SkyLight
ReflectionCaptureActors	Folder
SphereReflectionCapture10	SphereReflectionCap
Sky and Fog	Folder
Atmospheric Fog	AtmosphericFog
BP_Sky_Sphere	Edit BP_Sky_Spher
StaticMeshes	Folder
Chair	StaticMeshActor
Chair	StaticMeshActor
Floor	StaticMeshActor
Floor	StaticMeshActor
Statue	StaticMeshActor
Table	StaticMeshActor
Volumes	Folder
GlobalPostProcessVolume	PostProcessVolume
Landscape	Landscape
LandscapeGizmoActiveActor	LandscapeGizmoAct
SphereReflectionCapture	SphereReflectionCar

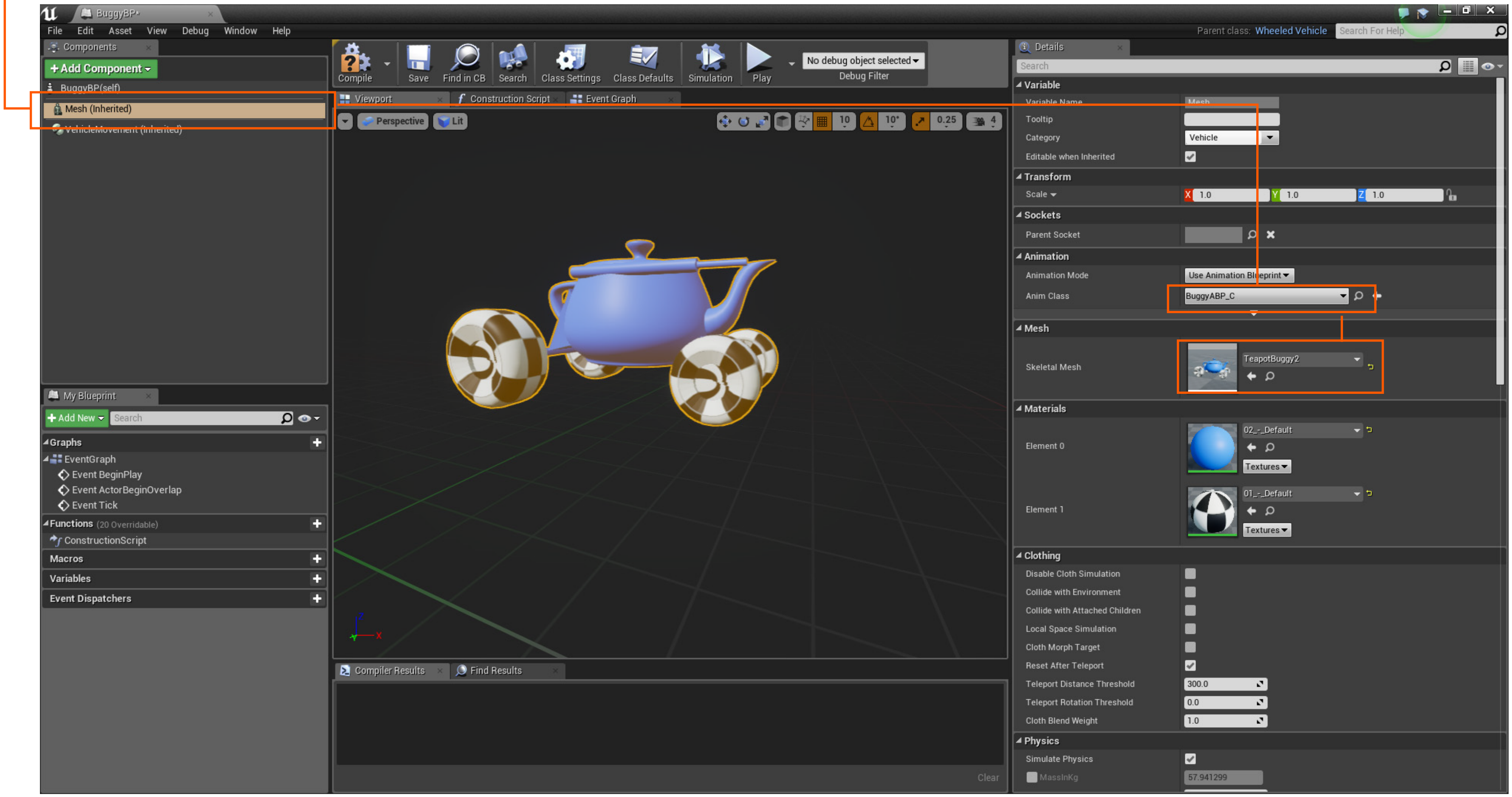
17 actors

Select an object to view details.

Career Technical Education
SoCal ROC

OPEN THE "MASTER BLUEPRINT"

17. Click "Mesh Inherited" and set both the Animation Blueprint you made and the Skeletal Mesh into the Vehicle Blueprint.



Variable
Variable Name: Mesh
Tooltip:
Category: Vehicle
Editable when Inherited: ☑

Transform
Scale: X 1.0 Y 1.0 Z 1.0

Sockets
Parent Socket:

Animation
Animation Mode: Use Animation Blueprint
Anim Class: BuggyABP_C

Mesh
Skeletal Mesh: TeapotBuggy2

Materials
Element 0: 02_-_Default
Element 1: 01_-_Default

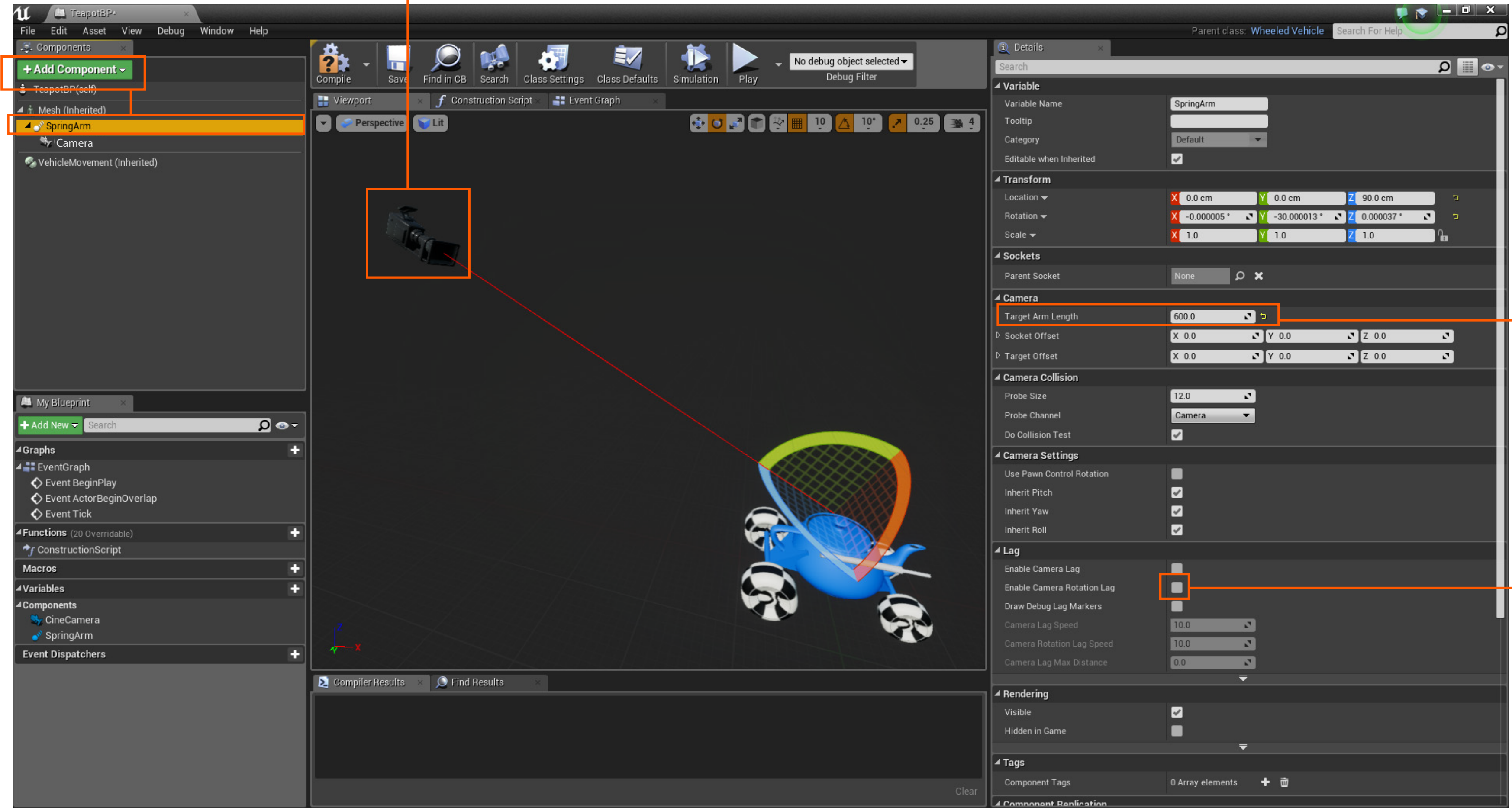
Clothing
Disable Cloth Simulation: □
Collide with Environment: □
Collide with Attached Children: □
Local Space Simulation: □
Cloth Morph Target: □
Reset After Teleport: ☑
Teleport Distance Threshold: 300.0
Teleport Rotation Threshold: 0.0
Cloth Blend Weight: 1.0

Physics
Simulate Physics: ☑
□ MassInKg: 57.941299

Connect a Camera Component” to the vehicle using a “Spring Arm Component”

18. Create a Spring Arm,move to top of vehicle, angle upwards and set to 600

19. Create “Camera”. The camera should automatically stick to the end of the Spring arm.



Variable

- Variable Name: SpringArm
- Tooltip:
- Category: Default
- Editable when Inherited: ☑

Transform

- Location: X 0.0 cm, Y 0.0 cm, Z 90.0 cm
- Rotation: X -0.000005 °, Y -30.000013 °, Z 0.000037 °
- Scale: X 1.0, Y 1.0, Z 1.0

Sockets

- Parent Socket: None

Camera

- Target Arm Length: 600.0
- Socket Offset: X 0.0, Y 0.0, Z 0.0
- Target Offset: X 0.0, Y 0.0, Z 0.0

Camera Collision

- Probe Size: 12.0
- Probe Channel: Camera
- Do Collision Test: ☑

Camera Settings

- Use Pawn Control Rotation: ☐
- Inherit Pitch: ☑
- Inherit Yaw: ☑
- Inherit Roll: ☑

Lag

- Enable Camera Lag: ☐
- Enable Camera Rotation Lag: ☐
- Draw Debug Lag Markers: ☐
- Camera Lag Speed: 10.0
- Camera Rotation Lag Speed: 10.0
- Camera Lag Max Distance: 0.0

Rendering

- Visible: ☑
- Hidden in Game: ☐

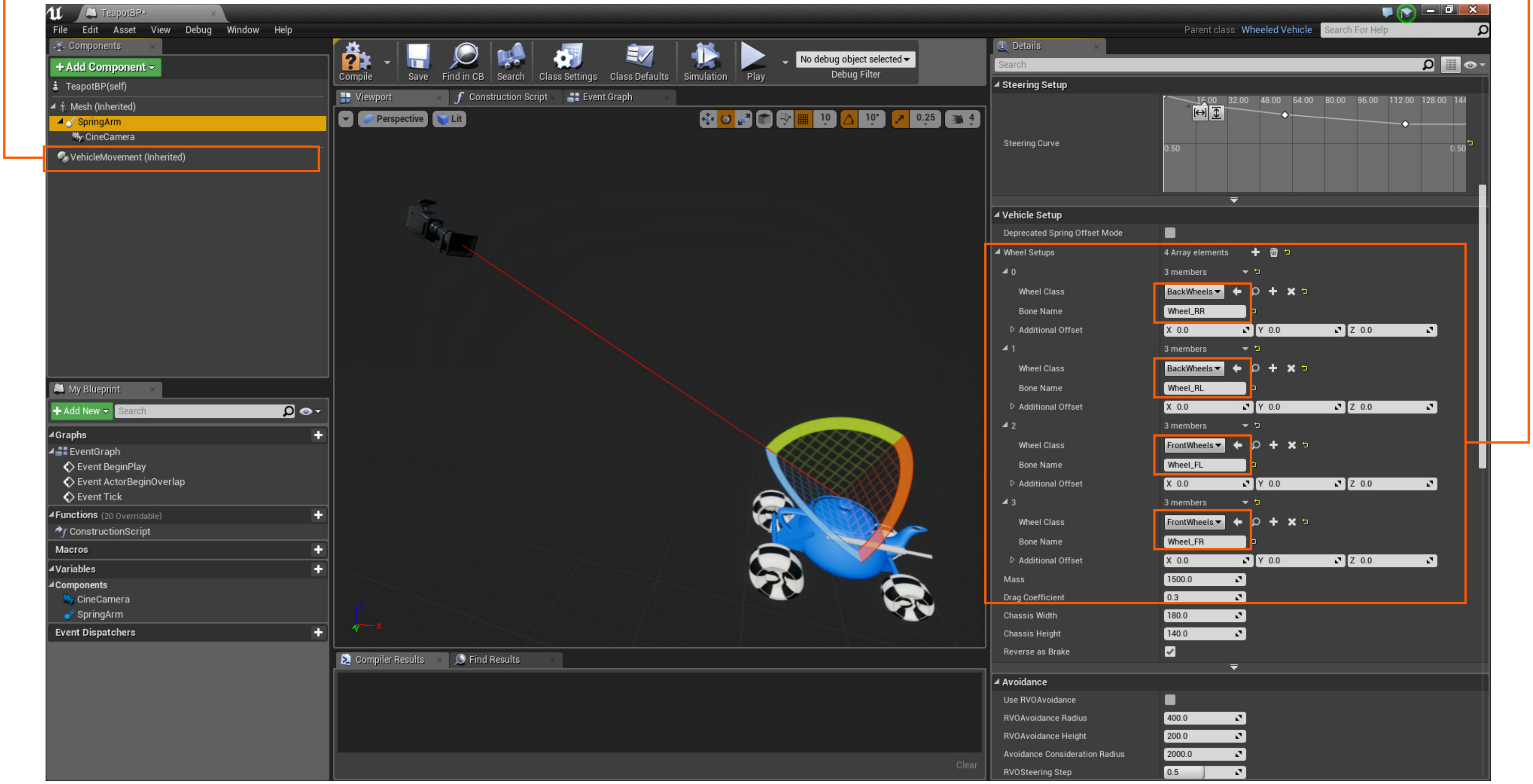
Tags

- Component Tags: 0 Array elements

Component Replication

Now you will need the names of the wheels that you labeled in 3D Max

20. Click on the vehicle body. Select “Vehicle Movement” and fill in the settings. Be sure to check your spelling. Cap’s matter. List them in the order they appear in Vehicle Skeleton.



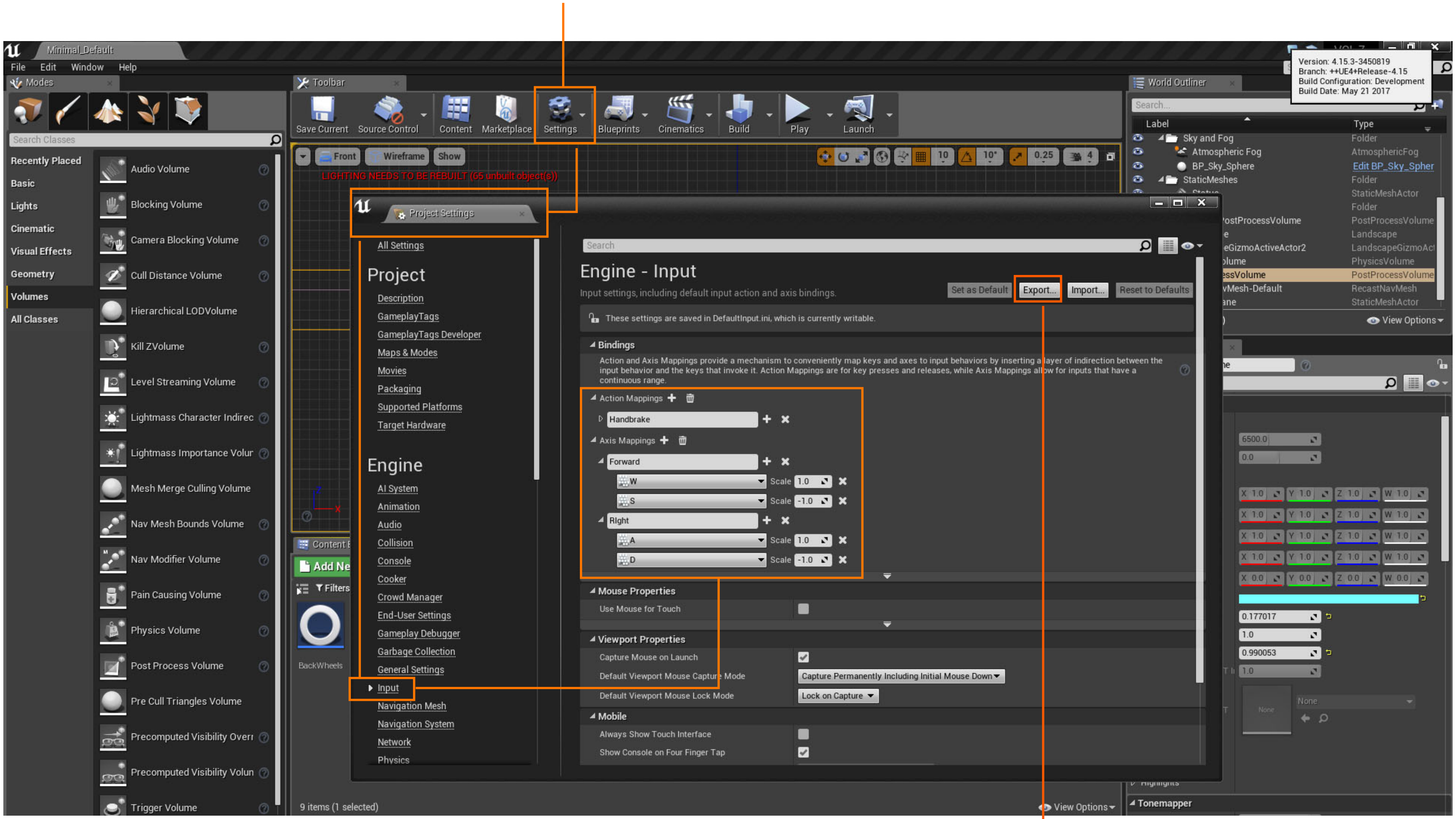
Vehicle Setup

Deprecated Spring Offset Mode	□
Wheel Setups	4 Array elements
0	3 members
Wheel Class	BackWheels
Bone Name	Wheel_RR
Additional Offset	X 0.0 Y 0.0 Z 0.0
1	3 members
Wheel Class	BackWheels
Bone Name	Wheel_RL
Additional Offset	X 0.0 Y 0.0 Z 0.0
2	3 members
Wheel Class	FrontWheels
Bone Name	Wheel_FL
Additional Offset	X 0.0 Y 0.0 Z 0.0
3	3 members
Wheel Class	FrontWheels
Bone Name	Wheel_FR
Additional Offset	X 0.0 Y 0.0 Z 0.0
Mass	1500.0
Drag Coefficient	0.3
Chassis Width	180.0
Chassis Height	140.0
Reverse as Brake	☑
Use RVOAvoidance	□
RVOAvoidance Radius	400.0
RVOAvoidance Height	200.0
Avoidance Consideration Radius	2000.0
RVOSteering Step	0.5

COMPILE & SAVE

The Vehicle Blueprint is almost complete. First set up directions for steering and brake the vehicle

21. Open Project Settings and enter the info for driving the vehicle using keyboard strokes.



Engine - Input

Input settings, including default input action and axis bindings.

Set as Default | Export... | Import... | Reset to Defaults

These settings are saved in DefaultInput.ini, which is currently writable.

Bindings

Action and Axis Mappings provide a mechanism to conveniently map keys and axes to input behaviors by inserting a layer of indirection between the input behavior and the keys that invoke it. Action Mappings are for key presses and releases, while Axis Mappings allow for inputs that have a continuous range.

Action Mappings
- Handbrake

Axis Mappings
- Forward
  - W Scale 1.0
  - S Scale -1.0
- RIght
  - A Scale 1.0
  - D Scale -1.0

Mouse Properties
- Use Mouse for Touch ☐

Viewport Properties
- Capture Mouse on Launch ☑
- Default Viewport Mouse Capture Mode: Capture Permanently Including Initial Mouse Down
- Default Viewport Mouse Lock Mode: Lock on Capture

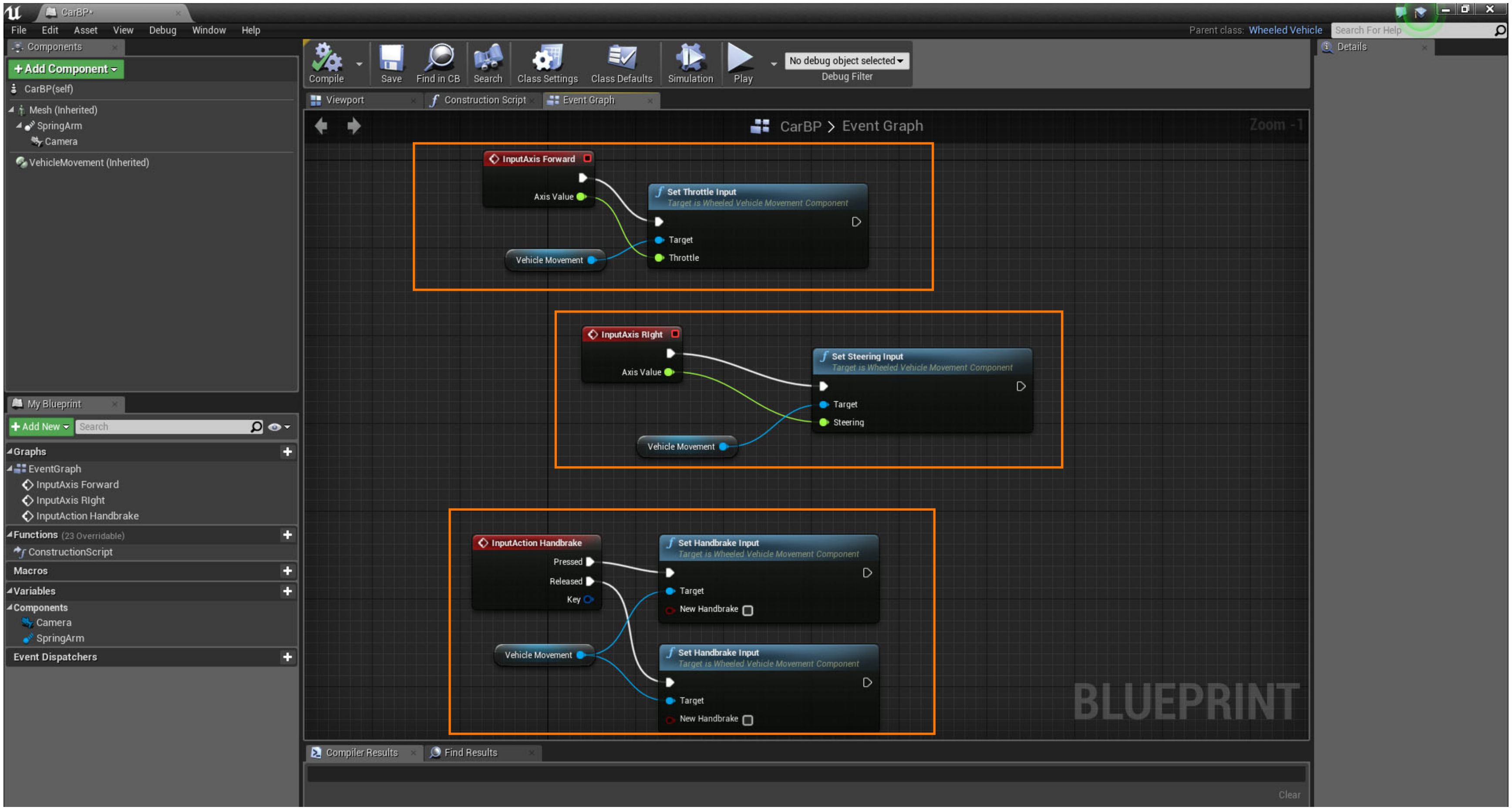
Mobile
- Always Show Touch Interface ☐
- Show Console on Four Finger Tap ☑

Version: 4.15.3-3450819
Branch: ++UE4+Release-4.15
Build Configuration: Development
Build Date: May 21 2017

Export the input to your Content Folder

Next you'll enter the input information into the Vehicle Blueprint Event Graph

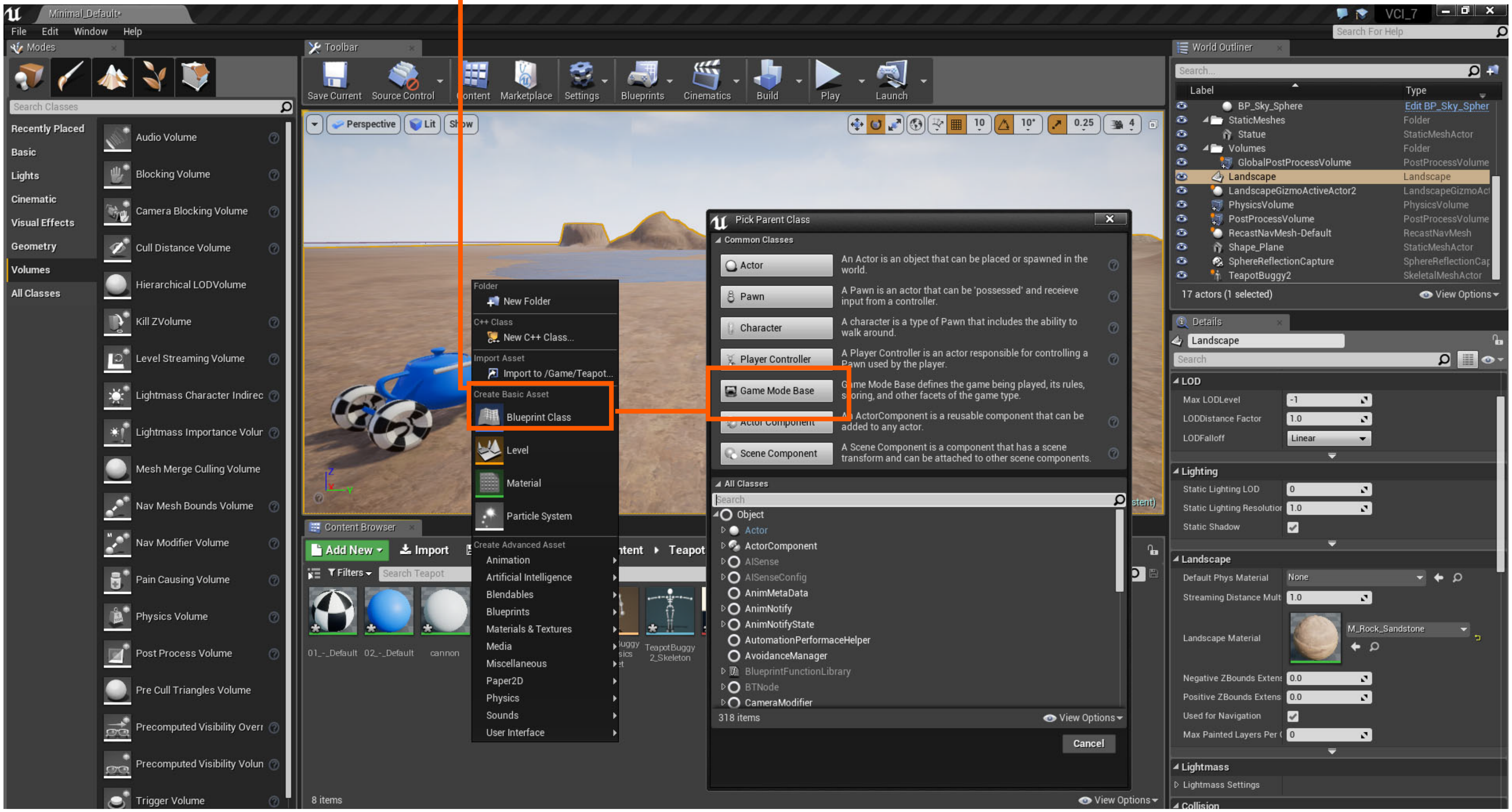
22. Create and link the nodes.



SAVE AND COMPILE

Now you will set up the game mode telling the game to control the vehicle

23. Right click in Content -Blueprint Class - Game Mode Base and create a new Game Mode.



An Actor is an object that can be placed or spawned in the world.

A Pawn is an actor that can be 'possessed' and receieve input from a controller.

A character is a type of Pawn that includes the ability to walk around.

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

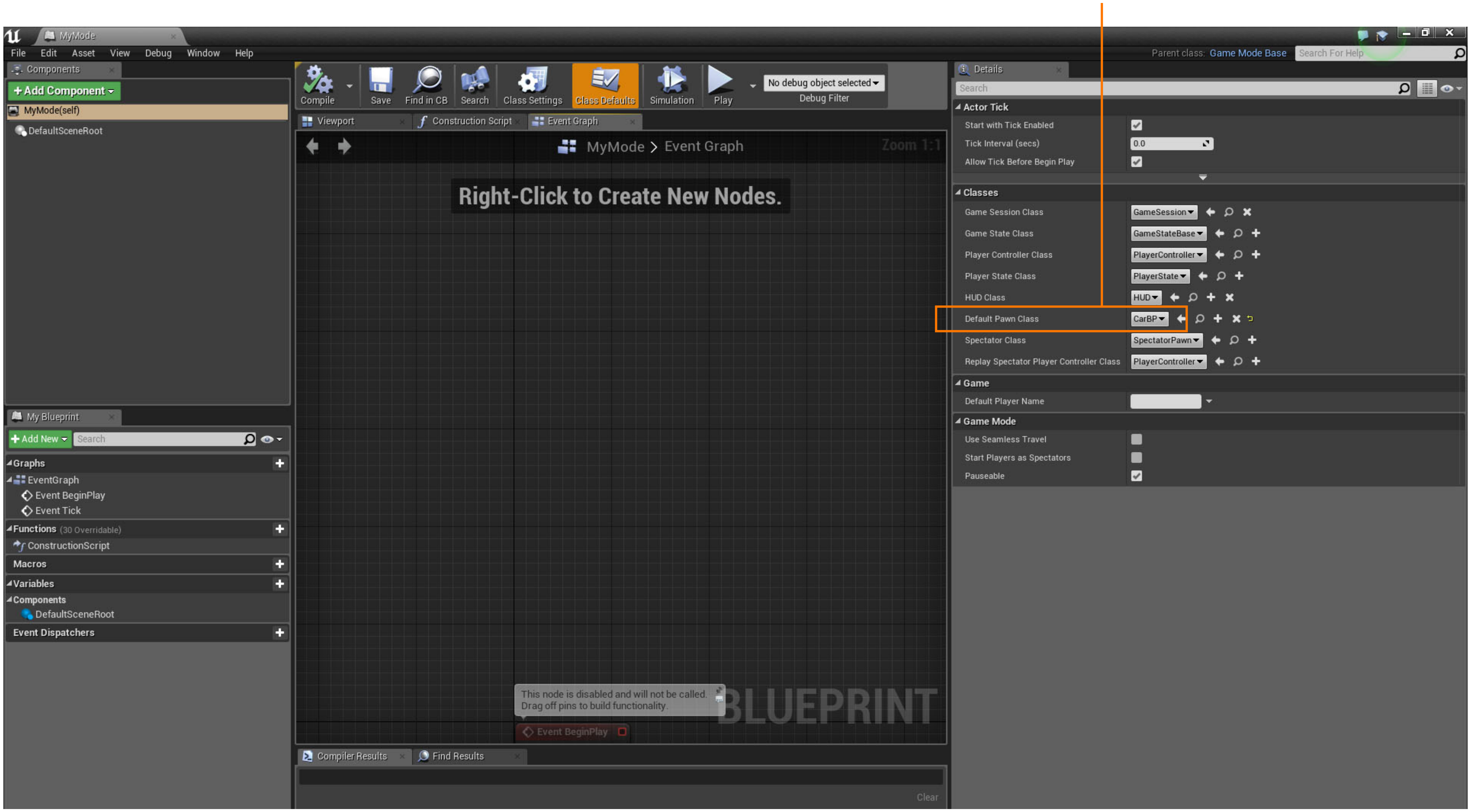
Game Mode Base defines the game being played, its rules, scoring, and other facets of the game type.

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

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

24. Name the new Game Mode and open it.

25. Place your Master Blueprint in the slot. Compile & Save.



Right-Click to Create New Nodes.

This node is disabled and will not be called.
Drag off pins to build functionality.

Actor Tick
Start with Tick Enabled ☑
Tick Interval (secs) 0.0
Allow Tick Before Begin Play ☑

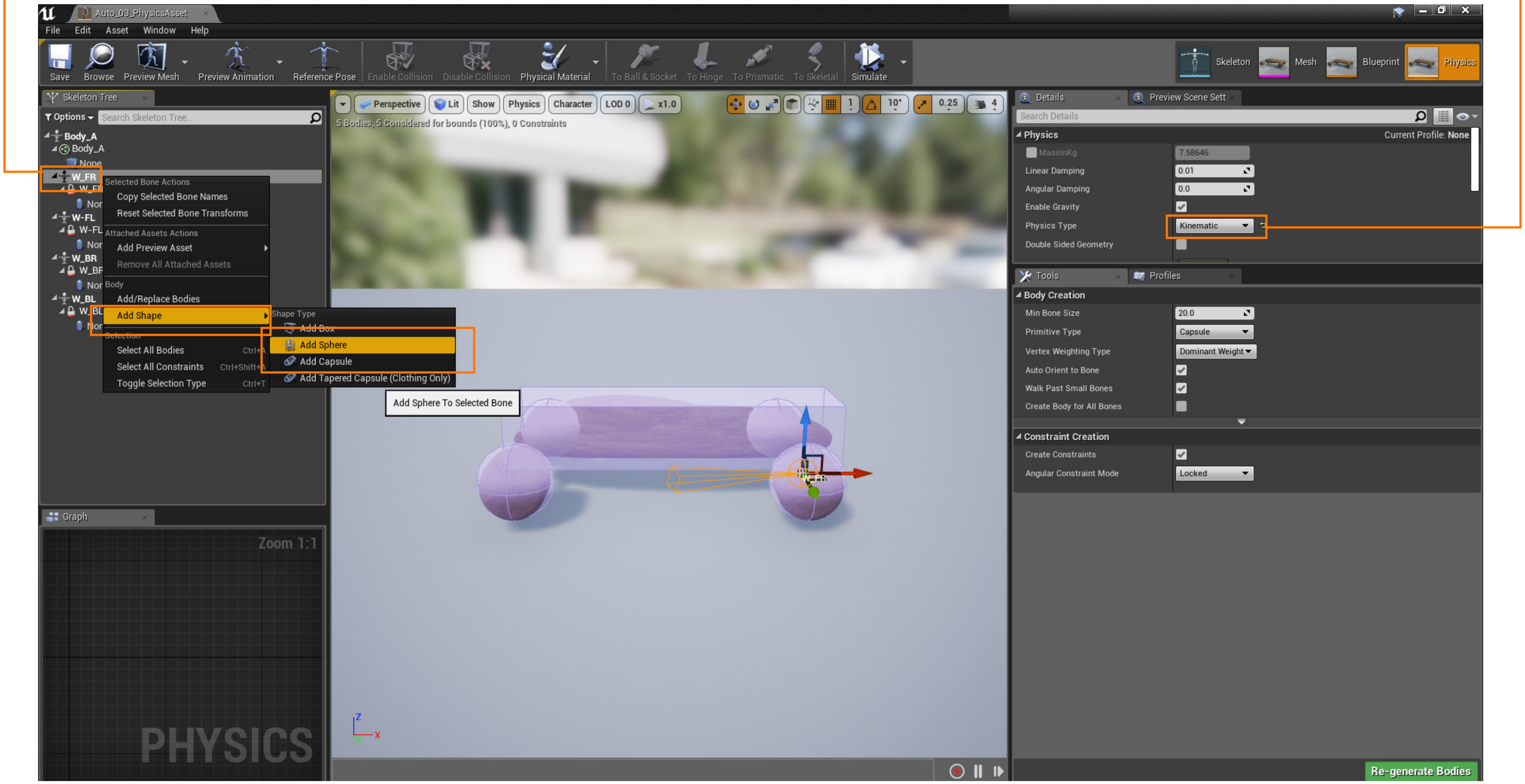
Classes
Game Session Class GameSession
Game State Class GameStateBase
Player Controller Class PlayerController
Player State Class PlayerState
HUD Class HUD
Default Pawn Class CarBP
Spectator Class SpectatorPawn
Replay Spectator Player Controller Class PlayerController

Game
Default Player Name

Game Mode
Use Seamless Travel ☐
Start Players as Spectators ☐
Pauseable ☑

25. Select a wheel (showing it collision shape) then select Add Shape and pick Sphere. Then delete the Capsule. Do this for all four wheels.

26. Set all wheels to "kinematic".



Physics — Current Profile: None

- MassInKg: 7.58646
- Linear Damping: 0.01
- Angular Damping: 0.0
- Enable Gravity: ☑
- Physics Type: Kinematic
- Double Sided Geometry: □

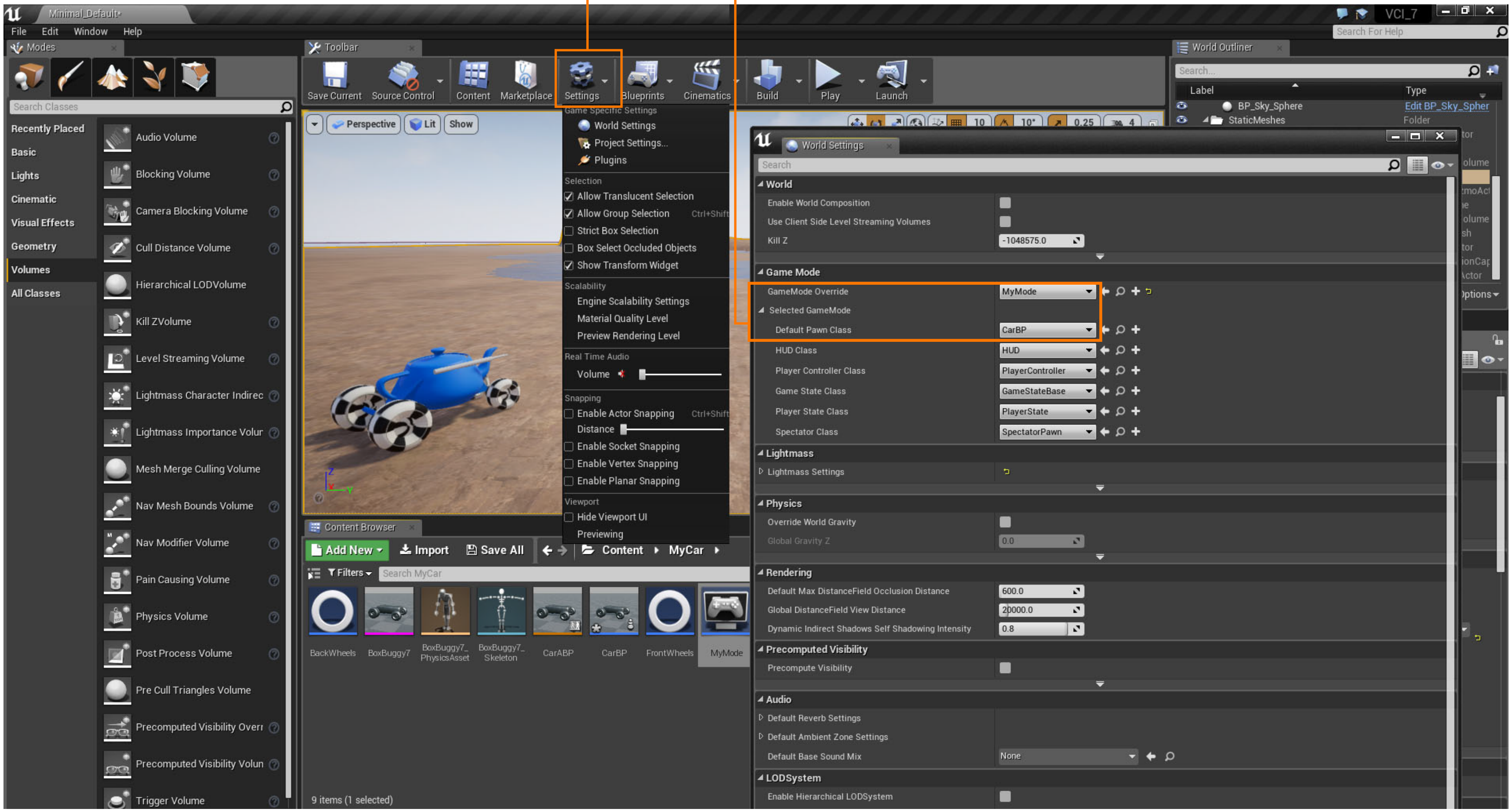
Body Creation

- Min Bone Size: 20.0
- Primitive Type: Capsule
- Vertex Weighting Type: Dominant Weight
- Auto Orient to Bone: ☑
- Walk Past Small Bones: ☑
- Create Body for All Bones: □

Constraint Creation

- Create Constraints: ☑
- Angular Constraint Mode: Locked

27. Last,open World Settings and set Game Mode to your Vehicle Blueprint.



World Settings

World
- Enable World Composition ☐
- Use Client Side Level Streaming Volumes ☐
- Kill Z: -1048575.0

Game Mode
- GameMode Override: MyMode
- Selected GameMode
  - Default Pawn Class: CarBP
  - HUD Class: HUD
  - Player Controller Class: PlayerController
  - Game State Class: GameStateBase
  - Player State Class: PlayerState
  - Spectator Class: SpectatorPawn

Lightmass
- Lightmass Settings

Physics
- Override World Gravity ☐
- Global Gravity Z: 0.0

Rendering
- Default Max DistanceField Occlusion Distance: 600.0
- Global DistanceField View Distance: 20000.0
- Dynamic Indirect Shadows Self Shadowing Intensity: 0.8

Precomputed Visibility
- Precompute Visibility ☐

Audio
- Default Reverb Settings
- Default Ambient Zone Settings
- Default Base Sound Mix: None

LODSystem
- Enable Hierarchical LODSystem ☐

If it doesn’t work, check this list of possible problems:

1. Vehicle won’t move forward but wheel spins. Collision box on your vehicle extends below the base of wheels and into the ground causing the wheels to spin but the car not to move .

or

There is something in front of the vehicle blocking forward movement.

2. Vehicle drops under landscape. The Spring Arm connecting the camera to the vehicle is below the level of the vehicle causing the vehicle to drop under the landscape.

3. Vehicle does not respond - Vehicle is not facing in the “X” (red arrow in the perspective viewport) direction when built in 3D Max.

4. Wheels won’t turn - Naming of the wheels and body set in 3D Max are misspelled in the Unreal BP wheel set up.

5. Vehicle turns left when it should turn right - Recheck your Project Settings-Input Axis.

6. Vehicle does not touch ground or is embedded in ground - Check the Physics Asset. Make sure wheels are set to “Kinematic”. Check in Wheel Blueprints if correct sizes for tire are entered. Try making the radius number smaller.

7. Vehicle skids around - Lower the “Steering Angle” in the Front Wheel Blue Print to 20

8. Vehicle flys about - Set your landscape to ”0” in the Z (up) direction.

9. Vehicle digs in. Go back to 3D Max and change the body pivot point.

Weight distribution of a vehicle is important to its control, as it affects a variety of its characteristics, like handling, acceleration, and traction. Different vehicle types will require different weight distributions depending on their intended use. For the purposes of game development, these characteristics can also define what style of game you are making, whether that is an arcade-style racer, a simulation, or even a hybrid of the two. Altering Center of Mass enables you to change the weight distribution of your vehicle.

In games, the primary use of Center of Mass is for vehicles, but it can also be used for large Physics Bodies encapsulating an irregular shape. In your Physics Asset , you will often use one large Physics Body to define the majority of the Mass for the vehicle (or large object). The Center of Mass will be generated at the center of this Physics Body, which can make the vehicle handle oddly, so you can adjust the Center of Mass to account for where the mass of the vehicle is really located.