Simple Projectile Motion Simulation

- Create a projectile sprite (for example, a basketball).
- Create variables for gravity, time, initial velocity, initial y position and initial x position.
- Set time to 0, gravity to -9.8 and velocity to be a slider (double click on a variable to show its slider).
- Set the initial x and initial y positions to be the current x and y positions of the projectile as follows:

```
set y0 to y position
set x0 to x position
```

- As a start, let's just get the projectile to move in the y direction according to its initial position, its initial velocity and gravity. Build an expression to describe the projectile's y position in terms of time:

```
set y to y0 + v0 * t + 1 / 2.0 * g * t * t
```

- Place this expression in a repeat until block with the condition “touching edge.”
- Make t change by 0.1 seconds for each repetition.
- Your code should now look something like the following:

![Code block with expressions and variables]

- Run the program several times with different values for v0, g and the amount t changes by.
- Now, let’s add movement in the x direction to simulate simple projectile motion. Add a variable to represent the projectile’s angle and set it to be a slider.
- Create variables for the x and y components of velocity.
- Using trigonometry, separate the initial velocity into its x and y components as follows (sin and cos are in the sqrt dropdown in the operators section):
• Change the expression for the y position to use the y component of velocity (Yv0 in my example).
• Add an expression to the projectile’s x position in terms of time and add it to your repeat until loop:

```
set x to x0 + Xv0 * t
```

• If you start your projectile at the bottom left of the screen, you should see it soar across the screen realistically!
• Here is what my final code looks like:

Extension ideas
Control the angle with the mouse, draw a cannon to shoot the projectile out of, keep track of the angle that resulted in the furthest travel distance, provide a randomly-placed target for the user to aim for, let the user control the initial velocity with keyboard arrows.