

FUNGAR Kit

MOBILE UPDATE



IMPORTANT V2 Mobile/Snow Level Update:



NEW V2.0 FEATURES:

- Mobile controls
- Playable Snow level
- Camera tilt option
- Penguin ragdoll
- Music track

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.”

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.”

WATCH THE WALKTHROUGH

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 (6 cars including different body types and wheels which can be mixed as you like, tracks are modular to allow for many track designs.)

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

W h a t

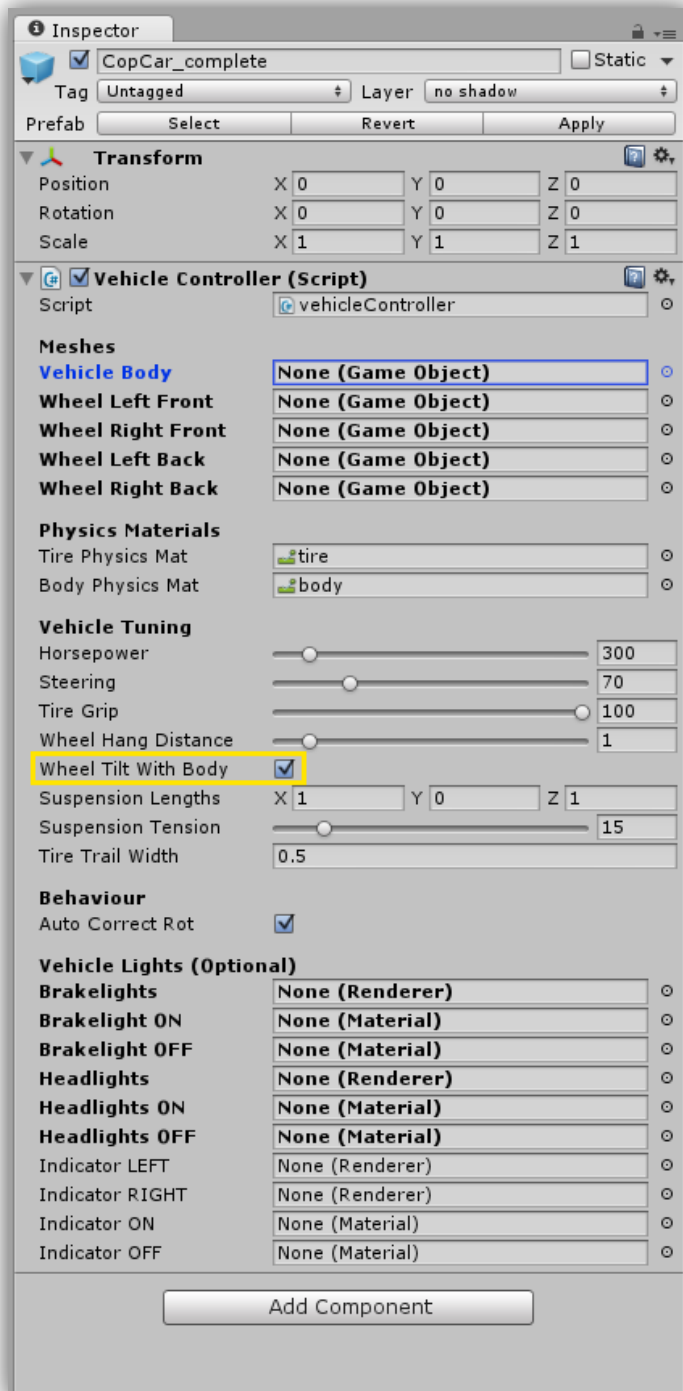
- Car parts (body and wheels) we supply 6 car types and 6 wheel sizes)
- Something to drive on.

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."

What

-> Script: Vehiclecontroller: V2.0 Mobile/Snow Level update



Drag this script to the node containing your car body and wheel.

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.”

This is where you define your car parts (drag in the body and wheels to the corresponding five meshes slots).

You can also tune your car here (tooltips available)

Wheel Tilt with Body: if set to true, when the body swings around, the wheels can swing with that rotation, turn it off/on, and steering left/right hard to see the difference.

Note: If you have tight suspension it should be true, if you have loose suspension and the body swings around a lot, then it would be better to have this set to false

UPDATE V2.0:



Mobile implementation:

In build settings, switch platform to a mobile device (Android or iOS)

If the MobileInput prefab is in the scene, mobile input should automatically enable, and mobile control buttons should appear. Mobile input can be turned off/on by going to 'Mobile Input' > Enable/Disable

In the Inspector Window, the camera will now have an option to set 'Finger Pinch Speed' which lets the player pinch their screen to zoom in and out

If you are building for iOS you will need to download and install Xcode

If you are building for Android, you need to install the Android SDK:

<https://developer.android.com/sdk/index.html#Other>

and install the Java Development Kit:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

After you have installed those, go to Edit (or File in newer versions) > preferences > external tools, and make sure you tell Unity where to find both the SDK and JDK directories, they will look something like this:

C:/Users/YourName/AppData/Local/Android/sdk

C:/Program Files/Java/jdk1.7.0_79

Your Android device will need Developer options unlocked. On the Android device open up your main system Settings. Scroll down, all the way to the bottom to find About Phone (or About Tablet.) Scroll down again and find the entry with Build number.

Start tapping on the Build number entry, Android will now pop up a message informing you that in x amount of clicks, you will become a Developer. Keep tapping away until the process is complete.

Your Android device will need a usb driver to communicate with Unity (I had to go on the device to apps > settings > about device to find what the model number is):

<http://developer.android.com/tools/extras/oem-usb.html>

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."

After installing the driver, choose to allow device debugging and trust the computer (an alert popped up on my android) (settings > developer options > USB debugging)

On your device, go to the App or Google Play store and install the latest Unity Remote

With the device plugged in, and Unity Remote running, go to Edit > Project Settings > Editor, and under Unity Remote, beside Device, choose your device. Now when you press play, you should be able to test the game in the editor, using your device as the input

To make a build, in the build settings, make sure you add the scenes that will be used. In the Player settings, make sure you enter a Company Name, and Product name at the top, as well as in the Bundle Identifier, and then choose 'build and run'

Snow Level:

Penguins: Run away and hit rag doll.

The penguin has a 'Target Avoid' and a 'See Distance'. Drag your car to the 'Target Avoid', and the penguin will run away from your car when it is close enough to see. If the penguin runs away and 1.5x the 'see distance' it will calm down and either stand around dancing, or wander back to near its original placement

Snowman: Watch and explode

The snowman has a 'See Distance' and a 'Target Obj' drag your car to the 'Target Obj' and choose a number for how far he should see. When the 'Target Obj' is close enough for the snowman to see, the snowman will pop out of the ground and rotate to face the target

Camera Tilt: follows terrain

The Camera has a new setting called 'Tilt With Slopes' when true the camera will tilt with the car when it drives up a wall, and works better when driving upside down, but will not stay as flat when driving over bumps

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.

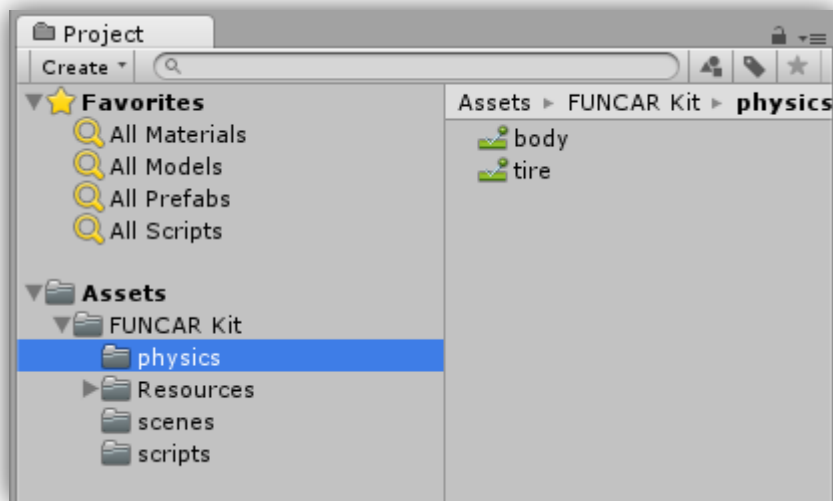
BACKUP YOUR PROJECT BEFORE UPDATING!

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

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.”

-->Physics Materials: tire & body

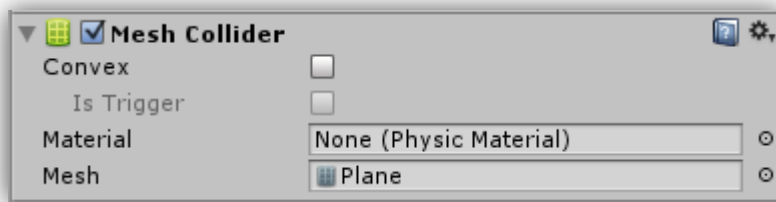


Add these to the correct slots in the Vehiclecontroller script.
You can create your own or edit these by selecting the physics material.

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.”

->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.

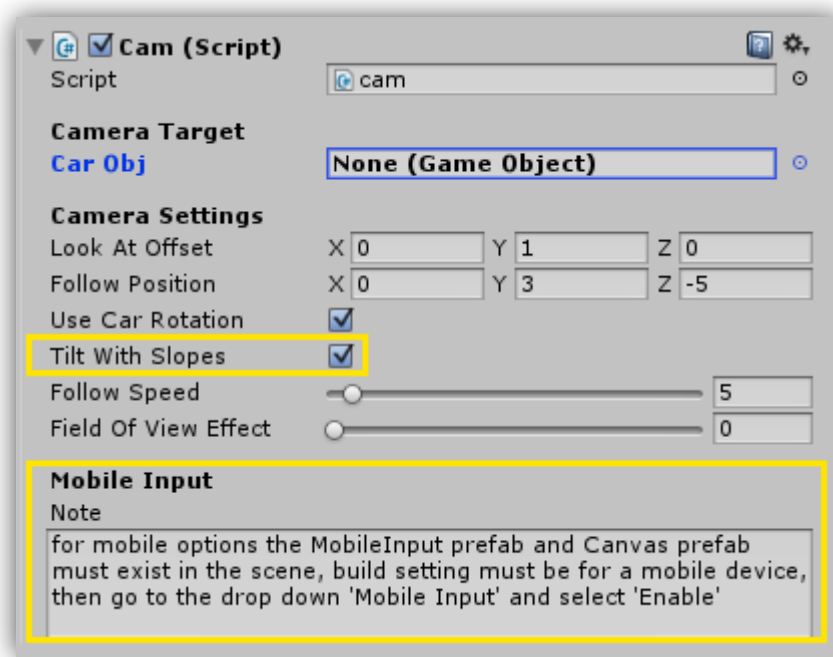
You will also have a mesh collider and filter added for you on runtime.

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

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.”

->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.

V2.0 Update:

- ✓ <Tilt With Slopes> angles the camera to the terrain under the car
- ✓ <Mobile Input> Mobile option notes

➔ 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!

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."

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

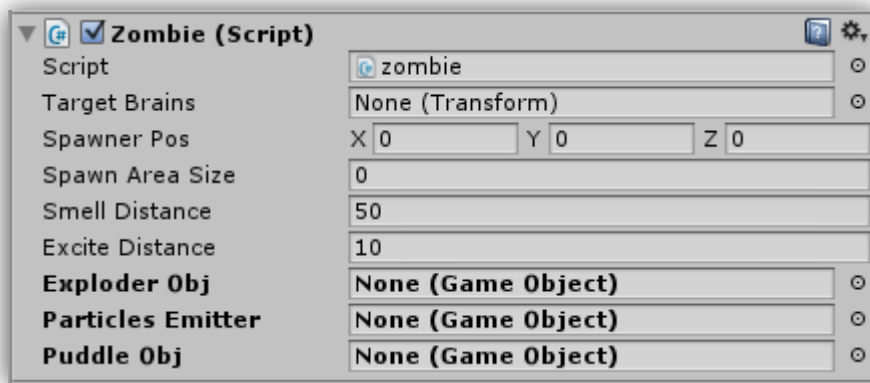
Baddie Particle List: the explosion for each baddie type (in the same order to match the baddie list)

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

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.”

->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

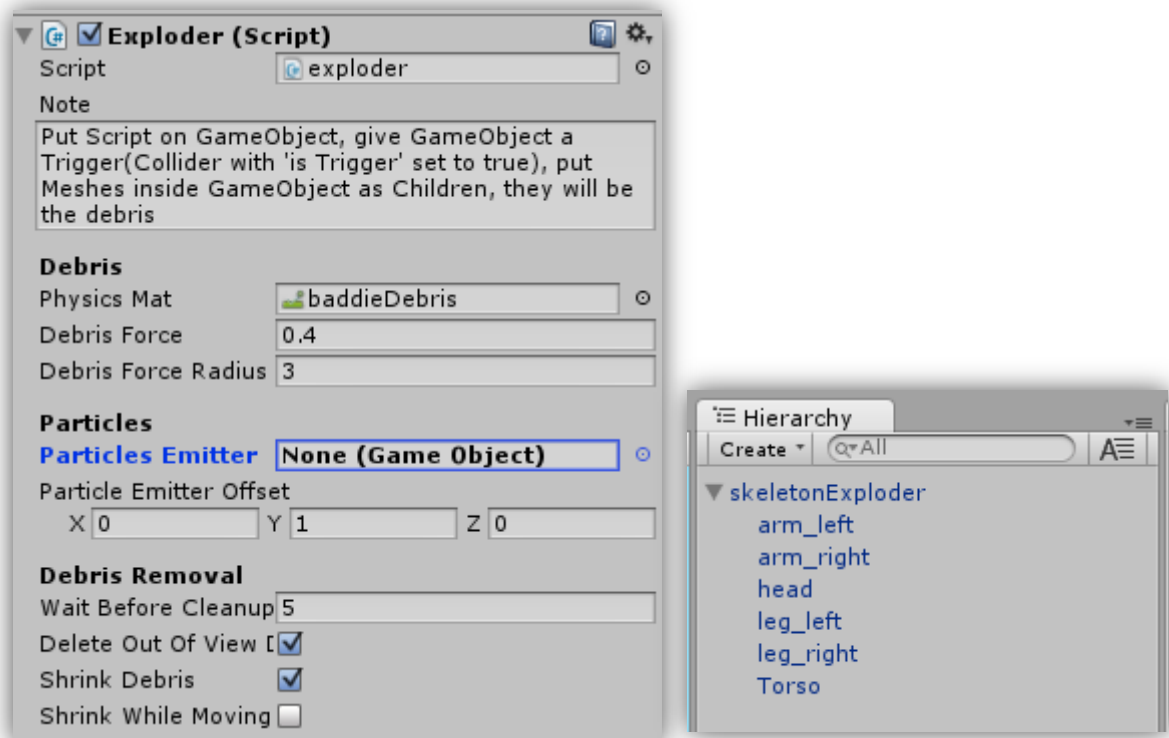
Excite Distance: distance where zombie is close enough to the car that it becomes excited and randomly walks faster than normal

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

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.”

->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)

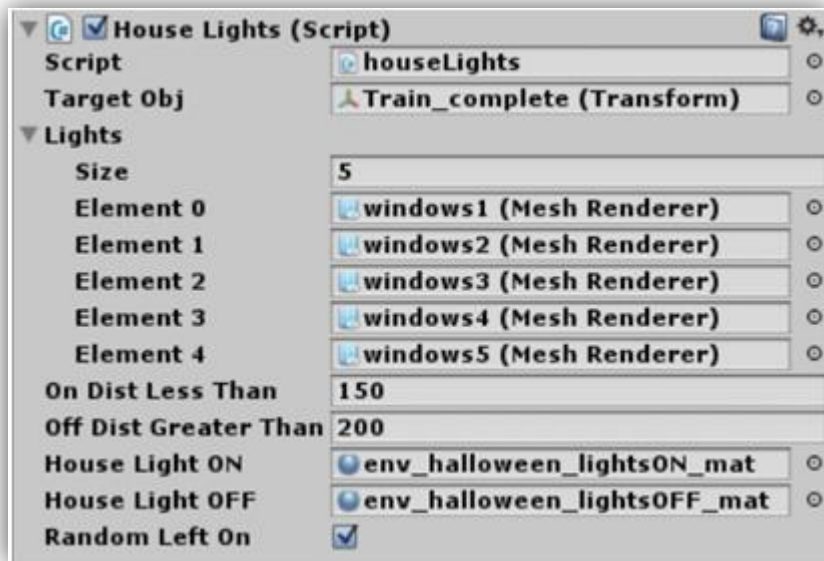
BACKUP YOUR PROJECT BEFORE UPDATING!

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

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)

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.”

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.”

What experiences?

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.

BACKUP YOUR PROJECT BEFORE UPDATING!

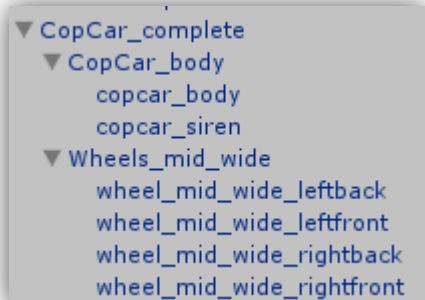
“If you have tuned your car or camera to perfection some of our fixes and improvements may affect these settings.”



-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 adding.

-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 it's 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)

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."