

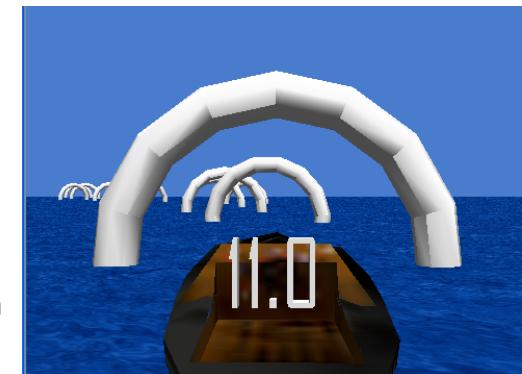
## Making a Boat Racing Game in Alice



Tutorial for Chapter: 5

## Making a Boat Racing Game in Alice

This tutorial will teach you to make a boat racing game in Alice, in which you steer a boat with the arrow keys through several arches, trying to beat the clock to the finish. You must drive through each arch before you get to the finish line, and a score will keep track of how many arches you've driven through.



## Setting Up the Scene

This tutorial will teach you how to make a boat racing game in Alice. First, open up a **water** template world in Alice. Then, from the vehicles folder in the Object Gallery, add a **motorboat** object to your world. After you've positioned your boat, drop a dummy object at your camera and rename it **originalView**, to save the camera view.



## Setting Up the Scene

Using the object positioning buttons, turn your boat so that the front of it is pointing away from the camera, and move it closer to the camera until it looks something like this:



## Setting Up the Scene

Next, go into the Shapes folder in the object gallery and find the **Torus** object. Click and drag a torus into your world. Using the object positioning buttons, first resize the torus until it is pretty large, then turn it until half of it is underwater. Then move it so that it is a little in front of your boat, as if your boat is about to drive through it.



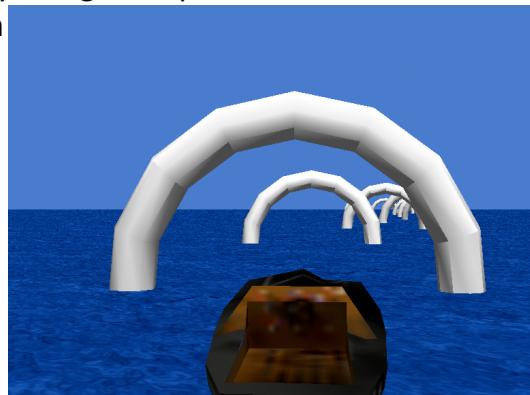
## Setting Up the Scene

Now, using your purple camera moving arrows, move your camera forward for a few seconds, and then drag, drop, resize, and turn another torus.



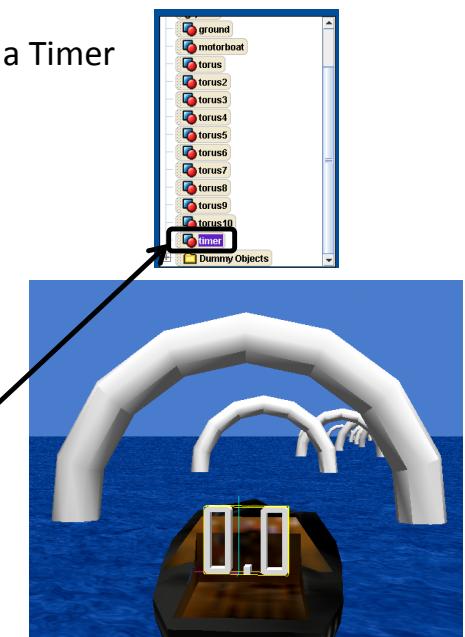
## Setting Up the Scene

Now repeat the last step until you have ten torus's in your world. The path that you move your camera along, and drop the torus's on, will be the racing path for your motorboat. You can make it perfectly straight, or put some twists and turns in it if you want. When you're done, move your camera back to originalView.



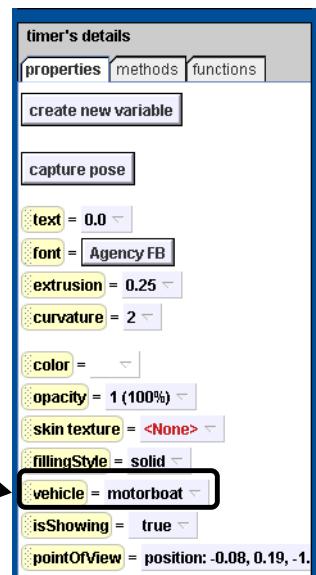
## Making a Timer

Now we'll add a timer to the world that will time the boat as it races through the arches. Add a 3D Text object from the object gallery into your world, and make it say **0.0**. Then rename it in your object tree to **timer**. Place it on top of your boat.



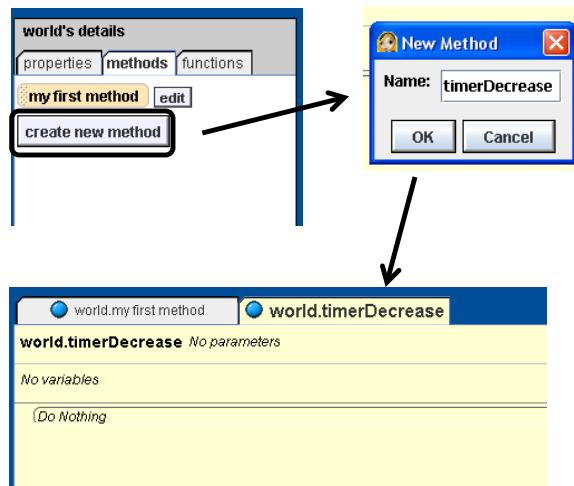
## Making a Timer

Since the boat will move through the world while it's racing, we want the timer to be able to move with it. So, go to the timer's properties tab and set its vehicle property to motorboat.



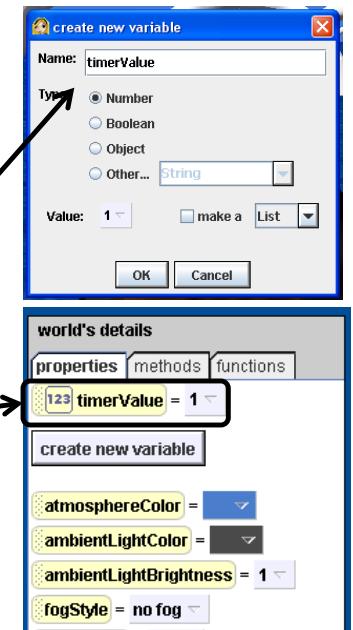
## Making a Timer

Now we need to create a method that will count down the amount of time that is left. Create a new world method called **timerDecrease**.



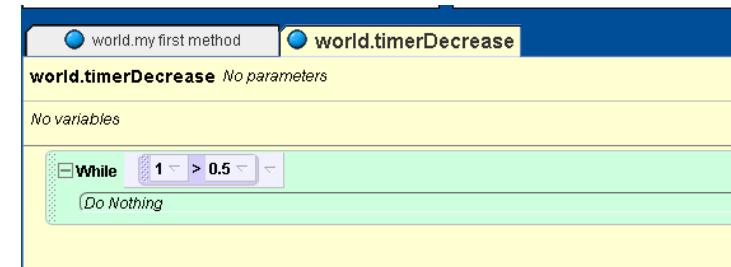
## Making a Timer

Now we need to write code to make our timer work. First, we need to create world variable to save the number value of the time that is left. Go to the world's properties tabs, and create a new variable, of the type number, and name it **timerValue**. It should be a world variable, because we will be using it in more than one place in Alice.



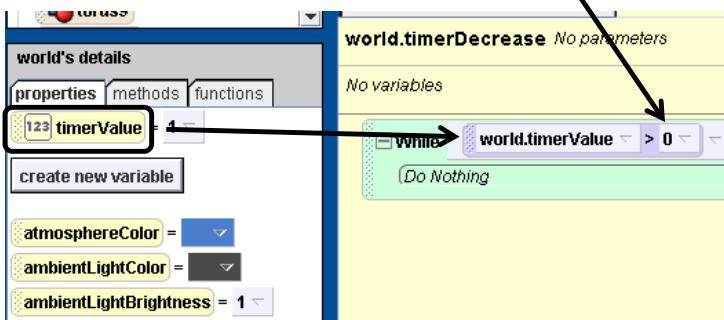
## Making a Timer

We want the timer to keep counting down as long as there is time left. To put this into code, get a **While** loop and drag it into the code, and select true. If **timerValue** is greater than zero, that means there's time left, so that will be the condition of the While loop. Go to the world's functions, get **a>b**, and drop it onto the **true** part of the While loop. Pick any two numbers from the drop down menu.



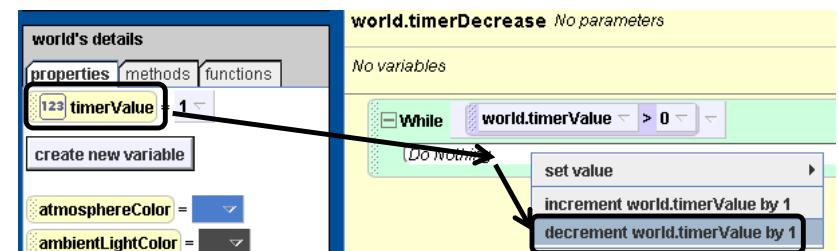
## Making a Timer

From the world properties pane, grab **timerValue** and drop it onto the first number at the top of the **While** loop. Then, change the second number to zero.



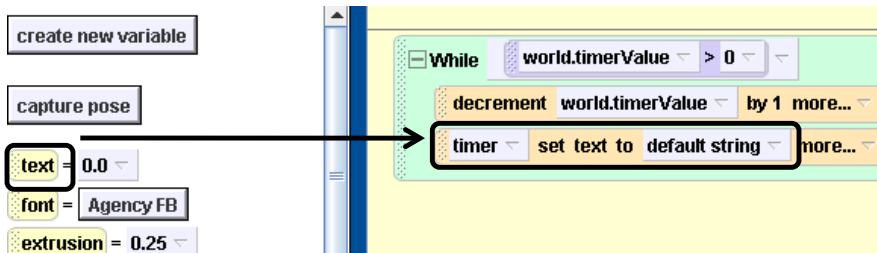
## Making a Timer

The next thing we want is for the timer to count down. To do this, we need **timerValue** to subtract one from its value. So, drag **timerValue** from the world's properties pane into the while loop. Then select **decrement world.timerValue by 1**.



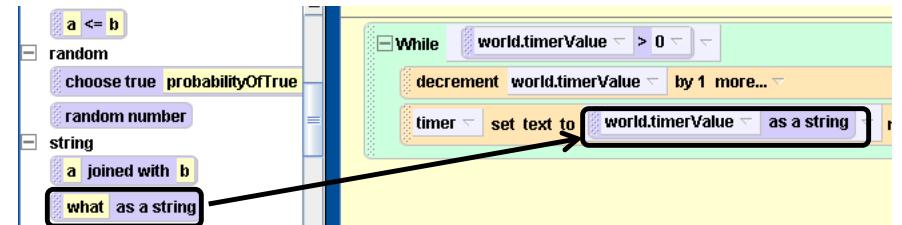
## Making a Timer

Now, we need to tie our 3D-Text to the **timerValue** variable, so that the 3D-Text shows how much time there is left. Click on **timer** in your object tree, and go to the properties tab. Click and drag **text** into your While loop and choose **default string**. Set the duration of this command to zero.



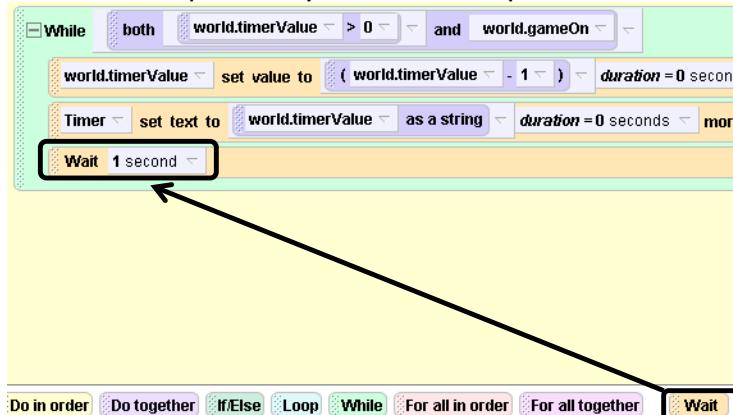
## Making a Timer

Now, click on **world** in the object tree and go to the **functions** tab. Scroll down to **what as a string**. Click and drag that onto **default string** in your code. In the drop down menu, select **timerValue**.



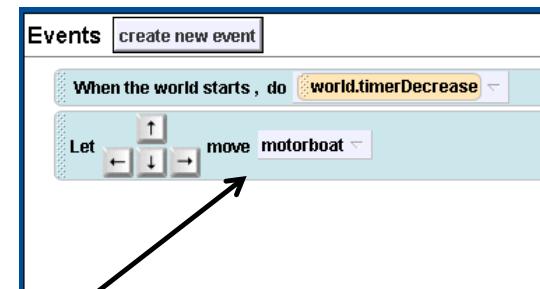
## Making a Timer

The last step to making our timer is to make sure that it takes exactly one second for the timer to count down one second. To do this, click and drag a **Wait** from the bottom of your method editor, and drop it into your **While** loop. Then select **1 second** in the drop down menu.



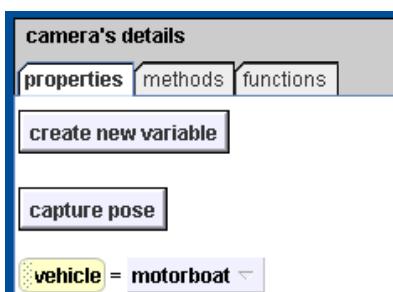
## Steering the Boat

So that we can test out our timer, we need to be able to steer our boat through the rings. To do this, we need an event. Click **create new event** in the event editor, and select **Let the arrow keys move**. Then change it from **camera** to **motorboat**.



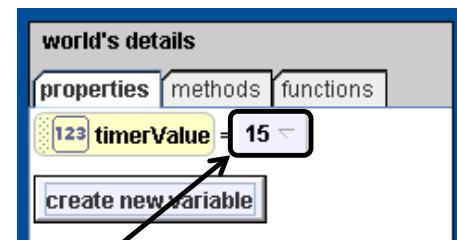
## Steering the Boat

The last thing you need before you test out the timer is to allow the camera to follow the boat. Click on **camera** in the object tree and go to its **Properties** tab. Then change its **vehicle** from **world** to **motorboat**.



## Making a Timer

Now you need to figure out how much time to put on your timer, so that your game is challenging. My timer counts down 15 seconds, and if I race through the arches perfectly, I can beat the clock. First, set your timer by changing **timerValue**'s value in the world properties tab.

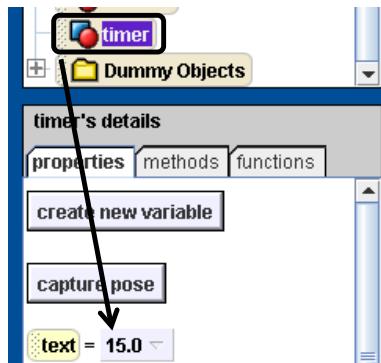


## Making a Timer

When you change the value of `timerValue`, you also have to change the beginning text of your timer to match that number. To do this, click on `timer` in the object tree and go to **properties**. Next to `text`, change it to the number that you have set `timerValue` to.

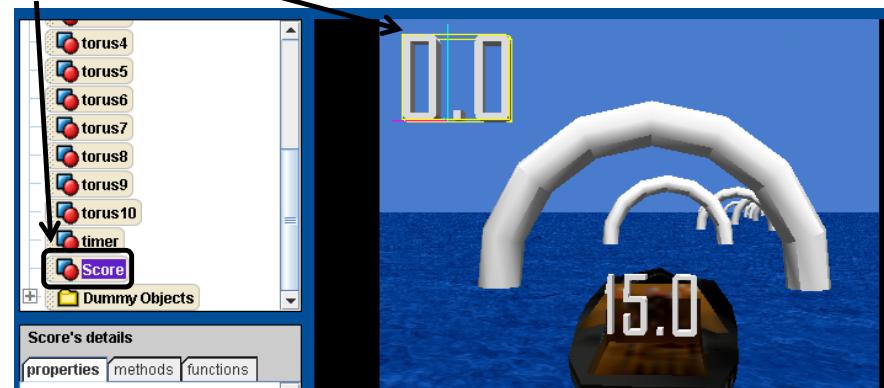
Change your `When the world starts` event to

`world.timerDecrease` and play your world to test it out. When you're done testing, change it back to `world.my first method`.



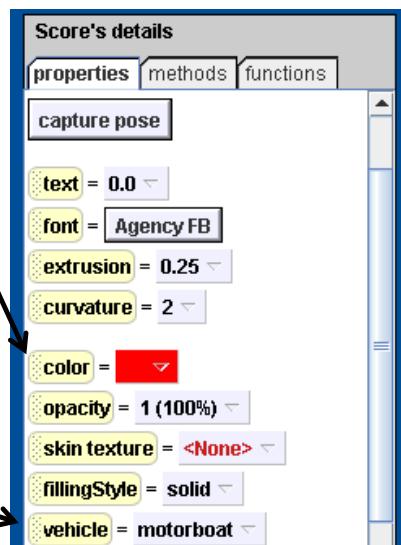
## Making a Score

Now we need a scorekeeper to keep track of whether you drive through all the rings. Add another 3D-Text object into your world that says `0.0`, and name it `score` in the object tree. Now, position it so that its in the upper left hand corner of your screen.



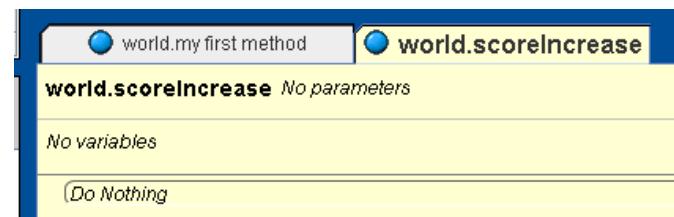
## Making a Score

Next, go to the score's **Properties** tab and go to `color`. Change the color to red so that the score shows up against the white arches. Then, set the score's `vehicle` to `motorboat` so that we can always see it, even when we move the motorboat.



## Making a Score

Now we need to create a method that will increase your score every time you drive the boat through an arch. Click on `world` in the object tree and create a new method called `scoreIncrease`.



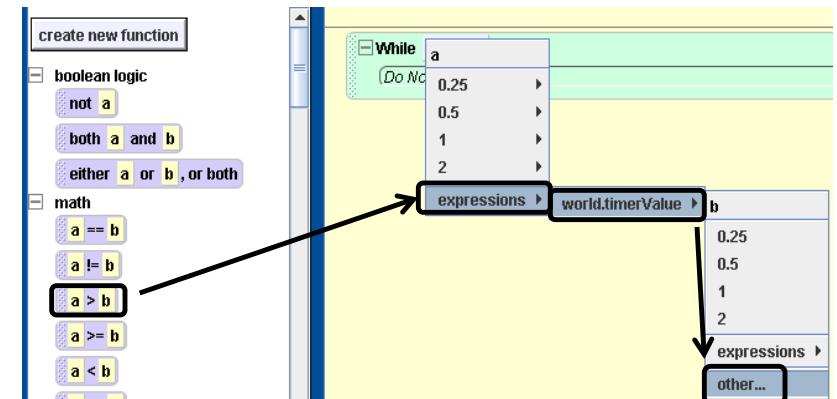
## Making a Score

We only want the score to increase while there is still time on the clock, so we will need a While loop. Drag and drop a While loop into your method. Select **true** on the drop down menu.



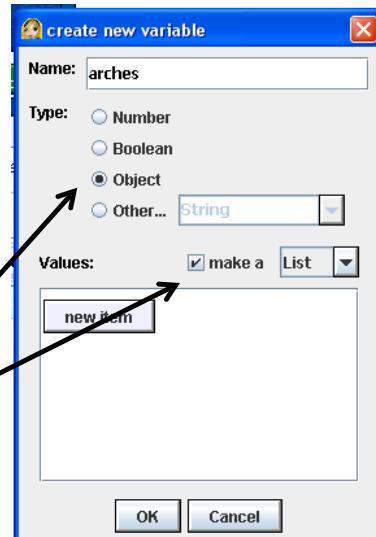
## Making a Score

To finish the while loop, go to the world functions tab and get **a > b**, and drop it on top of the **true** in the while loop. In the drop down menu select **expressions**, then **world.timerValue** then **other...**, and type in **0** into the calculator.



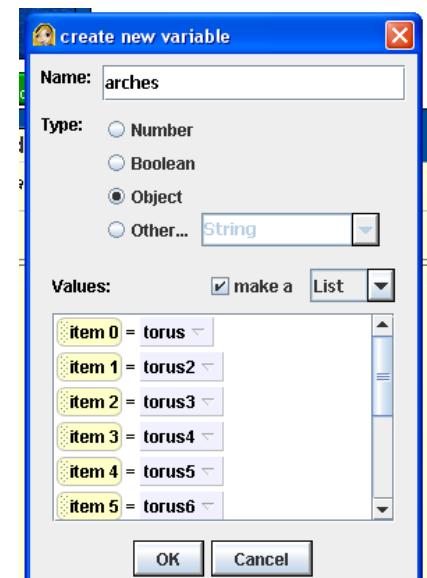
## Making a Score

We want this method to continuously be checking to see if the motorboat is inside any of the arches. To do this, we will need to make the arches into a list. Go to the world's properties tab and click **create new variable**. Select **Object**, then name it **arches**. Then click the **make a List** box.



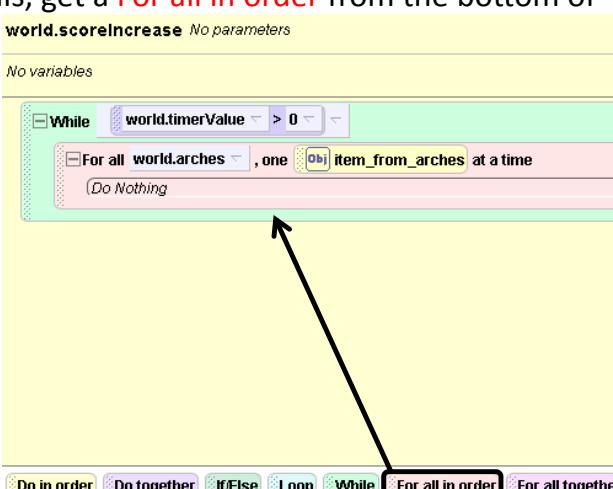
## Making a Score

Add each of the arches into the list. Since there are 10 arches, and the first arch will be item 0, your list should go up to item 9. So, click the **new item** button until your list goes up to item 9, and put each **torus** object into the list. When your list is all set up, click **OK**.



## Making a Score

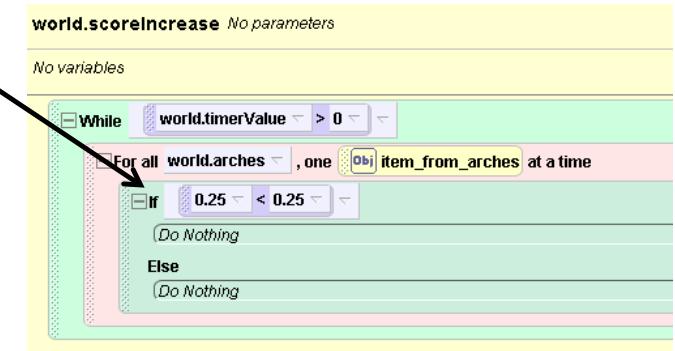
Now that we have a list, we can have this method constantly search through it to see if the motorboat is inside any of the arches. To do this, get a **For all in order** from the bottom of your method editor, and drop it into your while statement. Choose **expressions**, and **world**.**arches** in the drop down menu.



The screenshot shows a Scratch script editor. A green 'While' loop is active. Inside the loop, there is a light blue 'For all' loop. The 'For all' loop has 'world.arches' as its condition and 'one item\_from\_arches at a time' as its iteration. Inside the 'For all' loop, there is a light pink 'Do Nothing' block. At the bottom of the script editor, there is a toolbar with buttons: 'Do in order', 'Do together', 'If/Else', 'Loop', 'While', 'For all in order' (which is highlighted in black), and 'For all together'.

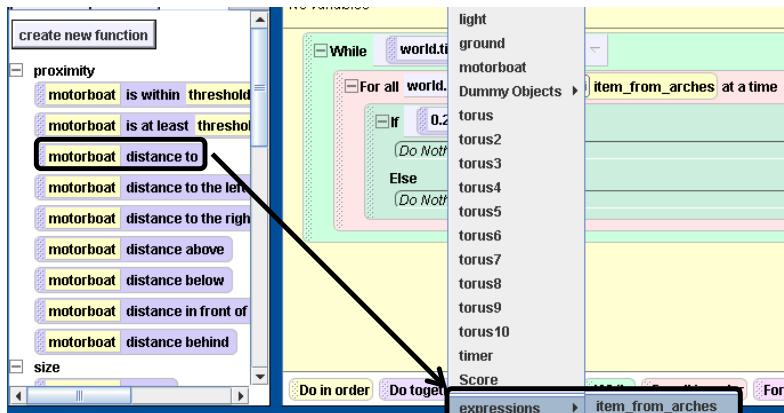
## Making a Score

Now we need to write in our code to add a point to the score if the motorboat is inside an arch. Get an **If/Else** from the bottom of your method editor, and drop it into the **For all in order**. Select **true**. Then go to the world's functions tab, and get an **a**<**b**, and drop it on top of the true. Choose any two numbers.



## Making a Score

Now, click on **motorboat** in the object tree and go to its **functions** tab. Get a **motorboat distance to**, and drop it on top of the first number in the **a**<**b**. Select **expressions**, and **item\_from\_arches**. Then, change the second number to **1.5**.



## Making a Score

Now, we want the score to add a point if it finds that you are inside an arch. To do this, click on **world** in the object tree and go to the **properties** tab. Click on **create new variable**, and name it **scoreValue**. Make sure **Number** is selected, then click **Ok**.



Now click and drag **scoreValue** into your If/ Else under If. Select increment world.scoreValue by 1.

Then, in the world's properties tab, set the value of **scoreValue** to 0.

### Making a Score

```

world.scoreIncrease No parameters
No variables

While [world.timerValue > 0]
  For all world.arches [one Obj item_from_arches] at a time
    If [motorboat distance to item_from_arches < 1.5]
      increment world.scoreValue by 1 more...
    Else
      (Do Nothing)
  End
End
  
```

**world's details**

- properties
- methods
- functions

123 timerValue = 15  
Obj arches = torus, torus2, tori  
123 scoreValue = 0

### Making a Score

Now, we need to set the text of the 3D Text object to the variable **scoreValue**. Click on **Score** in the object tree and go to its **properties** tab. Click on **text** and drag it under your **increment** command. Then select **default string**.

```

world.scoreIncrease No parameters
No variables

While [world.timerValue > 0]
  For all world.arches [one Obj item_from_arches] at a time
    If [motorboat distance to item_from_arches < 1.5]
      increment world.scoreValue by 1 more...
      Score set text to default string more...
    Else
      (Do Nothing)
  End
End
  
```

### Making a Score

Now, go to the world's functions tab and scroll down to **what as a string**. Click and drag that on top of **default string** in your code. Then select **expressions** and **world.scoreValue**. Finally, change the duration of that command to 0.

```

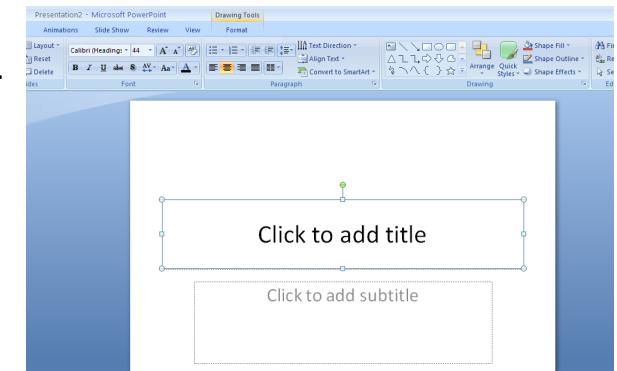
world's details
properties methods functions
  a >= b
  a < b
  a <= b
random
choose true probabilityOfTrue
random number
string
a joined with b
what as a string

world.scoreIncrease No parameters
No variables

While [world.timerValue > 0]
  For all world.arches [one Obj item_from_arches] at a time
    If [motorboat distance to item_from_arches < 1.5]
      increment world.scoreValue by 1 more...
      Score set text to world.scoreValue as a string duration = 0 seconds
    Else
      (Do Nothing)
  End
End
  
```

### Making Instructions

Now that we've written our score method, we're going to add in instructions into our world so that the user knows the rules of the game. To do this, you will need either Powerpoint, or Paint. I will use Powerpoint. Open up one of the two programs.

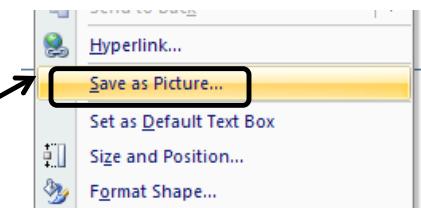


## Making Instructions

Insert a text box onto your Powerpoint slide and with a font size of about 30, type out these instructions:

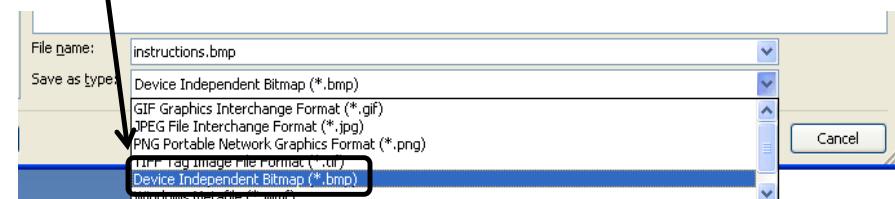
To win this game, you must steer the boat through each ring and beat the clock. You receive one point for each ring, and there are 10 rings, so if your score is less than 10 at the end, you lose!

Then, right click on the text box and select **Save as Picture**.



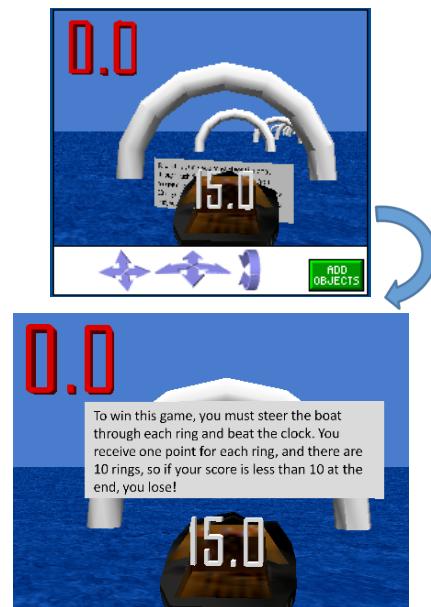
## Making Instructions

Now, name this picture **instructions**, and underneath that where it says **Save as type:**, choose **.bmp**. This is so the background of the instructions ends up being white, and not see-through, as it would with a **.png** type.



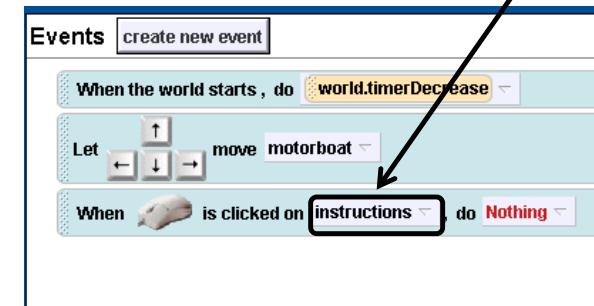
## Making Instructions

Now we will put these instructions into the Alice world. Go to **File** in your Alice world, and select **Make Billboard**. Find your **instructions**, click on it, and select **Import**. Now your instructions will appear in your world. Go to the **add objects** screen so that you can use the object positioning buttons to move the instructions over your boat, and closer to the camera.



## Making Instructions

Now what we will do is make it so that when the instructions are clicked on, the game will start. We will need an event for this, so leave the **Add Objects** area and click **create new event**. Choose **When the mouse is clicked on something**. Change the first anything to **instructions**.



## Making Instructions

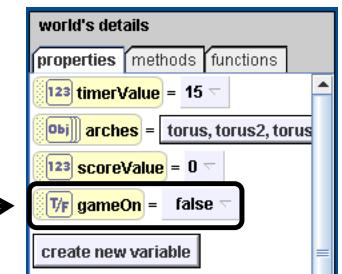
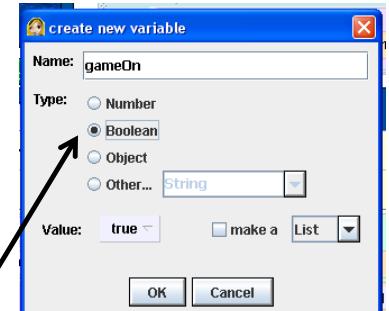
Now, get a **Do together** from the bottom of your method editor and drag and drop it onto the second **anything**. The first thing we want in there is for the instructions to disappear so the user can see the whole screen. Go to the **instructions' properties** tab, and click and drag **isShowing** to the **Do together**. Select **false** when you drop it. Then, change the duration on that command to 0.



## Starting the Game

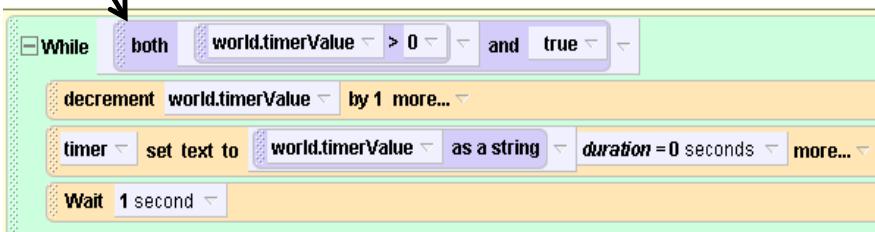
Now, we need to make sure that the timer counts down when the game starts, and stops if you get to the finish before time runs out. To do this, we will create another variable. Go to the world's properties tabs and click **create new variable**.

Name it **gameOn** and select **Boolean**. Then, set its value to **false**.



## Starting the Game

Now we need to go back into the **timerDecrease** method to tell the timer only to count down when **world.gameOn** is true. Go into that method, and then go to the world's functions tab. Drag and drop a **both a and b** on top of the **world.timerValue > 0**. Select **true**.

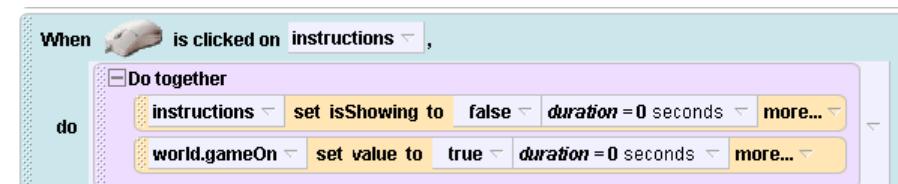


## Starting the Game

Now, click on **true** and select **expressions**, and **world.gameOn**.

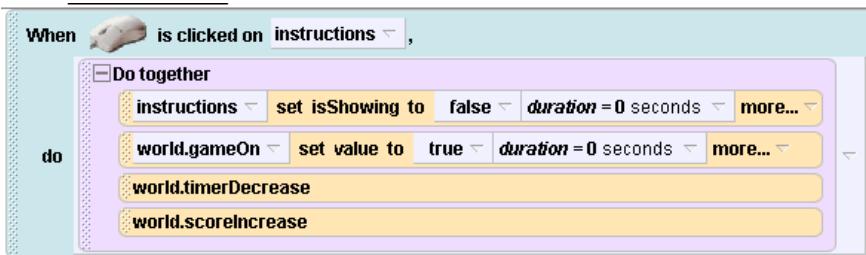


Now, we need to make it so that when the instructions are clicked on, **gameOn** turns to true so that the timer will start counting. To do this, click and drag **world.gameOn** from the world's properties tab, and drop it into the **Do together** under the **When mouse is clicked** event. Also, change the duration of that command to 0.



## Starting the Game

Now we need the `scoreIncrease` and the `timeDecrease` methods to start when the instructions disappear. So, drag both of those from the world's methods tab up into the event's **Do together**. Now we've got everything in place to actually play the game. Test your world to make sure everything works.



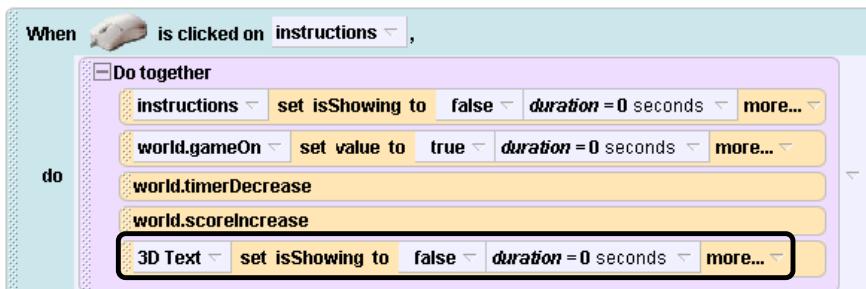
## Starting the Game

One last thing to add to your world is an instruction to the user to click on the instructions when they are ready to play. For this, add in a 3D Text object that says **Click Instructions To Start**. Position it above your motorboat.



## Starting the Game

Now, go to the new 3D Text object properties pane, and click and drag `isShowing` into the **Do together** under the **When mouse is clicked on** event. Select `false`, and change the duration to 0. Now, it will also disappear when the instructions are clicked on.



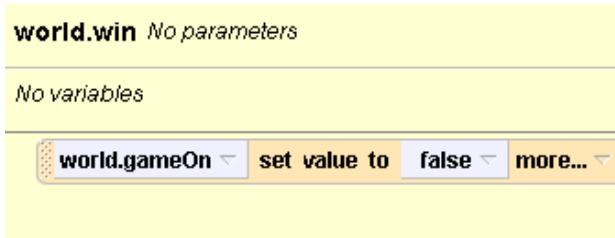
## When the Game Ends

Now we need to write some methods for what to do when the game ends. We'll need to write a winning method, and a losing method. They should both be very short. Let's start with the winning method. Create a new world method and name it `win`.



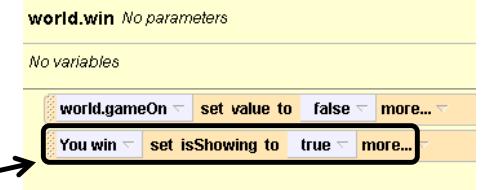
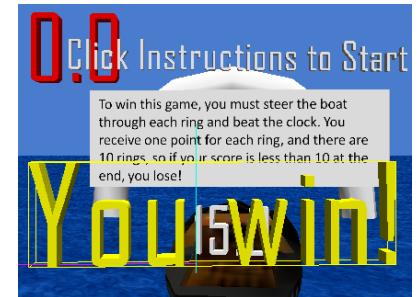
## When the Game Ends

The first thing to do in this method is set `world.gameOn` to false so that the timer stops counting down. To do that, go to the world's properties tab, and click and drag `world.gameOn` into the method. Set its value to **false**.



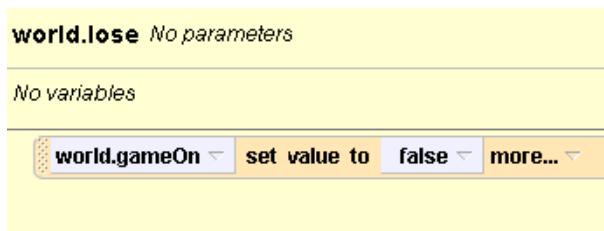
## When the Game Ends

Then, we will add a text object into the world that says **You win!**. Place it on the motorboat, and set its vehicle to **motorboat**, change its color to something other than white, and change its name to **You Win** in the object tree. Then, set its `isShowing` to **false** so that it's invisible. After that, click and drag its `isShowing` into the `win` method, and select **true**.



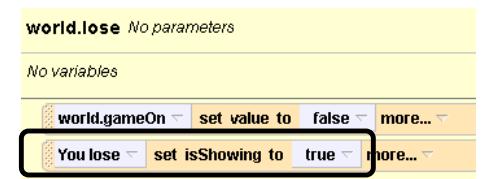
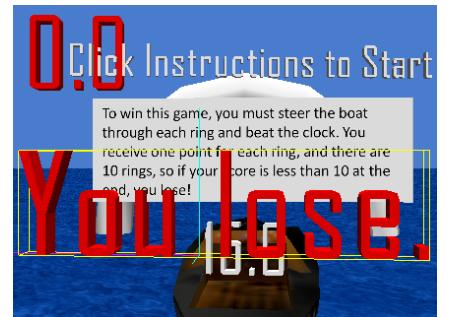
## When the Game Ends

Now we will do the exact same thing for a `world.lose` method. Create the method, and name it **lose**. Then go to the world's properties pane and click and drag `world.gameOn` into the method. Then, set it to false.



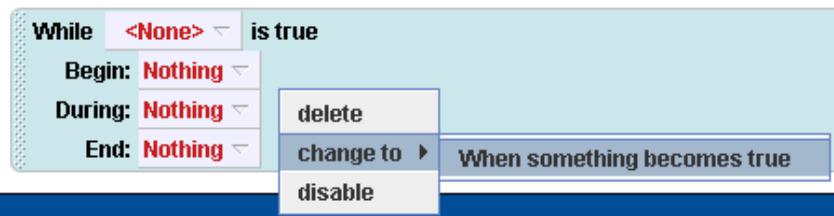
## When the Game Ends

Now we need to add a 3D Text object that says **You Lose**. Add the object, position it on the boat, change its color, and name it **You Lose** in the object tree. Set its vehicle to **motorboat**, and set its `isShowing` to **false** so that it is invisible, and then click and drag its `isShowing` into the `lose` method, and select **true**. Now we have methods for when the game ends, win or lose.



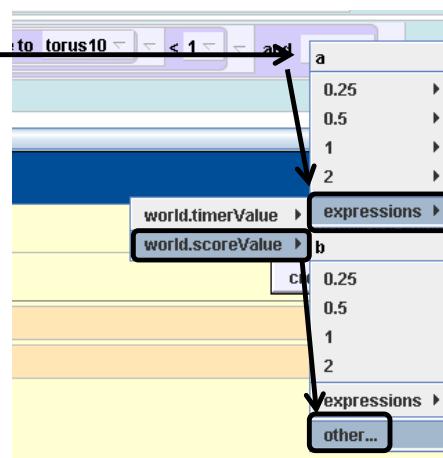
## When the Game Ends

The next thing we need to do is make events that will decide whether `world.win` or `world.lose` will execute at the end of the game. Click **create new event**, and select **While something is true**. Then right click on it, and select **change to**, and then **When something becomes true**.



## When the Game Ends

Now, get an `a==b` from the world's functions tab, and drag and drop it onto the second **true**. Choose **expressions**, then `world.scoreValue`, then `other....`. Type in **10**.



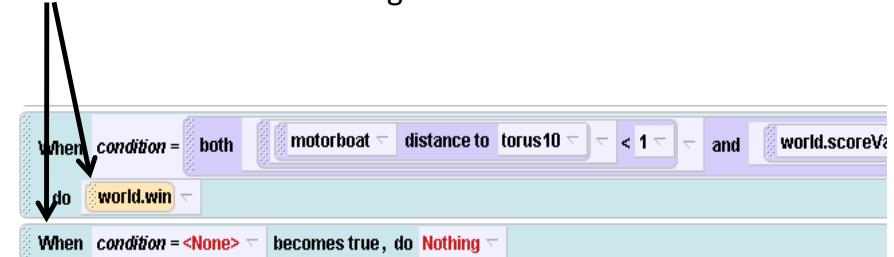
## When the Game Ends

Now, go to the world's functions tab, and click and drag **both a and b** onto the condition of the new event. Select **true** and **true**. Now click and drag an `a<b` onto the first **true**, and select **1** and **1**. Then go to the motorboat's functions tab, and click and drag `motorboat distance to` onto the first **1**. Select `torus10`.



## When the Game Ends

Now, drag a `world.win` from the world's methods tab, and drop it onto the **Nothing** on the event. Now, create a new **While something is true** event, and change it to **When something becomes true**. This will be an event for when the user loses the game.



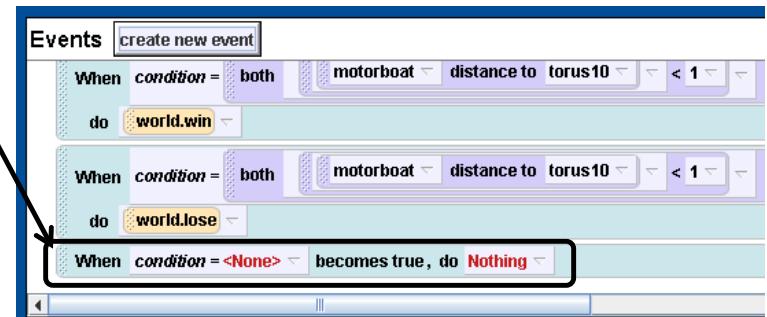
## When the Game Ends

Now, follow the exact same steps for this event as you did for the world.win **When something becomes true** event. The only difference will be, instead of getting an **a==b** for the second true, get an **a<b**. This is because if your score is less than 9 when you get to the last arch, that means you missed at least one arch and you will lose the game. Also, drag **world.lose** over nothing instead of **world.win**.



## When the Game Ends

Now we need one more event for the last game end possibility. If the timer ever counts down to 0, that means that the user has not made it to the finish line in time, because the timer will always stop counting down when the user gets to the finish line. Therefore, if **timerValue** ever equals 0, the user has lost. Create one more **While something is true** event, and change it to **When something becomes true**.



## When the Game Ends

Now, get an **a==b** from the world's functions tab, and drop it onto the **true**. Select **expressions**, then **world.timerValue** then **0**. Then, get **world.lose** from the world's methods tab, and drop it on top of **Nothing**.



Now you've finished your boat racing game! If you want, you can make changes to your game to make it harder. You can add arches, or shorten the amount of time given at the beginning. You can change the arches colors, and add different amounts of points for each color of arch. You can also add in background objects to decorate your world if you wish.

