Project Design 2

Skeet Shooter

Problem Definition

Have a game that simulates the sport of skeet shooting. There will be a gun in the bottom right corner of the screen, and the birds will come across the screen towards the left. The user moves the gun left and right to shoot the flying bird, every bird hit will increase the score.

Design Overview

Skeet will be displayed in frames to show the movement of the game. The game will start with the gun in the bottom right corner. The gun will be put on the screen from the draw\*() open gl as will all the other shapes. The gun will be able to rotate left and right by pressing the left and right keys. When the right key is pressed the rotation of the rectangle will increase in degrees. The left Arrow will decrease degrees. The gun will accelerate the longer a key is held. This will be done by having a variable called acceleration which is getting increased each frame. When going right the degrees will increase per frame by the variable acceleration. When the key is let go then the acceleration will be reset to 0.

The gun will shoot bullets out of the center of the gun. This is done because the center of the rectangle will already be defined when it is drawn. The bullet will go at the angle the gun is already pointed. Only 5 bullets can be in the play area at a time. The bullets will be assigned a number in an array and when the bullet is shot the variable ammo will be decreased by 1. Ammo will increase when either a pigeon is hit or a bullet leaves the specified play area. When ammo reaches 0 the user will no longer be able to use the space bar to shoot the bullets until ammo is increased.

Clay pigeons will appear at random locations within a minimum and maximum location on the Y axis and will move at a random degree angle between a minimum and maximum degrees. The speed of the pigeon will also will also have a random speed between a minimum and maximum number.

To determine if a bullet hits the distance between the pigeon and bullet will be determined using Pythagorean theory. IF the distance between the two objects center is close enough than the bullet and pigeon will disappear, the left score will increase and the ammo count will be decreased. If the pigeon leaves the play area then the right score will be increased.

Output

|  |  |  |
| --- | --- | --- |
| Element | Image | Description |
| Sample of game |  | Left score will be how many times the play hits the pigeon and the right will be how many times the pigeon successfully escapes. The pigeons will be circle and the gun will be a rectangle. |
| Gun |  | Rectangle gun in bottom right corner |
| Pigeon |  | Pigeon flies from left side of screen to the right at random angles |
| Score |  | Score in top left and right |
| Bullets |  | Consists one pixel that shoots from the center of the gun |

Input

|  |  |
| --- | --- |
| Input | Description |
| Left Arrow Key | Will rotate the gun to the left |
| Right Arrow Key | Will rotate the gun to the right |
| Space Bar | Will shoot bullets from the gun |

Errors

There will be no Errors that will be displayed all other input will be ignored.

Algorithms

bullet BulletFire()

Ammo = 5

If space pressed AND ammo >= 1

Ammo - -

drawBullet()

if bullet.getx() > screenMax OR bullet.gety() > screenMax OR pigeonHitt == true)

Ammo+ +

Draw Pigeon

startLocation = random (screenMin, screenMax)

pigeon.setx =0

pigeon.sety = startLocation

pigeon launcher()

time = randomDouble(0,3)

if (time == currentFrame)

drawPigeon()

dx = random(3, 6)

pigeon.setx += dx

pigeon.sety +=dy

if minDistance() == true

reset() //erase bullet and pigeon

gun rotateGun()

if leftkey == true

accelerationLeft++

rotation -=accelerationLeft

else

accelerationLeft = 1;

if rightkey == true

accelerationRight++

rotation += accelerationRight

else

accelerationRight = 1

Files

There will be no files used in this game.

API

These functions are used as open GL and are found under uidDraw.cpp

|  |  |  |
| --- | --- | --- |
| Name | Description | Parameters |
| drawCircle | Draws a circle on the screen | -center The center of the circle  -radius The radius of the circle  -points How many points the circle will have  -rotation The way the circle is rotating. |
| drawNumber | Draws the numbers on screen | -topLeft Top left corner of rectangle for it to be drawn.  -number The number to draw |
| DrawRect | Draws a rectangle on the screen | -center The center of the rect  -width The total width of it  -height The total height of it  -rotation how much it rotates in degrees |
| drawDot | Draws a point on the screen | -point The location of the dot |
| Random | Generates a random number | -min The minimum boundary  -max The maximum boundary |

These functions are used in uiInteract.

|  |  |  |
| --- | --- | --- |
| Name | Description | Parameters |
| Interface | Sets the game to play | Argc  Argv  title |
| Run | Game starts moving | Callback When called back a new frame is drawn. |
| Key\*Callback | Functions that need to be called | Key  X,y |

Errors

Internal errors will be the only errors in this game.

|  |  |  |
| --- | --- | --- |
| Name | Condition | Handle |
| More than 5 bullets | Ammo < 0 OR Ammo > 5 | Assert |
| Error rotating gun to far | Rotation < 0 OR Rotation > 90 | Assert |
| Pigeon goes out of boundary | Pigeon.center > screen boundary | Assert |

Data Structure

Gun

-gun

-degrees

-acceleration

+draw()

+gunRotate

Pigeon

 Bullet

-pigeon

-dx -dy

-time

 +launcher()

+draw()

+minDistance()

+reset()

 -bullet

-ammo

-dx -dy

 +bulletFire()

+minDistance()

+draw()

 +draw()

+keepScore()

-score1

-score2

 Score