National Institute of Technology, Silchar

Department of Computer Science & Engineering

Computer Graphics (B.Tech 4th Sem Project)

On

A Game Based on OpenGL in C++

Submitted by:

Jeet Jain (15-1-051)

Aditya Bhargav (15-1-034)

Vishal Vikram Singh (15-1-)

Monish Revuri (15-1-070)

Submitted To:

Mr. Badal Soni

Assistant Professor, Dept. of CSE

Contents

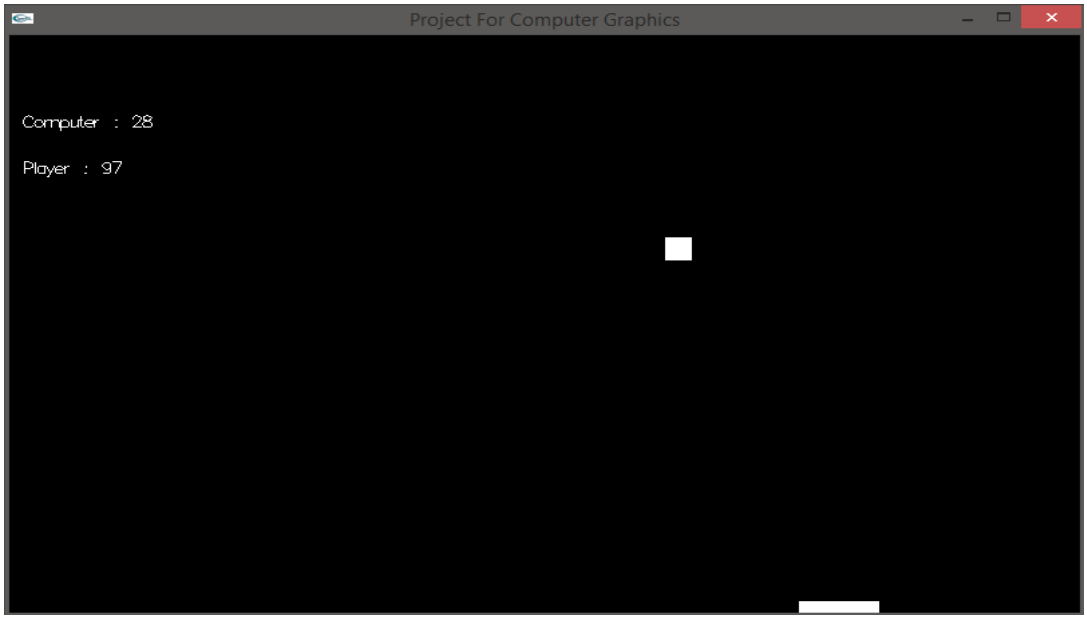
Objective

The objective of designing this game is to learn different features and functions of OpenGL to implement 2D motion and interaction of different objects within the window. The C++ language is used for writing the code.

Acknowledgement

We would like to express our special thanks of gratitude to Mr. Badal Soni Sir who gave us this opportunity to do this wonderful project ‘A Game Based on OpenGL in C++’ which made us doing a lot of research and we came to know about so many new things for which we are really thankful to him.

Introduction to Game



It is a very basic version of Pin Ball game in which the player need to assure that the ball should not touch the bottom wall of window. For doing this he is provided with a horizontal slider which can slide horizontally with the motion of mouse or arrow keys. If ball hits on the slider, score of Player increases otherwise score of Computer increases.

Code Behind the Scene

1. **Drawing The Rectangles:** We used function glVertex2f (float x, float y) inside glBegin (GL_QUADS); and glEnd() to draw the rectangular ball and the player's slider.

```
31
32  void DrawRectangle(RECTA  rect)
33  {
34      glBegin(GL_QUADS);
35      glVertex2f(rect.left, rect.bottom);
36      glVertex2f(rect.right, rect.bottom);
37      glVertex2f(rect.right, rect.top);
38      glVertex2f(rect.left, rect.top);
39      glEnd();
40  }
41
```

2. **Controlling Speed of Ball:** The Timer function take care of speed of ball. The Game as it starts, asks user for speed of ball value of which is fed into glutTimerFunc(int, func, val) which registers a timer callback to be triggered in a specified number of milliseconds.

```
42
43  void Timer(int v)
44  {
45      ball.left += Xspeed;
46      ball.right += Xspeed;
47      ball.top += Yspeed;
48      ball.bottom += Yspeed;
49      glutTimerFunc(speed, Timer, 1);
50  }
51
```

3. **Collision & Reflection of Ball:** The Test_Ball_Wall() function check the coordinates of ball with respect to wall and returns the direction of further motion of ball.

```
67
68  int Test_Ball_Wall(RECTA  ball, RECTA wall)
69  {
70      if (ball.right >= wall.right)
71          return FROM_RIGHT;
72      if (ball.left <= wall.left)
73          return FROM_LEFT;
74      if (ball.top <= wall.top)
75          return FROM_TOP;
76      if (ball.bottom >= wall.bottom)
77          return FROM_BOTTOM;
78
79      else return 0;
80  }
```

These lines in Render function determines the further motion of Ball after collision with wall.

```
189
190     if (Test_Ball_Wall(ball, wall) == FROM_RIGHT)
191         Xspeed = -delta;
192
193     if (Test_Ball_Wall(ball, wall) == FROM_LEFT)
194         Xspeed = delta;
195
196     if (Test_Ball_Wall(ball, wall) == FROM_TOP)
197         Yspeed = delta;
198
199     if (Test_Ball_Wall(ball, wall) == FROM_BOTTOM)
200     {
201         Yspeed = -delta;
202         pcResult += 1;
203     }
204
```

4. **User Input Detection:** The user can control the slider either by mouse or by key board. The inputkey() and MouseMotion() are used to detect the user inputs

```
104     //Key Board Message
105     void inputKey(int key, int x, int y) {
106
107         switch (key) {
108         case GLUT_KEY_LEFT:
109             break;
110         case GLUT_KEY_RIGHT:
111             break;
112         case GLUT_KEY_UP:
113             break;
114         case GLUT_KEY_DOWN:
115             break;
116         }
117     }
129     static int mouse_x = 0;
130     void MouseMotion(int x, int y)
131     {
132         mouse_x = x;
133     }
134
```

5. **Scoring:** If the ball touches the bottom the score of computer increases and if player manages to bring slider between ball and bottom wall, the score of player increases.

```
82  bool Test_Ball_Player(RECTA ball, RECTA player)
83  {
84
85      if (ball.bottom >= player.top && ball.left >= player.left && ball.right <= player.right)
86      {
87          playerResult++;
88          return true;
89      }
90
91      return false;
92  }
93
```

```
187     if (Test_Ball_Wall(ball, wall) == FROM_BOTTOM)
188     {
189         Yspeed = -delta;
190         pcResult += 1;
191     }
```

6. **The Main() Function:** The main() do the basic initialization and calls to the appropriate functions to start their basic utility. Finally, it runs the Game Loop which keeps on running till terminated from inside a function or forcefully.

```
204
205  int main(int argc, char** argv)
206  {
207      std::cout << "Enter the Ball Transition Speed: ";
208      std::cin >> speed;
209      std::cout << "Speed:"<<speed;
210      glutInit(&argc, argv);
211      glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
212      glutInitWindowSize(800, 500);
213      glutInitWindowPosition(0, 0);
214      glutCreateWindow("Project For Computer Graphics");
215      Setting();
216      glutDisplayFunc(Render);
217      glutIdleFunc(Render);
218      glutTimerFunc(1, Timer, 1);
219      glutReshapeFunc(reshape);
220      glutKeyboardFunc(keyboard);
221      glutSpecialFunc(inputKey);
222      glutPassiveMotionFunc(MouseMotion);
223      glutMainLoop();
224      return 0;
225  }
```

Conclusion: The Game runs smoothly on PC with OpenGL graphics. If you find any bug please report us.