**Breakout Game**

Build

**The ball**

* Use the Sprite library in PyGame to make the Ball
* Set the dimensions, a speed that the ball is going to move at, its initial position, the angle that it will move at and its colour.
* Get its attributes – the corner x, y coordinates

Bounce when collide with the paddle:

* Collisions can be detected using the sprite.collide function in Pygame
* Change the direction, make it move at an angle of 180 subtract the angle it is moving at currently, subtract another angle
* The ‘other angle’ makes the bounce special depend on where on the paddle has the ball collided.

This is calculated by doing:

Centre x coordinate of the paddle – the centre x coordinate of the ball at the time of collision

* If the collision is with left, right or top if the screen, then the ‘other angle’ is zero

Movement:

* Update the x, y coordinate of the corner of the ball in relation to the speed and direction at that instant
* Using Trigonometry:

X coordinate = speed \* sin(angle)

Y coordinate = speed \* cos(angle)

Add these to the current coordinate for it move.

* When a collision happens, the ball should bounce (change direction) and then move as mentioned above
* If the y coordinate of the ball is past that of the paddle (past the bottom edge), the player has lost a life.

**Paddle**

* Use the Sprite library in PyGame to make the Paddle
* Set the dimensions, its initial position and its colour
* Get its attributes – the corner x, y coordinate

Movement

* Get the mouse position and set the x coordinate of the paddle as: the x coordinate of the mouse position add half the width of the paddle. (This way the mouse aligns with the centre of the paddle)
* Y coordinate would be a constant as it can only move horizontally
* If the mouse goes out the screen the paddle should remain at the edge of the screen (when x coordinate > screen width)

**Brick**

* Use the Sprite library in PyGame to make the Brick
* Set the dimensions, its initial position and its colour
* Choose a random colour from a list every time a brick is made
* Get its attributes – the corner x, y coordinate
* Have 8 rows of 20 bricks
* If the ball collides with the brick make the ball bounce and eliminate the brick it touched
* For every eliminated brick the user scores 10 points

**Outputs**

* During the game, display the points the user scores alongside their remaining lives
* At the end of the game (when all lives are lost or all bricks are eliminated), display ‘Game Ended’ and the total points scored

Installing

The program should be converted to a standalone executable application and made available for users to install on their device.