CGM Project

Bouncing Football

Group 8

Bhavesh C : S20190010034

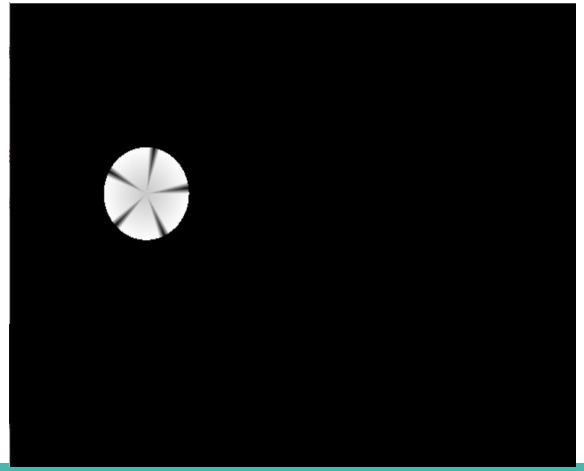
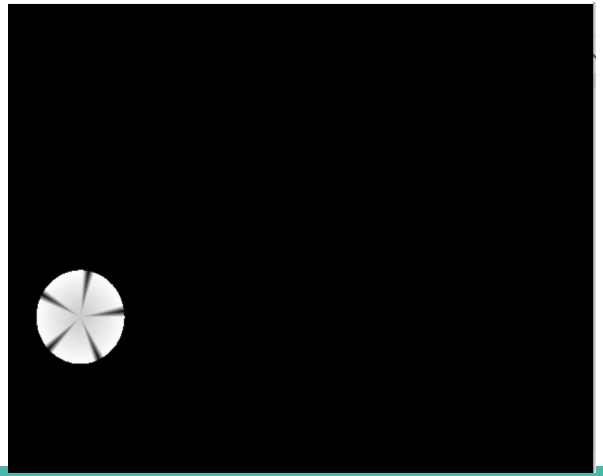
Khadyothan D : S20190010040

Gireesh C : S20190010036

Karthik K: S20190010100

Problem Statement

Implement the bouncing football animation.


Solution

The solution to the problem statement can be achieved by applying relevant transformations, in our case translation. To move further, let’s understand what transformations are and how do we apply them.

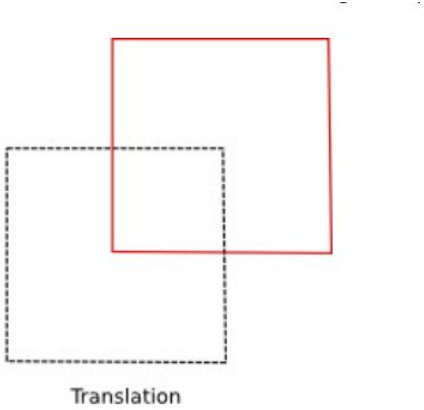
Transformations

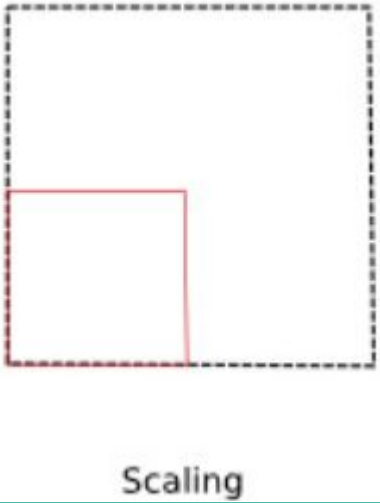
Transformations when applied change the input coordinates to obtain new set of coordinates and the change is dependent upon what we wish to achieve.

Some of the Transformations you know are:

1. Translation
2. Scaling
3. Rotation

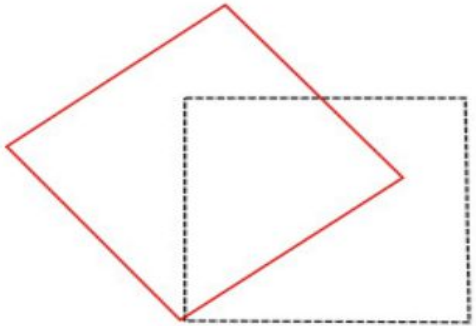
Translation - Given input (x, y) and translation parameters (tx, ty) we get the output (x+tx, y+ty)





Scaling - Given input (x, y) and scaling parameters (a, b) we get the output (ax, by)

Rotation - Given the axis of rotation and angle of rotation we apply this transformation to get the rotated coordinates.



Rotation

We apply these Transformations in our project in the modelling transformation. We use GL_MODELVIEW to start the modelling transformation, in which we will use the Translation accordingly to implement the bouncing.

glTranslatef(x, y, x)

- In Opengl, once we have specified that we work with GL_MODELVIEW mode and load the identity matrix, we specify the translation parameters for the ball's center to move using glTranslatef function.
- It takes in 3 parameters, and moves the the current position by the amount specified by the parameters in their respective directions.
- In our project to move the ball, we have actually moved the center of the ball and then constructed the ball every time we moved the center.

Constructing the Football

- In our project we have worked with 2D Projection. So the shape of football is close to a circle.
- To Construct the circle we have used the GL_TRIANGLE_FAN mode, which uses series of triangles with one common vertex to construct a circle.
- This common vertex is the center of the circle.
- To Specify the other coordinates of the triangles, we use the parametric equation of the circle = (rcos(**θ**), rsin(**θ**))
- First we decide the number of segments of triangles used to construct the circles and decide the vertices of these triangles using the parametric equation.

Football Texture

- To Make the ball look somewhat similar to a football, we tried to apply some texture to it.
- We used the fact that since the circle is constructed by the triangles.
- We applied coloring in such a way that out of every ten triangles we have one triangle which is colored black while the others are colored white.
- We ended up with a texture that looks similar to a football.

The Working

- First we set up some parameters required which specify the ball's characteristics and also the environment in which it moves.
- We specify the radius and position of the ball. Also we specify the boundary conditions namely, Xmax, Xmin, Ymax, Ymin, so that the ball moves within the specified conditions only.
- We set-up a timer which redisplays the display function every 30 milliseconds.
- At the end of every iteration, we calculate the new coordinates of the center for the next iteration and then render these coordinates when the timer expires and the display function is triggered again.

- The ball has speeds (xspeed, yspeed) which specify the speed in which the ball is moving along the respective axes.
- Every time the display function is called, we construct the circle using the center and then calculate the new center of the circle using (xspeed, yspeed), once the time expires and the display function is called again the circle is constructed using this new center and then again the same process is repeated, that is, calculating the center for the next iteration etc.
- We specify the coordinates of the circle center each iteration using the translation function. First we setup the identity matrix and then translate to circle's center obtained in previous iteration.
- We implement the timer functionality using glTimerFunc() function in opengl.


Thank
you