

CompSci 94

Classwork: Scorer and Timer

November 2, 2021



Prof. Susan Rodger

1) The Game

- Start with the game you built last time. You will add a scorer to keep track of the score, 1 pt for a white bunny, 2 pts for gray bunny, and 3 pts for the brown bunny. Then you will add a timer to give the player a fixed amount of time to play. The player wins if they get a specified number of points before the time runs out.
- Follow the following steps to enhance your game with a scorer and a timer.

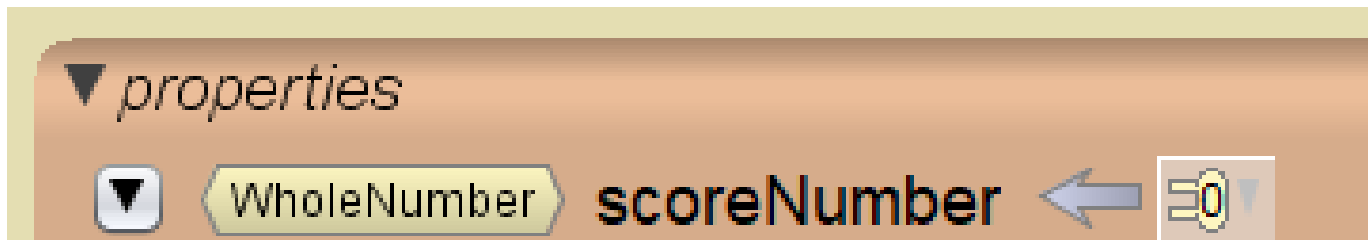
2) Create a Scorer

- In scene setup add a textModel from the shapes/text tab and name it scorer.
- Make it a contrasting color, like blue if you are using sand.
- Set its value to the string “0”
- Move the scorer to the top right side
- Then make it invisible by setting its opacity to 0



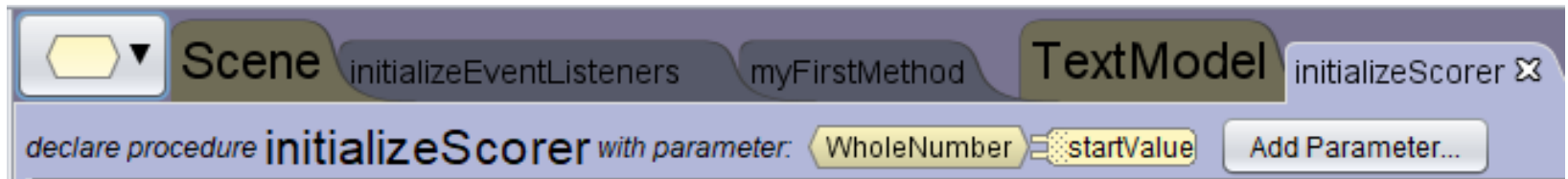
3) Add a textModel property of type WholeNumber named scoreNumber

- Set its initial value to 0



4) Add a **textModel** Procedure named **initializeScorer**

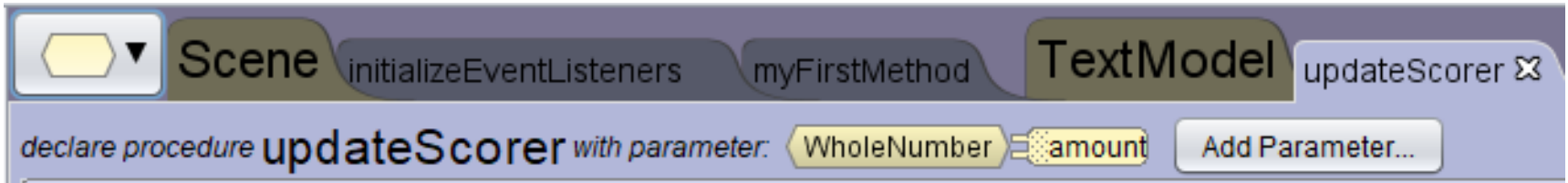
- Add one **wholeNumber** parameter named **startValue**



- For the body:
 - Set the `scoreNumber` to the `startValue`
 - Set the value for the Scorer `textModel` to “0”
 - Set the opacity of the Scorer to 1 – make it visible when the game is starting

5) Write the textModel procedure named updateScorer

- This procedure has one WholeNumber parameter named amount.



- This procedure should:
 - Update scoreNumber by amount
 - Update the text value that displays the score

6) Add the scorer to the game

- Click on the initializeEventListeners tab
- In the event for clicking on the tortoise, after the bunnies move down, call initializeScorer with 0.
- In the event for clicking on bunnies, **replace** the **tortoise say** with a call to updateScorer by 1
- In myFirstMethod, in the game loop that says “while true”, instead continue the game as long as the score is less than 10.
- After the while loop, have the tortoise say, game is over.

7) Play your game

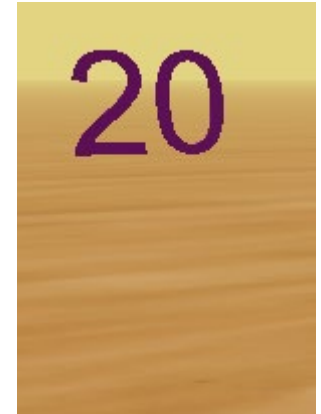
- The game should stop after you click on 10 bunnies.

8) Change game to count brown/gray bunnies worth more points

- In the bunny event, if the bunny you click on is gray then update score by 2, if brown then update the score by 3
- Otherwise update the score with 1 point.
- Play your game. Does clicking on the brown or gray bunny give more points?

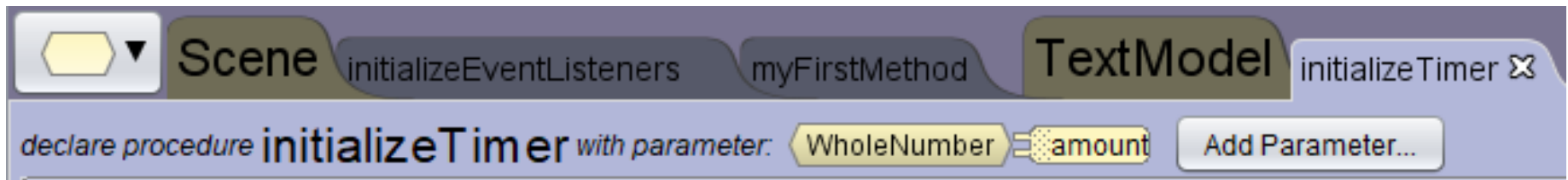
9) Add a Timer to the game

- In setup scene, click on the shapes/text tab and add a textModel named Timer. Make it a different color than the Scorer.
- Put the timer up in the air on the left side of the screen.
- Add a TextModel property of type wholeNumber named timerNumber set to 20.
- Add a TextModel property of type Boolean named TimerOn, set to FALSE



10) Write textModel procedure named InitializeTimer

- Add one parameter of type **WholeNumber** named **amount**



- Set TimerNumber to amount
- Display the new string value of the timer
- Make the opacity 1, to make the timer visible

11) Write TextModel procedure named updateTimer

- This procedure has NO parameters.



The screenshot shows a code editor with a tab labeled 'TextModel' and a sub-tab 'updateTimer'. The code being edited is:

```
declare procedure updateTimer
```

There is an 'Add Parameter...' button next to the procedure name.

- Update the timerNumber by subtracting 1
- Redisplay the new value of the timer.
- **THERE SHOULD NOT BE A LOOP IN THIS CODE!**

11) Update the game with the timer

- Click on the initializeEventListeners tab
- Need to initialize the timer in the tortoise event
 - Where the scorer is initialized, make that a doTogether and also initialize the timer to 20.
 - Also put a doTogether around where the tortoise says to click on the bunnies. At the same time, set the timerOn property to TRUE. That means the timer can start ticking down.

11) Update the game (cont)

- Click on the initializeEventListeners tab
- Create a new event
 - Click on addEventListener, then Scene Activation/Time, then addTimeListener, and pick a time of 1.0. This means every second this event will run again.



- For the body of the event, if the timerOn is TRUE, then update the timer.

12) Update myFirstMethod

- Change the while loop condition that plays the game. Now the game should continue while the score is less than 10 AND while the timer is greater than 0.
- AFTER the while loop that plays the game:
 - And after saying the game is over...
 - Tell the user either “you won” or “better luck next time”. The user won if they scored at least 10 points.

Now play the game!

- If you get 10 points before the game runs out, you win!
- Does the timer keep going? How do you fix that? Immediately after the while loop for the game ends, you'll need to set the TimerOn variable to False so the timer will stop.
- **If the game is too hard**, set your timeListener event to go off every 1.5 seconds. You will need to adjust the bunnies popping up to take 1.5 seconds total for the move up, delay and move down. That should make the game easier