Endless Car Chase Game Template

Game documentation and HowTo guide.

This document contains:

Package Description and features

Endless Car Chase is a fast paced game in which you must collect cash and avoid the police! The game is ready to release straight out of the box, and it can also be easily customized to make it even more engaging to your players. The game supports PC/Mac, iOS, Android, etc. It can be played with the keyboard, gamepad, or touch controls!

How to Play?

Steer your car left or right and collect cash while avoiding the police chasing you.

Features:
- Game ready for release straight out of the box, just build and play!
- Works on all platforms, PC, Mac, iOS, Android, etc
- Supports Keyboard, Gamepad, and Touch controls.
- Easily customizable with lots of options to control game difficulty.
- Great learning resource with commented scripts and documentation.
- All assets included: graphics, sounds, and code.

Current version 1.25

Update history

1.25 (10.04.2018)
- Option to toggle Direction Controls for keyboard/gamepad, which makes the car move in the direction of the input (left/right/up/down/diagonal), instead of rotating left/right only.
- Minor scene updates and fixes.

1.20 (28.12.2018)
- Added city demo scene with buildings, streets, trees, etc.
- Added spawning script that creates objects at certain points, closest to the player.
- Added two bike models, one of them with a turning handlebar.
- Option to make AI cars chase other cars periodically, for some extra chaos!
- Fixed mobile controls and removed unused assets.

1.15 (01.09.2018)
- Added option for Virtual Joystick controls on mobile.

1.10 (10.06.2018)
- Cars can move and rotate up and down along terrain.
- You can show health bars for each car.
- Obstacles and items are spawned along the terrain.

1.05 (25.04.2018)
- Cops cars now try to avoid other cop cars too, while chasing the player.
- Made rock obstacles deadly and kill instantly.
- Tweaked some attributes of spawn rate, car speeds, etc.
- Added hurt effect when cars are hit, and losing control before exploding.
- Added police pickup truck.

1.0 (7.04.2018)
- Initial version

Credits

The sounds are courtesy of the free sound project.

Music is Drum n Bass B Drive by Frank Nora (Public Domain)

Font is Maniac by Vladimir Nikolic (Public Domain)

Credits go to these authors for their great sound samples: NenadSimic, fins, Tristan, MentalSanityOff, blukotek,

Please rate my file, I'd appreciate it 🙂

Overview of the game's library contents

Let's take a look inside the game files. Open the main ECCAssets folder using Unity3D 5.5.0f3 or newer. Take a look at the project library, usually placed on the right or bottom side of the screen. Here are the various folders inside:

- **Animations:** Holds the animation clips made with Unity's built-in animation system.
- **FLA:** Holds the object graphics made with Flash CS3. These are vector graphics than can be easily scaled without loss of quality and then exported as PNG to be used in Unity.
- **Fonts:** Holds the font used in the game.
- **Prefabs:** Holds all the prefabs used in the game. These are distributed to various folders for easier access, Buttons, Enemies, Objects, etc. It also holds all the canvases in the game which are used to hold buttons and other UI elements.

- **Scenes:** The first scene that runs in the game is MainMenu. From this scene you can get to the Game scene.
- **Scripts:** Holds all the scripts used in the game. Each prefab contains one or more of these scripts.
- **Sounds:** Holds all the sounds used in the game. Cash, Item, etc
- **Textures:** Holds all the textures used in the game which are used as sprites in Unity.

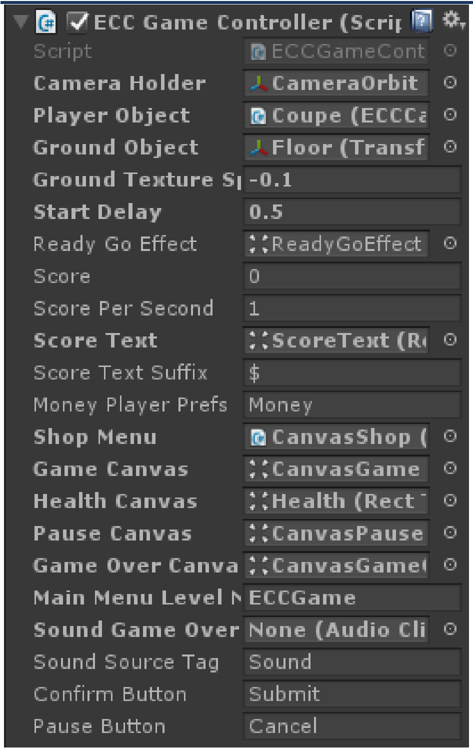
Customization Guide

Getting started

Endless Car Chase Game (ECC) is considered a complete project, and as such is supposed to work as the starting point of your planned game, rather than an addition to an existing project. That said, you may of course pick and choose some of the scripts/models to import into your existing project, but ECC works best as a starter kit which you can customize any part of to your liking.

The Game Controller

The Game Controller is the main prefab that controls all the progress of the game from start to finish. It controls the UI of the game, creates enemies and items and checks the game over condition.



ECC Game Controller (Scrip

Script	ECCGameCont
Camera Holder	CameraOrbit
Player Object	Coupe (ECCCa
Ground Object	Floor (Transf
Ground Texture Sp	-0.1
Start Delay	0.5
Ready Go Effect	ReadyGoEffect
Score	0
Score Per Second	1
Score Text	ScoreText (R
Score Text Suffix	$
Money Player Prefs	Money
Shop Menu	CanvasShop (
Game Canvas	CanvasGame
Health Canvas	Health (Rect T
Pause Canvas	CanvasPause
Game Over Canva	CanvasGame(
Main Menu Level N	ECCGame
Sound Game Over	None (Audio Cli
Sound Source Tag	Sound
Confirm Button	Submit
Pause Button	Cancel

Camera Holder – The camera object and the camera holder that contains it and follows the player.

Player Object – The player object assigned from the project folder or from the shop.

Ground Object - The ground object that follows the player position.

Ground Texture Speed – The speed at which the texture of the ground object moves, make the player seems as if it is moving on the ground.

Start Delay – How long to wait before starting the game. Ready? GO! time.

Ready Go Effect – The effect displayed before starting the game.

Score – The score of the player.

Score Per Second – How many points we get per second

Score Text – The text object that displays the score, assigned from the scene.

Score Text Suffix – Text that is appended to the score value. We use this to add a money sign to the score.

Money Player Prefs – The player prefs record of the total score we have (not high score, but total score we collected in all games which is used as money).

Shop Menu – The canvas menu that displays the shop where we can unlock and select cars.

Main Menu Level Name – The level of the main menu that can be loaded after the game ends.

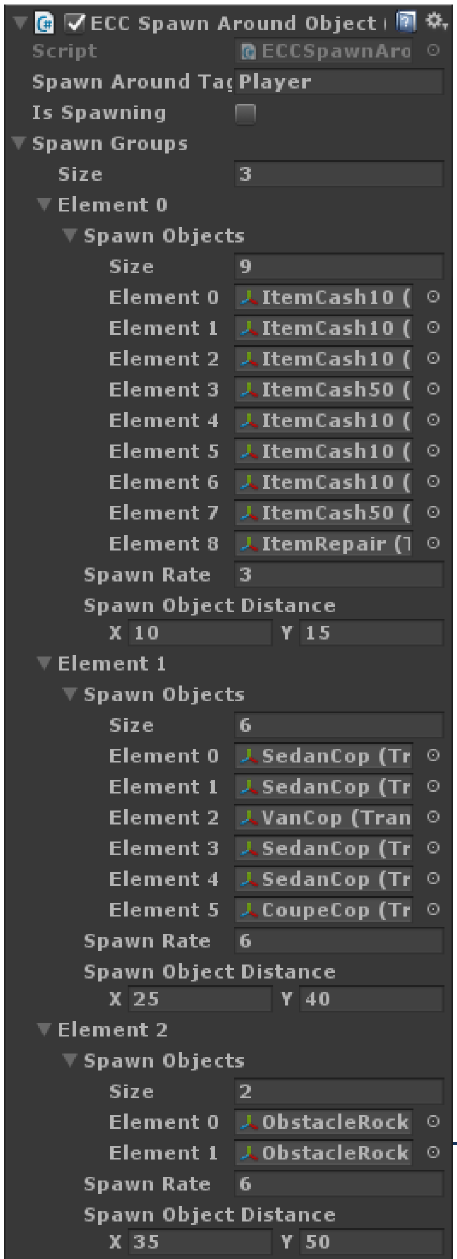
Confirm Button – The keyboard/gamepad button that will restart the game after game over.

Pause Button – The keyboard/gamepad button that pauses the game.

User Interface – Various canvases for the UI, assign them from the scene.

Sounds – Various sounds that play during the game.

Sound Source Tag – The audio source from which the Game Over sound plays.



```
ECC Spawn Around Object
Script                  ECCSpawnAro
Spawn Around Ta(        Player
Is Spawning             ☐
Spawn Groups
  Size                  3
  Element 0
    Spawn Objects
      Size              9
      Element 0         ItemCash10 (
      Element 1         ItemCash10 (
      Element 2         ItemCash10 (
      Element 3         ItemCash50 (
      Element 4         ItemCash10 (
      Element 5         ItemCash10 (
      Element 6         ItemCash10 (
      Element 7         ItemCash50 (
      Element 8         ItemRepair (
    Spawn Rate          3
    Spawn Object Distance
      X 10              Y 15
  Element 1
    Spawn Objects
      Size              6
      Element 0         SedanCop (Tr
      Element 1         SedanCop (Tr
      Element 2         VanCop (Tran
      Element 3         SedanCop (Tr
      Element 4         SedanCop (Tr
      Element 5         CoupeCop (Tr
    Spawn Rate          6
    Spawn Object Distance
      X 25              Y 40
  Element 2
    Spawn Objects
      Size              2
      Element 0         ObstacleRock
      Element 1         ObstacleRock
    Spawn Rate          6
    Spawn Object Distance
      X 35              Y 50
```

The Object Spawner

The Spawn-Around-Object component is responsible for creating objects around the player, such as enemies, pickup items, and obstacles. This component is attached to the Game Controller. You can choose not to use it if you want to have your level with pre-created objects in the scene.

Spawn Around Tag – The tag of the object around which other objects are spawned within a limited range.

Is Spawning – A toggle that turns spawning on and off.

Spawn Groups – An array of spawn groups. These can be enemy cars, pickup items, or obstacle rocks for example.

Spawn Objects – A list of all Objects that will be spawned.

Spawn Rate – The rate at which objects are spawned, in seconds.

Spawn Object Distance – The distance at which this object is spawned relative to the center object (taken from Spawn Around Tag).

Maximum Stars – The maximum stars we can get in a stage.

We have 3 different groups in this spawner; the first group creates an item every 3 seconds, starting with small cash, then a bigger cash stack, and finally a repair item before looping from the start of the list. The items are created at a random distance ranging between 10 to 15 points from the central object (the player).

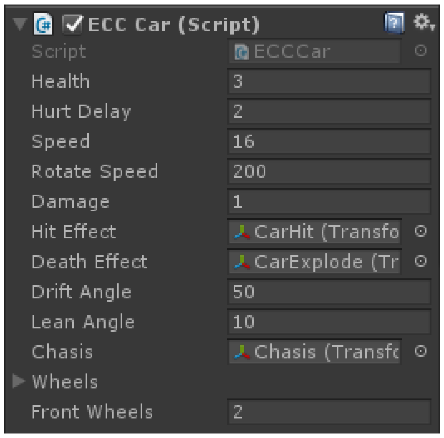
The next group creates an enemy every 6 seconds, starting with 2 sedans, then a van, then 2 sedans, and finally a coupe before looping from the start of the list. The enemies are created at a random distance ranging between 25 to 40 points from the central object (the player).

The final group creates an obstacle every 6 seconds, two types of rocks. The obstacles are created at a random distance ranging between 35 to 50 points from the central object (the player).

You can add your own spawn groups and they can be anything you want, not just enemies, or obstacles, or items.

The Car

The cars in the game use the ECCCar component to define their attributes. There are several things you can control to make them behave differently.


ECC Car (Script)

Script	ECCCar
Health	3
Hurt Delay	2
Speed	16
Rotate Speed	200
Damage	1
Hit Effect	CarHit (Transfo
Death Effect	CarExplode (Tr
Drift Angle	50
Lean Angle	10
Chasis	Chasis (Transfc
Wheels	
Front Wheels	2

Health – The health of the player. If this reaches 0, the player dies.

Hurt Delay – When the car gets hit and hurt, there is a delay during which it cannot be hit again.

Speed – The speed of the player, how fast it moves player. The player moves forward constantly.

Rotate Speed – How quickly the player car rotates, in both directions.

Damage – The damage this car causes when hitting other cars. Damage is reduced from Health.

Hit Effect – The effect that appears when this car is hit by another car or by an obstacle.

Death Effect – The effect that appears when this car dies.

Drift Angle – The slight extra rotation that happens to the car as it turns, giving a drifting effect.

Lean Angle – The slight side tilt that happens to the car chassis as the car turns, making it lean inwards or outwards from the center of rotation.

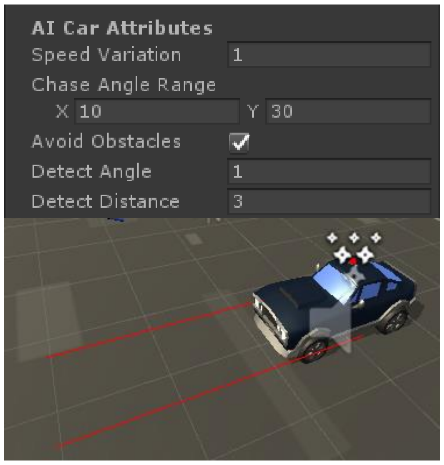
Chassis – The chassis object of the car which leans when the car rotates.

Wheels – The wheels of the car which rotate based on the speed of the car. The front wheels also rotate in the direction the car is turning.

Front Wheels – The front wheels of the car also rotate in the direction the car is turning.

Chassis – The chassis object of the car which leans when the car rotates.

The same car component is used for the player as well as the computer controlled cars. There are several AI attributes you can control for each car.



```
AI Car Attributes
Speed Variation     1
Chase Angle Range
  X 10              Y 30
Avoid Obstacles     ✓
Detect Angle        1
Detect Distance     3
```

Speed Variation – A random value that is added to the base speed of the AI car, to make their movements more varied.

Chase Angle Range – A random value that is to the chase angle to make the AI cars more varied in how to chase the player.

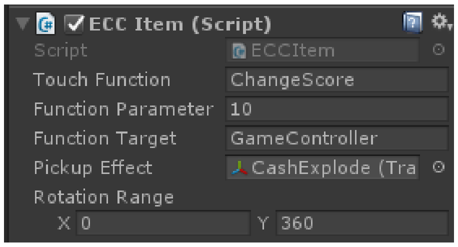
Avoid Obstacles – Make AI cars try to avoid obstacles. Obstacle are objects that have the ECCObstacle component attached to them.

Detect Angle – The width of the obstacle detection area for this AI car.

Detect Distance – The forward distance of the obstacle detection area for this AI car.

Items and Obstacles

Items are objects that can be picked up by the player. An item can be money that the player collects, or a repair kit that increases health. Here's how it works:



ECC Item (Script)	
Script	ECCItem
Touch Function	ChangeScore
Function Parameter	10
Function Target	GameController
Pickup Effect	CashExplode (Tra
Rotation Range	
X 0	Y 360

Touch Function – The function that is called when this item is picked up. For example it can be "ChangeScore" or "ChangeHealth".

Function Parameter – The parameter that will be passed with the function. For example 10 with "ChangeScore" will add 10 points to the score.

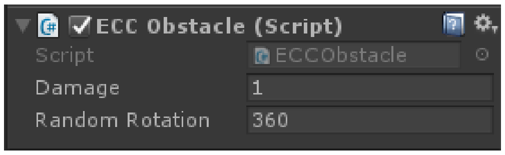
Function Target – The tag of the target object that the function will play from. For example "GameController" will look for an object tagged with that name.

As an example to the three variables above, I can call the function "ChangeHealth" with a parameter of 1, targeting the "GameController" tag, and this will add 1 health to the player. Another example is calling the function "ChangeScore" with a parameter of 50, targeting the "GameController" tag, which will add 50 to the game score when picked up.

Pickup Effect – The effect that is created at the location of this item when it is picked up.

Rotation Range – A random rotation given to the object only on the Y axis.

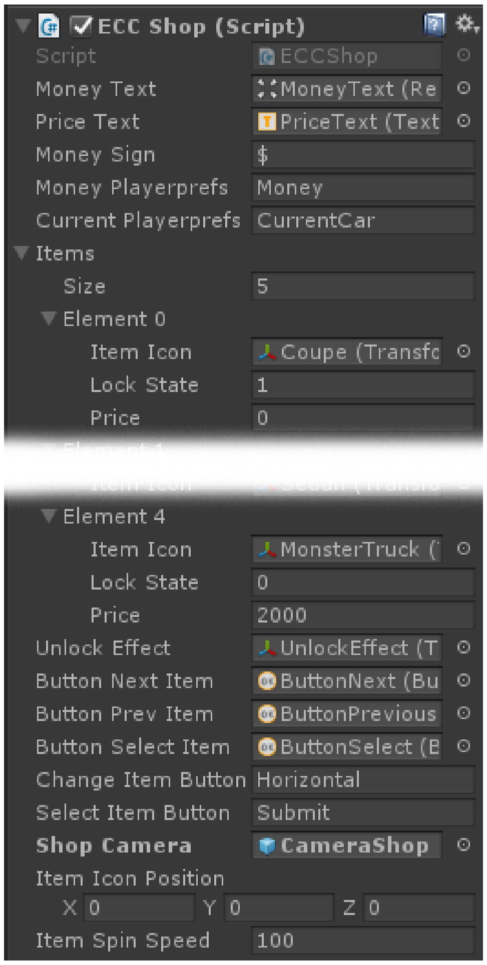
Obstacles are simpler, they only have a single parameter to set the damage value caused to the car that touches them.



ECC Obstacle (Script)	
Script	ECCObstacle
Damage	1
Random Rotation	360

The Shop

This script displays items that can be unlocked with in-game money. Each item has a 3D icon that spins while showing the price of the item if it is still locked. Locked items are also darkened.



ECC Shop (Script)

Script	ECCShop
Money Text	MoneyText (Re
Price Text	PriceText (Text
Money Sign	$
Money Playerprefs	Money
Current Playerprefs	CurrentCar
Items	
Size	5
Element 0	
Item Icon	Coupe (Transfo
Lock State	1
Price	0
Element 1	
Item Icon	Sedan (Transfo
Element 4	
Item Icon	MonsterTruck (
Lock State	0
Price	2000
Unlock Effect	UnlockEffect (T
Button Next Item	ButtonNext (Bu
Button Prev Item	ButtonPrevious
Button Select Item	ButtonSelect (B
Change Item Button	Horizontal
Select Item Button	Submit
Shop Camera	CameraShop
Item Icon Position	X 0 Y 0 Z 0
Item Spin Speed	100

Money Text – The text object that displays the money we have.

Price Text – The text object that displays the price of the current item.

Money Sign – The sign that appears next to the money text.

Money Player Prefs – The player prefs record of the money we have.

Current Player Prefs – The player prefs record of the current item we selected.

Items – A list of items in the game, their names, the scene they link to, the price to unlock it, and the state of the item (-1 - locked and can't be played, 0 - unlocked no stars, 1-X the star rating we got on the item.

Item Icon - The object representing this item, can be a 2D icon or a 3D object**.**

Lock State - Is the item locked or not? 0 = locked, 1 = unlocked.

Price - How much money we need to unlock this item.

Unlock Effect - The effect that appears when we unlock this item.

Buttons - Buttons for going to the next/previous items, and selecting items.

Shop Camera - The camera object that is turned on when this shop is activated.

Item Icon Position - Buttons for going to the next/previous items, and selecting items.

Item Spin Speed - The rotation speed of the currently selected item in 3D space.

UnityAds Integration

Since Unity 5.2 UnityAds integration has been simplified, here's how you can have full screen video ads in your game.

This video shows a quick process of integrating UnityAds into your project. In the example we used one of my templates, but it works on all my other templates too.

https://www.youtube.com/watch?v=EQNTgfV35DU

Here is what we did in the process:

1. Sign in to your Unity account in order to allow Unity Services such as UnityAds to be activated.

2. Open Build Settings and switch the platform to one of the supported ones (iOS, Android).

3. Download Puppeteer's UnityAds package from: puppeteerinteractive.com/freebies/PUPUnityAds.unitypackage

4. Drag the downloaded package into your Unity project, and import it. This UnityAds prefab can be used to display ads every several minutes.

5. Drag the prefab into any scene where you want ads to be shown. Make sure to save changes.

6. The time check is shared between all prefabs in all scenes, so you will never show too many ads.

7. The final step is to activate UnityAds services and get your unique project ID.

8. Open the services window and choose your organization, then click create.

9. Choose UnityAds from the list and turn it On.

10. Choose age group for your project (Will affect the nature of ads shown), and save changes.

11. While working on your project keep Test Mode activated. But when you are ready to release the final project, switch Test Mode off.

12. That's it! Now when you start the game, an ad will be shown after 3 minutes. The ad will never appear during gameplay or post-game screen. Instead, it will wait until the next level load (restart, main menu, etc) and then show the ad.

Before releasing a game, make sure you uncheck **Enable Test Mode.**

For more info about integrating UnityAds read this:

http://unityads.unity3d.com/help/monetization/integration-guide-unity

It is highly advised, whether you are a designer or a developer to look further into the code and customize it to your pleasing. See what can be improved upon or changed to make this file work better and faster. Don't hesitate to send me suggestions and feedback to puppeteerint@gmail.com

Follow me on twitter for updates and freebies!

Good luck with your modifications!