**CMPSC 121 Project Report**

**Project #1**

**sdh5378 - Shane Hagan**

**02-05-17**

**Code**

**Original Code:**

# Project 1

import math

print("The intention of this project is to allow a user to enter specific values to see if the Red Bird, from the game Angry Birds, will hit the structure")

# First, we want to write code that will allow the user to input his / her own values

line = input("\nEnter the initial velocity (in m/s): ")

Velocity = float(line)

line = input("\nEnter the angle of the Red Bird with respect to the horizontal axis (in degrees): ")

Angle = float(line)

line = input("\nEnter the ground distance to the structure (in meters): ")

GroundDistance = float(line)

# Come back and change so user can adjust height of launch

print("\nFor right now, the Red bird will be launched off a 5 meter tall slingshot")

line = input("\nEnter the height of the structure your're launching the Red Bird at (in meters): ")

StructureHeight = float(line)

# After all the values have been entered, we will begin to write the code to calculate whether it hit or miss

# Since this problem deals with both the x and y directions, we will split it up accordingly

print("\nLet's see if the Red Bird hit the structure...")

# Time it took to reach the structure

'''

First, we need to seperate the vectors of x and y. Since acceleration is 0 m/s^2 in x direction, so the equation can be simplified

We will be using equation 3 (d=Vit + 1/2at^2), which simplifies to

d=Vit. After rearranging for time we get t=d/Vi

'''

AngleRad = ((Angle \* math.pi) / 180)

Vx = (Velocity \* (math.cos(AngleRad)))

Vy = (Velocity \* (math.sin(AngleRad)))

Time = GroundDistance / Vx

print("\nThe Red Bird reached the structure in:",round(Time, 5),"s.")

# Now, we need to determine the final velocity as it hits the structure

'''

First we need to find the final velocity in both x and y directions, then use pythagorean theorem

to determine the final velocity. Final velocity in x direction will be same as initial due to there

being 0 m/s^2 acceleration in the x direction.

'''

Vfx = Vx

Vfy = Vy + (Time \* -9.8)

Vf = math.sqrt(math.pow(Vfx,2) + math.pow(Vfy,2))

print("\nThe Red Bird was traveling at a velocity of",round(Vf, 5),"m/s")

# Next will be to determine the angle, respective to the x axis

FinalAngle = math.degrees(math.atan(Vfy/Vfx))

print("and at an angle of",round(abs(FinalAngle), 5),"below the horizontal.")

# Height at which the Red Bird hits the structure

'''

We take the time we got from the horizontal distance and plug it into the y

direction equation to see if the Red Bird actually hits the structure. Use equation

Yf = Yi + Viyt + 1/2ayt^2

'''

StructureHit = 5 + (Vy \* Time) + (.5 \* -9.8 \* (math.pow(Time, 2)))

print("\nThe Red bird was at a height of",round(StructureHit, 5),"m from the ground when it reached the intended structure")

# To get the final verdict of whether it hit the structure or not, we will use if and else statements

if StructureHit < 0:

print("\nThe Red Bird was short and did not hit the structure!")

if 0 < StructureHit <= StructureHeight:

print("\nSuccess! The Red Bird hit the structure!")

if StructureHeight < StructureHit:

print("\nThe Red Bird went over the structure and missed!")

### ALL FINAL ANSWERS WERE ROUNDED TO 5 DECIMAL PLACES ###

**Metric Switch Code:**

# Project 1

import math

print("The intention of this project is to allow a user to enter specific values to see if the Red Bird, from the game Angry Birds, will hit the structure")

# First, we want to write code that will allow the user to input his / her own values

**line = input("\nEnter the initial velocity (in ft/s): ")**

Velocity = float(line)

line = input("\nEnter the angle of the Red Bird with respect to the horizontal axis (in degrees): ")

Angle = float(line)

**line = input("\nEnter the ground distance to the structure (in feet): ")**

GroundDistance = float(line)

# Come back and change so user can adjust height of launch

**print("\nFor right now, the Red bird will be launched off a 5 foot tall slingshot")**

**line = input("\nEnter the height of the structure your're launching the Red Bird at (in feet): ")**

StructureHeight = float(line)

# After all the values have been entered, we will begin to write the code to calculate whether it hit or miss

# Since this problem deals with both the x and y directions, we will split it up accordingly

print("\nLet's see if the Red Bird hit the structure...")

# Time it took to reach the structure

'''

First, we need to seperate the vectors of x and y. Since acceleration is 0 m/s^2 in x direction, so the equation can be simplified

We will be using equation 3 (d=Vit + 1/2at^2), which simplifies to

d=Vit. After rearranging for time we get t=d/Vi

'''

AngleRad = ((Angle \* math.pi) / 180)

Vx = (Velocity \* (math.cos(AngleRad)))

Vy = (Velocity \* (math.sin(AngleRad)))

Time = GroundDistance / Vx

print("\nThe Red Bird reached the structure in:",round(Time, 5),"s.")

# Now, we need to determine the final velocity as it hits the structure

'''

First we need to find the final velocity in both x and y directions, then use pythagorean theorem

to determine the final velocity. Final velocity in x direction will be same as initial due to there

being 0 m/s^2 acceleration in the x direction.

'''

Vfx = Vx

Vfy = Vy + (Time \* -9.8)

Vf = math.sqrt(math.pow(Vfx,2) + math.pow(Vfy,2))

**print("\nThe Red Bird was traveling at a velocity of",round(Vf, 5),"ft/s")**

# Next will be to determine the angle, respective to the x axis

FinalAngle = math.degrees(math.atan(Vfy/Vfx))

print("and at an angle of",round(abs(FinalAngle), 5),"below the horizontal.")

# Height at which the Red Bird hits the structure

'''

We take the time we got from the horizontal distance and plug it into the y

direction equation to see if the Red Bird actually hits the structure. Use equation

Yf = Yi + Viyt + 1/2ayt^2

'''

StructureHit = 5 + (Vy \* Time) + (.5 \* -9.8 \* (math.pow(Time, 2)))

**print("\nThe Red bird was at a height of",round(StructureHit, 5),"ft from the ground when it reached the intended structure")**

# To get the final verdict of whether it hit the structure or not, we will use if and else statements

if StructureHit < 0:

print("\nThe Red Bird was short and did not hit the structure!")

if 0 < StructureHit <= StructureHeight:

print("\nSuccess! The Red Bird hit the structure!")

if StructureHeight < StructureHit:

print("\nThe Red Bird went over the structure and missed!")

### ALL FINAL ANSWERS WERE ROUNDED TO 5 DECIMAL PLACES ###

**Height Adjust Code:**

# Project 1

import math

print("The intention of this project is to allow a user to enter specific values to see if the Red Bird, from the game Angry Birds, will hit the structure")

# First, we want to write code that will allow the user to input his / her own values

line = input("\nEnter the initial velocity (in m/s): ")

Velocity = float(line)

line = input("\nEnter the angle of the Red Bird with respect to the horizontal axis (in degrees): ")

Angle = float(line)

line = input("\nEnter the ground distance to the structure (in meters): ")

GroundDistance = float(line)

**# ALLOW USER TO CHANGE HEIGHT**

**line = input("\nEnter the height of the slingshot (in meters): ")**

**Height = float(line)**

line = input("\nEnter the height of the structure your're launching the Red Bird at (in meters): ")

StructureHeight = float(line)

# After all the values have been entered, we will begin to write the code to calculate whether it hit or miss

# Since this problem deals with both the x and y directions, we will split it up accordingly

print("\nLet's see if the Red Bird hit the structure...")

# Time it took to reach the structure

'''

First, we need to seperate the vectors of x and y. Since acceleration is 0 m/s^2 in x direction, so the equation can be simplified

We will be using equation 3 (d=Vit + 1/2at^2), which simplifies to

d=Vit. After rearranging for time we get t=d/Vi

'''

AngleRad = ((Angle \* math.pi) / 180)

Vx = (Velocity \* (math.cos(AngleRad)))

Vy = (Velocity \* (math.sin(AngleRad)))

Time = GroundDistance / Vx

print("\nThe Red Bird reached the structure in:",round(Time, 5),"s.")

# Now, we need to determine the final velocity as it hits the structure

'''

First we need to find the final velocity in both x and y directions, then use pythagorean theorem

to determine the final velocity. Final velocity in x direction will be same as initial due to there

being 0 m/s^2 acceleration in the x direction.

'''

Vfx = Vx

Vfy = Vy + (Time \* -9.8)

Vf = math.sqrt(math.pow(Vfx,2) + math.pow(Vfy,2))

print("\nThe Red Bird was traveling at a velocity of",round(Vf, 5),"m/s")

# Next will be to determine the angle, respective to the x axis

FinalAngle = math.degrees(math.atan(Vfy/Vfx))

print("and at an angle of",round(abs(FinalAngle), 5),"below the horizontal.")

# Height at which the Red Bird hits the structure

'''

We take the time we got from the horizontal distance and plug it into the y

direction equation to see if the Red Bird actually hits the structure. Use equation

Yf = Yi + Viyt + 1/2ayt^2

'''

**StructureHit = Height + (Vy \* Time) + (.5 \* -9.8 \* (math.pow(Time, 2)))**

print("\nThe Red bird was at a height of",round(StructureHit, 5),"m from the ground when it reached the intended structure")

# To get the final verdict of whether it hit the structure or not, we will use if and else statements

if StructureHit < 0:

print("\nThe Red Bird was short and did not hit the structure!")

if 0 < StructureHit <= StructureHeight:

print("\nSuccess! The Red Bird hit the structure!")

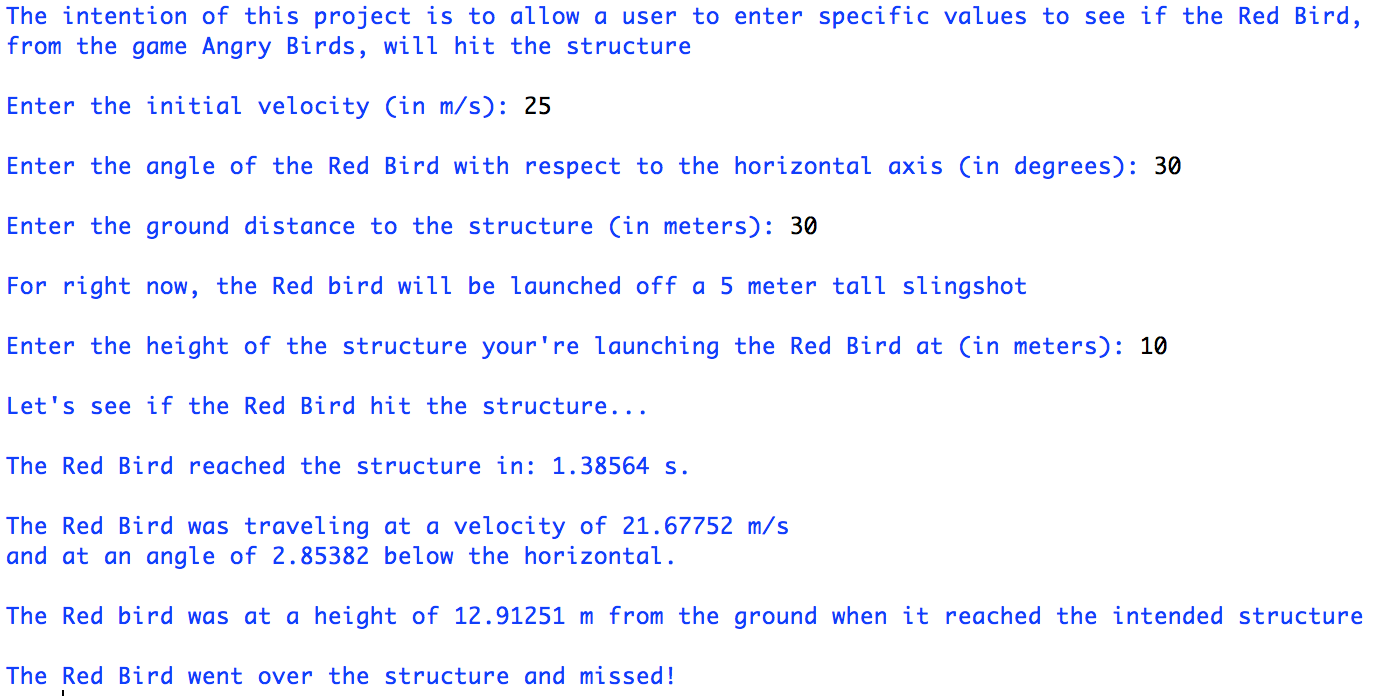
if StructureHeight < StructureHit:

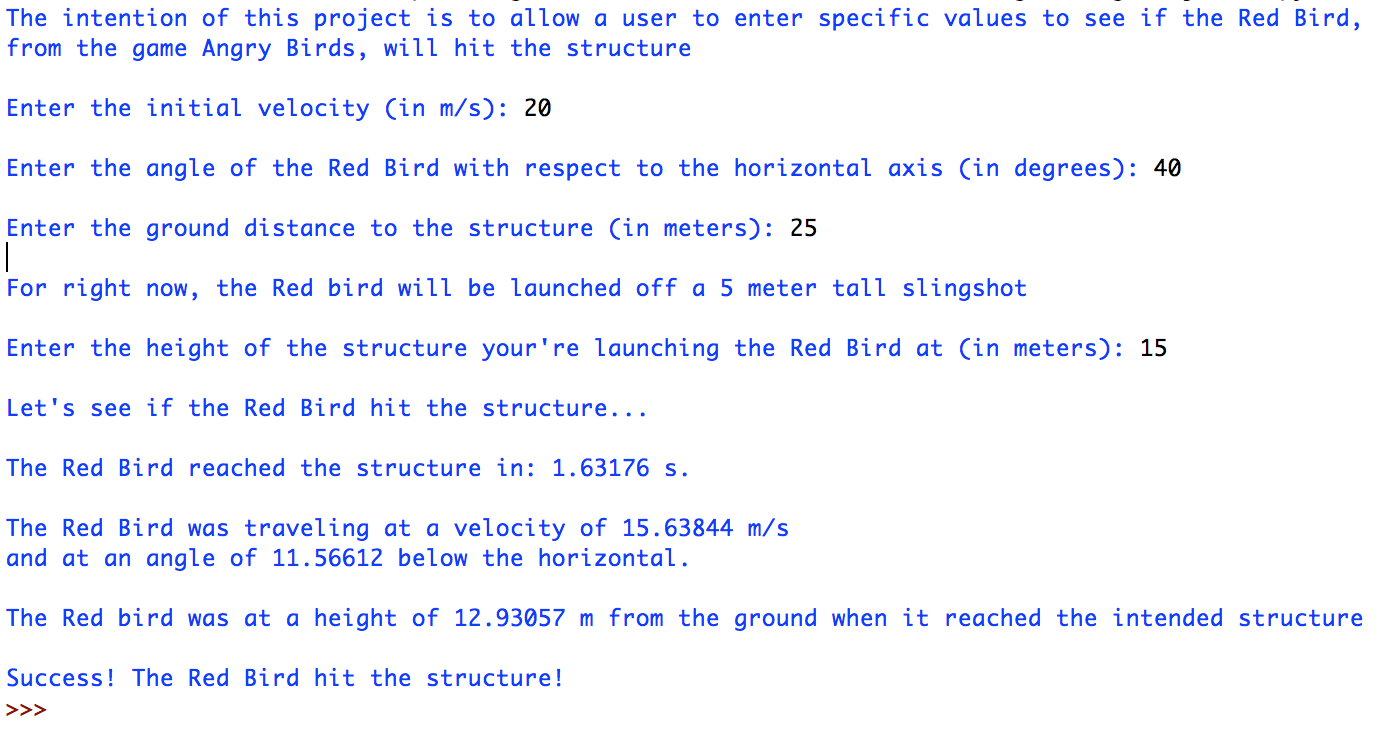
print("\nThe Red Bird went over the structure and missed!")

### ALL FINAL ANSWERS WERE ROUNDED TO 5 DECIMAL PLACES ###

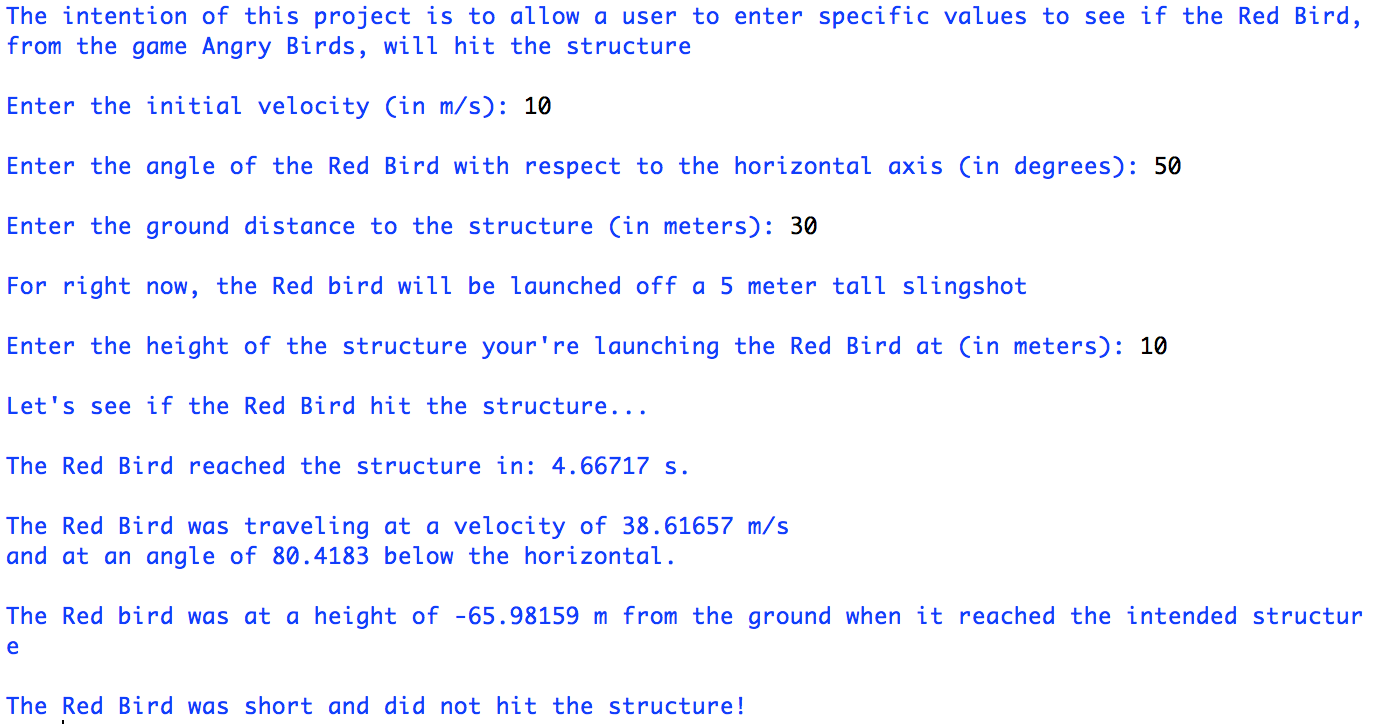
**Sample Runs**

Original Code:

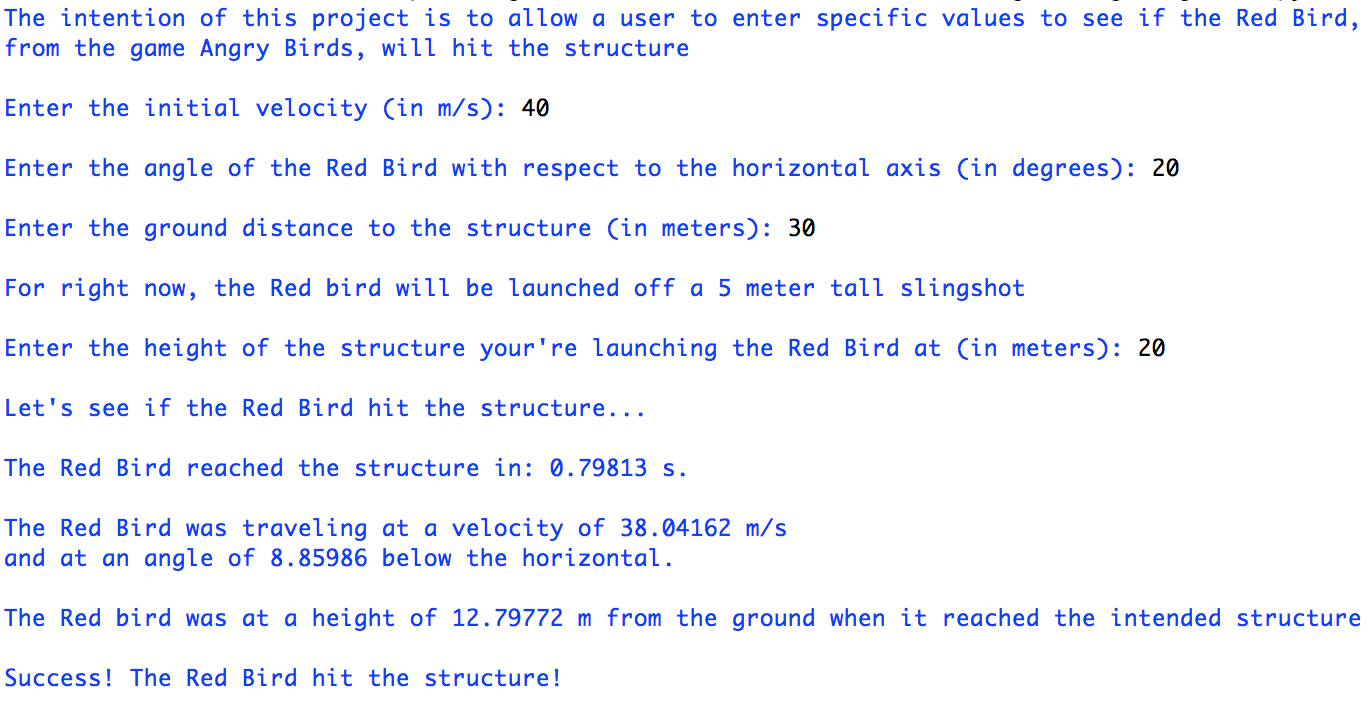
**Test 1**

**Test 2**

**Test 3**

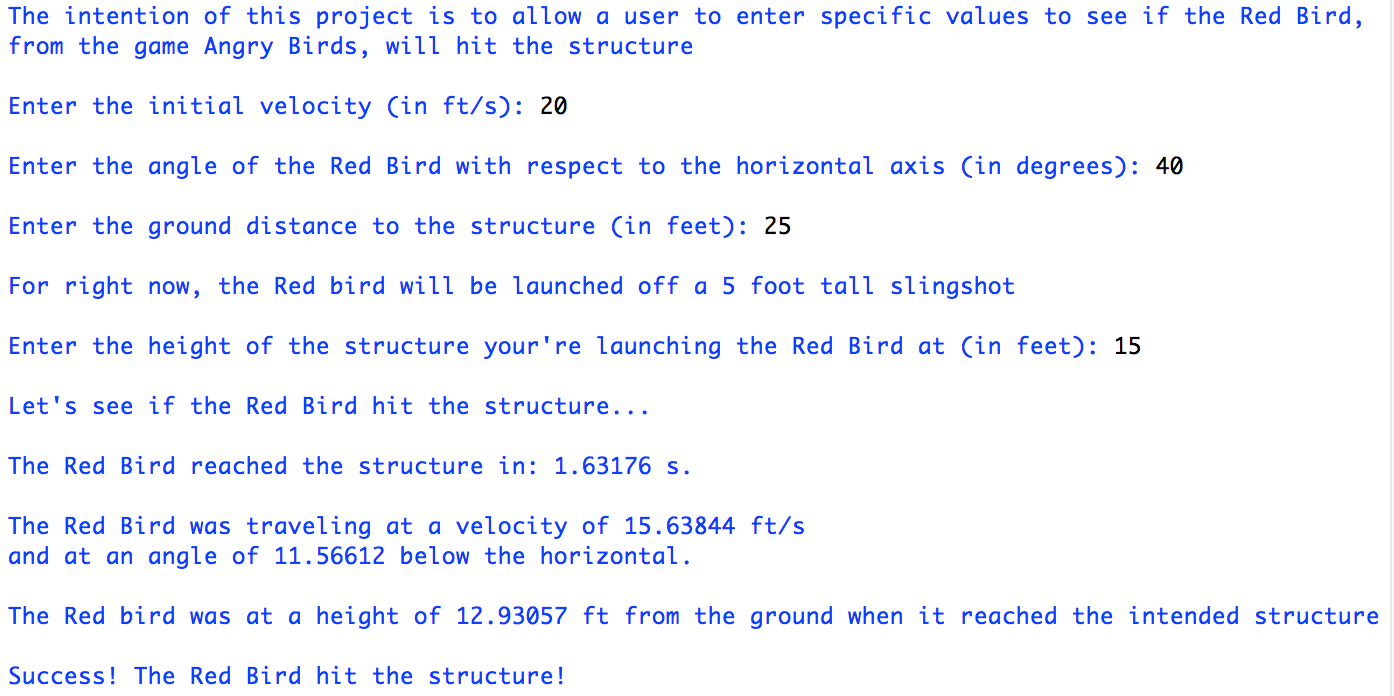
****

**Test 4**

****

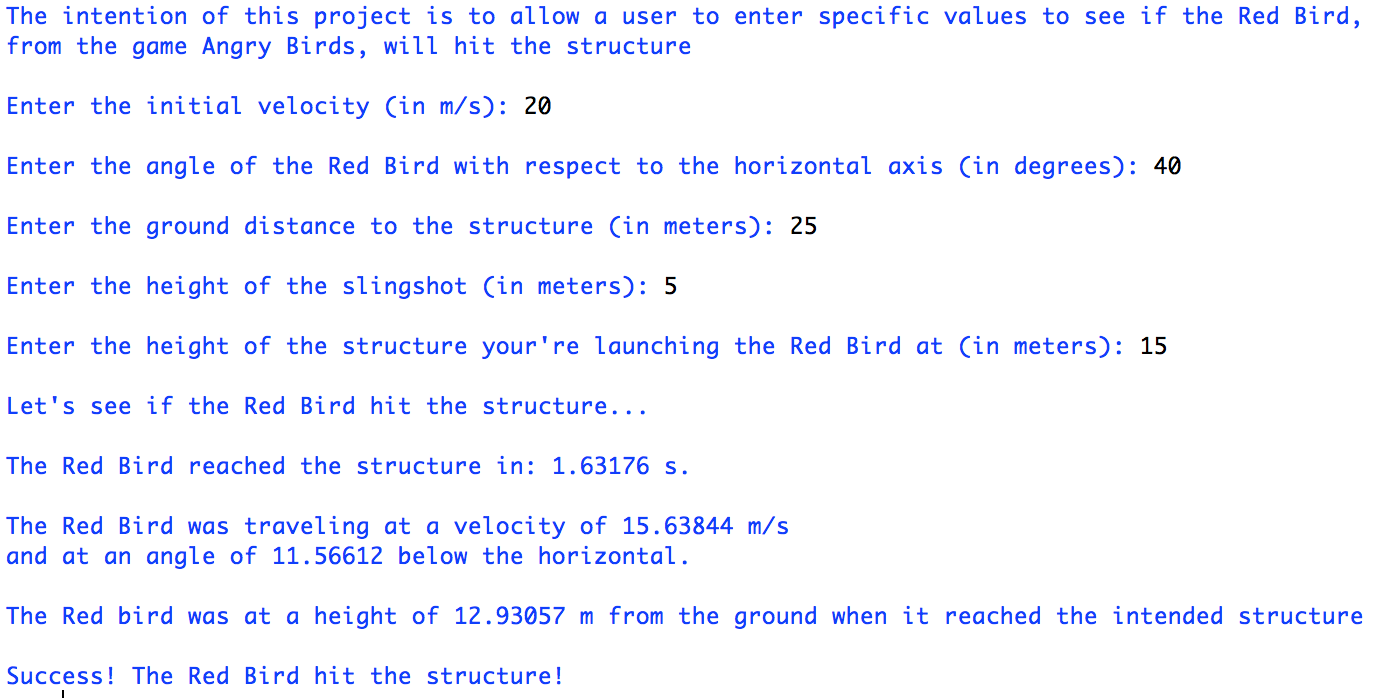
**Unit Switch:**

**Test 1**

****

**User Height Input:**

**Test 1**

****

**Discussion**

Q. How many hours did this program take?

A. This program only took me a little over 3-4 hours total I’d say. Perfecting the code and making sure everything ran smooth was key.

Q. What type of problems did you run into while programming this Project? How did you resolve them?

A. The only true problems were trying to figure out how all the numbers came to be, and how to work out all the equations. Other than that, the project was pretty simple.

Q. How could you enhance the program?

A. You could allow the user to input more information, or expand it beyond a video game and try to see this in a real world instance.