Name: _____ Date: _____

Advanced Computer Math Writing Classes

LightBike Game

Part A: Trail Class

1. Create a class called `Trail`

2. Field:
   a. **Integer** called `timer` – it will represent if the `Trail` is currently "active" or not.

3. Constructor:
   a. Build a constructor that is sent a **Boolean** parameter to represent if the `Trail` is blue.
   b. Create a test to determine the truth value of the **Boolean** variable.
      i. If it is true…
         1. Set the image to the blue trail (`trail1.png`)
      ii. If it is false…
         1. Set the image to the red trail (`trail2.png`)
   c. Set the `timer` variable to 10.

4. Methods
   a. `act()`
      i. Subtract 1 from the `timer`.
      ii. If the `timer` is below zero and the `Trail` is touching a `Bike`, it should remove the `Bike` it is touching from the screen.

Part B: Bike Class

1. **Create a class called** `Bike`

2. **Fields:**
   a. **Boolean to represent if the bike is blue**
   b. **String to represent the bike's control:** `up`
   c. **String to represent the bike's control:** `down`
   d. **String to represent the bike's control:** `left`
   e. **String to represent the bike's control:** `right`

3. **Constructor:**
   a. **Build a constructor that is sent a Boolean parameter to represent if the** `Bike` **is blue.**
      i. **The constructor should store this information in its permanent field variable.**
   b. **Create a test to determine the truth value of the Boolean variable.**
      i. **If it is true…**
         1. **Set the image to the blue bike (**`bike1.png`**)**
         2. **Set its controls to…**
            a. **"W" =** `up`
            b. **"S" =** `down`
            c. **"A" =** `left`
            d. **"D" =** `right`
      ii. **If it is false…**
         1. **Set the image to the red bike (**`bike2.png`**)**
         2. **Set its controls to…**
            a. **"up" =** `up`
            b. **"down" =** `down`
            c. **"left" =** `left`
            d. **"right" =** `right`

4. **Methods:**
   a. `buildTrail()`
      i. **Takes in no parameters**
      ii. **Build a** `Trail` **object with the Boolean parameter of the** `Bike`**'s current color**
      iii. **Ask the world to place the** `Trail` **on the screen at the same location of the** `Bike`
   b. `movement()`
      i. **Make the bike move directly forward at a speed of 5**
      ii. **Check if its** `up/down/left/right` **controls are being pressed.**
         1. **If** `up` **string is being pressed, make the** `Bike` **face up.**
         2. **If** `down` **string is being pressed, make the** `Bike` **face down.**
         3. **If** `left` **string is being pressed, make the** `Bike` **face left.**
         4. **If** `right` **string is being pressed, make the** `Bike` **face right.**
         5. **NOTE: DO <u>NOT</u> USE THE** `turn()` **COMMAND!**
   c. `act()`
      i. **call** `buildTrail()` **method**
      ii. **call** `movement()` **method**

Part C: World Class

1. **Create the** `World`

2. **Method:**
   a. `buildBikes()`
      i. **Takes in no parameters**
      ii. **Build two different** `Bike` **objects: one which is blue and one which is red.**
         1. **HINT: REMEMBER TO SEND OVER A BOOLEAN PARAMETER**
         2. **The blue** `Bike` **should be placed…**
            a. $\frac{1}{10}$**th of the way across the screen**
            b. **halfway down the screen.**
         3. **The red** `Bike` **should be placed…**
            a. $\frac{9}{10}$**th of the way across the screen**
            b. **halfway down the screen**
            c. **Set its rotation to 180 so it faces the correct starting direction**

3. **Constructor:**
   a. **The size of the** `World`**: 1000 x 775**
   b. **When the** `World` **is built, it should call a method** `buildBikes()`**.**