

Princess & Dragon

Part 3: A Knight Comes Riding In—Cameras & Events



By Elizabeth Liang
under the direction of
Professor Susan Rodger
Duke University
June 2010

Introduction

Welcome to Part 3 of the Princess & Dragon tutorial. In Part 2 we covered how to change camera views in an animation, create and edit object and world level methods, and how to change properties.

Part 1: Objects

Part 2: Methods & Properties

Part 3: Cameras & Events

Part 4: Billboards, Sound, & 3D-Text

In Part 3 we'll add more animation so that a knight will come in on a horse.

This will cover more on camera views as well as introduce you to new events and functions.

Step 1: Adding Objects

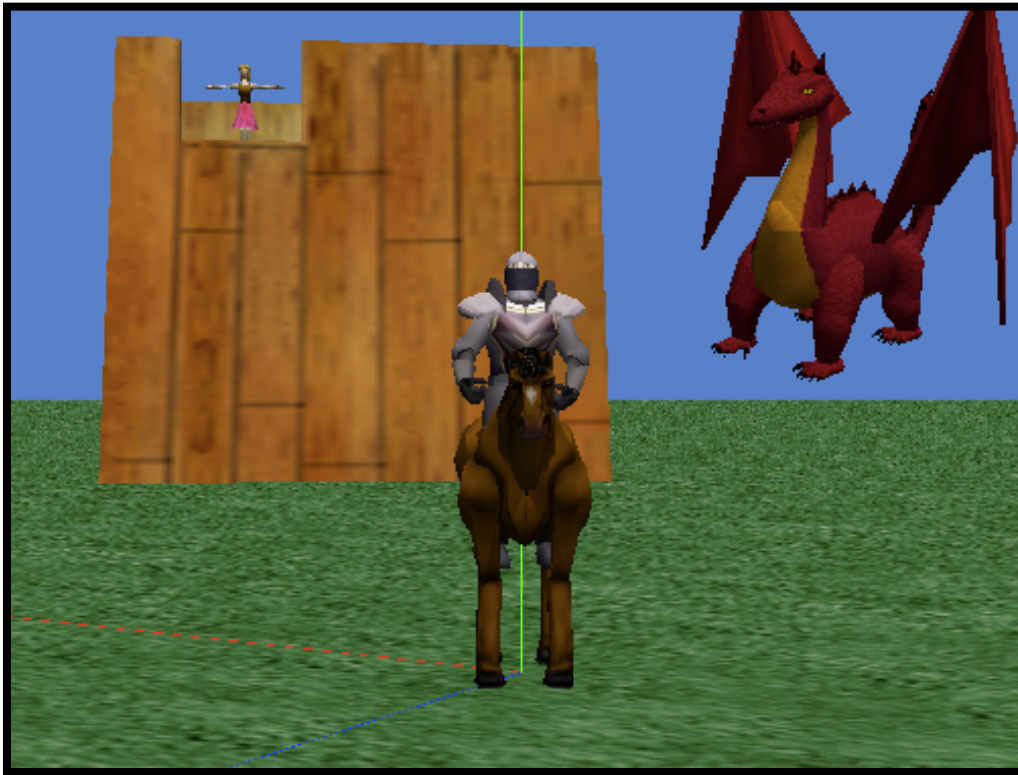
First a knight and horse must be added to the scene.



Click on the **Add Objects** button to go to the Adding Object Screen. Click on the **Medieval** gallery folder and add a **horse** and a **knight**.



Step 1: Positioning Objects

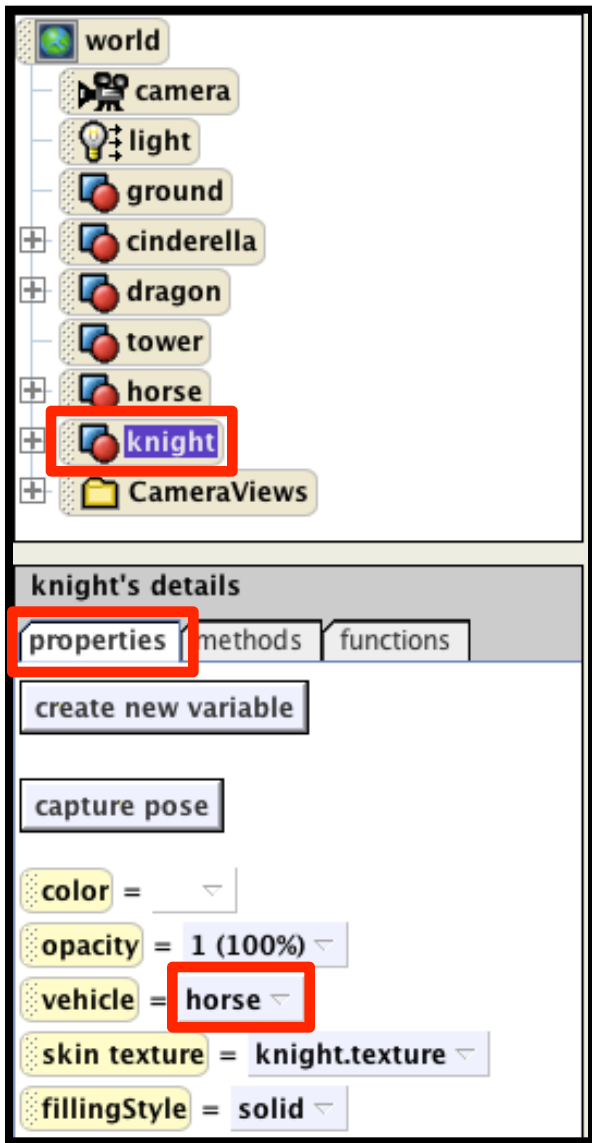


Use the **positioning tools** to **turn** the **horse** and **knight** toward the camera.

Then **move** the **knight** so that he is in the middle of the **horse** with the **mouse** tool.

Finally **move** the **knight up** until he looks like he is sitting on the **horse**.

Step 1: Vehicle Reviewed

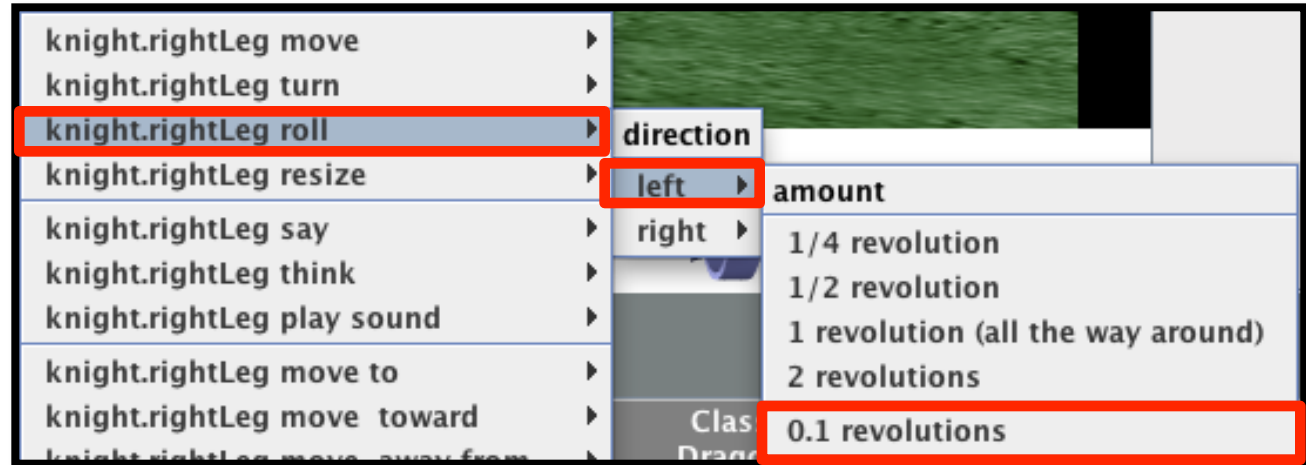
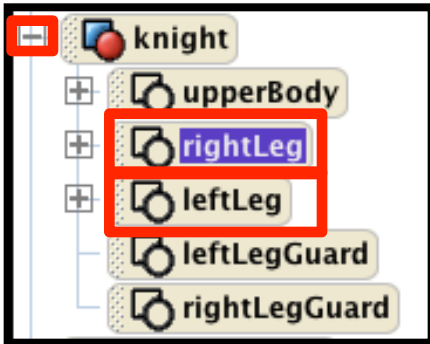


To glue the knight to horse, click on **knight** in the object tree. Under **properties**, set the **vehicle** of the **knight** to the **horse, the entire horse**.

Remember that changing the vehicle property this way is an instant change to the property and part of the set up of the world. There is no dragging in a line of code so it is not part of the animation.



Step 1: Positioning Objects Continued



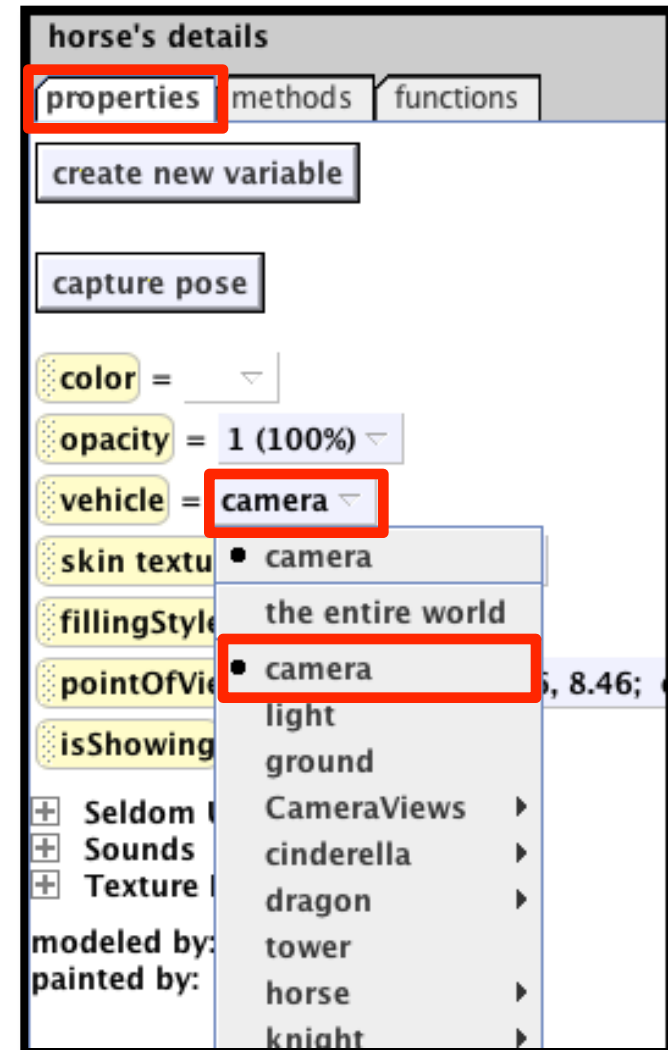
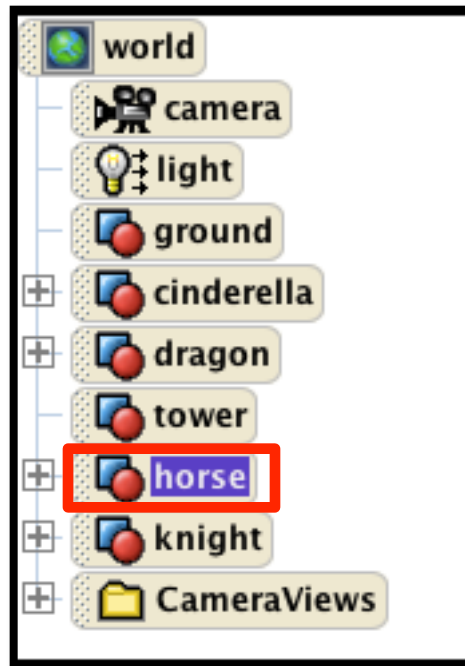
Right Click on the knight's right leg in the object tree select roll, left 0.1 revolutions.

Now Roll the left leg right 0.1 revolutions. This will move the legs of the knight so he looks more like he is riding.



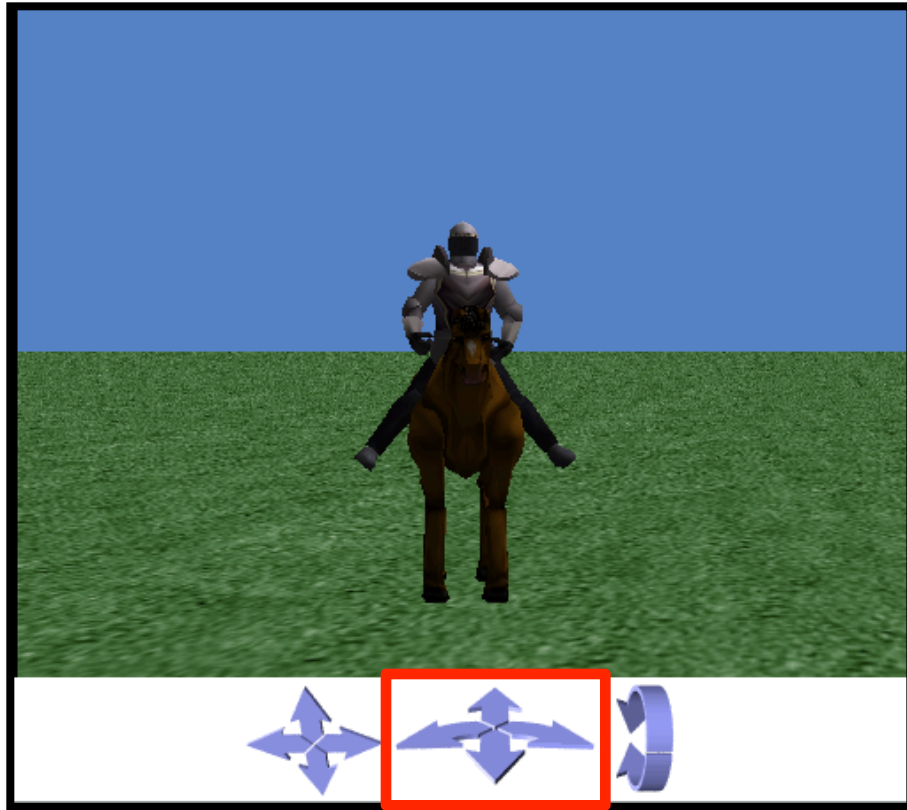
Step 1: Vehicle Review Continued

Right click on the **horse** in the object tree and under **properties**, change the **vehicle** of the horse from the **world** to the **camera**.

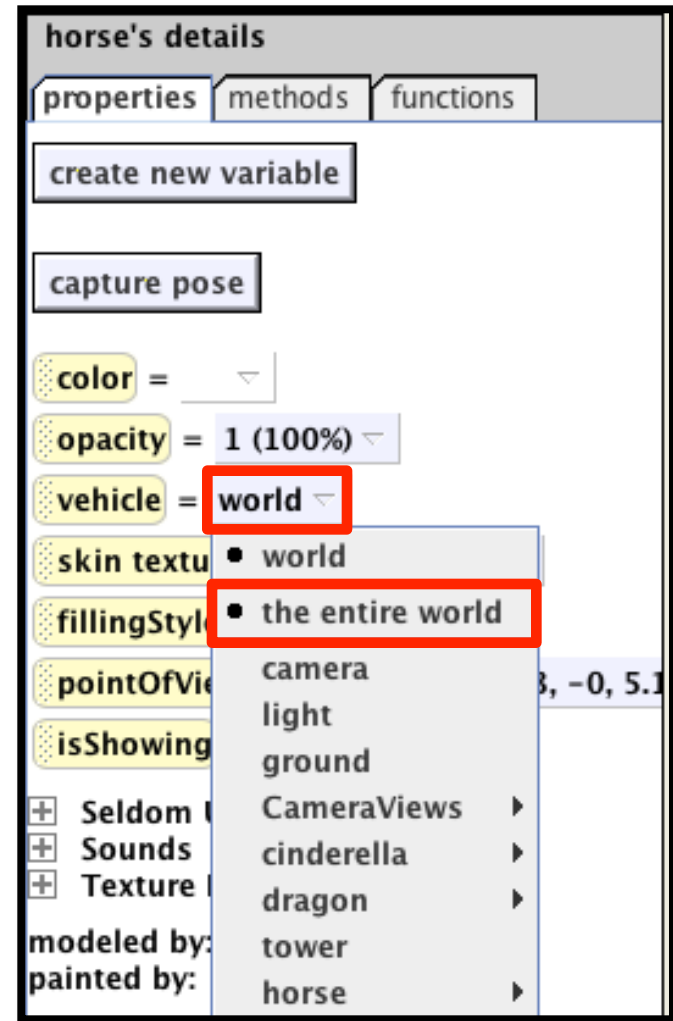


The trick here is to glue the horse to the camera so that when the camera view changes in the scene editor, the horse (and knight) moves with it. When we are done creating a new camera view we can simply 'unglue' the horse.

Step 1: Camera Positioning

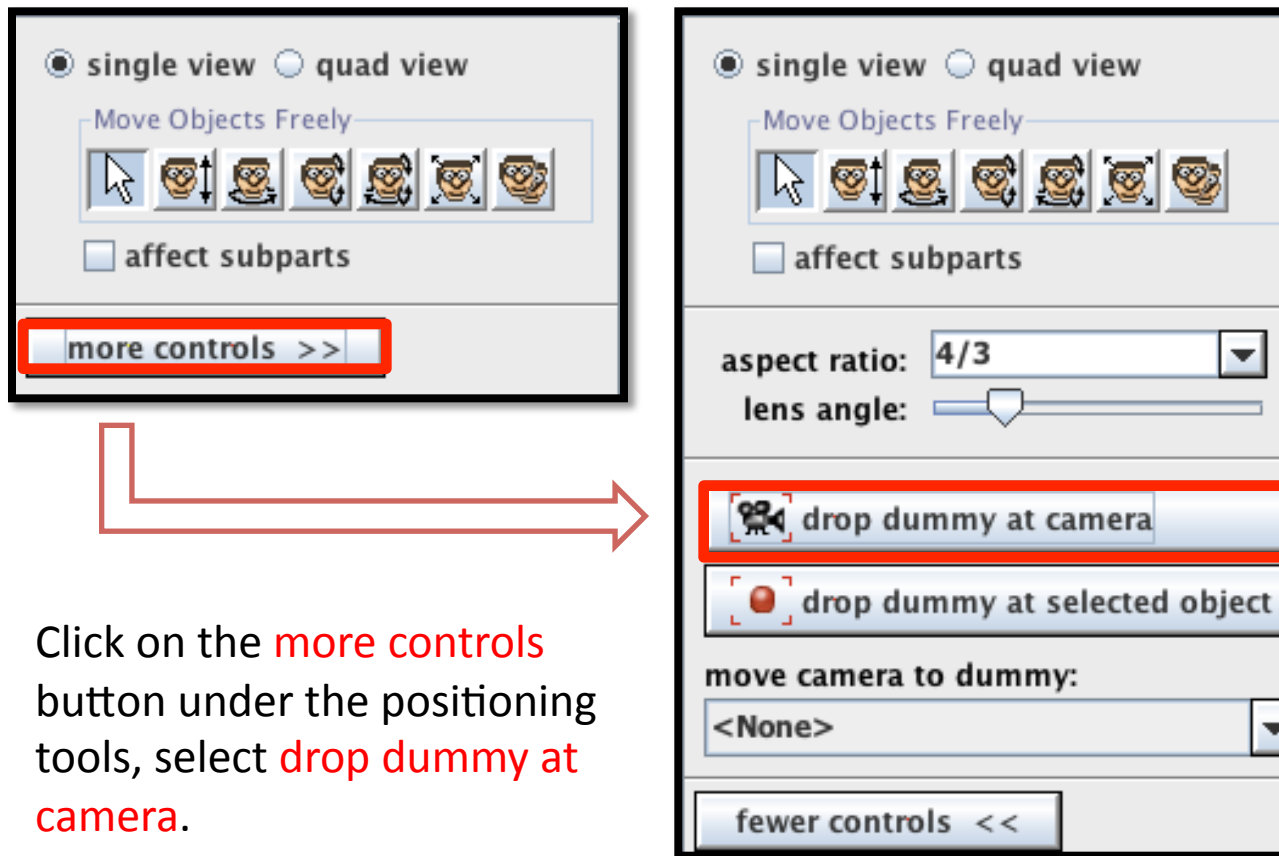


Drive the camera to the right until you can no longer see any of the other objects. Now you can 'unglue' the horse from the camera by setting the horse's vehicle back to the entire world.



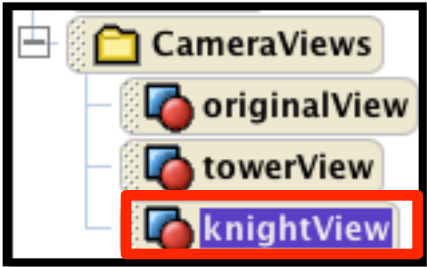
Step 1: Dummy Object Review

Remember that a Dummy Object allows us to save the position of the camera and return to it. It's like bookmarking a camera view. We did this with the originalView and towerView. Now we want to add a bookmarked camera view of the knight and horse into the CameraViews folder.

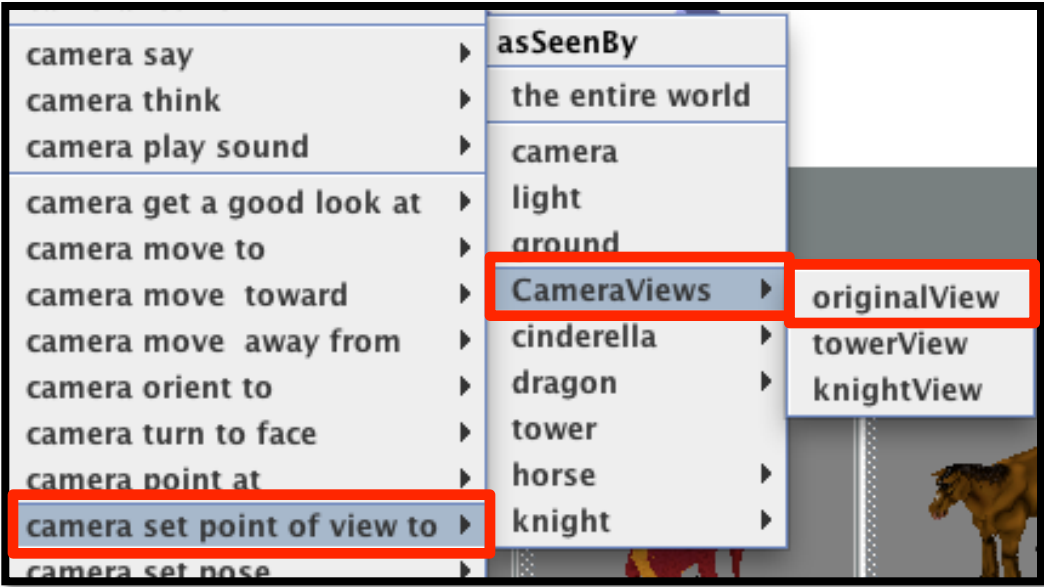


Click on the **more controls** button under the positioning tools, select **drop dummy at camera**.

Step 1: Dummy Object Review



The **dummy object** will appear in the **CameraViews** folder in the object tree. **Rename** it **knightView**.



Right click on the **camera** in the object tree and select **camera set point of view to**, **CameraViews**, **originalView**.

Click on **DONE**. Now it's time to animate!



Step 2: Change Camera View

camera's details

properties **methods** functions

- camera think
- camera play sound
- camera get a good look at
- camera move to
- camera move toward
- camera move away from
- camera orient to
- camera turn to face
- camera point at
- camera set point of view to**
- camera set pose
- camera stand up

First we want the **camera** to swing over to the **knight** during the animation. Drag a **camera set point of view to method** into the very end of **my first method**.

asSeenBy

the entire world

camera

light

ground

CameraViews

cinderella

dragon

originalView

towerView

knightView

Select **CameraViews** and then **knightView**.

camera set point of view to originalView more...

world.capturePrincess

camera set point of view to knightView more...

Step 2: knightRescue World Method

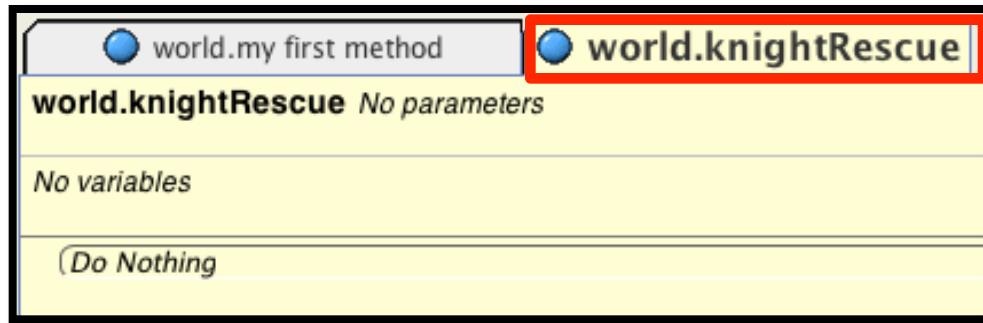


Now let's create a new world level method. In this method the knight will ride over to the princess and yell at the dragon to drop her. Then the knight will catch the princess.

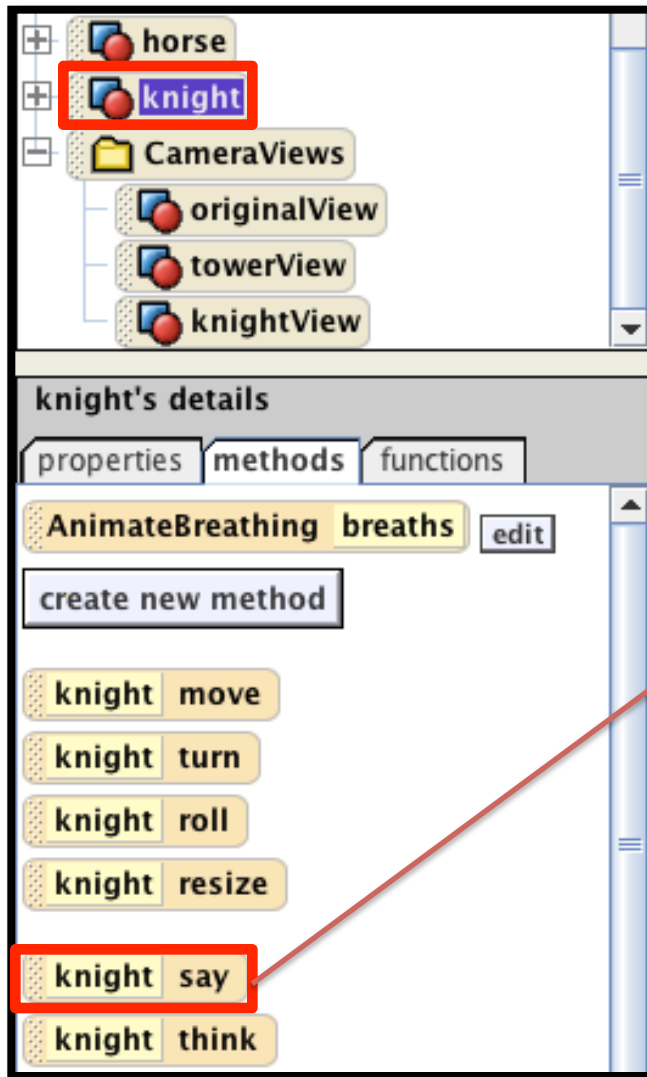
Click on **world** in the **object tree** and under methods, click on **create new method**.



Name it **knightRescue**.

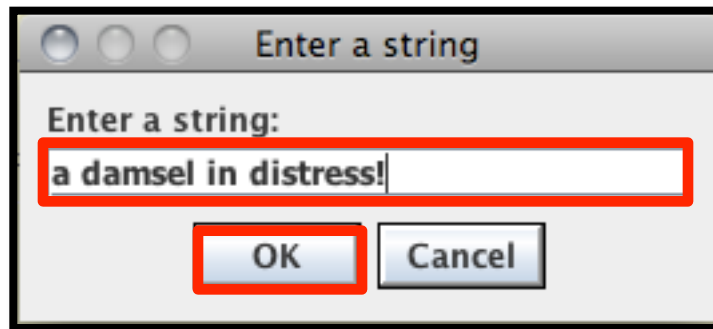
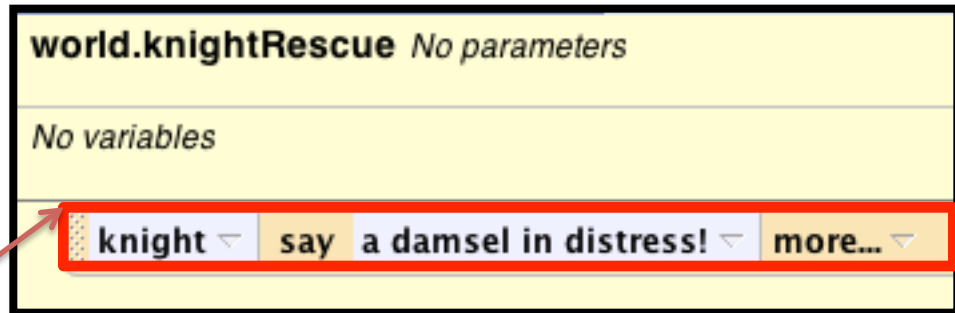


Step 2: Animation



Click on **knight** in the object tree. Drag in a **knight say method** into the new **knightRescue** method. He will call out **a damsel in distress!**

You will notice the AnimateBreathing method in the knight's methods. We won't use it in this tutorial but some objects come with more complicated methods unique to that object built in.



Step 2: Animation Continued

The screenshot shows the object tree on the left with 'horse' selected. Below it, the 'horse's details' panel is open to the 'methods' tab. A list of methods is shown, with 'horse move' and 'horse turn to face' highlighted with red boxes. A red arrow points from the 'horse move' method to the 'world.knightRescue' script area.

Click on the **horse** in the object tree.

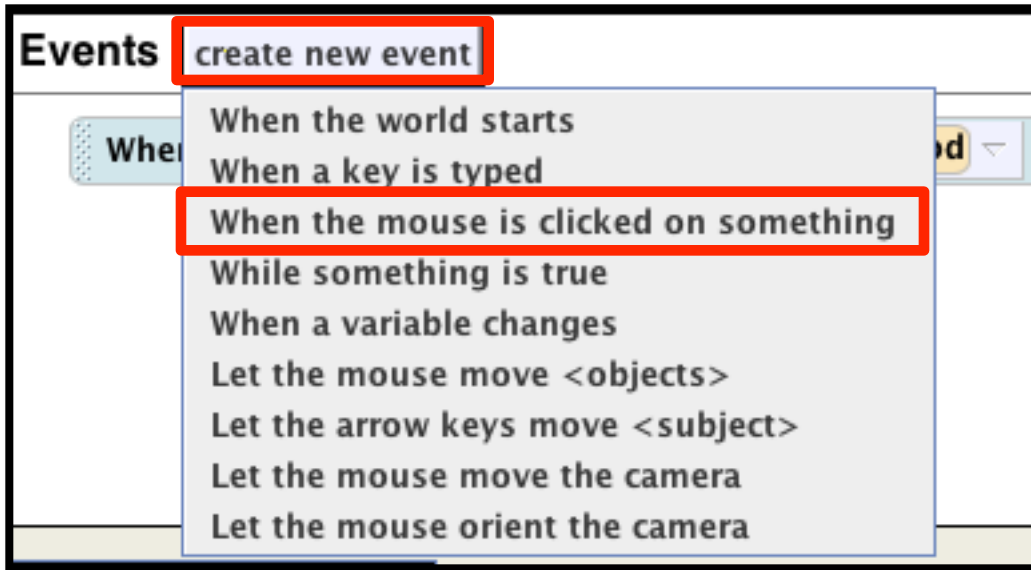
The screenshot shows a target selection menu with a list of objects. 'cinderella' is selected, and its sub-menu is open, showing 'the entire cinderella' as the selected target. Other options include 'the entire world', 'camera', 'light', 'ground', 'CameraViews', 'dragon', 'tower', 'horse', and 'knight'.

Drag in a **turn to face** method and select **cinderella, the entire cinderella**.

Then have the **horse move forward 15 meters**.

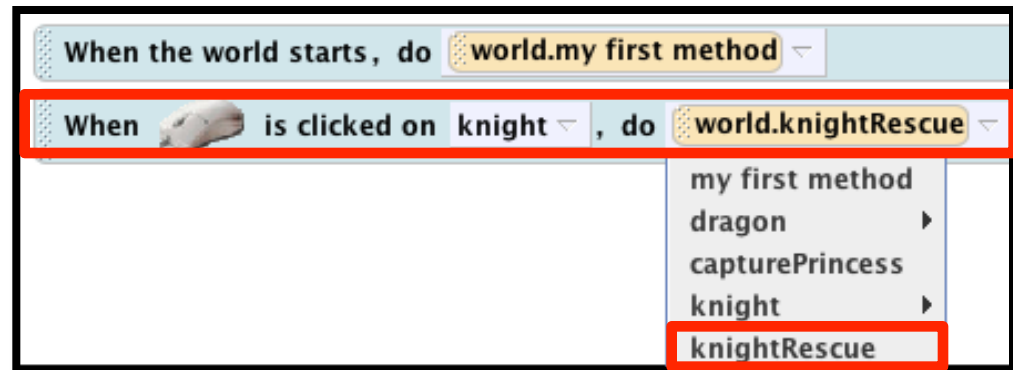
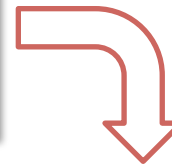
The screenshot shows the script area for 'world.knightRescue'. It contains three script blocks. The second block is 'horse turn to face cinderella' and the third block is 'horse move forward 15 meters', both highlighted with red boxes. The first block is 'knight say a damsel in distress!'.

Step 2: Mouse Click Event



To test the method, we will create a new type of event.

Click on **create new event** in the events editor and select **When the mouse is clicked on something**. Select **the entire knight** and **knightRescue** for the blanks.



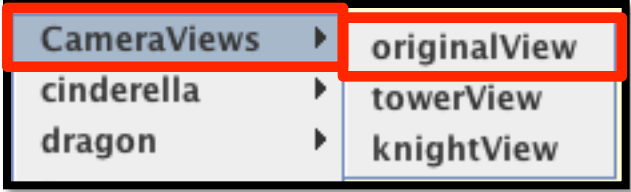
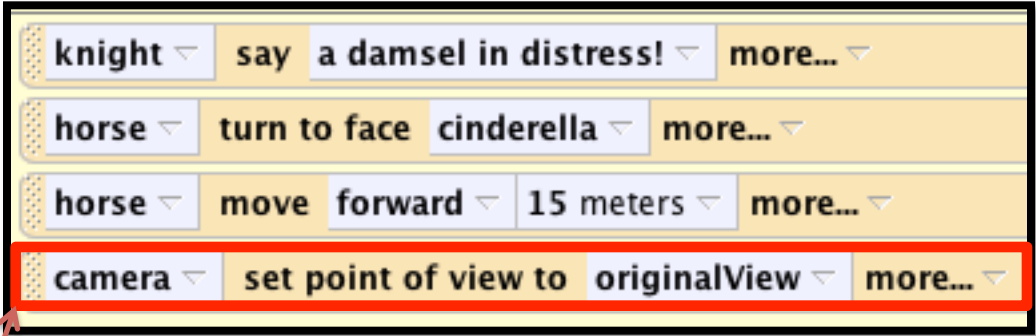
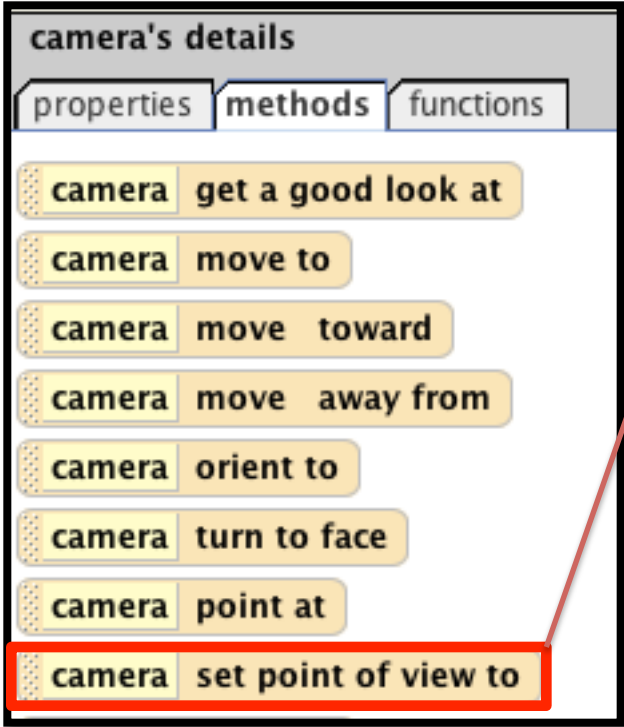
Click **Play** and once the camera moves to the **knight** click on him to see the method play.

Step 2: Animation Continued



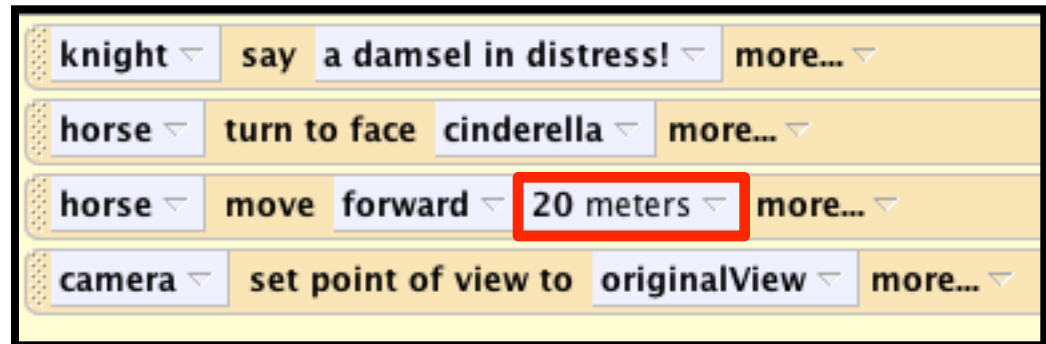
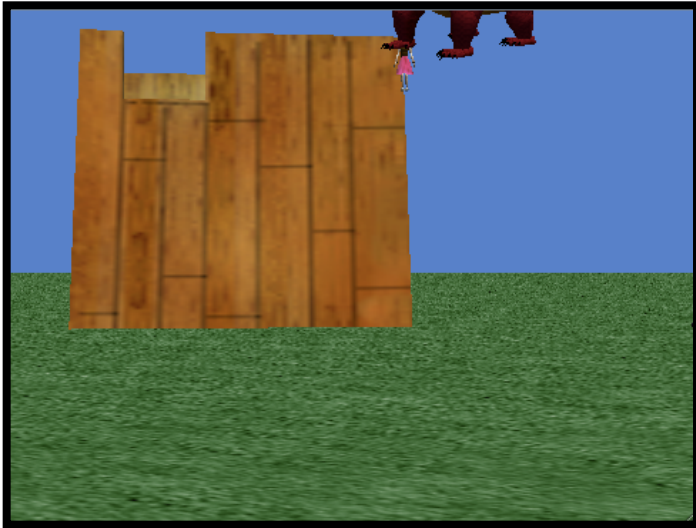
The animation looks fine so far but we want to make sure that the knight makes it to the tower. Let's change the camera view and check.

Click on camera in the object tree and drag in a set point of view to method.

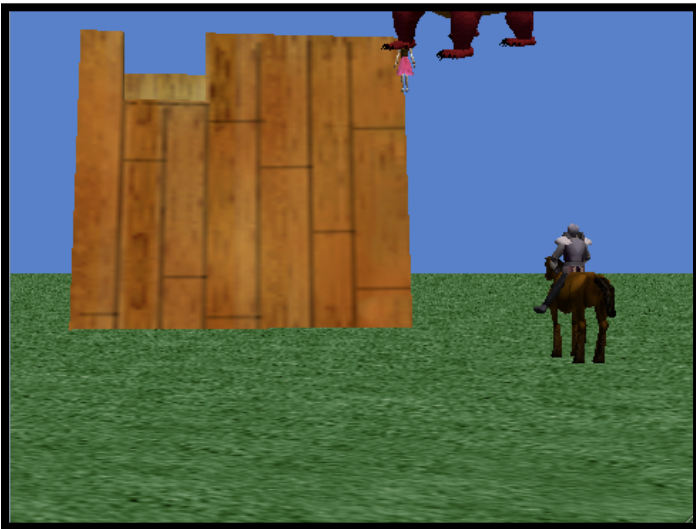


Select CameraViews, originalView. Now press Play.

Step 2: Testing



Depending on where you moved the camera when creating the new view, the horse and knight may be in the final shot of the animation.



If not, change the amount that the **horse moves forward** from **15** to **20** and test again, the knight and horse should be in view when the animation ends.



Step 3: New Camera View

We can also create moving camera scenes by tying the camera to an object. Drag in a **camera set point of view to method** between the two **horse** methods and select the **knight, upperBody, neck, head, helmet, faceGaurd**.

camera's details
properties | methods | functions

- camera get a good look at
- camera move to
- camera move toward
- camera move away from
- camera orient to
- camera turn to face
- camera point at
- camera set point of view to**

```
knight < say a damsel in distress! < more... <  
horse < turn to face cinderella < more... <  
camera < set point of view to knight.upperBody.neck.head  
horse < move forward < 20 meters < more... <  
camera < set point of view to originalView < more... <
```

world.knightRescue No parameters

No variables

- asSeenBy
 - the entire world
 - camera
 - light
 - ground
 - CameraViews
 - cinderella
 - dragon
 - tower
 - horse
 - knight**
 - rightLeg
 - leftLeg
 - leftLegGuard
 - rightLegGuard
 - upperBody**
 - the entire upperBody
 - breastplate
 - rightArm
 - leftArm
 - neck**
 - the entire neck
 - head**
 - the entire head
 - helmet**
 - the entire helmet
 - faceGuard**

Step 3: New Camera View Continued

camera's details

properties methods functions

create new variable

capture pose

vehicle world

pointOfView = position: -4.26,

knight say a damsel in distress! more...

horse turn to face cinderella more...

camera set point of view to knight.upperBody.neck.head more...

camera set vehicle to knight more...

horse move forward 20 meters more...

camera set point of view to originalView more...

camera set vehicle to world more...

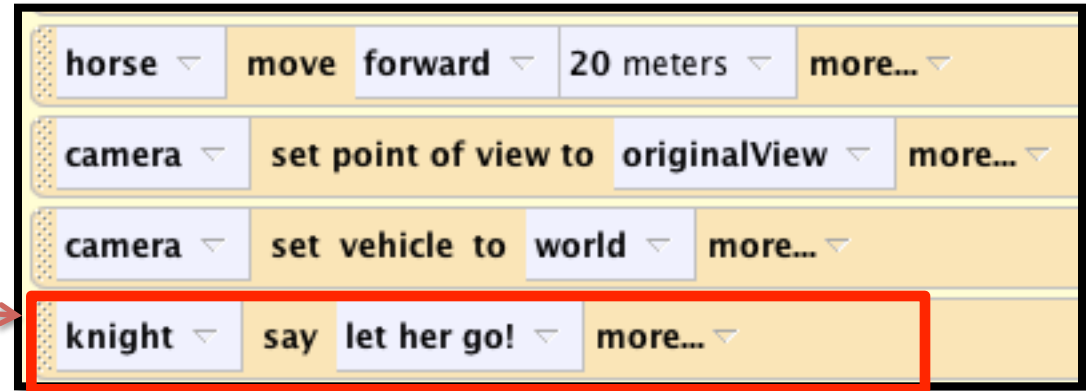
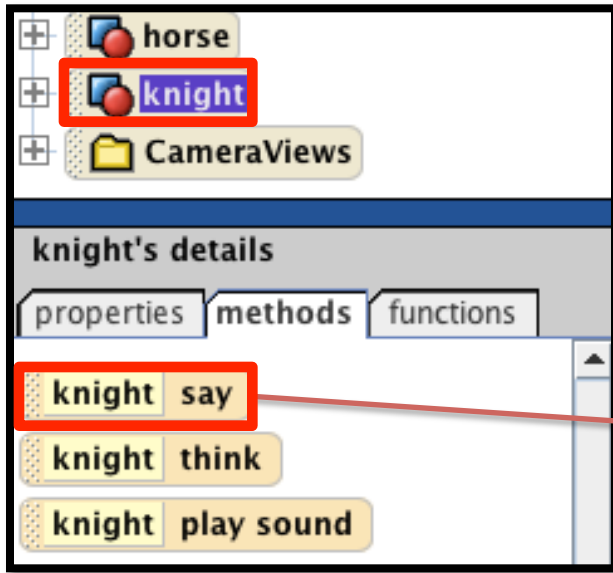
Once again the vehicle property will be used to tie the camera to the knight as he is moving.

Drag the camera vehicle property into the code after the set point of view to knight's faceguard command and select the entire knight. Drag in another one after set point of view to originalView and set the vehicle to world.

Press play to see what the knight sees!

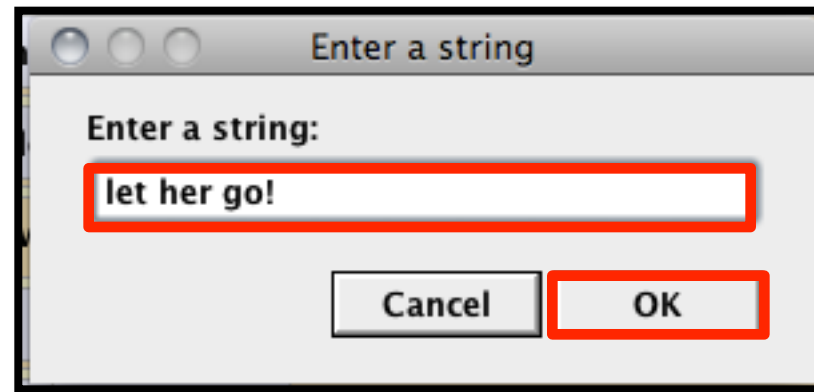


Step 3: Animation



Click on the **knight** in the object tree and scroll down until you find the **knight say** method.

Drag it into the method and have him say **“let her go!”**



Step 3: Animation



Now let's finish the animation and rescue the princess.

Click on **dragon** in the object tree and find the list of **methods**.



Drag in the **turn to face method** and select **the entire knight**.

Drag in a **move forward** command and for now select **1 meter**.

Step 3: Functions

dragon's details

properties | methods | **functions**

- dragon distance above
- dragon distance below
- dragon distance in front of**
- dragon distance behind

dragon turn to face knight more...

dragon move forward 1 meter more...

object

- camera
- light
- ground
- CameraViews ▶
- cinderella ▶
- dragon ▶
- tower
- horse ▶
- knight ▶**

- the entire knight**
- upperBody ▶
- rightLeg ▶
- leftLeg ▶
- leftLegGuard
- rightLegGuard

Since we do not know how far the dragon is from the knight we will use a function.

A function in Alice does not change the world like a method but it gives us information about the world like the distances between two objects.

Click on the **functions** tab in the details pane. Find the **dragon distance in front of** function under the **proximity** functions.

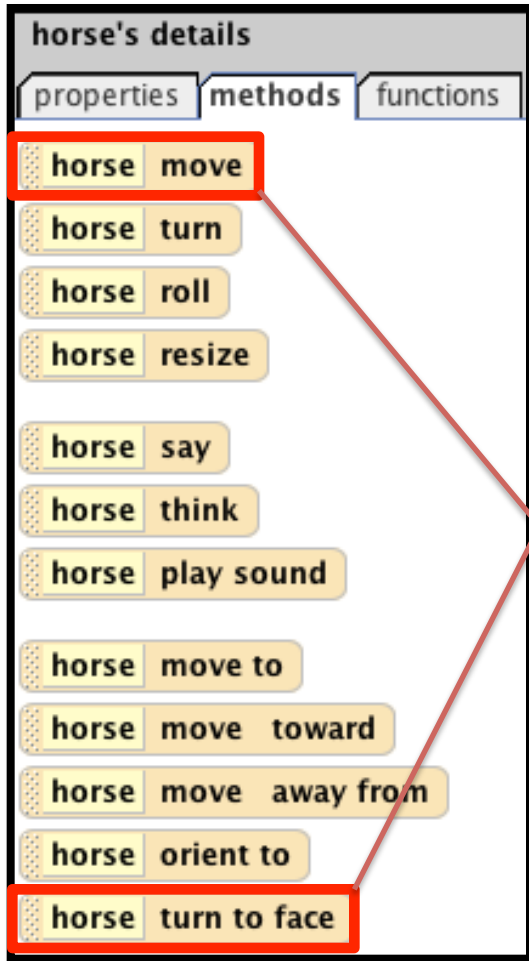


dragon move forward

dragon distance in front of knight more...

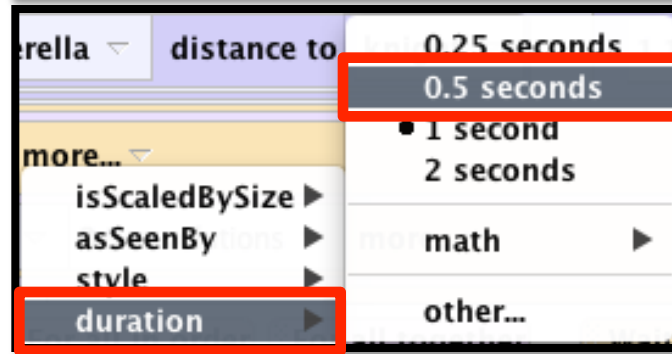
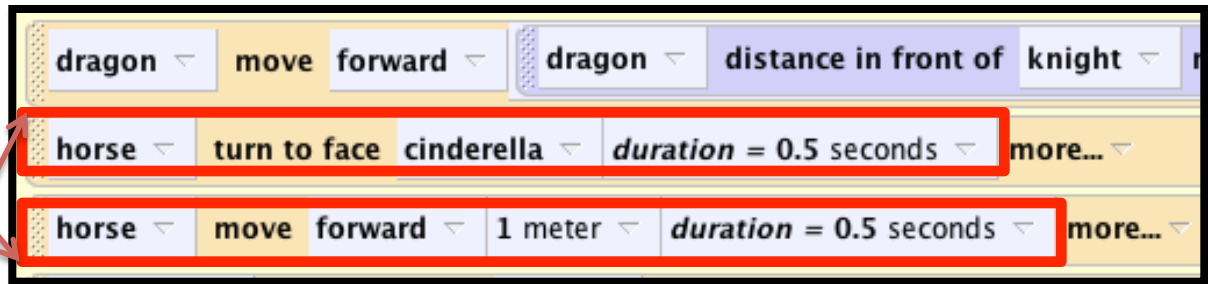
Drag it into the code over the **1 meter** until there is a green box. Drop and select **knight, the entire knight**.

Step 3: Animation Continued



Click on **horse** in the object tree and drag in a **turn to face** command. Select **the entire Cinderella**.

Following that, drag in a **horse move forward** command and select **1 meter**.



Set the **duration** of both commands to **0.5 seconds** by clicking on **more...** at the end of each line.

We want the horse to bring the knight to Cinderella, so we will use a function since we do not know the exact distance.

Step 3: Function Continued

The screenshot shows a software interface with an object tree on the left. The 'knight' object is selected and highlighted with a red box. Below it, the 'CameraViews' folder is visible. The 'knight's details' panel is open, showing three tabs: 'properties', 'methods', and 'functions'. The 'functions' tab is selected and highlighted with a red box. It contains four function entries: 'knight distance above', 'knight distance below', 'knight distance in front of', and 'knight distance behind'. The 'knight distance in front of' entry is highlighted with a red box. A red arrow points from this entry to the '1 meter' dropdown in the block below.

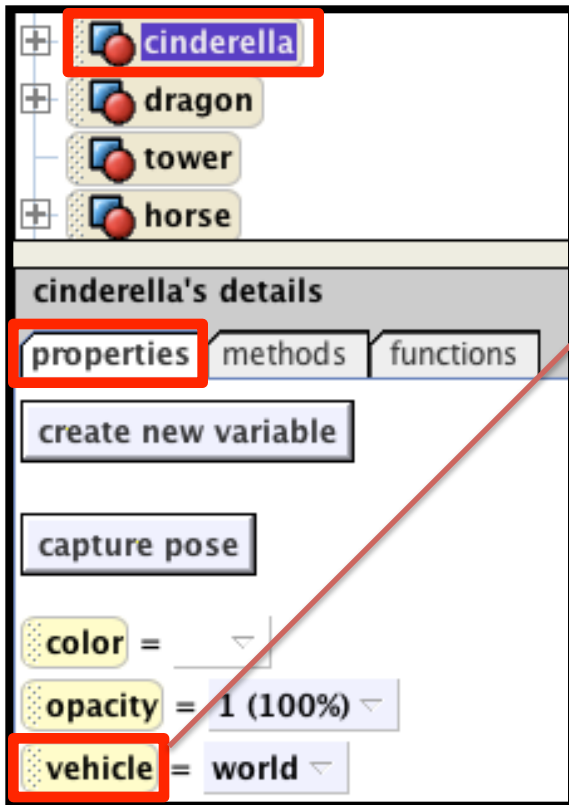
Click on **knight** in the object tree and then find his list of **functions**. Drag the **knight distance in front of** function over the **1 meter** and select the entire **Cinderella**.

The screenshot shows a block with several parameters: 'horse', 'move forward', '1 meter', 'duration = 0.5 seconds', and 'more...'. The '1 meter' dropdown is highlighted with a green box. A red arrow points from the 'knight distance in front of' function in the object tree to this dropdown.



The screenshot shows a block with parameters: 'move forward', 'knight', 'distance in front of', 'cinderella', and 'more...'. The 'knight', 'distance in front of', and 'cinderella' dropdowns are highlighted with a red box, indicating the result of the drag-and-drop action.

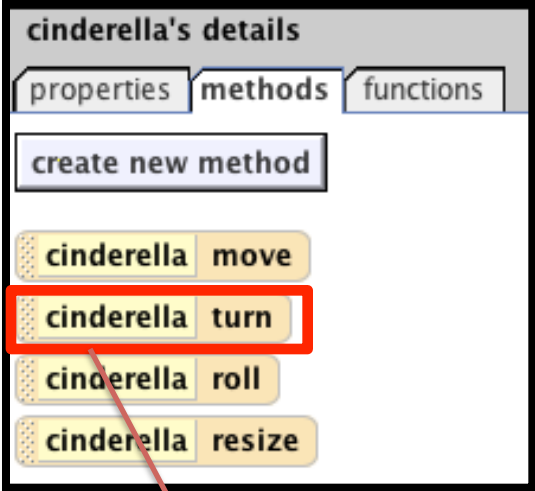
Step 3: Animation Continued



Drag in Cinderella's vehicle property into the method and set it to the entire world. This will 'unglue' her so that when the dragon moves, she will no longer move with it.

Now drag in a Do together. We will now drag in commands to make Cinderella fall into the arms of the knight and the dragon fly away.

Step 3: Animation Continued

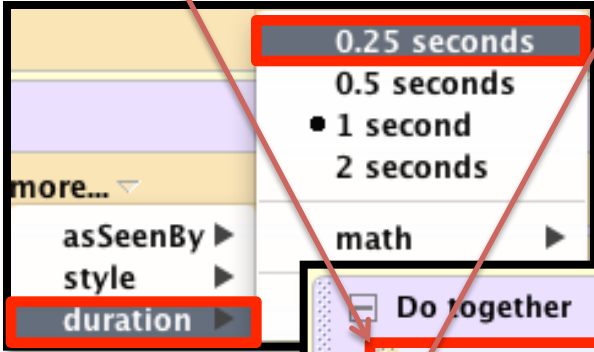


A screenshot of the 'cinderella's details' menu. The 'methods' tab is selected. A 'create new method' button is at the top. Below it are several method blocks: 'cinderella move', 'cinderella turn' (highlighted with a red box), 'cinderella roll', and 'cinderella resize'.



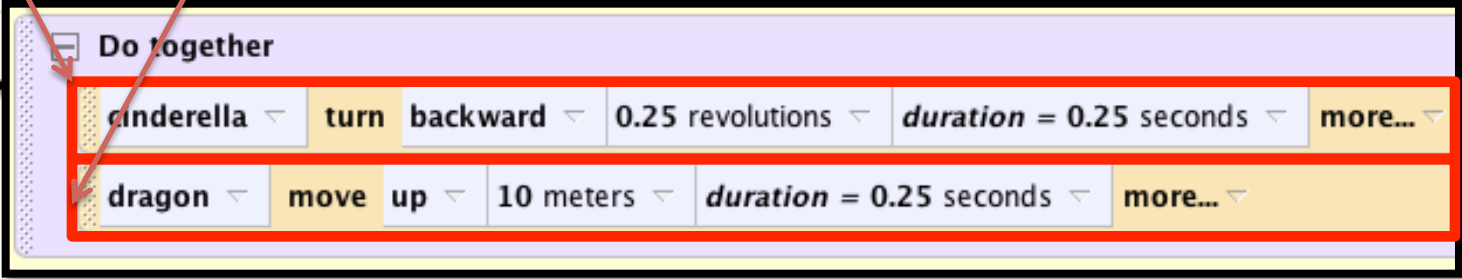
A screenshot of the 'dragon's details' menu. The 'methods' tab is selected. A 'create new method' button is at the top. Below it are several method blocks: 'dragon move' (highlighted with a red box), 'dragon turn', 'dragon roll', and 'dragon resize'.

Drag two lines inside the **Do together**.
One line to have **Cinderella turn backward ¼ revolution**.
Another line to have the **dragon move up 10 meters** (remember to click on the **dragon** in the object tree for a list of its methods).



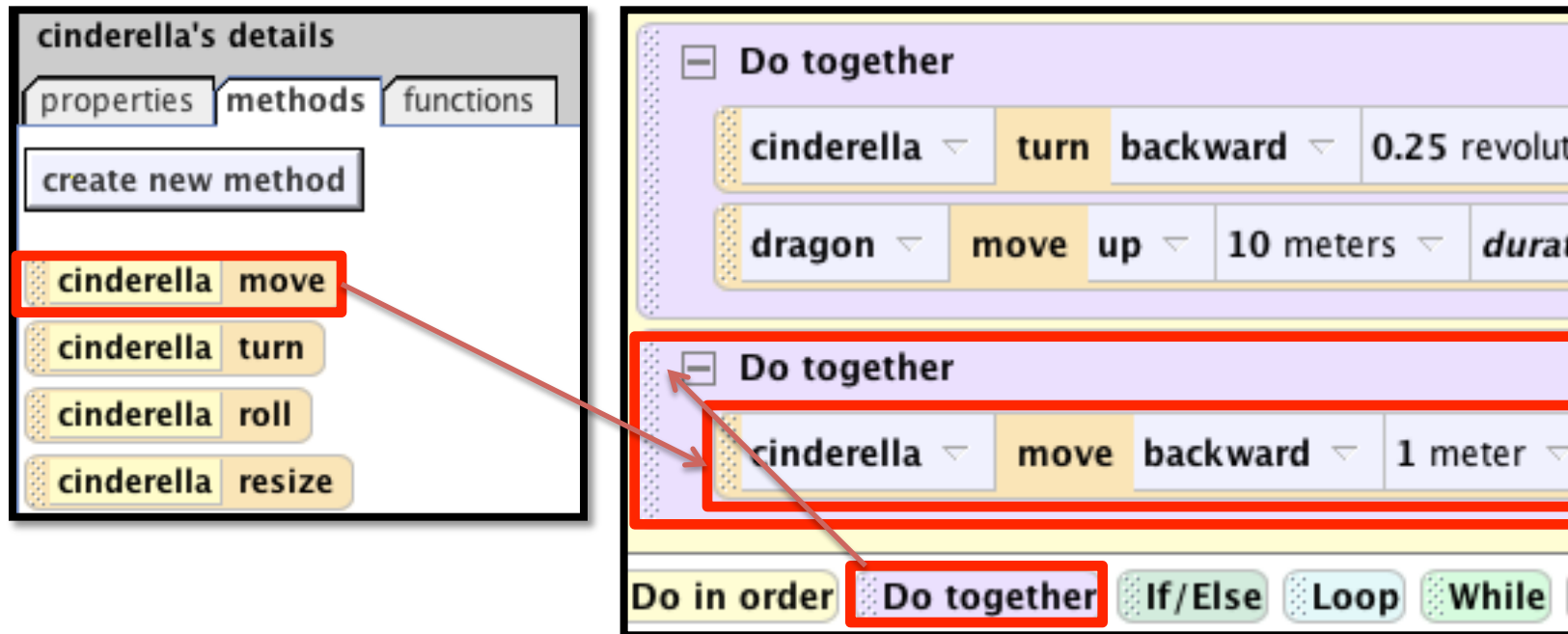
A screenshot of a dropdown menu for setting duration. The 'duration' option is highlighted with a red box. The menu items are: '0.25 seconds' (highlighted with a red box), '0.5 seconds', '1 second', '2 seconds', 'math', 'asSeenBy', 'style', and 'duration'.

Click on the **more...** after both commands and Set the **duration** for **0.25 seconds**.



A screenshot of a 'Do together' block. It contains two lines of code. The first line is 'cinderella turn backward 0.25 revolutions duration = 0.25 seconds more...'. The second line is 'dragon move up 10 meters duration = 0.25 seconds more...'. The entire block is highlighted with a red border.

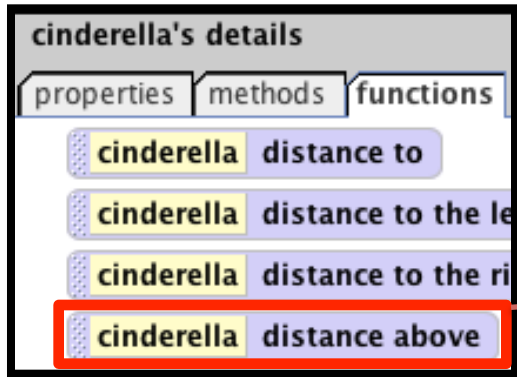
Step 3: Animation Continued



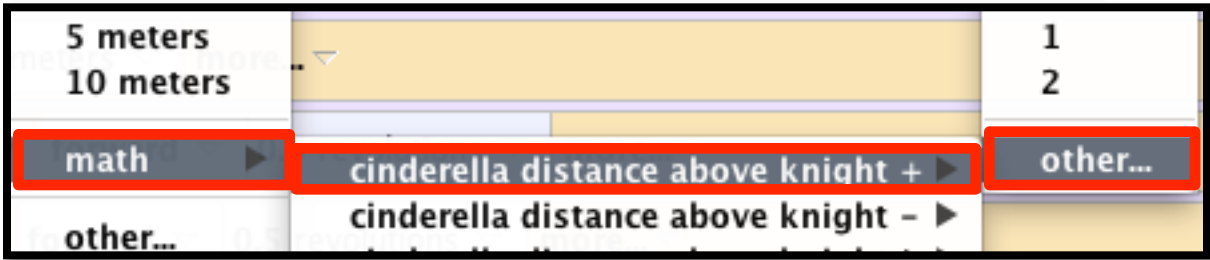
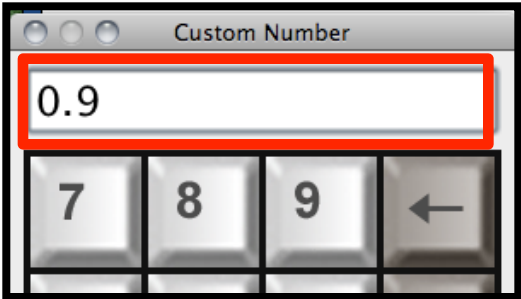
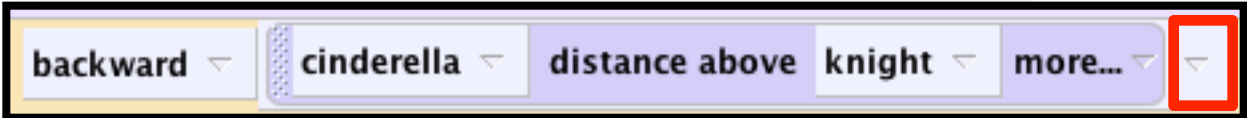
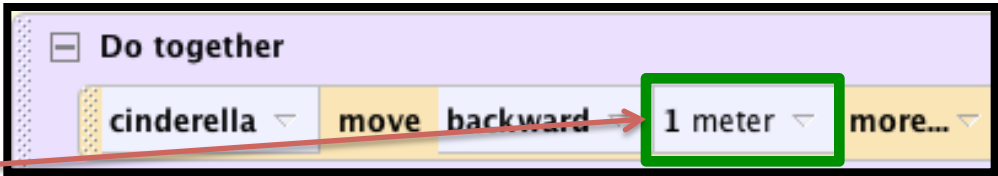
Play the world at this point. You will see that Cinderella is left on her back in the air. Because animation happens relative to the object we need to move her backward to make her fall.

Drag in another **Do together** and into that, drag a **Cinderella move backward** command (remember to click **Cinderella** in the object tree to get a list of her methods). Set it right now for **1 meter**. We will use another function to determine how far she will fall.

Step 3: Function Continued

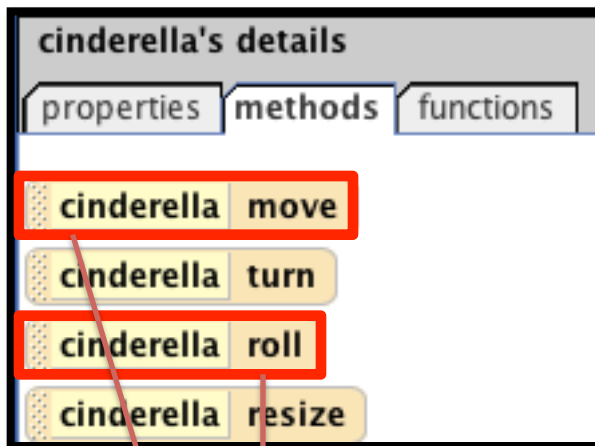


Drag in Cinderella's distance above function onto the 1 meter in the move backward command.



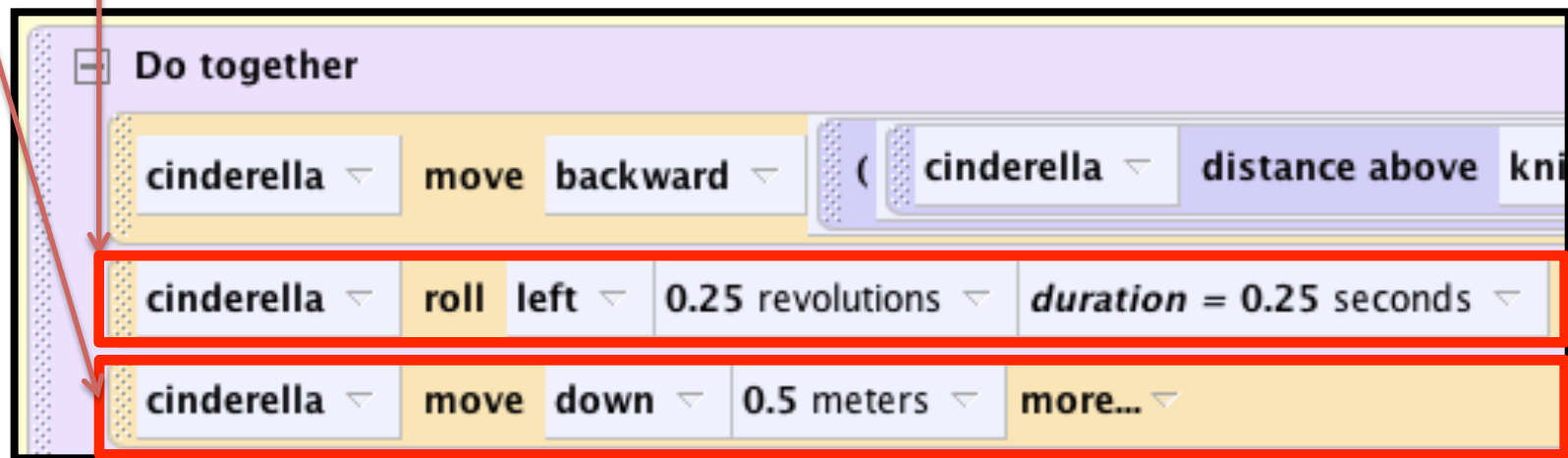
Modify the amount she falls by clicking on the arrow following the Cinderella distance above knight function. Select math, Cinderella distance above knight + and select other. Punch in 0.9.

Step 3: Animation Continued



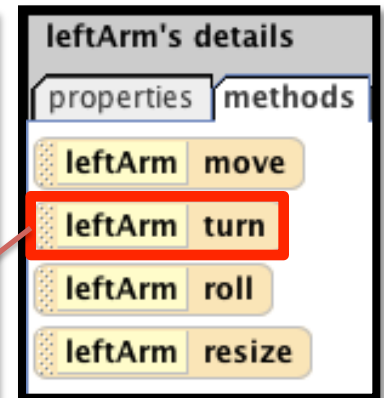
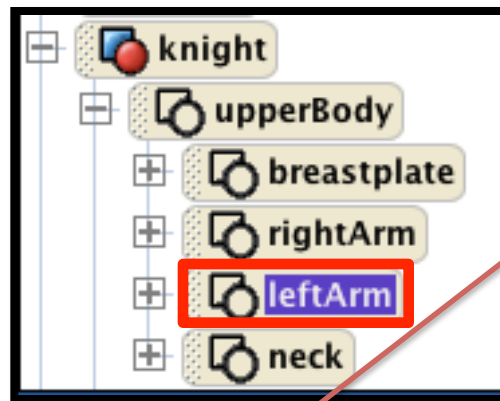
Drag in a **Cinderella roll left** command set it for $\frac{1}{4}$ revolution and duration 0.25 seconds.

Drag in another **Cinderella move** command for **down $\frac{1}{2}$ meter**.



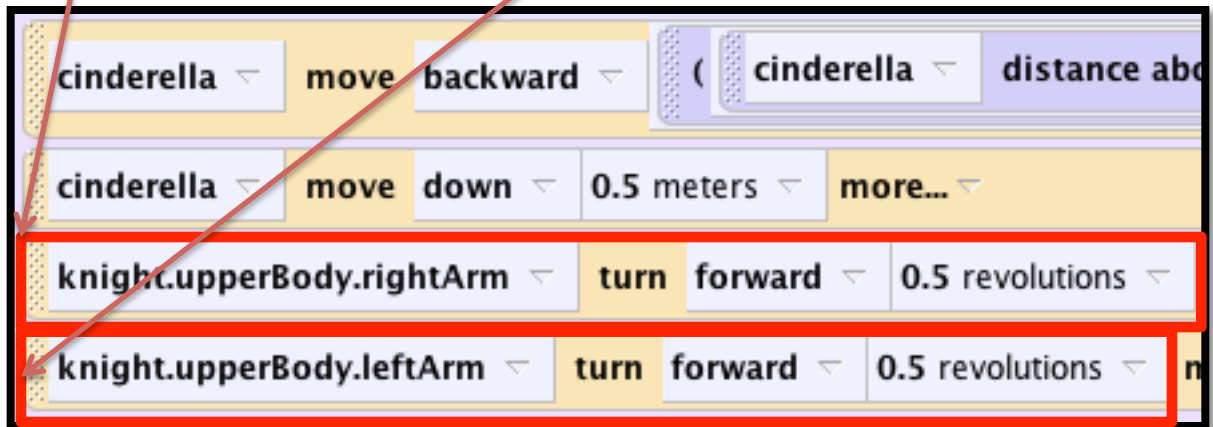
The next step of the animation is to raise the knight's arms to catch her.

Step 3: Animation Continued



First find the arms of the **knight** under his **upperBody** in the object tree.

For each arm, drag in a **turn forward** command, select $\frac{1}{2}$ revolution.



Press **Play** to see the final animation!

Final Code

```
knight say a damsel in distress! more...
horse turn to face cinderella more...
camera set point of view to knight.upperBody.neck.head.helmet.faceGuard more...
camera set vehicle to knight more...
horse move forward 20 meters more...
camera set point of view to originalView more...
camera set vehicle to world more...
knight say let her go! more...
dragon turn to face knight more...
dragon move forward dragon distance in front of knight more...
horse turn to face cinderella duration = 0.5 seconds more...
```

This slide and the next slide contain the final code for `world.knightRescue`.

Congratulations!

The screenshot displays a sequence of animation events in a game engine interface. The events are as follows:

- horse** → **move forward** → **knight** → **distance in front of cinderella** → **more...** → **duration = 0.5 seconds**
- cinderella** → **set vehicle to world** → **more...**
- Do together** (collapse icon):
 - cinderella** → **turn backward** → **0.25 revolutions** → **duration = 0.25 seconds** → **more...**
 - dragon** → **move up** → **10 meters** → **duration = 0.25 seconds** → **more...**
- Do together** (collapse icon):
 - cinderella** → **move backward** → **(cinderella distance above knight more... + 0.9)** → **more...**
 - cinderella** → **roll left** → **0.25 revolutions** → **duration = 0.25 seconds** → **more...**
 - cinderella** → **move down** → **0.5 meters** → **more...**
 - knight.upperBody.rightArm** → **turn forward** → **0.5 revolutions** → **more...**
 - knight.upperBody.leftArm** → **turn forward** → **0.5 revolutions** → **more...**

Congratulations on finishing Part 3! Part 4 will teach you how to embellish your animations so that they are more Engaging. We will use billboards, sound, and 3D text to make this world come to life!

