

PROJECT REPORT

PADDLE BALL

LAB BATCH 26

ACKNOWLEDGEMENTS

We would like to thank our TA, Zibran Shaikh, for his indispensable help and guidance which were essential to the completion of this project.

We would also like to thank our instructor, Prof. D. B. Phatak, for introducing us to the delightful pursuit of programming.

PROJECT STATUS

The Paddle Game project has been successfully completed. We have achieved the goals we have set up ourselves. With successful teamwork and coordination, we have almost matched it's parent game BattlePong.

Introduction

The Game has been already described in our previous report and the rough draft.

The game involves a ball restricted in a rectangular arena reflecting against the walls. There are two paddles on two opposite sides whose job is to prevent the ball from hitting the wall behind them. When a player fails to protect his wall, his health decreases by a substantial amount. When health of one of the player comes down to zero, the winner will be declared. The game also involves special powers which are invoked on hitting a certain arena of the wall. The player can then use this special power to damage his opponent. An additional feature of score has been included, which reflects the user's proficiency with the game .

The code

1. Variables
2. ApiMain()
3. The computer paddle
4. The ball motion
5. The user paddle
6. Graphics
7. Special powers
8. ApiEnd()

1) **Variables**

All the required libraries have been added at the top. The variables used refer to the various objects in the game.

The character 'in' is used to give input to the user paddle.

'ballx' and 'bally' refer to the x and y coordinates of the ball. vx and vy are the instantaneous velocities of the ball.

'xmin', 'xmax', 'ymin', 'ymax' are the parameters of the game arena.

'padmid' is the middle position of the computer paddle.

H_play and H_comp are health of the user and computer respectively.

vx_max .. bound the range of speed of the ball.

PowerFlag and powerFlag1 refer to the special powers.

x1, x2 denote the area which invokes special powers.

t_total and t_left is for the AI to move the computer paddle.

Padmid2 is the middle position of the user paddle.

Padv2 is computer paddle velocity.

The _strike variables are for the computer paddle to know the position of ball and decide its motion.

Kb_char is for the user paddle to take input.

2. **ApiMain**

In this the paddle game window is opened immediately.

Then the rectangle arena is created and the infinite while loop is opened.

The user paddle motion code to take input continuously from the terminal (without pressing enter) is written.

Then the ballmove and paddlemve functions are called which do the ballmotion and

the computer paddle motion.

Then the health bars are defined for the user as well as the computer.

Then the computer paddle and the ball are drawn and redrawn after frequent intervals.

Then the code for special power starts.

Then there is the code for user paddle motion which calls another function paddlemove2.

3. **The computer paddle**

The paddlemove function in apimain is responsible for the computer paddle motion.

The computer calculates the final position of the ball when it reaches its wall. Then it calculates an appropriate velocity to reach that position and does so.

The computer takes into account all the reflections the ball is going to undergo.

The code is such that the paddle is stationary when ball is going towards the user paddle.

4. **Ball motion**

The ballmove function in apimain governs the ball motion.

The ball collides perfectly against the walls of the arena. It is done by reversing either the x or y component of velocity as required.

But on collision with paddle, there is an element of irregularity (coded to be). The ball on hitting the upper part of the paddle reflects such that it gets a drag in the upward direction. Similarly the lower part.

5. **User paddle**

The code (studied and taken from the internet) takes the input from key board into the terminal. Then it executes the commands as per the key pressed.

But since the code is in an infinite loop, it repeats the process and by pressing a key, the computer keeps on executing till another input is given.

6. **Graphics**

The graphics part includes creating the game window and the arena.

Code is written to draw and redraw the objects at their new positions. The health bars are shown to decrease at every miss.

7. **Special powers**

When the ball hits the special power area, random powers are given to the player.

The special powers included are:

Decrease the paddle length of opponent. (3)

Increase your paddle length. (2)

Reverse the ball direction in midcourse. (1)

A number corresponding to the power you have been granted is printed outside the game area in your court on the top side. On pressing a certain key, your special power is executed, and the number disappears, indicating that the power has been executed.

8. ApiEnd()

The ApiEnd() function is a function to handle any clean up duties that is included in the SimpleWindow class. It is used to display the user's score on the terminal.

CONTROLS

W == user paddle starts moving up.

S == user paddle starts moving down.

Spacebar == user paddle stops.

J == execute special power.

CREDITS

Delay(t) which we have used to delay the processing by t microseconds was copied from

<http://www.newton.dep.anl.gov/askasci/comp99/CS042.htm>

To get continuous input from keyboard while remaining the while (1) loop, we took a block of code from:

<http://www.gidforums.com/t-14843.html>