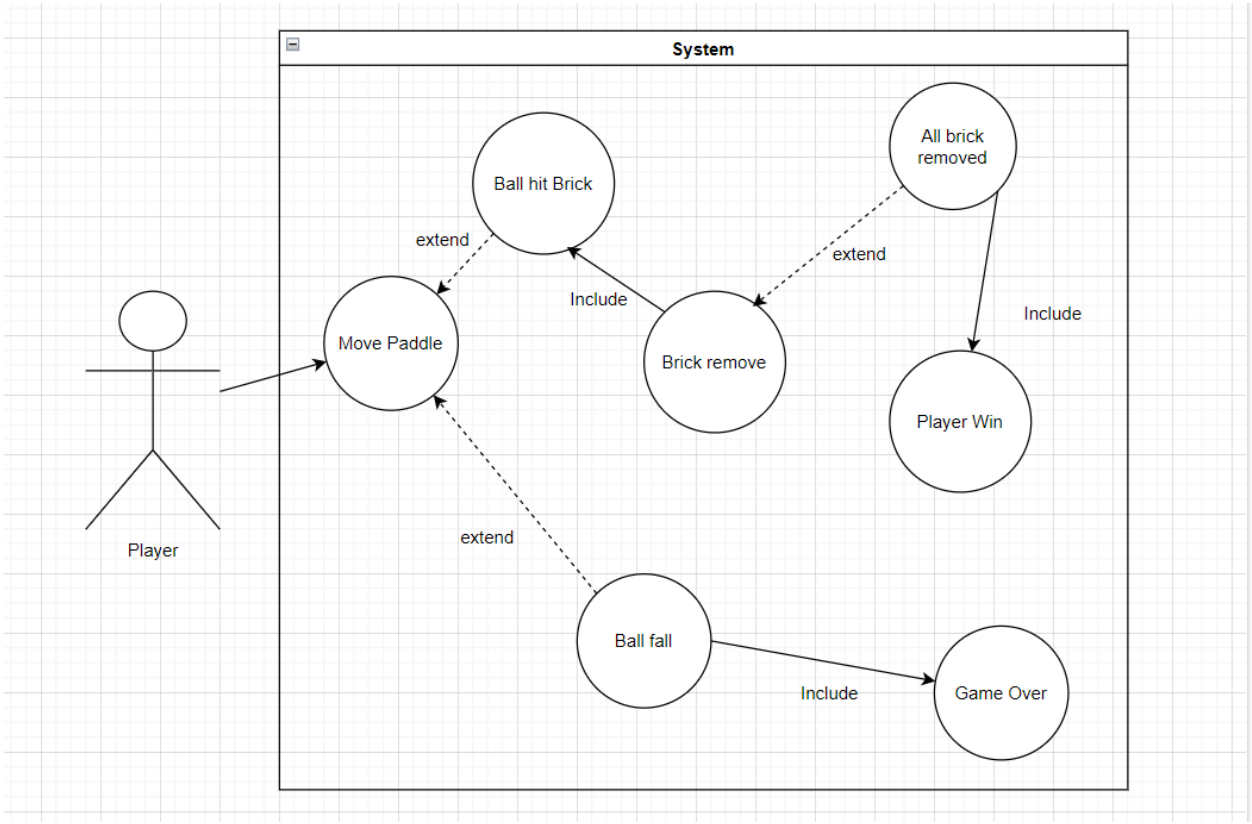
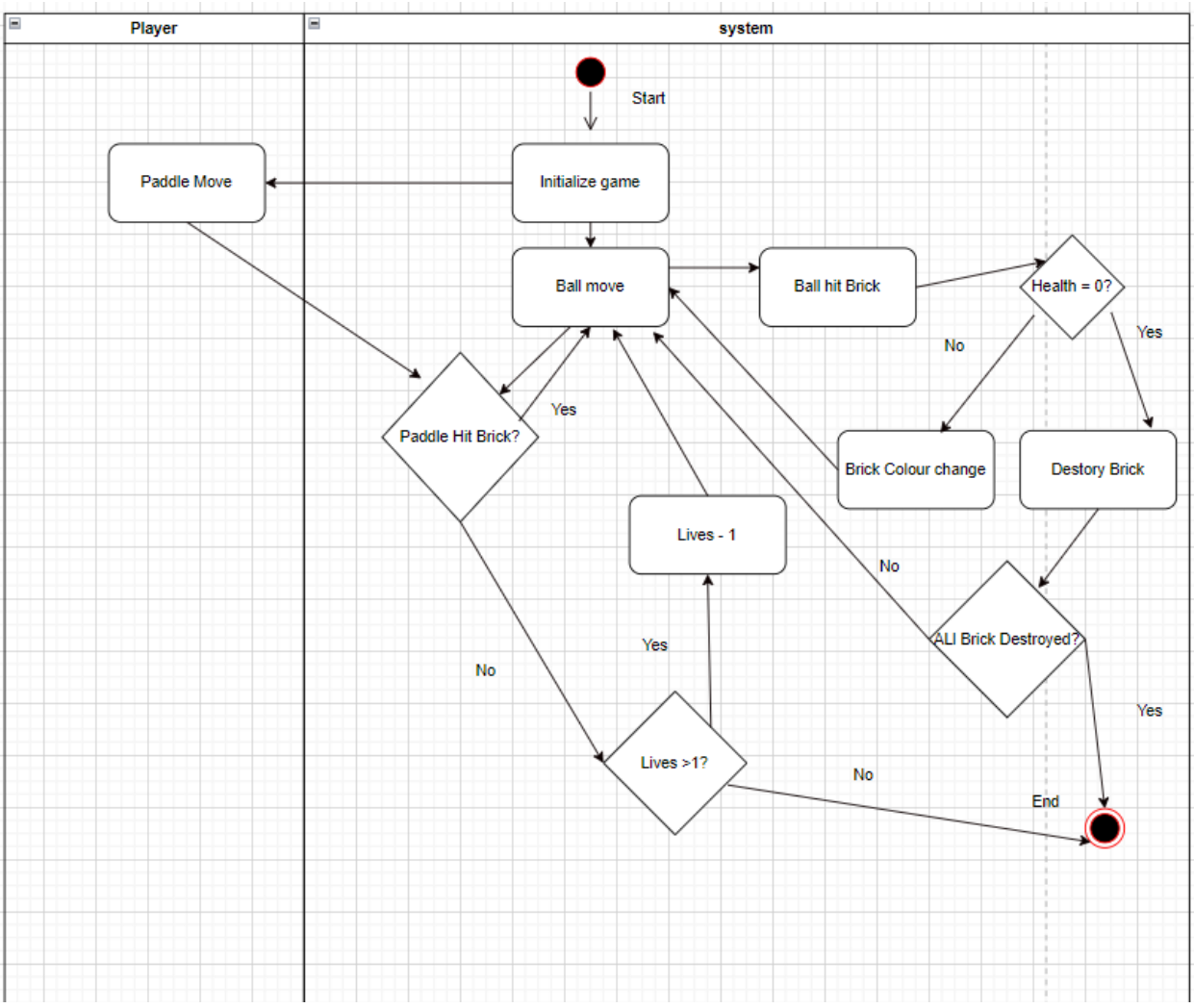
Project Documentation

Project Description:
The program I'm making is a game called Breakout. Basically, Breakout is a game where players have to destroy lines of brick using balls that are bouncing off a paddle. The game objective is to clear all the bricks without making the ball fall over.

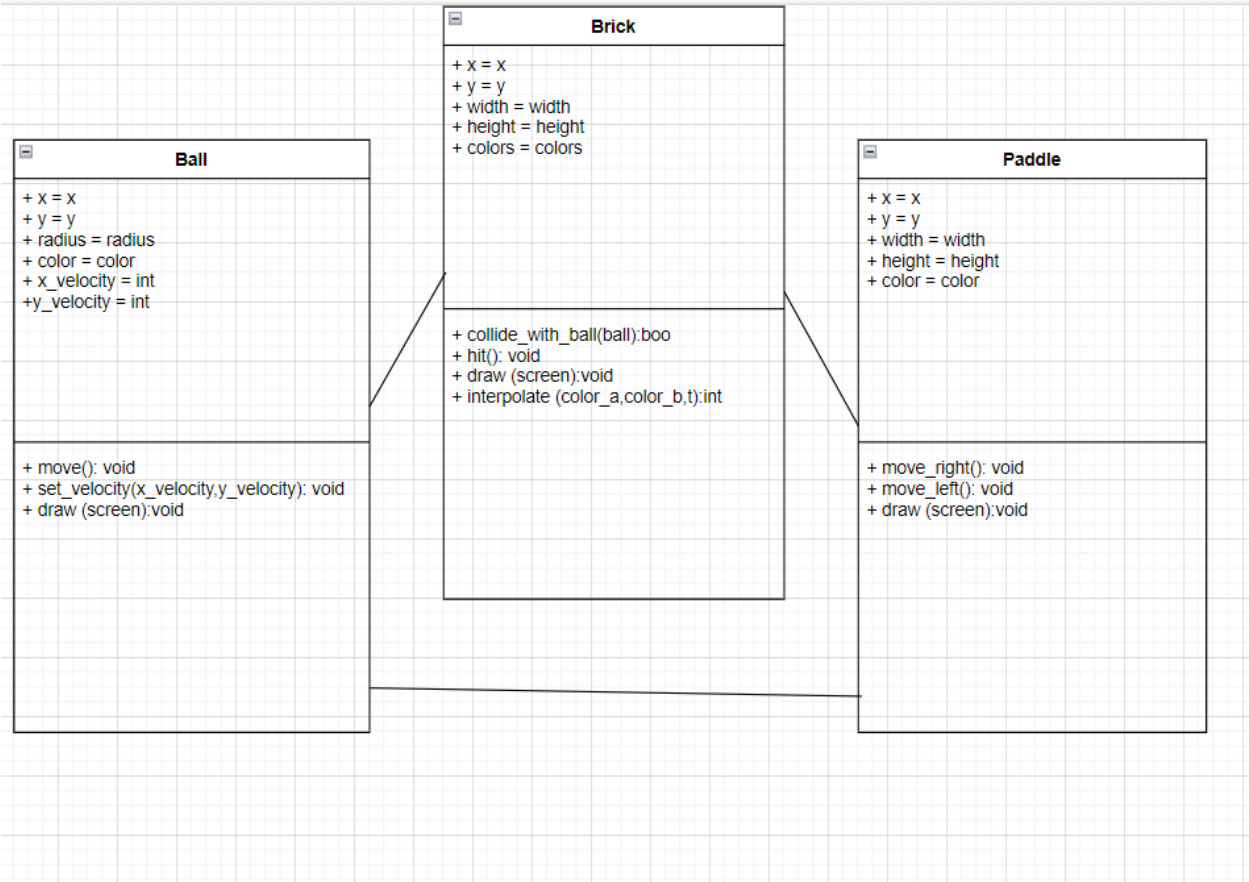
Use-case Diagram:



Activity Diagram:



class Diagram:



Modules:

-Pygame: Pygame is a modules that are used to make game using python language
-Math: Python build-in module which extends the list of mathematical function

Essential Algorithm:

```python
#function to handle ball collision with wall
def ball_collision(ball):

    if ball.x - Ball_radius <= 0 or ball.x + Ball_radius >= screen_Width:
        ball.set_velocity(ball.x_velocity * -1, ball.y_velocity)
    if ball.y + Ball_radius >= screen_Height or ball.y - Ball_radius <= 0:
        ball.set_velocity(ball.x_velocity, ball.y_velocity * -1)


#generate blocks of brick
def generate_bricks(rows, cols):
    gap = 4
    brick_width = screen_Width // cols - gap
    brick_height = 40

    bricks = []
    for row in range(rows):
        for col in range(cols):
            brick = Brick(col * brick_width + gap * col, row * brick_height +
                          gap * row, brick_width, brick_height, 2, [(0, 0, 255), (255, 0, 255)])
            bricks.append(brick)

    return bricks
```

```python
#function to handle the ball and paddle collision
def paddle_ball_collision(paddle,ball):
    if not (ball.x <= paddle.x + paddle.width and ball.x >= paddle.x):
        return
    if not (ball.y + ball.radius >= paddle.y):
        return

    paddle_center = paddle.x + paddle.width/2
    distance_to_center = ball.x - paddle_center

    percent_width = distance_to_center / paddle.width
    angle = percent_width * 90
    angle_radians = math.radians(angle)

    x_velocity = math.sin(angle_radians) * ball.Velocity
    y_velocity = math.cos(angle_radians) * ball.Velocity * -1

    ball.set_velocity(x_velocity, y_velocity)
```

```
#Allocation of keys to move the paddle and stop it if it goes outside of the window
keys = pygame.key.get_pressed()

if keys[pygame.K_LEFT] and paddle.x - paddle.Velocity >= 0:
    paddle.move_left()
if keys[pygame.K_RIGHT] and paddle.x + paddle.width + paddle.Velocity <= screen_Width:
    paddle.move_right()

for brick in bricks[:]:
    brick.collide_with_ball(ball)
    #Remove brick if it's health is zero
    if brick.health <= 0:
        bricks.remove(brick)

# when the ball hit the bottom of the screen the life amount will be reduced
if ball.y + ball.radius >= screen_Height:
    lives -= 1
    ball.x = paddle.x + paddle.width/2
    ball.y = paddle.y - Ball_radius
    ball.set_velocity(0, ball.Velocity * -1)
```

Screenshot of application:


Breakout
Lives: 2


Breakout
Game Over!
Lives: 1

Lesson learned/ reflection:
While working on this project, I've become more familiar with python and pygame. There are still a lot of things that can be added on this project such as level, variety of brick that are spawned and the amount of health that the brick can have. However, my coding skill is not really that great yet so I'm unable to implement those in this project . I'll still need to learn harder in the future so I can code better and create a better program.