Game Programming Exploration Homework

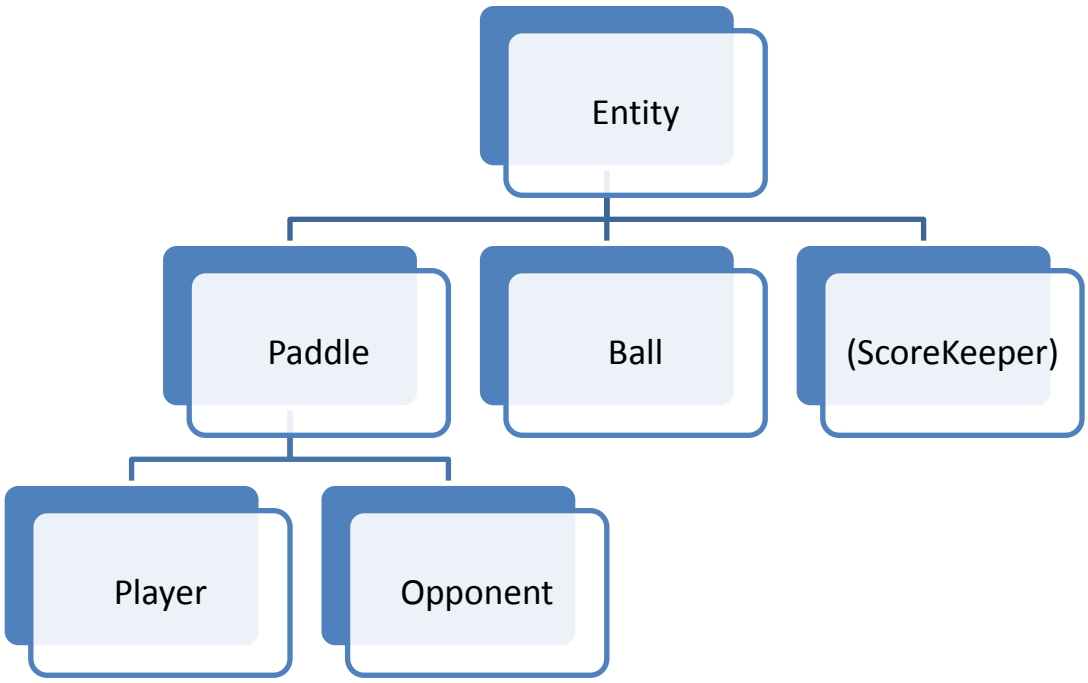
We're going to create a simple **ScoreKeeper** object to interact with our Ball object. We have already written the interaction code for you, so you just need to make the **ScoreKeeper** object.

To start, download processing from http://www.processing.org/

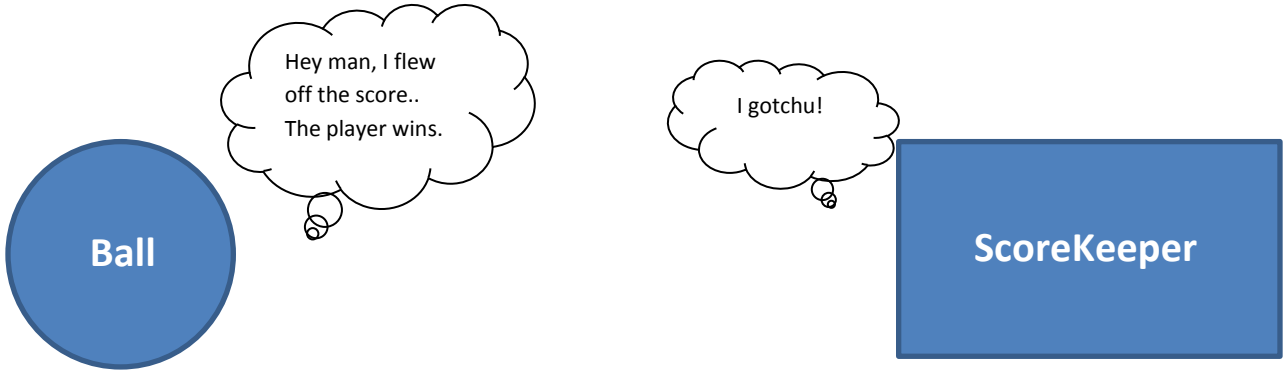
You'll also need the files we made during the exploration session (available as a .zip file):

pong.pde, Entity.pde, Ball.pde, Paddle.pde, Player.pde, Opponent.pde

Recall that we set up a **Game Entity Hierarchy** as shown below:



Your task (if you choose to accept it) will be to create the **ScoreKeeper** object. As you can see, we are treating it as an Entity so we can include it in our update and render routine in **pong.pde**. As you may have noticed, we have updated **Ball.pde** to include a win/loss routine, and we have updated **pong.pde** to construct the **ScoreKeeper** object properly. As a result, **the game won't compile properly until you implement the ScoreKeeper object**. Our **Ball** object is going to communicate to the score keeper:



The ball does this by storing a **Reference** to the **ScoreKeeper** object (much like how **LinkedListNodes** store References to other **Nodes**). You can see this by looking at the **Ball** constructor. We then can "let the score keeper know" that something happened by calling methods on our **ScoreKeeper** reference, and the keeper can react properly (I.E. change its' score).

Therefore we must write the following methods in **ScoreKeeper** for it to compile. You can write the method stubs, compile, and fill in the rest as you go:

public ScoreKeeper()	Constructs a ScoreKeeper. Calls the Entity super() constructor to specify and x, y, width, and height. Both width and height can be 0. X and Y will determine where we display our score.
public void update()	We have to implement this(since we promised we were an Entity), but nothing goes in here because at least for our purposes, ScoreKeeper doesn't update based on the game loop. (You could make it move or something though, in which case you would use this)
public void render()	This is where we render the score. Processing as a built in "magic function" called text(String, float, float), which takes a String to render, a x position, and a y position.
public void opponentWin()	This is called by our ball signaling that the opponent has won a round. Increment the appropriate score here.
public void playerWin()	This is called by our ball signaling that the player has won a round. Increment the appropriate score here.

I hope you enjoyed the exploration session! Feel free to send me (necha@cs.washington.edu) any cool things you make in the future. I'd love to see them!

Game programming is an awesomely complicated and deep topic. It's also very fun! Remember, that most game "cool" effects are actually simple to implement and are mostly tricks. Check out this video for some topics on how to make the game "look" better:

https://www.youtube.com/watch?v=Fy0aCDmgnxg

-Aaron Nech