## Functions Used

**DrawBall**

**void DrawBall(uint8\_t x, uint8\_t y, bool clear)**

Input: x and y coordinates of ball position, type: signed char

Draw or clear parameter, type: boolean

Returns: Nil

Called by:

* PongInit()
* ResetBall()
* MoveBall(uint8\_t x\_direction, uint8\_t y\_direction)

Task

Determines the position of the ball to be drawn or cleared depending on the clear parameter.

True = clear ball.

False = draw ball.

Function uses ClearPixel(x, y) and DrawPixel(x, y) from lcd.h

**DrawPaddleP1(uint8\_t y)**

**void DrawPaddleP1(uint8\_t y)**

Input: y coordinates of paddle 1, type: signed char

Returns: Nil

Called by:

* TestDrawing()
* PongInit()
* MovePaddleP1(uint8\_t direction)

Task

Takes the y-coordinate from the called function. Draws paddle at provided y-axis.

Function uses DrawPixel(x, y) from lcd.h

**DrawPaddleP2(uint8\_t y)**

**void DrawPaddleP2(uint8\_t y)**

Input: y coordinates of paddle 2, type: signed char

Returns: Nil

Called by:

* TestDrawing()
* PongInit()
* MovePaddleP2(uint8\_t direction)

Task

Takes the y-coordinate from the called function. Draws paddle at provided y-axis.

Function uses DrawPixel(x, y) from lcd.h

**DrawScores(uint8\_t score\_p1, uint8\_t score\_p2)**

**void DrawScores(uint8\_t score\_p1, uint8\_t score\_p2)**

Input: Scores of player 1 and 2, type: signed char

Returns: Nil

Called by:

* PongInit()

Task

Draws the scores player 1 and player 2.

Function calls CursorPos(unsigned char x, unsigned char y), PutcharLCD(char c),from lcd.h

**DrawGameOver(uint8\_t is\_p1\_winner)**

**void DrawGameOver(uint8\_t is\_p1\_winner)**

Input: Player status (P1 or P2), type: boolean

Returns: Nil

Called by:

* TestDrawing()
* HandleScores()

Task

Draws “GAME OVER” when called with a logical true argument.

Function calls CursorPos(unsigned char x, unsigned char y) and PutcharLCD(char c) from lcd.h

**PongInit()**

**void PongInit()**

Input: Nil

Returns: Nil

Called by:

* main()
* MoveBall(uint8\_t x\_direction, uint8\_t y\_direction)

Task

Initialises the pong graphics to the screen, used at the beginnig after the all initialisations.

Function calls DrawScores(g\_score\_p1, g\_score\_p2), DrawPaddleP1(g\_p1\_position), DrawPaddleP2(g\_p2\_position), DrawBall(g\_ball\_position\_x, g\_ball\_position\_y, false) in pong.c

**ResetBall()**

**void ResetBall()**

Input: Nil

Returns: Nil

Called by:

* HandleScores()

Task

Resets the ball to the centre of the screen after the ball is missed by any of the players and reaches the end of the screen (i.e: x == 0 or x == LCD\_WIDTH)

Function calls DrawBall(g\_ball\_position\_x, g\_ball\_position\_y, true) in pong.c

**HandleScores()**

**void HandleScores()**

Input: Nil

Returns: Nil

Called by:

* DynamicsTick ()

Task

Keep track of scores, if the ball reaches the left boundary (i.e. x == LCD\_WIDTH) player1 scores a point.If the ball reaches the right boundary (i.e. x==0) player 2 scores a point. If any of the players score 5 (i.e. NUM\_ROUNDS) the game ends.

Function calls DrawGameOver(uint8\_t is\_p1\_winner) and ResetBall().

**HandleScores()**

**void HandleScores()**

Input: Nil

Returns: Nil

Called by:

* DynamicsTick ()

Task

Keep track of scores, if the ball reaches the left boundary (i.e. x == LCD\_WIDTH) player1 scores a point.If the ball reaches the right boundary (i.e. x==0) player 2 scores a point. If any of the players score 5 (i.e. NUM\_ROUNDS) the game ends.

Function calls DrawGameOver(uint8\_t is\_p1\_winner) and ResetBall().

**MovePaddleP1(uint8\_t direction)**

**void MovePaddleP1(uint8\_t direction)**

Input: Direction based on user input

Returns: Nil

Called by:

* main()

Task

Moves paddle up or down based on input argument.

**MovePaddleP2(uint8\_t direction)**

Same as *MovePaddleP2(uint8\_t direction)* for paddle 2.

**MoveBall(uint8\_t x\_direction, uint8\_t y\_direction)**

**void MoveBall(uint8\_t x\_direction, uint8\_t y\_direction)**

Input: Direction of ball to be bounced towards, determined in DynamicsTick()

Returns: Nil

Called by:

* DynamicsTick()
* TestDynamics()

Task

Moves the direction of the ball based on the x and y values provided by DynamicsTick().

**DynamicsTick**

Input: Nil

Returns: Nil

Called by:

* DynamicsTick()
* TestDynamics()

Task

DynamicsTick is continuously called in the main loop to decide whether the ball will rebound vertically straight, diagonally up/down depending on which third of the panel has made contact and the top or bottompanel of the screen set the x (vertical) and y (horizontal) displacement direction, by setting the variables y\_direction and y\_direction.

**DynamicsTick**

Input: Nil

Returns: Nil

Called by:

* DynamicsTick()
* TestDynamics()

Task

DynamicsTick is continuously called in the main loop to decide whether the ball will rebound vertically straight, diagonally up/down depending on which third of the panel has made contact and the top or bottompanel of the screen set the x (vertical) and y (horizontal) displacement direction, by setting the variables y\_direction and y\_direction.