BREAK OUT GAME

```
import  pygame
pygame.init()
HEIGHT=700
WIDTH=700
FPS = 60
COLUMN = 10
ROW = 6

#COLORS
WHITE=(255,255,255) #rgb
BLACK=(0,0,0)
RED=(255,0,0)
GREEN=(80,175,90)
BLUE=(60,160,200)



screen=pygame.display.set_mode((HEIGHT , WIDTH))
pygame.display.set_caption("Breake out game".title())
clock= pygame.time.Clock()
run =True
#paddle class
class Paddle():
   def __init__(self):
       self.width = int(WIDTH/10)
       self.height = 20
       self.x =int(WIDTH/2) - int(self.width/2)
       self.y=HEIGHT -50
       self.speed=10
       self.rectangle=pygame.Rect(self.x,self.y,self.width,self.height)

  # DRAWING PADDLE
   def Draw_paddle(self):
       pygame.draw.rect(screen,WHITE,self.rectangle)
```

```
    # PADDLE MOVEMENT
    def Paddle_Movement(self):
        key= pygame.key.get_pressed()
        if key[pygame.K_LEFT] and self.rectangle.left>0:
            self.rectangle.x -= self.speed
        if key[pygame.K_RIGHT] and self.rectangle.right<WIDTH:
            self.rectangle.x+=self.speed


# BALL CLASS
class Ball():
    def __init__(self, x,y):
        self.radius=10
        self.x = x
        self.y = y
        self .rectangle=pygame.Rect(self.x,self.y,self.radius*2,self.radius*2)
        self.dx = 3
        self.dy = -3
        self.game_status = 0

    # Draw Ball
    def Draw_Ball(self):
         pygame.draw.circle(screen,BLUE,(self.rectangle.x,
self.rectangle.y),self.radius)
    # Ball movement
    def Ball_Movement(self):
        self.rectangle.x += self.dx
        self .rectangle.y  += self.dy

        # wall colession
        if self.rectangle.x <0 or self.rectangle.x >WIDTH:
            self.dx *=-1
        if self.rectangle.y <0 :
            self.dy*= -1
        if self.rectangle.bottom > HEIGHT:
            self.game_status =-1
        if self.rectangle.colliderect(paddle.rectangle) and self.dy >0:
            self.dy *= -1

        #Brick Colision
        all_done = True
        row_num = 0
        for row in Bricks_block.bricks:
            col_num = 0
            for br in row:
                if self.rectangle.colliderect(br):
                    if abs(self.rectangle.bottom - br.top) <5 and self.dy > 0:
                        self.dy *= -1
                    if abs(self.rectangle.top - br.bottom) <5 and self.dy <=0 :
                        self.dy *= -1
                    if abs(self.rectangle.left - br.right) <5 and self.dy <=0:
                        self.dx *= -1
                    if abs(self.rectangle.right - br.left ) <5 and self.dy >0:
                        self.dx *= -1


                    Bricks_block.bricks[row_num][col_num] = [0,0,0,0]
                    if   Bricks_block.bricks[row_num][col_num]  != [0,0,0,0] :
                        all_done = False

                col_num += 1
            row_num += 1



        if all_done:
          self.game_status = 1

        return self.game_status

class Bricks():
        def __init__(self):
            self.width = int(WIDTH/COLUMN)
            self.height =30
        def Create_Bricks(self):
            self.bricks = []
            for row in range (ROW):
                brick_row=[]
                for col in range (COLUMN):
                    brick_x = col * self.width
                    brick_y = row * self.height
                    brick = pygame.Rect(brick_x,brick_y,self.width,self.height)
                    brick_row.append(brick)
                self.bricks.append(brick_row)

        def Draw_Bricks(self):
            for row in self.bricks:
                for br in row:
                    pygame.draw.rect(screen,GREEN,br)
                    pygame.draw.rect(screen, BLACK, br,3)
```

```
paddle=Paddle()
ball=Ball(paddle.x + int(paddle.width/2) ,paddle.y  -11 )
Bricks_block= Bricks()
Bricks_block.Create_Bricks()
while run:
    clock.tick(FPS)
    paddle.Draw_paddle()
    pygame.display.update()
    screen.fill(BLACK)
    paddle.Paddle_Movement()
    ball.Draw_Ball()
    Bricks_block.Draw_Bricks()
    game_status=ball.Ball_Movement()
    if game_status == -1:
        screen.fill(BLACK)
        font= pygame.font.SysFont(None,50)
        text= font.render("GAME OVER",True,BLUE)
        text_rectangle = text.get_rect( center=(WIDTH / 2, HEIGHT / 2 ))
        screen.blit(text,text_rectangle)
    if game_status == 1:
            screen.fill(BLACK)
            font= pygame.font.SysFont(None,50)
            text= font.render(" WIN ",True,BLUE)
            text_rectangle = text.get_rect( center=(WIDTH / 2, HEIGHT / 2 ))
            screen.blit(text,text_rectangle)

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            run=False



pygame.quit()
```