# **ANGRY BIRDS GAME**

## ***Mohammad Anas 22k-4362***

## ***Huzefa Tayyab 22k-4188***

***Section E***

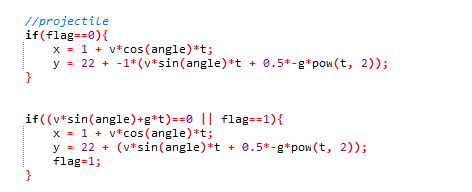
## **Project Report**

The main idea of the project is to develop a user-friendly game which mimics the popular game known among masses, as “Angry Birds”. The game’s story revolves around birds whose eggs are stolen by some pigs. These birds are now in anguish and would have to be launched into the air to strike the pigs off their pedestals.

There are multiple stages in the game, as the user progresses the game becomes more challenging with the increase of pigs.

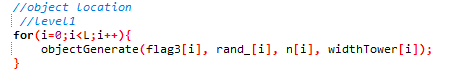
**Main features:**

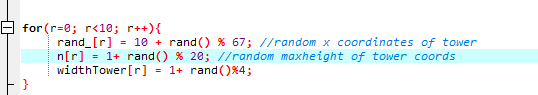
There are a few features which implement the original idea which is of the projectile being launched into the air. As the user selects the speed and angle of the bird the program cleverly calculates the trajectory of the projectile motion of the bird with integrated equations of projectile motion. These equations return values in the form of x and y coordinates which are in turn passed through the *gotoxy() function* to display the bird to the user.

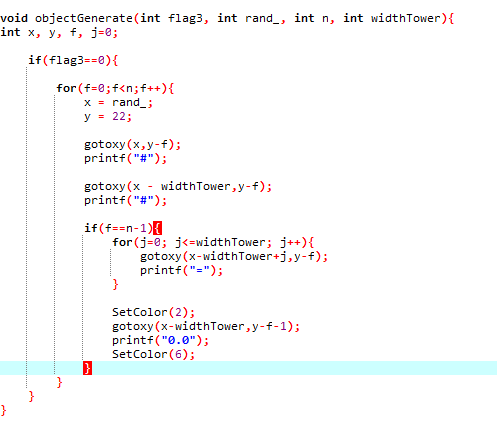


**Primary features:**

Dynamic Tower Generation:





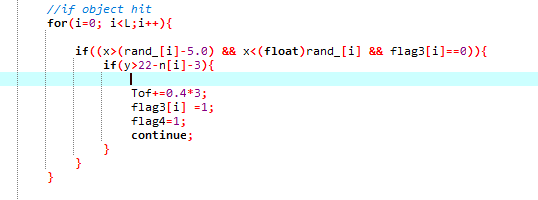


Explanation:

In the above snippets of code, random values of height, width and x-coordinate of generation are allocated to each tower through the use of rand() functions. Then the object generate function is called in a loop to traverse through each object/tower.

In the objectGenerate() function, the particular tower is placed using previously allotted values while y remains 22 to ground these towers. The function runs until counter ‘f’ equals height of tower ‘n’. In each iteration, “#” is printed at x coordinate of tower and (x-width) coordinate where y coordinate reduces to increase height of the tower until f reaches n to complete the tower as a whole. Pigs are then placed at the end of the loop on the top of the tower.

Tower Hit Detection:





Explanation:

In the above code, First, the range is checked between the tower and few units before the tower to identify if x, y coordinates of the bird lie in the region. If these conditions hold true then the objectHit() function is called to show the animation and conduct all the protocols need to be executed in the case tower is hit.

In the objectHit() function, tower is initially generated similar to objectGenerate() function but the end value of loop or height of tower in other words is reduced to debilitate the tower, while “#” is continously printed as the tower falls to give the tower breaking effect.

**Problems encountered:**

* The equations used to perform projectile motions return values in float type however, on the command interface values are printed on discrete locations. For example, value needed to be printed on coordinates (1,1.5) would print on (1,1) which significantly disturbs the motion of the projectile.
* Functions return can only return 1 value which considerably hindered in the process of making the game. Alternate method of using pass by reference would increase the complexity of the code while also increasing the number of loops used increasing execution time, reducing overall efficiency of the game.

**Contributions:**

* Huzefa helped up build up the logic behind projectile motion with equations and different sources referring to it including videos and important equations. Plus,

Modelled and helped visualize the coordinate system and how it would print through gotoxy function.

* Me, Anas did the implementation of the code and integrated those equations plus, connected important functions together as well as linked Frontend to Backend.