Overview

- This tutorial will show how to create a game of catch with a score.
- A ball will be thrown several times and the player moves the glove to catch it.

Set Up

- Open a new Alice world with any template
- Click
- Click on sports in the local gallery
- Add a baseball and a baseballglove2
- Re-size both objects to make them a little bigger.
- Move the baseballglove2 so it is right in front of the camera. Right click on it in the object tree and select methods, orient to, camera

Position the ball 20 feet away: Orienting the ball

- Right click on the baseball in the object tree and select methods, move to, baseballglove2. It should look like the picture to the right.
- Right click baseball again and select methods, move, 20m, forward. The ball will probably move in the wrong direction. That is because is oriented in a different direction than the camera. Click undo.
- Right click the baseball again and select orient to, camera. This will cause the ball to always move according to our perspective.
Position the ball 20 feet away: Part two

- Try moving the ball again. Right click baseball again and select methods, move, 20m, forward.
- Now it should look like the bottom picture.
- Now we know when the gameplays, if the ball moves backward 20 meters, it will end up in the glove.
- Click 

Let the User Control the Glove

- Click and drag move into the during part of the while loop.
- Select right, other, and type .1.
- Set the style to abruptly and duration to .1 seconds. It should look like the while loop in the picture below.
- Now drag the entire while loop over to the clipboard to make a copy. (see picture below)

Let the User Control the Glove (Cont’d)

- When you click create new event, there is an option to Let the arrow keys move <subject> but that will cause the right and left arrow keys to turn the glove instead of moving it.
- Instead, click when a key is typed. Change any key to right.
- Right click on the instruction and click change to, while a key is pressed.
- Click on the baseballglove2 in the object tree, then methods tab.

Let the User Control the Glove (Cont’d)

- Now drag the entire while loop over to the clipboard and put the copy below the first one. Change both of the right’s to left.

Let the User Control the Glove (Cont’d)

- When the world starts, do workMyFirstMethod

- Now drag the entire while loop over to the clipboard and put the copy below the first one. Change both of the right’s to left.
Save the starting position of the ball

- Click on the baseball in the object tree. Click and more controls.
- You will notice two button appear. The first one, drop dummy at camera, you have probably already used in another tutorial. It saves the current location of the camera. The second one, drop dummy at selected object, you may not have used yet. It works the same way except it saves the location of whatever object is selected in the object tree instead.
- Click drop dummy at selected object. Rename the dummy startPosition. Click done.

Setting up the Pitch

- Under the baseball methods tab click and drag move to into world.my first method. Select startPosition and set the duration to 0.
- Now drag in the move method. Select backward, 20 meters. This will move the ball back to the glove.
- Play your world. The ball should be pitched right to the glove.

Making the Pitch Random: create a variable

- We want the ball to be thrown from a different spot each time to make it more difficult to play
- We will have the ball start at a random distance from the original position each time. Click create a new variable. Call it randomDistance and select number
- Click and drag this new variable to in between the two lines of code. Select set value, 1.

Making the Pitch Random: Make a Random Number

- Click on the world in the object tree and click functions. Scroll down to random.
- Drag in the random number to where the 1 is. Click more, minimum, other, -.5. Click more again, maximum, other, .5.
Making the Pitch Random (cont’d)

• Click on the baseball again in the object tree. Click and drag in move right below the line we just made. Select right, expressions, randomDistance. Set the duration to 0 and asSeenBy to camera. It should look like the picture below.

<table>
<thead>
<tr>
<th>Object</th>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseball</td>
<td>move to startPosition</td>
<td>duration: 0 seconds</td>
</tr>
<tr>
<td>randomDistance</td>
<td>set value to</td>
<td>random number:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>minimum: -2.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>maximum: 2.5</td>
</tr>
<tr>
<td>baseball</td>
<td>move right</td>
<td>randomDistance:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>duration: 0 seconds</td>
</tr>
<tr>
<td></td>
<td></td>
<td>asSeenBy: camera</td>
</tr>
</tbody>
</table>

• Test your world. It should throw to a different spot each time. You may need to change the minimum and maximum for the random number so the ball moves enough to make it difficult, but doesn’t go to far to the left or right of the screen.

Display the Score

• We want the score to be displayed.
• Click  and scroll over the right side of the local gallery and click create 3D text. Type “Score: 0”
• Rename the new 3D text Object “score”
• Resize it and move it to the top right corner of the screen like in the picture
• Click

Incrementing the Score: Part 1

• Now, under the methods tab of score, click create new method, call it incrementScore
• Under the properties tab, click create new variable. Name it value, select number, and set the value to 0. Then click and drag value into the new incrementScore method. Select increment score.value by 1

Incrementing the Score: Part 2

• Now we want to display the new score. Click and drag in text from the properties and select default text. Click on world, then functions and drag in a joined with b and select default text again.
• Change the first default text to “Score: ”
• If you try to drag in score.value, Alice won’t let you. That is because when we set the text of a 3D text object we need to set it to a string (a line of text). So, we need to convert value, which is a number, to a string so Alice can display it.
• Drag in what as a string from the world functions into the other default text. Select score and value.
Create checkForCatch method: Using an If statement

- After the ball is thrown, we want to see if the glove caught it. We will do this by checking how close the ball is to the glove.
- Click world, methods, then create new method. Name it checkForCatch.
- Drag in an If and select true.
- Under the world functions tab, drag a<b into the true spot. Select 1 for both a and b.
- Under the baseball functions tab, drag in distance to into the first 1. Select baseballGlove2. Change the 2nd 1 to 0.3

Create checkForCatch: Setting the Conditions

- Now go the score methods tab and drag in incrementScore into the first do nothing
- If the baseball is not caught we want it to keep moving so it is out of sight. Under the ball methods, drag in move below the else. Select backward, 5 meters.

CheckForCatch: Testing

- Click the world, methods tab. Drag in CheckForCatch to the bottom.
- Play your world and move the glove. If the ball is caught, the score should increase to one. If it is not, the ball will keep going. Click restart to play multiple times

Making Adjustments

- If it doesn’t catch the ball when it looks like it should, you may need to change the checkForCatch method. Increase the .3 to make it easier or decrease it to make it harder (This will make it only catch if it hits the very middle of the glove)
- If you don’t need to change it or you have extra time, try making the throw have a more realistic arc, movement by having the ball moving up then down while it moves backward 20m
Repeating until you win

- We want to keep throwing the ball until it has been caught 3 times.
- Drag a while into my first method, select true
- Drag the instructions that we already have into the do nothing.
- Go to the world functions and drag in a<b. Select expressions, score.value, and 3. This is what my first method should look like so far.

### Winning the Game

- When the ball has been caught 3 times, we want to congratulate the player.
- Click the score in the object tree and under the properties tab drag text to below the while loop. Click other and type “You Win!”

### Extra Exercises

- If you didn’t do the extra step of making the throw more realistic on slide 17, do that now.
- Make the catch more realistic by having the ball stop a little bit earlier so it doesn’t overlap with the glove. You can do this by either changing where the ball starts, or how far it travels. You might also need to change your check for catch method by increasing the distance allowed.
- Add a message that says “Miss!” every time the ball is not caught.

---

Finished!

- Now your game is ready. Click play to test it out.
Challenge Exercise

- Add a timer that counts down so the player has to catch the ball three times within 5 seconds. If they don’t, the game should stop and a losing message should be displayed.
- This will be similar to how you kept track of the score. You will need to create a 3D text and a variable. Since each throw takes about 1 second, you can decrement the timer inside the while loop of my first method. Just make sure you set the duration of any methods you add to zero so you don’t slow it down.
- The while loop will now have two conditions to keep running because we want the game to stop when you win or run out of time.