Subject: Find_My_Vehicle
Posted by robbyke on Tue, 04 Oct 2011 22:10:22 GMT
View Forum Message <> Reply to Message

How do i get the VehicleGameObj converted to a normal GameObject?

Subject: Re: Find_My_Vehicle
Posted by snazy2000 on Thu, 06 Oct 2011 06:38:33 GMT
View Forum Message <> Reply to Message

what do you mean?

Subject: Re: Find_My_Vehicle
Posted by danpaul88 on Thu, 06 Oct 2011 14:10:53 GMT
View Forum Message <> Reply to Message

```
VehicleGameObject pVehicleObj = Find_My_Vehicle();
GameObject* pObject = dynamic_cast<GameObject*>(pVehicleObj);
```

Or...

```
GameObject* pObject = dynamic_cast<GameObject*>(Find_My_Vehicle());
```

Or...

```
GameObject* pObject = (GameObject*)Find_My_Vehicle();
```

(PS: If you didn't already figure it out, VehicleGameObject *is* a GameObject, since it's a derived class)

(PPS: Yes, I know dynamic casting a subclass to its superclass is redundant since it's always going to succeed, just illustrating my point)

Subject: Re: Find_My_Vehicle
Posted by robbyke on Fri, 07 Oct 2011 00:28:34 GMT

View Forum Message <> Reply to Message

ok that works although i dont understand how it works i gues i understand it somehow on some way XD im just a nooby in programmin

Subject: Re: Find_My_Vehicle
Posted by Omar007 on Fri, 07 Oct 2011 07:27:12 GMT

View Forum Message <> Reply to Message

VehicleGameObject is derived from GameObject.
This means VehicleGameObject is everything GameObject is +more.

The code for this would be
class VehicleGameObject : public GameObject{...};
*note: In the scripts.dll source, there might be an inheritance in between or multiple inheritance. I did not check this code so it may look a bit different in the real source. This does however reflect how VehicleGameObject is a GameObject

You can ofcourse have this stacked
class X{...};

//Y is X +more
class Y : public X{...};

//Z is Y +more
class Z : public Y{...};
As Z derives from Y, and Y from X, Z is also an X.

Or multiple
class A{...};
class B{...};

class C : public A, B{...};
C is both an A and B, BUT A is not a B and B is not an A in this case.

http://www.cprogramming.com/tutorial/lesson20.html

Then you also get things like access modifiers and the virtual keyword that all affect inheritance behavior, but you can search that up when you need it

Subject: Re: Find_My_Vehicle
Posted by Gen_Blacky on Fri, 07 Oct 2011 07:32:36 GMT

View Forum Message <> Reply to Message

look at VehicleGameObj.h for all the stuff you can do with vehicles game objects.