```
IMPORT pygame
IMPORT random

# Initializing pygame
pygame.init()

# Window setup
SET window_width TO 800
SET window_height TO 600
SET window TO pygame.display.set_mode((window_width, window_height))
SET background TO pygame.image.load("background.jpg")
SET background_width TO 3840
SET background_height TO 2160
SET background_default_x TO -1920
SET background_default_y TO -1080

# Window Caption
pygame.display.set_caption("Memory Game")

# Variables
SET clock TO pygame.time.Clock()
SET fps TO 144
SET x_center TO window_width/2
SET y_center TO window_height/2
SET center TO x_center, y_center
SET button_height TO 50
SET INPUT_box_width TO 400
SET INPUT_box_height TO 50
SET character_set TO
"QWERTYUIOPASDFGHJKLZXCVBNMqwertyuiopasdfghjklzxcvbnm1234567890"
SET word TO ""
SET INPUT_text TO ""
SET mouse_position TO pygame.mouse.get_pos()
SET mouse_handled TO False
SET sound_handled TO False
SET sound_handled_position TO [0, 0, 0, 0]

# Fonts
SET largeText TO pygame.font.Font("roboto.ttf", 60)
SET mediumText TO pygame.font.Font("roboto.ttf", 40)
SET smallText TO pygame.font.Font("roboto.ttf", 20)
SET warningText TO pygame.font.Font("roboto.ttf", 10)



# Colours
SET white TO (255, 255, 255)
```

```
SET darker_white TO (150, 150, 150)
SET black TO (0, 0, 0)
SET grey TO (75, 95, 110)
SET dark_grey TO (35, 35, 35)
SET red TO (255, 0, 0)
SET blue TO (15, 30, 50)
SET lighter_blue TO (50, 95, 130)
SET light_blue TO (105, 215, 250)

# Sounds
SET button_hover_sound TO pygame.mixer.Sound("Button Hover.ogg")
SET button_press_sound TO pygame.mixer.Sound("Button press.ogg")


# Text renderer
DEFINE FUNCTION text_objects(text, font, colour):
   SET textSurface TO font.render(text, True, colour)
   RETURN textSurface, textSurface.get_rect()

# Button renderer
DEFINE FUNCTION button(button_text, button_x, button_y, button_width, button_height,
action=None):
   SET mouse_handled AS global
   SET mouse_position AS global
   SET sound_handled AS global
   SET sound_handled_position AS global

   # Mouse events
   IF mouse_position != pygame.mouse.get_pos():
      SET mouse_handled TO True
   SET mouse_position TO pygame.mouse.get_pos()
   SET mouse_click TO pygame.mouse.get_pressed()

   # Sets mouse_handled to false IF mouse isnt clicked
   IF mouse_click[0] EQUALS 0:
      SET mouse_handled TO False

   # If mouse position is not position of button that handled sound

   IF not (sound_handled_position[0] > mouse_position[0] > sound_handled_position[1] and
sound_handled_position[2] > mouse_position[1] > sound_handled_position[3]):
      SET sound_handled TO False

   # Button
   IF button_x + button_width > mouse_position[0] > button_x and button_y +
button_height > mouse_position[1] > button_y:
      IF not sound_handled:
```

```
            SET sound_handled TO True

            # Get position of button that handled the sound
            SET sound_handled_position[0] TO button_x + button_width
            SET sound_handled_position[1] TO button_x
            SET sound_handled_position[2] TO button_y + button_height
            SET sound_handled_position[3] TO button_y

            # Play hover sound
            pygame.mixer.Sound.play(button_hover_sound)

        # Button rectangle
        pygame.draw.rect(window, light_blue, (button_x, button_y, button_width,
button_height))
        pygame.draw.rect(window, lighter_blue, (button_x + 2, button_y + 2, button_width - 4,
button_height - 4))

        # Button text
        SET TextSurf, TextRect TO text_objects(button_text, mediumText, white)
        SET TextRect.center TO (button_x + button_width/2, button_y + button_height/2)
        window.blit(TextSurf, TextRect)

        IF mouse_click[0] EQUALS 1 and action is not None and not mouse_handled:
            # Play press sound
            pygame.mixer.Sound.play(button_press_sound)
            action()
    ELSE:
        # Button rectangle
        pygame.draw.rect(window, lighter_blue, (button_x, button_y, button_width,
button_height))

        # Button text
        SET TextSurf, TextRect TO text_objects(button_text, mediumText, light_blue)
        SET TextRect.center TO (button_x + button_width/2, button_y + button_height/2)
        window.blit(TextSurf, TextRect)

# Splashscreen
DEFINE FUNCTION splashscreen():
    SET timer TO 2 * fps
    WHILE True:
        # If exit button pressed
        FOR event IN pygame.event.get():
            IF event.type EQUALS pygame.QUIT:
                quit()

        # Mouse position
        SET mouse_position TO pygame.mouse.get_pos()
```

```
    SET background_x TO mouse_position[0] + background_default_x
    SET background_y TO mouse_position[1] + background_default_y

    # Background
    window.blit(background, (background_x, background_y))

    # Splashscreen text
    SET TextSurf, TextRect TO text_objects("Sun Woo's Memory Game", largeText,
light_blue)
    SET TextRect.center TO (center)
    window.blit(TextSurf, TextRect)

    # Display update
    pygame.display.update()
    clock.tick(fps)

    # Timer
    timer -= 1
    IF timer EQUALS 0:
      menu()

# Menu
DEFINE FUNCTION menu():
  WHILE True:
    # If exit button pressed
    FOR event IN pygame.event.get():
      IF event.type EQUALS pygame.QUIT:
        quit()
    # Mouse position
    SET mouse_position TO pygame.mouse.get_pos()
    SET background_x TO mouse_position[0] + background_default_x
    SET background_y TO mouse_position[1] + background_default_y

    # Background
    window.blit(background, (background_x, background_y))

    # Menu text
    SET TextSurf, TextRect TO text_objects("Memory Game", largeText, light_blue)
    SET TextRect.center TO (x_center, 50)
    window.blit(TextSurf, TextRect)

    # Start button
    button("Start", 10, 215, 100, button_height, difficulty_selection)

    # Instructions button
    button("Instructions", 10, 275, 220, button_height, instructions)
```

```
        # Quit button
        button("Quit", 10, 335, 80, button_height, quit)

        # Display update
        pygame.display.update()
        clock.tick(fps)

# Select diffuculty
DEFINE FUNCTION difficulty_selection():
    WHILE True:
        # If exit button pressed
        FOR event IN pygame.event.get():
            IF event.type EQUALS pygame.QUIT:
                quit()
        # Mouse position
        SET mouse_position TO pygame.mouse.get_pos()
        SET background_x TO mouse_position[0] + background_default_x
        SET background_y TO mouse_position[1] + background_default_y

        # Background
        window.blit(background, (background_x, background_y))

        # Select diffuculty text
        SET TextSurf, TextRect TO text_objects("Select difficulty", largeText, light_blue)
        SET TextRect.center TO (x_center, 50)
        window.blit(TextSurf, TextRect)

        # Easy button
        button("Easy", x_center - 45, y_center - 100, 90, button_height, easy)

        # Medium button
        button("Medium", x_center - 75, y_center, 150, button_height, medium)

        # Hard button
        button("Hard", x_center - 45, y_center + 100, 90, button_height, hard)

        # Back button
        button("Back", x_center - 350, y_center + 200, 100, button_height, menu)

        # Display update
        pygame.display.update()
        clock.tick(fps)

# Instructions
DEFINE FUNCTION instructions():
    WHILE True:
        # If exit button pressed
```

```
    FOR event IN pygame.event.get():
      IF event.type EQUALS pygame.QUIT:
        quit()
    # Mouse position
    SET mouse_position TO pygame.mouse.get_pos()
    SET background_x TO mouse_position[0] + background_default_x
    SET background_y TO mouse_position[1] + background_default_y

    # Background
    window.blit(background, (background_x, background_y))

    # Instructions text
    SET TextSurf, TextRect TO text_objects("How to play", largeText, light_blue)
    SET TextRect.center TO (x_center, y_center - 200)
    window.blit(TextSurf, TextRect)

    SET TextSurf, TextRect TO text_objects("A random set of characters will be displayed on
the screen FOR a set amount of time.", smallText, light_blue)
    SET TextRect.center TO (x_center, y_center - 100)
    window.blit(TextSurf, TextRect)

    SET TextSurf, TextRect TO text_objects("Easy: 5 seconds", smallText, light_blue)
    SET TextRect.center TO (x_center, y_center - 50)
    window.blit(TextSurf, TextRect)

    SET TextSurf, TextRect TO text_objects("Medium: 3 seconds", smallText, light_blue)
    SET TextRect.center TO (x_center, y_center - 25)
    window.blit(TextSurf, TextRect)

    SET TextSurf, TextRect TO text_objects("Hard: 1 seconds", smallText, light_blue)
    SET TextRect.center TO (x_center, y_center)
    window.blit(TextSurf, TextRect)

    SET TextSurf, TextRect TO text_objects("After the timer is up type the characters IN to
the INPUT box.", smallText, light_blue)
    SET TextRect.center TO (x_center, y_center + 50)
    window.blit(TextSurf, TextRect)

    # Back button
    button("Back", x_center - 350, y_center + 200, 100, button_height, menu)

    # Display update
    pygame.display.update()
    clock.tick(fps)

# Easy
```

```
DEFINE FUNCTION easy():
   random_word()

   global INPUT_text

   SET timer TO 10 * fps

   WHILE True:
      # If exit button pressed
      FOR event IN pygame.event.get():
         IF event.type EQUALS pygame.QUIT:
            quit()

      # Mouse position
      SET mouse_position TO pygame.mouse.get_pos()
      SET background_x TO mouse_position[0] + background_default_x
      SET background_y TO mouse_position[1] + background_default_y

      # Background
      window.blit(background, (background_x, background_y))

      # Display word
      SET TextSurf, TextRect TO text_objects(f"The word is: {word}", largeText, light_blue)
      SET TextRect.center TO (center)
      window.blit(TextSurf, TextRect)

      # Display timer
      SET TextSurf, TextRect TO text_objects(f"Starting IN {round(timer/fps, 1)}...",
mediumText, light_blue)
      SET TextRect.center TO (x_center, y_center + 50)
      window.blit(TextSurf, TextRect)

      # Display update
      pygame.display.update()
      clock.tick(fps)

      # Start game when timer reaches 0
      timer -= 1
      IF timer EQUALS 0:
         SET INPUT_text TO ""
         game()

# Medium
DEFINE FUNCTION medium():
   random_word()

   global INPUT_text
```

```
    SET timer TO 5 * fps

    WHILE True:
        # If exit button pressed
        FOR event IN pygame.event.get():
            IF event.type EQUALS pygame.QUIT:
                quit()
        # Mouse position
        SET mouse_position TO pygame.mouse.get_pos()
        SET background_x TO mouse_position[0] + background_default_x
        SET background_y TO mouse_position[1] + background_default_y

        # Background
        window.blit(background, (background_x, background_y))

        # Display word
        SET TextSurf, TextRect TO text_objects(f"The word is: {word}", largeText, light_blue)
        SET TextRect.center TO (center)
        window.blit(TextSurf, TextRect)

        # Display timer
        SET TextSurf, TextRect TO text_objects(f"Starting IN {round(timer/fps, 1)}...",
mediumText, light_blue)
        SET TextRect.center TO (x_center, y_center + 50)
        window.blit(TextSurf, TextRect)

        # Display update
        pygame.display.update()
        clock.tick(fps)

        # Start game when timer reaches 0
        timer -= 1
        IF timer EQUALS 0:
            SET INPUT_text TO ""
            game()

# Hard
DEFINE FUNCTION hard():
    random_word()

    global INPUT_text

    SET timer TO 3 * fps

    WHILE True:
        # If exit button pressed
```

```
        FOR event IN pygame.event.get():
            IF event.type EQUALS pygame.QUIT:
                quit()
        # Mouse position
        SET mouse_position TO pygame.mouse.get_pos()
        SET background_x TO mouse_position[0] + background_default_x
        SET background_y TO mouse_position[1] + background_default_y

        # Background
        window.blit(background, (background_x, background_y))

        # Display word
        SET TextSurf, TextRect TO text_objects(f"The word is: {word}", largeText, light_blue)
        SET TextRect.center TO (center)
        window.blit(TextSurf, TextRect)

        # Display timer
        SET TextSurf, TextRect TO text_objects(f"Starting IN {round(timer/fps, 1)}...",
mediumText, light_blue)
        SET TextRect.center TO (x_center, y_center + 50)
        window.blit(TextSurf, TextRect)

        # Display update
        pygame.display.update()
        clock.tick(fps)

        # Start game when timer reaches 0
        timer -= 1
        IF timer EQUALS 0:
            SET INPUT_text TO ""
            game()

# Word generation
DEFINE FUNCTION random_word():
    global word

    SET word TO ""

    SET i TO 0
    WHILE i < 6:
        word += random.choice(character_set)
        i += 1

# Game
DEFINE FUNCTION game():
    global INPUT_text
    global clicked
```

```
    SET INPUT_box_selected TO False

    WHILE True:
        # Mouse events
        SET mouse_position TO pygame.mouse.get_pos()

        # Events
        FOR event IN pygame.event.get():
            # If exit button pressed
            IF event.type EQUALS pygame.QUIT:
                quit()

            # Selecting the INPUT box
            IF event.type EQUALS pygame.MOUSEBUTTONDOWN:
                IF x_center + INPUT_box_width/2 > mouse_position[0] > x_center -
INPUT_box_width/2 and y_center + INPUT_box_height/2 > mouse_position[1] > y_center -
INPUT_box_height/2:
                    SET INPUT_box_selected TO True
                ELSE:
                    SET INPUT_box_selected TO False
            # Detect keys
            IF event.type EQUALS pygame.KEYDOWN:
                IF INPUT_box_selected:
                    IF event.key EQUALS pygame.K_BACKSPACE:
                        SET INPUT_text TO INPUT_text[:-1]
                    ELSEIF event.key EQUALS pygame.K_RETURN:
                        confirm()
                    ELSE:
                        IF len(INPUT_text) < 6:
                            INPUT_text += event.unicode

        # Mouse position
        SET mouse_position TO pygame.mouse.get_pos()
        SET background_x TO mouse_position[0] + background_default_x
        SET background_y TO mouse_position[1] + background_default_y

        # Background
        window.blit(background, (background_x, background_y))

        # Title text
        SET TextSurf, TextRect TO text_objects("Type the word below", mediumText,
light_blue)
        SET TextRect.center TO (x_center, y_center - 200)
        window.blit(TextSurf, TextRect)

        # Enter button
```

```
        button("Enter", x_center - 50, y_center + 200, 100, button_height, confirm)

        # Input box
        IF INPUT_box_selected:
            # Box
            pygame.draw.rect(window, light_blue, (x_center - INPUT_box_width/2, y_center -
INPUT_box_height/2, INPUT_box_width, INPUT_box_height))
            pygame.draw.rect(window, grey, (x_center - (INPUT_box_width - 4)/2, y_center -
(INPUT_box_height - 4)/2, INPUT_box_width - 4, INPUT_box_height - 4))

            # Text
            SET TextSurf, TextRect TO text_objects(INPUT_text, smallText, white)
            SET TextRect.center TO (x_center, y_center)
            window.blit(TextSurf, TextRect)
        ELSE:
            # Box
            pygame.draw.rect(window, dark_grey, (x_center - INPUT_box_width/2, y_center -
INPUT_box_height/2, INPUT_box_width, INPUT_box_height))

            # Text
            SET TextSurf, TextRect TO text_objects(INPUT_text, smallText, white)
            SET TextRect.center TO (x_center, y_center)
            window.blit(TextSurf, TextRect)

            # Warning
            SET TextSurf, TextRect TO text_objects("Warning: Text box is not selected",
warningText, red)
            SET TextRect.center TO (x_center, y_center + 40)
            window.blit(TextSurf, TextRect)

        # Display update
        pygame.display.update()
        clock.tick(fps)

# Confirm
DEFINE FUNCTION confirm():
    global word

    global INPUT_text

    IF INPUT_text EQUALS word:
        correct()
    ELSE:
        incorrect()

# Correct
DEFINE FUNCTION correct():
```

```
    WHILE True:
        # If exit button pressed
        FOR event IN pygame.event.get():
            IF event.type EQUALS pygame.QUIT:
                quit()

        # Mouse position
        SET mouse_position TO pygame.mouse.get_pos()
        SET background_x TO mouse_position[0] + background_default_x
        SET background_y TO mouse_position[1] + background_default_y

        # Background
        window.blit(background, (background_x, background_y))

        # Display correct
        SET TextSurf, TextRect TO text_objects("Correct", largeText, light_blue)
        SET TextRect.center TO (center)
        window.blit(TextSurf, TextRect)

        # Menu button
        button("Menu", 50, 500, 110, button_height, menu)

        # Quit button
        button("Quit", 660, 500, 90, button_height, quit)

        # Display update
        pygame.display.update()
        clock.tick(fps)

# Incorrect

DEFINE FUNCTION incorrect():

    global word



    WHILE True:
        # If exit button pressed
        FOR event IN pygame.event.get():
            IF event.type EQUALS pygame.QUIT:
                quit()

        # Mouse position
        SET mouse_position TO pygame.mouse.get_pos()
        SET background_x TO mouse_position[0] + background_default_x
        SET background_y TO mouse_position[1] + background_default_y
```

```
        # Background
        window.blit(background, (background_x, background_y))

        # Display incorrect
        SET TextSurf, TextRect TO text_objects("Incorrect", largeText, light_blue)
        SET TextRect.center TO (center)
        window.blit(TextSurf, TextRect)

        # Display answer
        SET TextSurf, TextRect TO text_objects(f"The correct answer was {word}", smallText,
light_blue)
        SET TextRect.center TO (x_center, y_center + 50)
        window.blit(TextSurf, TextRect)

        # Menu button
        button("Menu", 50, 500, 110, button_height, menu)

        # Quit button
        button("Quit", 660, 500, 90, button_height, quit)

        # Display update
        pygame.display.update()
        clock.tick(fps)

# Start splashscreen
splashscreen()

# Quit
DEFINE FUNCTION quit():
   pygame.quit()
```