```
import pygame
import random

# Initializing pygame
pygame.mixer.pre_init(44100, -16, 8, 2048)
pygame.mixer.init()
pygame.init()

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

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

# Variables
clock = pygame.time.Clock()
fps = 60
x_center = window_width/2
y_center = window_height/2
center = x_center, y_center
button_height = 50
input_box_width = 400
input_box_height = 50
character_set =
"QWERTYUIOPASDFGHJKLZXCVBNMqwertyuiopasdfghjklzxcvbnm1234567890"
int_list = "1234567890"
word = ""
input_text = ""
timer_text = ""
mouse_position = pygame.mouse.get_pos()
mouse_handled = False
sound_handled = False
sound_handled_position = [0, 0, 0, 0]

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

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

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


# Text renderer
def text_objects(text, font, colour):
    textSurface = font.render(text, True, colour)
    return textSurface, textSurface.get_rect()


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

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

    # Sets mouse_handled to false if mouse isnt clicked
    if mouse_click[0] == 0:
        mouse_handled = 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]):
        sound_handled = 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:
            sound_handled = True
            # Get position of button that handled the sound
            sound_handled_position[0] = button_x + button_width
            sound_handled_position[1] = button_x
            sound_handled_position[2] = button_y + button_height
            sound_handled_position[3] = button_y
            # Play hover sound
            pygame.mixer.fadeout(800)
            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
        TextSurf, TextRect = text_objects(button_text, mediumText, white)
        TextRect.center = (button_x + button_width/2, button_y + button_height/2)
        window.blit(TextSurf, TextRect)

        if mouse_click[0] == 1 and action is not None and not mouse_handled:
            # Play press sound
            mouse_handled = True
            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
        TextSurf, TextRect = text_objects(button_text, mediumText, light_blue)
        TextRect.center = (button_x + button_width/2, button_y + button_height/2)
        window.blit(TextSurf, TextRect)


# Splashscreen
def splashscreen():
    timer = 2 * fps
    while True:
        # If exit button pressed
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                quit()

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

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

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

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

        # Timer
        timer -= 1
        if timer == 0:
            menu()


# Menu
def menu():
    while True:
        # If exit button pressed
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                quit()

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

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

        # Menu text
        TextSurf, TextRect = text_objects("Memory Game", largeText, light_blue)
        TextRect.center = (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
def difficulty_selection():
    while True:
        # If exit button pressed
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                quit()

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

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

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

        # Easy button
        button("Easy", x_center - 45, y_center - 150, 90, button_height, easy)

        # Medium button
        button("Medium", x_center - 75, y_center - 50, 150, button_height, medium)

        # Hard button
        button("Hard", x_center - 45, y_center + 50, 90, button_height, hard)

        # Custom button
        button("Custom", x_center - 75, y_center + 150, 150, button_height, custom_input)

        # Back button
        button("Back", x_center - 350, y_center + 200, 100, button_height, menu)
        # Display update
        pygame.display.update()
        clock.tick(fps)


# Instructions
def instructions():
    while True:
        # If exit button pressed
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                quit()

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

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

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

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

        TextSurf, TextRect = text_objects("Easy: 10 seconds", smallText, light_blue)
        TextRect.center = (x_center, y_center - 50)
        window.blit(TextSurf, TextRect)

        TextSurf, TextRect = text_objects("Medium: 5 seconds", smallText, light_blue)
        TextRect.center = (x_center, y_center - 25)
        window.blit(TextSurf, TextRect)

        TextSurf, TextRect = text_objects("Hard: 3 seconds", smallText, light_blue)
        TextRect.center = (x_center, y_center)
        window.blit(TextSurf, TextRect)

        TextSurf, TextRect = text_objects("Custom: Set your own timer", smallText, light_blue)
        TextRect.center = (x_center, y_center + 25)
        window.blit(TextSurf, TextRect)
        TextSurf, TextRect = text_objects("After the timer is up type the characters into the
input box.", smallText, light_blue)
        TextRect.center = (x_center, y_center + 75)
        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
def easy():
    random_word()

    global input_text

    timer = 10 * fps

    while True:
        # If exit button pressed
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                quit()
            if event.type == pygame.KEYDOWN:
                if event.key == pygame.K_RETURN:
                    confirm()

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

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

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

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

        # Skip button
        button("Skip", x_center - 50, y_center + 100, 100, button_height, game)

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

        # Start game when timer reaches 0
        timer -= 1
        if timer == 0:
            game()


# Medium
def medium():
    random_word()

    global input_text

    timer = 5 * fps

    while True:
        # If exit button pressed
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                quit()
            if event.type == pygame.KEYDOWN:
                if event.key == pygame.K_RETURN:
                    confirm()

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

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

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

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

        # Skip button
        button("Skip", x_center - 50, y_center + 100, 100, button_height, game)

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

        # Start game when timer reaches 0
        timer -= 1
        if timer == 0:
            game()


# Hard
def hard():
    random_word()

    global input_text

    timer = 3 * fps

    while True:
        # If exit button pressed
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                quit()
            if event.type == pygame.KEYDOWN:
                if event.key == pygame.K_RETURN:
                    confirm()

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

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

        # Display word
        TextSurf, TextRect = text_objects(f"The word is: {word}", largeText, light_blue)
        TextRect.center = (center)
        window.blit(TextSurf, TextRect)
        # Display timer
        TextSurf, TextRect = text_objects(f"Starting in {round(timer/fps, 1)}...", mediumText,
light_blue)
        TextRect.center = (x_center, y_center + 50)
        window.blit(TextSurf, TextRect)

        # Skip button
        button("Skip", x_center - 50, y_center + 100, 100, button_height, game)

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

        # Start game when timer reaches 0
        timer -= 1
        if timer == 0:
            game()


# Custom input
def custom_input():
    global timer_text
    global clicked

    timer_text = ""

    timer_box_selected = False

    while True:
        # Mouse events
        mouse_position = pygame.mouse.get_pos()

        # Events
        for event in pygame.event.get():
            # If exit button pressed
            if event.type == pygame.QUIT:
                quit()
            # Selecting the input box
            if event.type == 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:
                    timer_box_selected = True
                else:
                    timer_box_selected = False
            # Detect keys
            if event.type == pygame.KEYDOWN:
                if timer_box_selected:
                    if event.key == pygame.K_BACKSPACE:
                        timer_text = timer_text[:-1]
                    elif event.key == pygame.K_RETURN:
                        custom()
                    elif str(event.unicode) in int_list:
                        if len(timer_text) < 2:
                            timer_text += event.unicode

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

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

        # Title text
        TextSurf, TextRect = text_objects("Type in amount", mediumText, light_blue)
        TextRect.center = (x_center, y_center - 200)
        window.blit(TextSurf, TextRect)

        # Enter button
        button("Confirm", x_center - 73, y_center + 200, 146, button_height, custom)

        # Timer box
        if timer_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
            TextSurf, TextRect = text_objects(timer_text, smallText, white)
            TextRect.center = (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
            TextSurf, TextRect = text_objects(timer_text, smallText, white)
            TextRect.center = (x_center, y_center)
            window.blit(TextSurf, TextRect)
            # Warning
            TextSurf, TextRect = text_objects("Warning: Text box is not selected", warningText, red)
            TextRect.center = (x_center, y_center + 40)
            window.blit(TextSurf, TextRect)

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


# Custom
def custom():
    random_word()

    global timer_text
    if timer_text == "":
        timer_text = "10"

    timer = int(timer_text) * fps

    while True:
        # If exit button pressed
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                quit()
            if event.type == pygame.KEYDOWN:
                if event.key == pygame.K_RETURN:
                    confirm()

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

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

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

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

        # Skip button
        button("Skip", x_center - 50, y_center + 100, 100, button_height, game)

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

        # Start game when timer reaches 0
        timer -= 1
        if timer == 0:
            game()


# Word generation
def random_word():
    global word
    word = ""

    i = 0
    while i < 6:
        word += random.choice(character_set)
        i += 1


# Game
def game():
    global input_text
    global clicked

    input_text = ""

    input_box_selected = False

    while True:
        # Mouse events
        mouse_position = pygame.mouse.get_pos()

        # Events
        for event in pygame.event.get():
            # If exit button pressed
            if event.type == pygame.QUIT:
                quit()
            # Selecting the input box
            if event.type == 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:
          input_box_selected = True
        else:
          input_box_selected = False
      # Detect keys
      if event.type == pygame.KEYDOWN:
        if input_box_selected:
          if event.key == pygame.K_BACKSPACE:
            input_text = input_text[:-1]
          elif event.key == pygame.K_RETURN:
            confirm()
          else:
            if len(input_text) < 6:
              input_text += event.unicode

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

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

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

    # Confirm button
    button("Confirm", x_center - 73, y_center + 200, 145, 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
      TextSurf, TextRect = text_objects(input_text, smallText, white)
      TextRect.center = (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
            TextSurf, TextRect = text_objects(input_text, smallText, white)
            TextRect.center = (x_center, y_center)
            window.blit(TextSurf, TextRect)

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

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


# Confirm
def confirm():
    global word
    global input_text

    if input_text == word:
        correct()
    else:
        incorrect()


# Correct
def correct():
    while True:
        # If exit button pressed
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                quit()

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

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

        # Display correct
        TextSurf, TextRect = text_objects("Correct", largeText, light_blue)
        TextRect.center = (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
def incorrect():
    global word

    while True:
        # If exit button pressed
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                quit()

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

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

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

        # Display answer
        TextSurf, TextRect = text_objects(f"The correct answer was {word}", smallText,
light_blue)
        TextRect.center = (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
def quit():
    pygame.quit()
```