NAME : AMMAAR AHMAD
ROLL No. - 1801CS08

GAME TITLE : P~~ING~~ADDLE PONG GAME

Different Procedure Used

(i) Clear_Screen
(ii) Move_Ball (Movement of Ball)
(iii) Check_Collision (Collision between paddle and ball)
(iv) Move_Paddle (Movement of Paddles)
(v) Reset_Ball_Position
(vi) Draw_Ball
(vii) Draw_ Paddles

Variables Used

Variable	Purpose
String 1	Game instructions
String 2	
String 3	
String 4	Printing final result of the game
String 5	
String 6	
Window_width	Dimension of window
Window_height	
Window_bounds	Edge checking
Time_Aux	Keep track of time for movement
Ball_Original_X	Original position of the ball at the start of the game
Ball_Original_Y	
Ball_X	Ball position at any instance (Top left corner position
Ball_Y	
Ball_Size	Size of ball (pixel)
Ball_velocity_X	Velocity of ball in x and y directions
Ball_velocity_Y	

Variable	Purpose
Paddle_left_X	Position of top left corner of the left peddle at any instance
Paddle_left_Y	
Paddle_right_X	Position of top left corner of the right peddle at any instance
Paddle_right_Y	
Paddle_width	Dimension of both the paddle
Paddle_height	
Paddle_velocity	Velocity of paddle on key pressed movement (Speed)

Code Segment

Start

(I) Initially printing the instructions of the game and waiting for the user to start the game

(II) Clearing the Screen for game

(III) Check_Time loop to continue it's operation after every 1/100 second. It runs until the ball cross the side boundaries of paddles

(IV) In this loop if 1/100 is not passed then it call itself for next instance
Otherwise various other procedure is called to update the position of ball and paddles and status of the game

(V) If Boundary is crossed by Ball then final result of the game is displayed and game ends

(I) Clear_Screen

- Clear the Screen every instance of time and set the configuration to videomode again
- Background colour is set as black (00H)
- Used to create a optical illusion of moving object

(II) Move_Ball (Movement of Ball)

- Change the position of the ball to new location of x and y coordinates
- If it hits the boundary of the window then reverses the direction of movement
- Call check_collision to check for collision between paddles and ball and reverse the x direction motion if collision occured
- Call Reset_Ball_Position to reset the ball position to center of the screen if it collides with side boundary of the window
  Note :- This feature is kept for future development of this game, to make it a multi point game rather than just one chance. It is not being currently used as the game end as soon as ball crosses the paddle boundaries first time
- Neg_Velocity_Y label to reverse the direction of movement of the ball on collision with top and bottom boundaries

(III) check_collision (Collision between paddle and ball)

- If ball and any paddle collides then ~~we~~ the x direction of movement is reversed
- Here only Axis Alligned Boundary Boxes collisions is being considered

(iv) Move_Paddle (Movement of Paddles)

- If there is no key pressed there is no movement required so exit
- Left paddle can be moved up by ~~[illegible]~~ 'w' or 'S' 'W'
- ~~Re~~ and moved down ~~+~~ by 's' or 'S'
- Right paddle can be moved up by 'o' or 'O' and moved down by 'l' or 'L'
- At one instance of time only one ~~mo~~ movement is allowed (either left paddle or right paddle can move) depending upon which key is pressed first
- If apart from these keys any other key is pressed ~~then~~ then there is no movement

(v) Reset_Ball_Position

- It reset the ball position to the center of the ball when the ball collides with the ~~[illegible]~~ sides boundary of the screen (not paddles)
  Note :- This feature is currently not being used in current version of the game

(vi) Draw_Ball

- Set the initial point of ball location (Top left corner of the ball)
- Expanding it horizontally to get ~~the~~ the shape of line
- Expanding it vertically to get the shape of the ~~sq~~ square ball
- Calling Draw_Ball every instance of time (1/100 second) to draw the ball at new location)

Draw _ Paddle

- Set the initial point of the left paddle (Top left corner of the paddle)
- Expanding it horizontally to get the shape of line
- Expanding it vertically to get the shape of rectangular paddle
- CALL ~~Dr~~ Draw_Paddle every 1/100 second of to draw the peddle at both sides of window. It is being called at every instance because every instance it clear screen is being called
- Similarly for the right paddle the peddle is drawn

Common Instruction Set used (Important Instructions)

	Instruction	Purpose
(i)	LEA DX STRING	⇒ To ~~pu~~ load the string to be displayed
(ii)	MOV AH 09H INT 21H	⇒ To display the content of DX register as a string
(iii)	MOV AH 00H INT 16H	⇒ Waiting for User input in game
(iv)	CALL	⇒ To call various subroutines to perform different functions
(v)	CMP	⇒ To compare various position of ball and peddles for collision ~~and~~
(vi)	JG, JL, JNG, JNL JE, JMP	⇒ Jump on various location conditionally or unconditionally
(vii)	RET	⇒ to return back
(viii)	ADD	⇒ Addition to update the position of ball and peddles
(ix)	MOV AX,	⇒ To move the content to AX register to perform various arithmetic operations as operation on 2 variables can't be performed directly
(x)		

(x) SUB ⇒ To move the paddles up

(xi) NEG BALL_VELOCITY_X,Y ⇒ To change the direction of movement of the ball

(xii) MOV AH, 01H
INT 16H ⇒ To check if any key is being pressed

(xiii) INC ⇒ To drew the ball and paddles

(xiv) MOV AH, 0CH ⇒ Set the configuration to writing in pixel
MOV AL, 0FH ⇒ Choose white as color
MOV BH, 00H ⇒ Set the page number
INT 10H ⇒ Execute the configuration

(xv) MOV AH, 00H ⇒ Set the configuration to video mode
MOV AL, 13H ⇒ Choose the video mode
INT 10H ⇒ Execute the configuration

(xvi) MOV AH, 00H ⇒ Set the configuration
MOV BH, 00H ⇒ To the background color
MOV BL, 00H ⇒ Choose Black as background color
INT 10H ⇒ Execute the configuration

(xvii) CMP, AL ⇒ To check whether the keyword input is 1 out of 4 keys

(xviii) MOV AH, 4CH
INT 21H } To exit program :- Terminate with return code

Suggestion to improve on the game

(i) We can make it a points game rather than one chance game by resetting a ball position to centre and awarding a point to opponent on misses We can introduce display of score of left and right paddle

(ii) It can be played as a multiplayer game between left paddle and right paddle. For this we have to introduce 2 independent input rather than 1 input per check time loop. We can also modify to give the control to player which has to defend the ball at the moment and used control signal for that

(iii) We can modify this game to make it single player and time limit. If player survive for some pre defined time he wins else he lost the game

(iv) We can improve the game to different difficulty by changing the speed of movement of ball as well as paddles. This can be sorted difficulty wise to unlock the game level by level.

HOW TO PLAY THE GAME

Press 'w' or 'W' to move left paddle up
Press 's' or 'S' to move left paddle down
Press 'o' or 'O' to move right paddle up
Press 'l' or 'L' to move right paddle down

AIM: To defend the ball, preventing it from crossing the rectangular area between two paddles (sides boundaries). If the ball crosses the particular paddle, the other paddle (opponent) win's the game.