Making a Timer

This is an modification of the July 2008 timer tutorial by Jenna Hayes
By Natalie Huffman
Under the direction of Susan Rodger
Duke University
June 2017

Open a new Alice world

• Choose “Grass”
• Go to setup scene
• Click on the shapes/text tab at the bottom of the page
• Select new TextModel()

Create a TextModel property

• Select TextModel ➔ Add TextModel Property from the drop down menu
• Value type: DecimalNumber
• Name: timeLeft
• Initializer: 0.0
Create a TextModel Procedure

• Select TextModel ➔ Add TextModel Property from the dropdown menu
• Name it initialize

Initialize

• This procedure will establish how much time our timer starts with, so we need a parameter
• Set the value type as DecimalNumber
• Set the name as amountOfTime

Create a TextModel Procedure

• Drag in the setTimeLeft method and choose “amountOfTime”
• setTimeLeft is the property we created for our textModel and it will be accessible from myFirstMethod

Create another TextModel procedure by selecting TextModel ➔ Add TextModel Procedure from the dropdown menu
• Name it “countDown”
countDown

- Drag in a while loop and choose true
- Then select Relational (DecimalNumber) \( \rightarrow \text{other} \rightarrow \text{this.timeLeft} \)
  - Custom DecimalNumber
- Select 0.0 for the custom number
- See the next slide for a picture

- Next, drag in setValue and select “hello”
- setValue changes the content displayed by the TextModel
- However, this procedure wants to be given a word, and we want to change the value to timeLeft, which is a DecimalNumber
- So instead, we select \( \text{other} \rightarrow \text{DecimalNumber} \rightarrow \text{this.timeLeft} \)
  - Custom TextString
- When the custom string box pops up, leave it blank and hit enter
- This lets us set the content equal to timeLeft + “”, so Alice will automatically convert timeLeft to a string
• Drag in setTimeLeft and select TimeLeft
• Then click the dropdown arrow → Math → this.timeLeft-?? → 1.0
• Drag in delay, and select 1.0

Hang on....

• Our timer ends at 1.0!
• Why?
• Go back to countDown
• The while loop only runs when timeLeft is greater than zero, so when timeLeft = 0, the loop ends
• But we want to reset the value of the TextModel one last time

Go back to myFirstMethod

• In myFirstMethod, select this.timer on the far left and drag in initialize
• Choose how much time you want your timer to start with! I chose 10.0
• Drag in countDown
• Run your code!

Debugging

• Right click on the setValue line and select “copy to clipboard”
• Click and drag from the clipboard image on the top right
• Put the line of code beneath the while loop
Now run your code!

• The timer should stop at 0.0, just like we want it to
• But we made the timer a DecimalNumber for a reason—what if we want to countdown by smaller numbers?
• We’ll choose .1

Go back to countDown

• Set the two instances of 1.0 (in setTimeLeft and delay) to .1
• Run your code!

Umm.....

• This doesn’t seem quite right
• You probably got something that looks like this

6.4000000000000013

• As the timer keeps going, the error gets bigger and bigger

Quick Explanation

• This is essentially caused by a rounding error
• Computers store numbers in binary (ones and zeroes)
• But the number .1 can’t be represented exactly in binary
• It’s kind of like trying to write 1/3 in decimal; you get .3333333333 repeating on forever
• Alice doesn’t subtract .1, it actually subtracts .0999999999 etc, so your timer is always slightly ahead of where it should be
Luckily, we can fix this!

- Alice has a built in set of rounding commands, found under the Math tab
- *Ceiling* always rounds up, *floor* always rounds down, and *round* follows normal rounding rules
- We’ll use *floor*, since our number is just a little bit too big
- But using *floor* on the number 9.9000000001 will round it all the way down to 9! That’s not what we want

• Click the arrow on the far right→Math→absolute value, round, ceiling, floor→floor→this.timeLeft

• Now click the inner arrow→Math→this.timeLeft-???
  →Custom DecimalNumber and enter .1

• Now there are three arrows! Click the middle one→Math→(this.timeLeft - 0.1)*???
  →10.0

• This gives us 99.000000001, and rounding will change it to 99.0, which is almost what we want!
• The only thing left to do is divide it back out
• Click the arrow on the far right→Math→floor((this.timeLeft - 0.1) * 10.0)/???
  →10.0

Your code should look like this

```
while this.timeLeft > 0.9 is true
  do in order
    this.setValue = this.timeLeft + 1
    this.setTimeLeft = floor(this.timeLeft / 10.0)
    this.delay = 0.1
  loop
```

Run your code

• Your timer should show up exactly like you wanted it to!