



---

**IMPORTANT V1.5 Halloween Update:**



**BACKUP YOUR PROJECT BEFORE UPDATING!**

"If you have tuned your car or camera to perfection some of our fixes and improvements may affect these settings."

**FUNGAR KIT**  
**HALLOWEEN UPDATE**

---

## WATCH THE WALKTHROUGH

[https://youtu.be/kHFpxK4F\\_1M](https://youtu.be/kHFpxK4F_1M)



This tool will allow you to get a heads start on your next car game.

Art assets are also supplied for you to use and try out. They can be used in your games (4 cars including Bodys and wheels which can be mixed as you like, tracks are modular to allow for many track designs).

NOTE: No part of this pack can be shared or resold.

- Car parts (body and wheels) we supply 4 car types and 4 wheel sizes)
- Something to drive on.

-> Script: Vehiclecontroller: V1.5 Halloween Update



Drag this script to the node containing your car body and wheel.  
This is where you define your car parts (drag in the body and wheels to the correct 5 slots).  
You can also tune your car here (tooltips available)

---

#### **UPDATE V1.5:**

Car Lights: using a separate mesh with a material for on and off states of the lights.  
All ON material states should use Emission on the standard material with a value of 1.

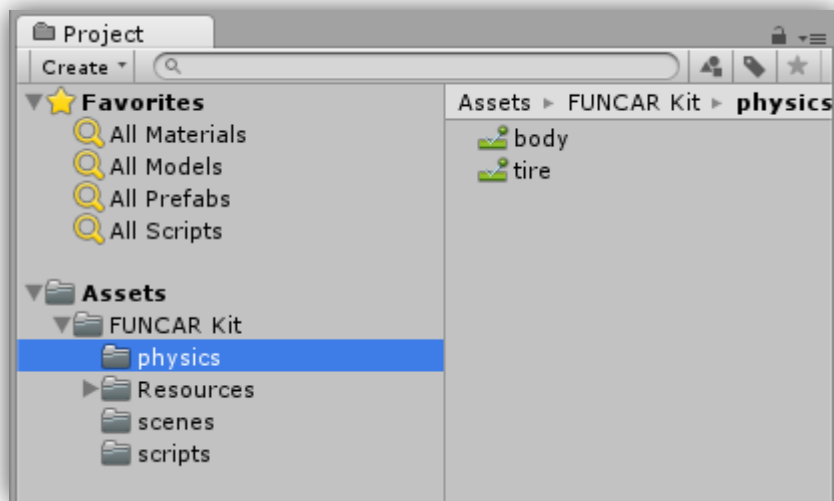
- Headlights: Press “L” to turn off or on the headlight material.
- Indicators: will flash as the car turns left or right
- Brake lights: will turn on as the car is braking

Trail Width: define the width of any tracks left behind tires.

---

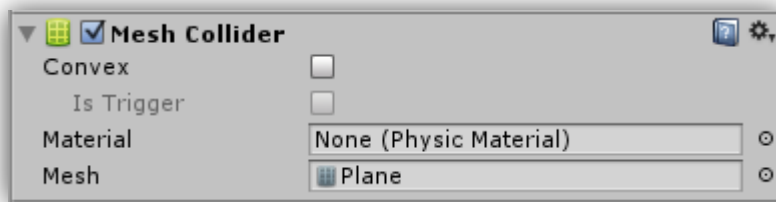


-->Physics Materials: tire & body



Add these to the correct slots in the Vehiclecontroller script.  
You can create your own or edit these by selecting the physics material.

->Script: Mergemesh:



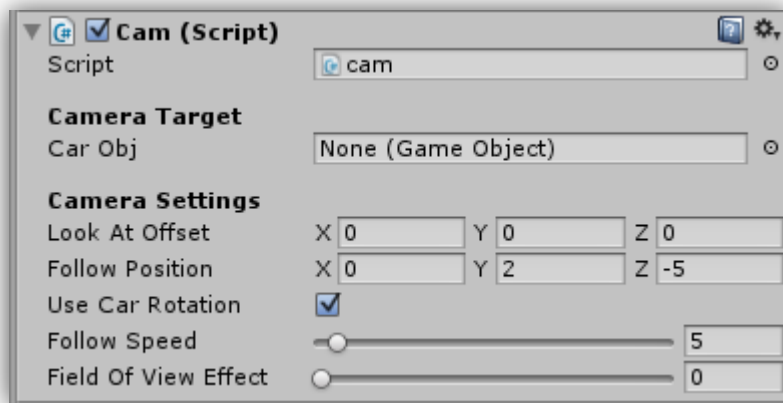
Drag this over the parent gameobject of your modular track. (This track parent node must be set to 0,0,0 in the world)

This will make a new mesh merging all the pieces under the parent gameobject at runtime and weld the connecting verticies.

You will also have a mesh collider and filter added for you on runtime.

NOTE: There is no need to fill in any of the available slots in this script.

->Script: Cam:



You can use your own script or use ours to manage the camera.  
Drag this script onto the camera.  
Tell the camera what to follow, drag your car (game object) onto the Car OBJ slot.  
Tooltips will help you tune the camera.

->All other scripts are procedural and part of these scripts above...

->So you're done. put your car above your track, press play and start driving!

---

---

## UPDATE V1.5: Additional Scripts

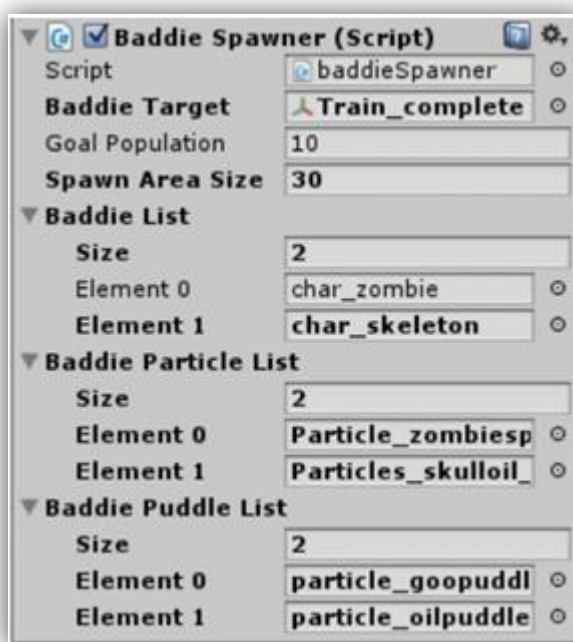


HINT: Mouse over for tooltips on the Exploder for a description of each setting.

HINT 2: The Exploder script can be used to explode any group of mesh game objects you drive into!

---

### ->Script: Baddie Spawner



**Baddie Target:** the car that zombies will walk towards

**Goal Population:** The total number of baddies it will spawn (and if they die they will respawn)

**Spawn Area Size:** the area (both width and length), making a square, that baddies will spawn within, the baddie spawner is in the middle of this area

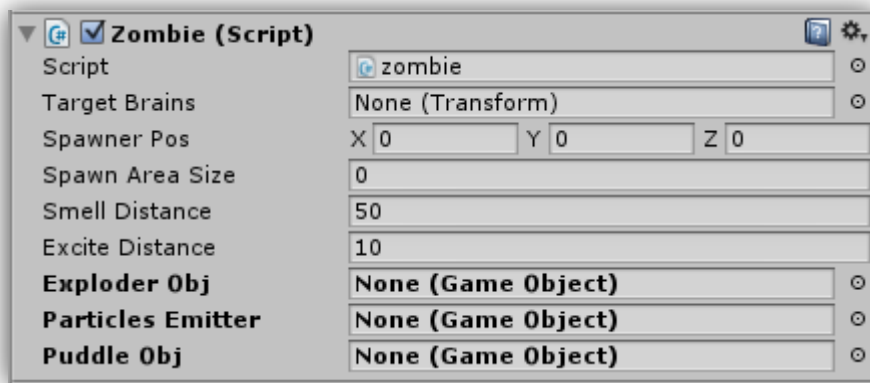
**Baddie List:** the number of baddie types, and each baddie prefab to place

**Baddie Particle List:** the explosion for each baddie type (in the same order to match the baddie list)

**Baddie Puddle List:** the puddle prefab for each baddie type (in the same order to match the baddie list)



->Script: Zombie



Variables not mentioned are carried over from the baddie spawner when the baddie spawner places the zombie/skeleton, they are only set to public for transparency.

(the skeleton is a re-skinned zombie)

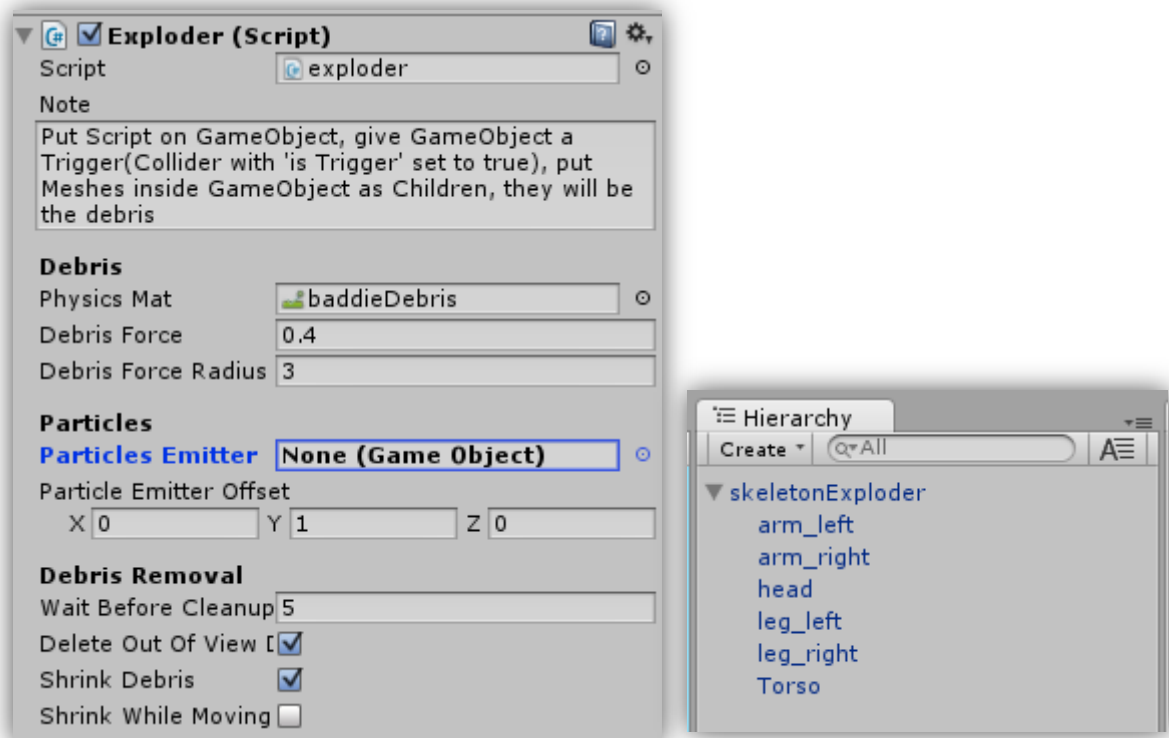
**Target Brains** == Baddie Target (set by baddie spawner)

**Smell Distance:** distance zombie can detect the car from, and start walking towards it

**Excite Distance:** distance where zombie is close enough to the car that it becomes excited and randomly walks faster than normal

**Exploder Obj:** the exploder prefab for this zombie (holds each body part that will go flying)

->Script: Exploder:



The Zombie also uses this script, but this script can be used to make your own objects explode.

Put the exploder script on a new gameobject, and give the gameobject a collider that is set to 'is Trigger'. Then put some children gameobjects inside it. When a collider moves into the trigger, it will 'explode' by sending all the child objects (debris) flying.

**Physics Mat:** the physics material that will be applied to the debris, so you can make them bouncy or not

**Debris Force:** how strongly the debris goes flying

**Debris Force Radius:** how big the area is that the Debris Force is applied from onto the debris

**Particle Emitter:** particle effect to place while the debris go flying

**Particle Emitter Offset:** offset the position of the particle emitter so it appears where you want in the explosion, depending on your pivot point it may not be centered by default

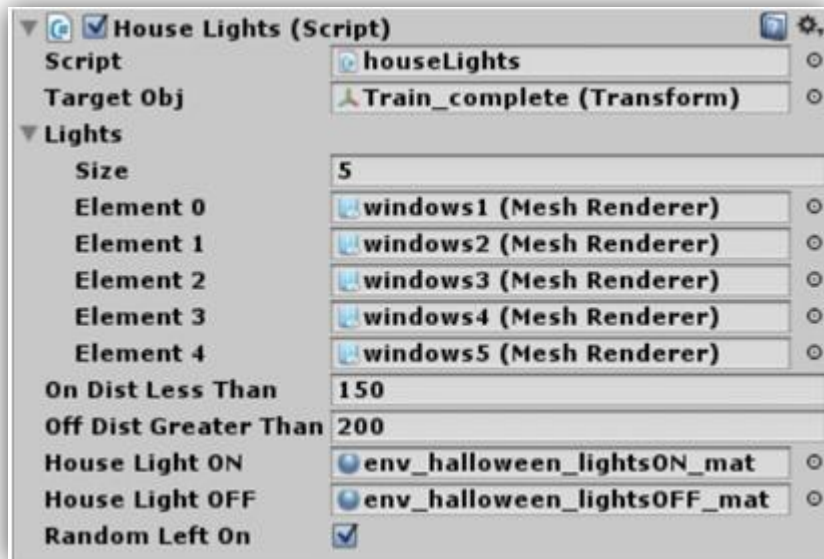
**Wait before cleanup:** how long to wait before debris can shrink to disappear (or disappear off screen)

**Delete Out of View:** debris will delete (after cleanup starts) when no camera can see them(the view in the editor counts as a camera seeing them)

**Shrink Debris:** shrink the debris and then delete them, so they can delete even while in view of a camera

**Shrink While Moving:** don't wait for debris to settle and stop moving before beginning to shrink them.

->Script: House Lights:



The House Lights Script is used on the street lamps and the Haunted Houses windows

**Target Obj:** your car object

**Lights:** the number of lights/windows it is controlling, and each light/window mesh

**On Dist Less Than:** when your car is closer than this distance, the light (or group of lights/windows) switches to the ON material

**Off Dist Greater Than:** when your car is farther away than this distance, the lights (or group of lights/windows) switches to the OFF material

**House Light ON:** the ON material to apply to the mesh renderer of the lights/windows

**House Light OFF:** the OFF material to apply to the mesh renderer of the lights/windows

**Random Left On:** if there are multiple lights/windows, when they switch to off, randomly one of them will be left on (so house can appear as if someone is still home, does nothing if there is only one light/window like a single lamp)

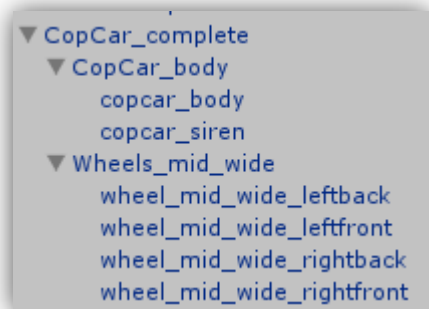
---

1. Make sure all empty gameobjects you use are reset before you add elements to them.
2. The gameobject with the mergemesh script added must be at 0,0,0 in the world or the collision will not work.
3. Use game objects for anything that has multiple parts. If you have a special camera linked to a certain car you can put the car and camera in a game object together.

-Placing your Body mesh(es) inside an empty game object (transforms reset = translate & rotate 0,0,0, scale 1,1,1)

-DO NOT add more than one mesh for a wheel" we recommend not putting them each in a gameobject unless you have to.

-Put the car and wheels inside a top node/parent empty gameobject. Like so:



-Drag the Vehiclecontroller script onto the top node/parent gameobject of the car.

-Drag the gameobject parts into the available slots in the inspector.

-Hover over the setting names to see tooltips.

-Have a mesh to drive on?

-If it's a single mesh it needs a mesh collider component added.

-If your environment is made out of modular pieces and you have snapped them all together (hold v, hover over an edge point and drag to the point you wish to snap to) then use our mergemesh script.

-mergemesh will create a collider on the parent gameobject, based on the meshes of its children. It will round vertices that are near each other to the same spot, eliminating small cracks and seams between meshes(if they are already close together)