Piñata Game: Keeping Score in Alice

By Maggie Bashford
Professor Susan Rodger
Duke University
July 2009
www.cs.duke.edu/csed/alice/aliceInSchools

Adding the Text Object

Click add objects and add a new 3D Text Object to the world.

When it asks what to say, type score: 0.0.

Adding the Text Object

Download the Alice World called piñataGameStart.a2w that goes along with this tutorial.

We will use this world to create a game that allows players to hit a swinging piñata with a bat. Many of the objects have already been created and placed in the world. We must create an object to display and update the score, and we will also write the code to play the game.

Click add objects and add a new 3D Text Object to the world.

When it asks what to say, type score: 0.0.

Adding the Text Object

Once you have created the new text object, right click on it in the object tree and rename it score.

Reposition the score where you can see it. Use resize and turn to face the camera.
Creating Variables

Click on score in your object tree, and then go to the properties tab. Click on the create new variable button.

Name the variable scoreValue, select the type Number, and click the arrow next to value to select other and type 0.

Initialize Method

Create a class-level method for score called initialize. Click on scoreValue in the properties tab, drag it into the initialize method, and set its value to 0.

Now go back to the properties tab and find text. Drag text into the initialize method, and set its value to score: 0.0. Change the duration for both to 0 seconds.

Add Points Method

Now we need to write a method that will add one point to scoreValue variable when the piñata is hit. Create another class-level method, called addPoints.

Find scoreValue in the properties tab and drag it into the method, and choose to increment it by 1. Change the duration to 0 seconds.
Add Points Method

Next drag text into the addPoints method, set its value to “score:” and set its duration to 0 seconds.

Next, click on world in the object tree, and find its functions tab.

Find a joined with b and drag it into where you typed “score:” and select default string for b.

Then drag what as a string into default string, and select score.scoreValue from expressions.

Set Up Method

Now let’s divide the game into three parts: the set up and instructions, the game play, and what happens when the game is over.

Click on world in the object tree and create a new world method called setUp. Now click on score in the object tree and drag its method initialize into the setUp method.

For fun, let’s swing the bat during the last instruction. Drag a Do together into the setUp method, then the last instruction “…to swing the bat” inside it. Then drag a Do in order below that.

Now let’s swing the bat. Click on baseballBat in the object tree and drag the turn method into the Do in order, then select backward and ¾ revolution. Set the duration to .25 seconds and the style to abruptly. Make a copy of this line, and change its direction to forward.
**Set Up Method**

Now let's prepare the player to start the game. Select instructions from the object tree and drag text three times into the setUp method outside of the Do together. First set the text to “Ready?”, then “Set...” then “Go!”

Now we want to hide the instructions. Drag in the isShowing property and set it to false, then change its duration to 0 seconds. The final step in the setUp method is to start the timer. Find timer in the object tree and drag its initialize method into the bottom of setUp and select the parameter world.time from expressions.

*The timer and its methods were created in the Timer Tutorial. See the Alice Repository for more information.*

**Play Game Method**

Now that the clock has been set, we need to swing the piñata. Create a new world method called playGame. Drag a while loop into the playGame method. Now click on the functions tab and drag a\(a > b\) into the condition, then select timer.timeLeft from expressions for a and 0 for b.

Now we want the piñata to move while the timer goes down, so drag a Do together into the while loop. Next drag two copies of a Do in order inside the Do together.

**Set Up Method**

Check your setUp method to see if you missed anything! You can test it by selecting it When the world starts, but remember to set it back to my first method when you are done!

**Play Game Method**

In the first Do in order, we will make the timer count down. Click on timer in the object tree, and click on the properties tab. Drag text into the first Do in order, and choose default string for now. Change its duration to 0 seconds. Then go back to the world functions and replace default string with what as a string. Select timer.timeLeft from expressions as the what.
Play Game Method

We want to count down the seconds, so drag **Wait** from the bottom of the screen, and select **1 second**. Next find **timeLeft** from the timer properties and drag it into the **Do in order**. Select decrement **timeLeft** by 1.

**Game Over Method**

To perform the post-game animation, create a new **world** method called **gameOver**. The first step after the game ends is to move the piñata back to the center. Click on **piñata** and drag the **turn** method into **gameOver**. Select **right** and **0.5 revolutions**, then change **asSeenBy** to **camera** and **duration** to **.5 seconds**.

**Game Over Method**

Click on **baseballBat** in the object tree and drag the **roll** method into the **If** statement. Select **right** then **other** and type **1.25 revolutions**.

**Play Game Method**

Now let's use the second **Do in order** to make the piñata swing! Go to the piñata methods and drag in **turn**, then select **right** and other to type in **.1 revolutions**. Now change the style to **abruptly**, duration to **.5 seconds**, and **asSeenBy** to camera. Right-click and make a copy of this line, and change the direction to **left**.

**Game Over Method**

A text object called **gameOverText** is included in the world, and it has already been set to say "¡Felicidades!". Go to its **properties** tab, then drag **isShowing** into the **If** statement and select **true**. Set the **duration** to **0 seconds**. The **explode** method has already been written to make the piñata burst. Find **explode** in the world methods and drag it into the **If** statement.

**Game Over Method**

If you don't like "¡Felicidades!" (which means "Congratulations!" in Spanish), you can change the text and color in the **gameOverText** properties.
**Game Over Method**

Now let’s create the animation for when the player loses. Drag a **Do Together** into the **Else**.

Find **baseballBat** in the object tree, and drag the **roll** method into the **Do Together**, then select **right** and **.25 revolutions**. Now make a **copy** of this line and change the object to the **piñata**.

Next, drag in **color** and set it to light gray with a duration of 0 seconds. In the final step, drag **isShowing** into the **Else**, select **true**, and set the duration to 0 seconds.

---

**Typing Accents**

**PC:** (use the keypad and make sure you have num lock on)

- á = Alt + 0225
- é = Alt + 0233
- í = Alt + 0237
- ó = Alt + 0243
- ù = Alt + 0250
- ñ = Alt + 0241
- ñ = Alt + 0161
- ñ = Alt + 0191

**Mac:**

- á = Option + e, then a
- é = Option + e, then e
- í = Option + e, then i
- ó = Option + e, then o
- ú = Option + e, then u
- ñ = Option + e, then n
- ñ = Option + 1
- ñ = Option + Shift + ?

These are just some of the many different foreign characters you can create. Try searching for “keyboard accent codes” to find more.

---

**My First Method**

Now we need to call the three methods we have made in **my first method**. Click on the **my first method** tab, then find the new methods and drag them in.

Do this for the **setup**, **playGame**, and **gameOver** methods. This is all that we will include in **my first method**.
Bat Hit Method

If you try playing the game, you will see that the piñata swings across the screen but there is no way to hit it. Create a new world method called batHit. This method will turn the bat toward the piñata, check to see if the piñata has been hit, then turn the bat back.

Drag an If/Else statement into the batHit method. Then find the world functions and drag a>b into the If condition, then select timer.timeLeft from expressions and 0.

Click on the baseballBat and drag its turn method into the If statement. Select backward and 0.25 revolutions, then change the duration to .3 and the style to abruptly.

Now we want to check that the bat has hit the piñata. Go to the world functions, find both a and b, and drag it into the inner If condition.

Drag a>b into both parts and choose any values.

Click on baseballBat and go to its functions tab. Drag baseballBat distance to the left of into the first number and choose piñata. Then drag baseballBat distance to the right of into the third number and choose piñata.

Bat Hit Method

Make a copy of the turn method, and change its direction to forward. Now drag another If/Else statement between the two turn methods.

Our piñata will be turning during a 1 second period. Try experimenting with the durations in this method to make it easier or harder to hit.

Next find Wait at the bottom and drag it below the second turn method, still inside the If statement. Select other and type .3 seconds.

Now click on piñata in the object tree and find its functions. Drag piñata's width into the two remaining spots.

But since the measurement is from the center, we must divide piñata's width by 2. Click the arrow next to it and select math, piñata's width /, then 2. Do this for both places where you dragged piñata's width.
Bat Hit Method

To finish the method, click on score in the object tree, then drag the addPoints method into the if statement. This will give the player points only when the bat has hit the piñata.

Events

Go to the world functions and drag $a > b$ into the condition. Select score.scoreValue from expressions then world.numWinningHits from expressions.

Click on timer in the object tree then go to the properties tab. Drag timeLeft into the do Nothing and select set value to and choose 0. This will end the game when the player reaches the required number of hits.

Events

Now we need to create two events: one to recognize when the player wins and another to swing the bat when the player presses the up key.

Click create new event in the top of the window, and select While something is true.

Right click on the blue area of the new event and select change to then When something becomes true.

Events

Now click create new event and select When a key is typed. Change any key to Up. Then click the arrow next to Nothing and change it to batHit.
Finishing Up

Now you are ready to play the game! Click play and try your best to break the piñata.

Keeping track of a score can be used in many games. Some games require a certain score to reach the next level. Others may keep track of the players' high scores.

You can create different events and methods to update your score. All you need to do is keep track of your score and update its display.