Vehicle Property and Inheritance

By Ruthie Tucker, under the direction of Professor Susan Rodger, Duke University, 2008

Description

• This tutorial will demonstrate how to use inheritance to save a modified object as a new, more intelligent class, to use in other worlds.

• It will also demonstrate how to set a vehicle property part way through an animation and end the property before the animation is complete.

Characters

The Space Scene, the Ground Roamer, The Lunar Lander, the Space Colonist and a Cow

Prerequisites

• In order to complete this example you need to know how to do the following
  – Intro Tutorial
  – Methods
  – Parameters
  – Loops
Step One: Vehicle Property

- In this animation we have an astronaut on a Lunar Lander. We want him to go to the Ground Roamer, ride around and come back to the Lunar Lander.

Getting started in your world

- What is the vehicle property?
  - The vehicle property allows one object to be attached to or “ride” another object.
  - Example = a man riding a horse
- What is Inheritance?
  - Inheritance allows you to teach an object to do things in one world and then save those new abilities to use in a different world.
  - Example = Teaching the horse to gallop

Starting code

- Because we are learning inheritance and vehicle property in this tutorial we need to code all of our methods at the class level.
- The following slide shows how your world level code should look when you are completely done.
Before coding vehicle property

• We need to create some simple actions for our space colonist
  – SpaceColonist.ClimbObject
  – SpaceColonist.Walk
  – SpaceColonist.GetOnTransport

• Unfortunately there is no quick way to do this. You simply need to mess around with his arms and legs until you get his actions too look right.

A Quick Side Note

• You will notice from the pictures in this presentation that the colonist is named “CleverColonist” instead of “SpaceColonist”

• This will make sense later on. So don’t worry about it right now

• At this point your character should still be named “SpaceColonist”

SpaceColonist.ClimbObject

• You do not have to code the climb action this way.
• This is simply an example of how to get the astronaut to climb down the ladder.
• If you are having trouble you can just copy this code

• The code is on the next slide

Code for SpaceColonist.ClimbObject

<table>
<thead>
<tr>
<th>Do in order</th>
</tr>
</thead>
<tbody>
<tr>
<td>cleverColonist</td>
</tr>
<tr>
<td>cleverColonist</td>
</tr>
<tr>
<td>cleverColonist</td>
</tr>
<tr>
<td>cleverColonist</td>
</tr>
</tbody>
</table>

Loop 4.5 times

<table>
<thead>
<tr>
<th>Do together</th>
</tr>
</thead>
<tbody>
<tr>
<td>cleverColonist</td>
</tr>
<tr>
<td>cleverColonist</td>
</tr>
<tr>
<td>cleverColonist</td>
</tr>
</tbody>
</table>

Do together

<table>
<thead>
<tr>
<th>Do together</th>
</tr>
</thead>
<tbody>
<tr>
<td>cleverColonist</td>
</tr>
<tr>
<td>cleverColonist</td>
</tr>
<tr>
<td>cleverColonist</td>
</tr>
</tbody>
</table>
Vehicle Property Continued

• Once you have created this method you can call it under World.MyFirstMethod to make the beginning of the animation
• You may repeat this step for spaceColonist.walk and SpaceColonist.GetOnTransport

Additional Code

• The following 4 slides show the code for SpaceColonist.Walk and SpaceColonist.GetOnTransport.
• If you do not want to try to create them on your own you can copy them from these slides

Code for SpaceColonist.Walk
(see next slide for second half)

<table>
<thead>
<tr>
<th>space_colonist.Walk</th>
<th>No parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do in order</td>
<td></td>
</tr>
<tr>
<td>While (( cleverColonist.distance to object ) &gt; 3.6 )</td>
<td></td>
</tr>
<tr>
<td>Do in order</td>
<td></td>
</tr>
<tr>
<td>cleverColonist.leg1 turn forward 0.15 revolutions duration = 0.5 seconds</td>
<td></td>
</tr>
<tr>
<td>cleverColonist.leg2 turn backward 0.15 revolutions duration = 0.25 seconds</td>
<td></td>
</tr>
<tr>
<td>cleverColonist.move amount = 0.5 meters toward target = object duration = 0.5 seconds</td>
<td></td>
</tr>
<tr>
<td>cleverColonist.spaceMan.leftArm turn forward 0.15 revolutions duration = 0.5 seconds</td>
<td></td>
</tr>
<tr>
<td>cleverColonist.spaceMan.rightArm turn backward 0.15 revolutions duration = 0.5 seconds</td>
<td></td>
</tr>
<tr>
<td>Do together</td>
<td></td>
</tr>
<tr>
<td>cleverColonist.leg1 turn forward 0.15 revolutions duration = 0.5 seconds</td>
<td></td>
</tr>
<tr>
<td>cleverColonist.leg2 turn backward 0.15 revolutions duration = 0.5 seconds</td>
<td></td>
</tr>
<tr>
<td>cleverColonist.move amount = 0.5 meters toward target = object duration = 0.5 seconds</td>
<td></td>
</tr>
<tr>
<td>cleverColonist.spaceMan.leftArm turn backward 0.15 revolutions duration = 0.5 seconds</td>
<td></td>
</tr>
<tr>
<td>cleverColonist.spaceMan.rightArm turn forward 0.15 revolutions duration = 0.5 seconds</td>
<td></td>
</tr>
<tr>
<td>Do together</td>
<td></td>
</tr>
<tr>
<td>cleverColonist.leg1 turn backward 0.15 revolutions duration = 0.5 seconds</td>
<td></td>
</tr>
<tr>
<td>cleverColonist.leg2 turn forward 0.15 revolutions duration = 0.5 seconds</td>
<td></td>
</tr>
<tr>
<td>cleverColonist.move amount = 0.5 meters toward target = object duration = 0.5 seconds</td>
<td></td>
</tr>
<tr>
<td>cleverColonist.spaceMan.leftArm turn backward 0.15 revolutions duration = 0.5 seconds</td>
<td></td>
</tr>
<tr>
<td>cleverColonist.spaceMan.rightArm turn forward 0.15 revolutions duration = 0.5 seconds</td>
<td></td>
</tr>
<tr>
<td>Do together</td>
<td></td>
</tr>
<tr>
<td>cleverColonist.leg1 turn backward 0.15 revolutions duration = 0.5 seconds</td>
<td></td>
</tr>
<tr>
<td>cleverColonist.leg2 turn forward 0.15 revolutions duration = 0.5 seconds</td>
<td></td>
</tr>
<tr>
<td>cleverColonist.move amount = 0.5 meters toward target = object duration = 0.5 seconds</td>
<td></td>
</tr>
<tr>
<td>cleverColonist.spaceMan.leftArm turn forward 0.15 revolutions duration = 0.5 seconds</td>
<td></td>
</tr>
<tr>
<td>cleverColonist.spaceMan.rightArm turn backward 0.15 revolutions duration = 0.5 seconds</td>
<td></td>
</tr>
<tr>
<td>Do together</td>
<td></td>
</tr>
<tr>
<td>cleverColonist.leg1 turn backward 0.15 revolutions duration = 0.5 seconds</td>
<td></td>
</tr>
<tr>
<td>cleverColonist.leg2 turn forward 0.15 revolutions duration = 0.5 seconds</td>
<td></td>
</tr>
<tr>
<td>cleverColonist.move amount = 0.5 meters toward target = object duration = 0.5 seconds</td>
<td></td>
</tr>
<tr>
<td>cleverColonist.spaceMan.leftArm turn forward 0.15 revolutions duration = 0.5 seconds</td>
<td></td>
</tr>
<tr>
<td>cleverColonist.spaceMan.rightArm turn backward 0.15 revolutions duration = 0.5 seconds</td>
<td></td>
</tr>
<tr>
<td>Do together</td>
<td></td>
</tr>
</tbody>
</table>

Second half of code for SpaceColonist.walk

cleverColonist.spaceMan.leftArm turn backward 0.15 revolutions duration = 0.5 seconds
cleverColonist.spaceMan.rightArm turn forward 0.15 revolutions duration = 0.5 seconds
cleverColonist.spaceMan.leftLeg turn forward 0.15 revolutions duration = 0.5 seconds
cleverColonist.spaceMan.rightLeg turn backward 0.15 revolutions duration = 0.5 seconds
cleverColonist.move amount = 0.5 meters toward target = object duration = 0.5 seconds
cleverColonist.spaceMan.leftArm turn backward 0.15 revolutions duration = 0.5 seconds
cleverColonist.spaceMan.rightArm turn forward 0.15 revolutions duration = 0.5 seconds
cleverColonist.spaceMan.leftLeg turn forward 0.15 revolutions duration = 0.5 seconds
cleverColonist.spaceMan.rightLeg turn backward 0.15 revolutions duration = 0.5 seconds
cleverColonist.move amount = 0.5 meters toward target = object duration = 0.5 seconds
cleverColonist.spaceMan.leftArm turn backward 0.15 revolutions duration = 0.5 seconds
cleverColonist.spaceMan.rightArm turn forward 0.15 revolutions duration = 0.5 seconds
cleverColonist.spaceMan.leftLeg turn forward 0.15 revolutions duration = 0.5 seconds
cleverColonist.spaceMan.rightLeg turn backward 0.15 revolutions duration = 0.5 seconds
cleverColonist.move amount = 0.5 meters toward target = object duration = 0.5 seconds
SpaceColonists.GetOnTransport

Now we are ready to create the Vehicle for “Space Colonist”

- Create a new method called SpaceColonist.RideTransport
- Click on SpaceColonist - Properties, drag vehicle to your Method Editor, value = the entire GroundRoamer

Vehicle Property Continued

Remember, in order to make the Colonist ride the GroundRoamer you must select “Colonist set vehicle to GroundRoamer” not the other way around.

After the vehicle is set you can command the Roamer and the Colonist will follow.
Now test it out

- Once your space colonist is set as a vehicle to the GroundRoamer you can test it out.
- Try moving the GroundRoamer around.
- If you did the code correctly the Space Colonist should ride along with it.

Vehicle Property continued

- Once our Space Colonist has had an explorative journey around the moon we want him to get off the GroundRoamer and walk away.

Vehicle Property Continued

Set Camera back to the entire world.
Write code to have the Colonist walk away from the Roamer.
You will notice that the Roamer stays put while the Space Colonist moves forward.

But Wait!

- What happens if we have the Roamer move instead of the Space Colonist? Try it.
- The Colonist moves with it.
- This is because originally, we set the Space Colonist as a vehicle to the Roamer.
- This means that the Colonist can do whatever he wants as long as the Roamer is stationary. But when the roamer moves, the Colonist must follow.
Now that’s Better

- Before we move the Colonist away from the Roamer he needs to be set back as a vehicle to the entire world. Now he can travel at will.

Vehicle Property Wrap Up

- You might have noticed that much of this tutorial was spent teaching the Space Colonist to do simple tasks like walking.
- Wouldn’t it be nice to use those things you taught him in another world without going through the teaching process all over again?

Welcome to Inheritance!

- Inheritance is a way to save a particular class and all its methods, to use again in another world.
- The first step is to rename your class.
- Right click on the class you want to save out and rename it whatever you like.
- In our case we will rename “Space Colonist” to “Clever Colonist”.

Your Screen should look like this
Inheritance Continued

- Once you have renamed your class you need to save it out.
- Right click again on “Clever Colonist” and scroll down to “Save Object”

Inheritance continued

- A pop up window like
- This should appear
- You should create a folder to store your
- New classes.
- Name it “Alice Classes”
- Click on save
- Note the name of the file has “.a2c”

Inheritance Continued

- Now open a new world in which to use your new class “CleverColonist”
- Once you are in your new world hit File/Import

Inheritance continued

- After clicking on “Import” your screen should look like this.
- Click your clever colonist guy and then “Import”.
- Your new class should then import into your new world.
Inheritance continued

- Our Space Colonist has just found himself in a magical new world
- This time instead of a lunar Lander our space hero has to climb a tree and ride a triceratops

Inheritance wrap up

- Fortunately since we have already saved all of these methods out in a new class, programming this is as simple as this.
- You are now qualified to use vehicle property and inheritance!

Hey why was there a Cow in the beginning?

- Helpful Alice tip
- If you want your object to walk towards something
- You can insert a random object and set "Is showing to false"
- In our first animation this was how the astronaut walked away from the Roamer

Code for is showing

- This is a simple way to move objects places without cluttering your world with place markers.
- And that's all folks!