

## **Monkey–Banana Problem**

**P Kartikeya Raj**

**2023001492**

### **Aim**

To simulate the Monkey–Banana problem using a goal-based agent in a 2D environment.

### **Agent**

Monkey (goal-based agent)

### **Goal**

Grab the banana and reach the door.

### **Environment**

- Single room
- Fully observable
- Deterministic
- Static
- Discrete (2D)

### **Objects**

Monkey, Box, Banana, Door, Walls

### **Actions**

- Move (left, right, up, down)
- Push box
- Climb box
- Grab banana

### **Initial State**

Monkey and box on the floor. Banana is out of reach.

### **Goal State**

Monkey has the banana and reaches the door.

## Code

```
import pygame
import sys

pygame.init()

# Window
WIDTH, HEIGHT = 800, 500
screen = pygame.display.set_mode((WIDTH, HEIGHT))
pygame.display.set_caption("Monkey Banana Room")
clock = pygame.time.Clock()

# Load images
def load_img(path, size):
    try:
        img = pygame.image.load(path)
        return pygame.transform.scale(img, size)
    except Exception as e:
        print("Image load error:", path)
        pygame.quit()
        sys.exit()

monkey_img = load_img("images/monkey.png", (60, 60))
monkey_banana_img = load_img("images/monkey_banana.png", (60, 60))
box_img = load_img("images/box.png", (80, 80))
banana_img = load_img("images/banana.png", (40, 40))
door_img = load_img("images/door.png", (110, 110))

# Game objects (Rectangles)
monkey = pygame.Rect(50, 350, 60, 60)
box = pygame.Rect(350, 350, 80, 80)
banana = pygame.Rect(380, 120, 40, 40)
```

```
door = pygame.Rect(700, 50, 80, 80)

monkey_on_box = False
has_banana = False

# Walls (room boundary)
walls = [
    pygame.Rect(0, 0, WIDTH, 10),           # top
    pygame.Rect(0, 0, 10, HEIGHT),          # left
    pygame.Rect(0, HEIGHT-10, WIDTH, 10),   # bottom
    pygame.Rect(WIDTH-10, 0, 10, HEIGHT)    # right
]

FONT = pygame.font.SysFont(None, 26)
MOVE = 5

def draw():
    screen.fill((200, 220, 240))

    # Draw walls
    for wall in walls:
        pygame.draw.rect(screen, (100,100,100), wall)

    # Draw objects
    if not has_banana:
        screen.blit(banana_img, banana)
    screen.blit(box_img, box)
    if has_banana:
        screen.blit(monkey_banana_img, monkey)
    else:
        screen.blit(monkey_img, monkey)
    screen.blit(door_img, door)

    # Info text
    info = FONT.render(
        "Arrows Move | P Push | C Climb | G Grab | R Reset",

```

```
        True, (0,0,0)
    )
screen.blit(info, (20, 20))
pygame.display.update()

def reset_game():
    global monkey, box, banana, monkey_on_box, has_banana
    monkey.topleft = (50, 350)
    box.topleft = (350, 350)
    banana.topleft = (380, 120)
    monkey_on_box = False
    has_banana = False

# Collision helper
def check_collision(rect):
    for wall in walls:
        if rect.colliderect(wall):
            return True
    return False

while True:
    clock.tick(60)

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()

    keys = pygame.key.get_pressed()

    dx = dy = 0
    if keys[pygame.K_LEFT]:
        dx = -MOVE
    if keys[pygame.K_RIGHT]:
        dx = MOVE
    if keys[pygame.K_UP]:
```

```

        dy = -MOVE

    if keys[pygame.K_DOWN] :
        dy = MOVE

# Push box and stick monkey

if keys[pygame.K_p] and monkey.colliderect(box) and not monkey_on_box:

    # Try moving box

    box.x += dx

    box.y += dy

    if check_collision(box):

        box.x -= dx

        box.y -= dy

        dx = dy = 0 # prevent monkey from moving into wall

    # Monkey sticks to box

    monkey.x += dx

    monkey.y += dy

else:

    # Move monkey normally

    monkey.x += dx

    monkey.y += dy

    if check_collision(monkey):

        monkey.x -= dx

        monkey.y -= dy

# Climb box

if keys[pygame.K_c] and monkey.colliderect(box):

    monkey_on_box = True

    monkey.bottom = box.top + 10

# Grab banana

if keys[pygame.K_g] and monkey_on_box and monkey.colliderect(banana):

    has_banana = True

# Reset game

if keys[pygame.K_r]:

    reset_game()

```

```

# Check for door (end game)
# Check for door (end game)

if has_banana and monkey.colliderect(door):
    # Display message

    screen.fill((0, 0, 0)) # black background

    message = FONT.render("Monkey got the banana! Game Ends!", True,
(255, 255, 0))

    screen.blit(message, (WIDTH//2 - message.get_width()//2, HEIGHT//2 -
message.get_height()//2))

    pygame.display.update()

    pygame.time.wait(2000) # wait 2 seconds

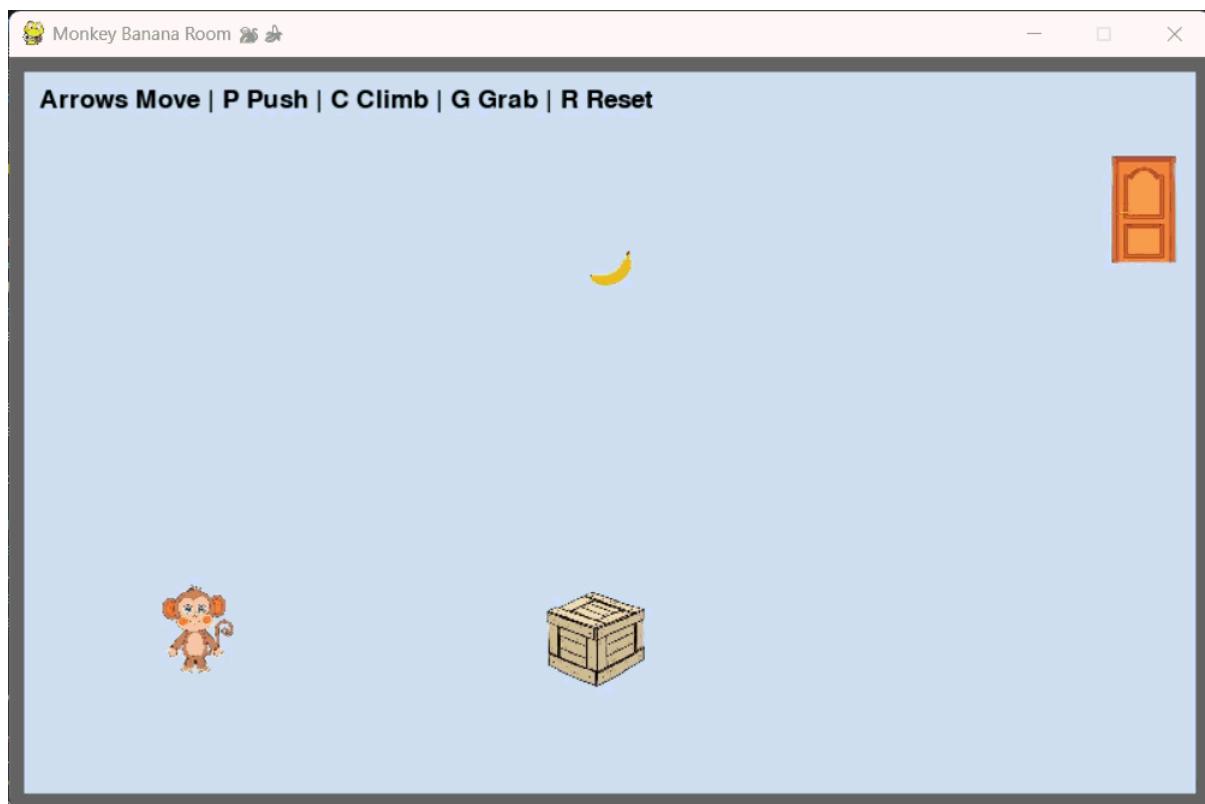
    pygame.quit()

    sys.exit()

draw()

```

## Output



.gif

Gameplay link: [GITHUB](#).