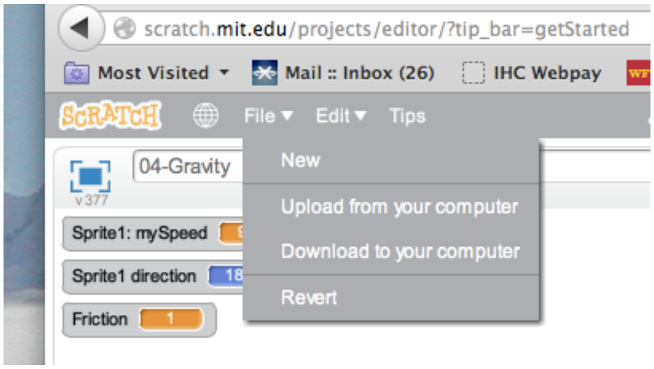
Survey of Engineering

Computer Science

Pinball Worksheet 3

Name______________________________

Date____________________ Team _________



1. Log on to Scratch
   http://scratch.mit.edu/
2. Select "Try it out."
3. Upload your file from last exercise.
4. **Define Gravity**

   4.1. Click the **Stage** area (lower left pane) and create two global variables called **Gravity** and **GravityAngle.** All the Sprites will be able to see and access this variable.

   4.2. In the Green Flag Event handler set **Gravity** to 0.1 and **GravityAngle** to 180.

   4.3. Make an event handler for the letter "**g**" that adds 0.01 to Gravity.

   4.4. Make an event handler for the letter "**t**" that sets gravity to 0.

   4.5. Test your event handlers to make sure they change the variables like you would expect.

   4.6. Click on the **Ball** Sprite Icon in the lower left pane to get back to its code.

   4.7. Make a user defined function (or Block) called **Gravity.**

   4.7.1. Go to the **More Blocks** command group and **Make a Block**.

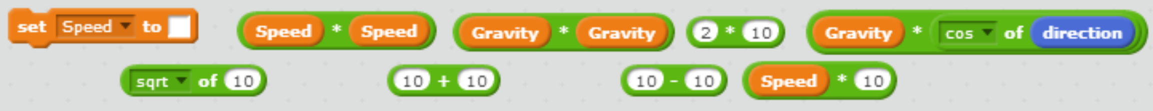
   4.7.2. Name this block **Gravity**. You should see a new block somewhere in your code called **Define Gravity.**

   4.7.3. Add a **Gravity** block to your **forever** loop.

   4.8. Use vector addition to determine Gravity's effect on the ball.

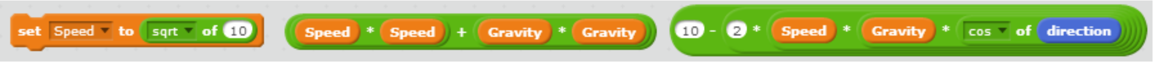
   4.8.1. You will use the Law of Cosines to calculate a new **Speed**: set **Speed** to sqrt(**Speed**^{2} + **Gravity**^{2} – 2***Speed*****Gravity***cos(**direction**)). Direction is a variable in the **Motion** group. This is complicated, so let's break it down into smaller steps.

   4.8.2. Drag the following to the code space and start assembling them



```
set Speed to
Speed * Speed
Gravity * Gravity
2 * 10
Gravity * cos of direction
sqrt of 10
10 + 10
10 - 10
Speed * 10
```

   4.8.3. Once all the correct variables are in the operators, start dragging the operator groups together covering up all the 10's. Order is important

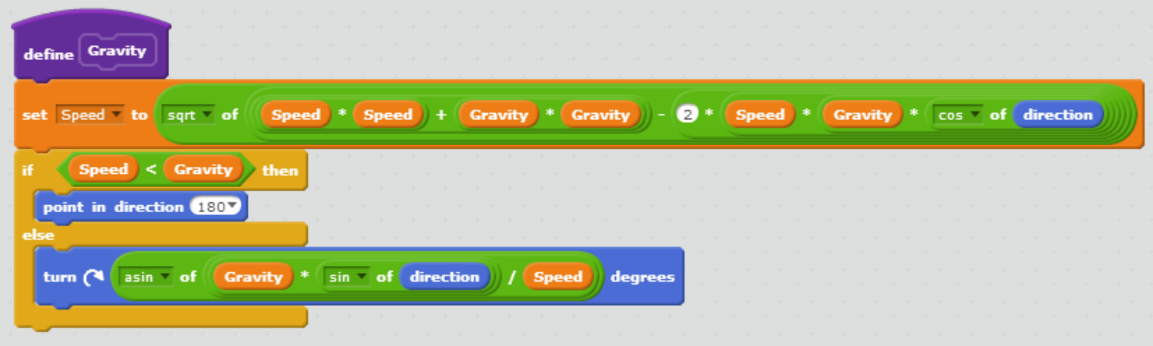


```
set Speed to sqrt of 10
Speed * Speed + Gravity * Gravity
10 - 2 * Speed * Gravity * cos of direction
```

4.8.4. The middle group drags to the right, and then everything goes into the sqrt function to finish the speed calculation.

4.8.5. We will use the Law of Sines and an **If Then Else** block to calculate **how far we need to turn**. If **Speed** is close to 0 (Speed < Gravity) the equation doesn't work well. In this case set **newAngle** to 180 (down). Otherwise (else), **turn** asin(**Gravity** * sin(**direction**)/**Speed**) degrees.

4.8.6. Your **define Gravity** code should look like something like this:



```
define Gravity
set Speed to (sqrt of ((Speed * Speed) + (Gravity * Gravity)) - (2 * Speed * Gravity * (cos of direction)))
if <(Speed) < (Gravity)> then
    point in direction 180
else
    turn ↻ ((asin of (Gravity * (sin of direction))) / (Speed)) degrees
```

4.9. Test your code. Make sure that you have called **Gravity** from your forever loop. Adjust gravity using the "g" and "t" keys and use the Speed and Direction controls you made earlier to see if gravity works as you expect it to.

Extra for experts:

- Make a function called Friction and move your code that implements friction into it.
- If you want a real challenge, define a variable called GravityAngle and modify the equations so that gravity can work in any direction.

5. Save your work.
   5.1. Find the File menu just above the output pane.
   5.2. Select **Download to your computer**.
   5.3. Give the file a name and store it in your chosen location.