CS101 PROJECT DETAILS:

As the CS101 project, we present the game -"Brick Breaker" , also called "E-Ball" .

Objective :

The user has to break all the bricks on the screen using the striker and a ball provided.

Bricks :

The bricks have been specially made using four polygons . And the color combinations for the bricks have been selected so as to make the bricks appear as in 3D. There are eight different types of bricks i.e of eight different colors.

Striker :

The striker has been presented as rectangular slab which can be controlled by the mouse scroll button. The striker has the limitations to move within the screen along the X-axis at the bottom of the screen only.

Game Screen :

The background of the game screen has been made by imprinting thin rectangles with varying positions and color. The details such as score and levels have been provided. A pause button has also been provided .

Welcome Screen :

The welcome screen has a special animation called -"The Newtons Cradle",
which signifies the transfer of momentum from from one ball to another.
The background of the screen has been made using random circles of random colors and radius.
The game name-"E-BALL" has been written using the concept of polygons.
This screen contains the menu which offers four options.
Play Game : When the user clicks on this button, a new canvas opens which contains the game.
Instructions : This contains the instructions for the user about how to play game.
About : This contains the information about the developers of the game.
Exit : This button lets the user close and exit the game.
Thanking Screen : This screen has a background of random lines with random colors. The word "THANKS" is printed using a moving square which is imprinted at required positions. This is then followed by a new background which is again made using moving rectangles with sinusoidally varying colors.
Finally the names of the developers are shown moving upwards and then disappearing.

Algorithm :

For confinement of the ball within the game screen the condition of the distance of the center of the ball being greater than radius has been used.
And for the collision of the ball with the bricks again the distance of the center of the ball from the sides of the bricks should be just less than the radius of the ball . As soon as the ball collides with the bricks , they disappear and the ball gets new velocity accordingly. For the disappearance of the bricks ,in the "struct brick" a hide function is declared which hides all the component polygons. An array is also defined which serves the purpose of storing whether a particular brick has been hit or not. The condition for the hit brick position is no longer evaluated. A function called "levelsetter" decides how many bricks are there in a given level. It also assigns the bricks a value of -1 which remains till the brick is hit by the ball. When all the bricks are hit the user is said to have won the game and is eligible to proceed to the next level. There are a total of seven levels with increasing difficulty in terms of no. of bricks and their positions. Finally if the user completes all the levels , thanking screen is put up and the game ends.

Initial Problems faced :

- **The striker move mechanism**

    1. **Keyboard mechanism used to get stuck or used to drag.**
    2. **Mouse click mechanism used to effect the efficiency of the program in a similar way as the keyboard mechanism.**
    3. **Hence, we used the scroll mechanism which worked fine.**

- **Coloring the canvas**
    1. **Earlier, we used a rectangle which used to color the canvas while it moved. But this made the program slow as the opening scree used to take around 3 seconds to open up completely.**
    2. **We solved this by using the beginframe and endframe commands.**
- **Colors**
    1. **The mono-colored brick and canvas looked quite awkward, so we knew we had to change it.**
    2. **For bricks, we made 4 polygons for each brick which gave it a 3-dimensional look.**
    3. **For the canvas a sine function as the argument for setColor was used that gave a very nice effect.**
- **NextEvent**
    1. **nextEvent command used to give a very slow response when initiated.**
    2. **Hence, we used checkEvent to correct it.**