```
import pygame
import random
from pygame import mixer


class Menu:
    def __init__(self, game):
        mixer.music.load('audio.mp3')
        mixer.music.play(-1)
        self.game = game
        self.mid_w, self.mid_h = self.game.DISPLAY_w / 2,
self.game.DISPLAY_h / 2
        self.run_display = True
        self.cursor_rect = pygame.Rect(0, 0, 20, 20)
        self.offset = -100

    def draw_cursor(self):
        self.game.draw_text('*', 15, self.cursor_rect.x,
self.cursor_rect.y)
        pygame.display.update()
        self.game.reset_keys()

    def blit_screen(self):
        self.game.window.blit(self.game.display, (0, 0))
        pygame.display.update()
        self.game.reset_keys()

class Round1Menu(Menu):
    def __init__(self, game):
        super().__init__(game)
        self.secret_number = None

    def play_game(self, level):
        if level == 1:
            self.secret_number = random.randint(1, 10)
            max_attempts = 3
            instruction_text = "I am thinking of a number between 1 and
10"
        elif level == 2:
            self.secret_number = random.randint(1, 30)
```

```python
            max_attempts = 5
            instruction_text = "I am thinking of a number between 1 and 
30"
        elif level == 3:
            self.secret_number = random.randint(1, 50)
            max_attempts = 5
            instruction_text = "I am thinking of a number between 1 and 
50"

        return self.secret_number, max_attempts, instruction_text

    def display_menu(self):
        level = 1
        attempts = 0
        max_attempts = 0
        secret_number = 0
        input_text = ''
        game_message = ''
        right_or_wrong = ''
        Starting_Attempts = ''
        Level_Instructions = ''
        game_over = False #Flag for game over

        while True:
            self.game.check_events()
            if self.game.ESCAPE_KEY:  # Allow going back if needed
                self.game.curr_menu = self.game.main_menu
                return

            self.game.display.fill(self.game.BLACK)

            if game_over:
                #Display Game Over Screen
                self.display_game_over()
                self.blit_screen()
                continue

            if attempts >= (max_attempts if level > 0 else 0):
                game_message = f"Sorry the number was 
{self.secret_number}."
```

```
                level = max(1, level - 1)
                attempts = 0
                input_text = ''
                self.secret_number, max_attempts, instruction_text =
self.play_game(level)
                self.game.draw_text(game_message, 20, self.game.DISPLAY_w
/ 2, self.game.DISPLAY_h / 3)

            if level > 3:
                game_over = True

            if attempts == 0:
                self.secret_number, max_attempts, instruction_text =
self.play_game(level)
                self.game.draw_text (f"Level {level} {instruction_text}",
20,self.game.DISPLAY_w / 2, self.game.DISPLAY_h / 4)
                self.game.draw_text (f"You have {max_attempts} attempts to
guess my number", 20,self.game.DISPLAY_w / 2, self.game.DISPLAY_h / 3)
            elif attempts == max_attempts - 1:
                hint_min = max(0, self.secret_number - 2)  # Lower bound
for hint
                hint_max = min(100, self.secret_number + 2)
                self.game.draw_text(f'Last attempt hint the number is
between {hint_min} and {hint_max}', 20,self.game.DISPLAY_w / 2,
self.game.DISPLAY_h / 4)
            else:
                remaining_attempts = max_attempts - attempts
                self.game.draw_text(f'You have {remaining_attempts}
attempts left', 20,self.game.DISPLAY_w/ 2, self.game.DISPLAY_h / 4)

            # Display game message
            self.game.draw_text(right_or_wrong, 20, self.game.DISPLAY_w /
2, self.game.DISPLAY_h /5)
            # Display input box
            self.game.draw_text("Your guess: " + input_text, 20,
self.game.DISPLAY_w / 2, self.game.DISPLAY_h / 2)

            # Check for user input
            for event in pygame.event.get():
                if event.type == pygame.QUIT:
                    self.game.running = False
                    return
                if event.type == pygame.KEYDOWN:
                    if event.key == pygame.K_RETURN:
                        if input_text:
                            try:
                                guess = int(input_text)

                                if level == 3:
                                    guess = random.randint(1,50)
                                    attempts += 1

                                else:
                                    attempts += 1
                                    print(f"Guess{guess}, Attempts:
{attempts}")


                                if guess < self.secret_number:
                                    right_or_wrong = "Your guess is too
low"
                                elif guess > self.secret_number:
                                    right_or_wrong = "Your guess is too
high"
                                else:
                                    right_or_wrong = "Congratulations You
guessed correctly"
                                    if level == 3:
                                        game_over = True
                                    else:
                                        level += 1  # Move to the next
level

                                        attempts = 0
                                        input_text = ''
                            except ValueError:
                                right_or_wrong = "Please enter a valid
number"
                        input_text = ''
                    elif event.key == pygame.K_BACKSPACE:
                        input_text = input_text[:-1]
                    else:
                        if level == 3 and input_text ==
str(self.secret_number):
                            # Prevent correct guess during  round 3
                            input_text += str(random.randint(1,50))
                        else:
                            input_text += event.unicode

            self.blit_screen()

    def display_game_over(self):
        mixer.music.stop()  # Stop any current music
        mixer.music.load('audio2.wav')
        mixer.music.play(-1)

        self.game.display.fill(self.game.BLACK)
        self.game.draw_text('Game Over', 50, self.game.DISPLAY_w / 2,
self.game.DISPLAY_h / 5)
        self.game.draw_text(f'The number was
{self.secret_number}',20,self.game.DISPLAY_w / 2, self.game.DISPLAY_h / 4)
        self.game.draw_text(f'I gave you your hint and you guessed
{self.secret_number} how foolish', 20, self.game.DISPLAY_w /
2,self.game.DISPLAY_h / 3)
        self.game.draw_text('HAHAHAHAH Enjoy your eternity in my prison',
20, self.game.DISPLAY_w / 2, self.game.DISPLAY_h / 2)

        # Update the display
        pygame.display.update()

        # Keep checking for events until the player decides to return to
the menu
        game_over_running = True
        while game_over_running:
            for event in pygame.event.get():
                if event.type == pygame.QUIT:
                    pygame.quit()
                    return
                if event.type == pygame.KEYDOWN:
                    if event.key == pygame.K_RETURN:
                        self.game.curr_menu = self.game.main_menu
                        self.game.playing = False
                        game_over_running = False  # Exit the game over
loop

            # Optionally, you can add a small delay to reduce CPU usage
            pygame.time.delay(100)



class MainMenu(Menu):
    def __init__(self, game):
        Menu.__init__(self, game)
        self.state = "Start"
        self.startx, self.starty = self.mid_w, self.mid_h + 30
        self.optionsx, self.optionsy = self.mid_w, self.mid_h + 50
        self.creditsx, self.creditsy = self.mid_w, self.mid_h + 70
        self.cursor_rect.midtop = (self.startx + self.offset, self.starty)

    def display_menu(self):
        self.run_display = True
        while self.run_display:
            self.game.check_events()
            self.check_input()
            self.game.display.fill(self.game.BLACK)
            self.game.draw_text('Mystery Digits', 50, self.game.DISPLAY_w
/ 2, self.game.DISPLAY_h / 4 - 20)
            self.game.draw_text("Start Game", 20, self.startx,
self.starty)
            self.game.draw_text("Options", 20, self.optionsx,
self.optionsy)
            self.game.draw_text("Credits", 20, self.creditsx,
self.creditsy)
            self.draw_cursor()
            self.blit_screen()

    def move_cursor(self):
        if self.game.DOWN_KEY:
            if self.state == "Start":
                self.cursor_rect.midtop = (self.optionsx + self.offset,
self.optionsy)
                self.state = 'Options'
            elif self.state == 'Options':
                self.cursor_rect.midtop = (self.creditsx + self.offset,
self.creditsy)
                self.state = 'Credits'
            elif self.state == "Credits":
                self.cursor_rect.midtop = (self.startx + self.offset,
self.starty)
                self.state = 'Start'
        elif self.game.UP_KEY:
            if self.state == 'Start':
                self.cursor_rect.midtop = (self.creditsx + self.offset,
self.creditsy)
                self.state = 'Credits'
            elif self.state == 'Options':
                self.cursor_rect.midtop = (self.startx + self.offset,
self.starty)
                self.state = 'Start'
            elif self.state == 'Credits':
                self.cursor_rect.midtop = (self.optionsx + self.offset,
self.optionsy)
                self.state = 'Options'

    def check_input(self):
        self.move_cursor()
        if self.game.START_KEY:
            if self.state == 'Start':
                self.game.playing = True  # This starts the game loop,
which has the button
            elif self.state == 'Options':
                self.game.curr_menu = self.game.options
            elif self.state == 'Credits':
                self.game.curr_menu = self.game.credits
            self.run_display = False

class OptionsMenu(Menu):
    def __init__(self, game):
        Menu.__init__(self, game)
        self.state = 'Volume'
        self.volx, self.voly = self.mid_w, self.mid_h + 20
        self.controlsx, self.controlsy = self.mid_w, self.mid_h + 40
        self.cursor_rect.midtop = (self.volx + self.offset, self.voly)

    def display_menu(self):
        self.run_display = True
        while self.run_display:
            self.game.check_events()
            self.check_input()
            self.game.display.fill(self.game.BLACK)
            self.game.draw_text('Options', 20, self.game.DISPLAY_w / 2,
self.game.DISPLAY_h / 2 - 30)
            self.game.draw_text("Volume", 15, self.volx, self.voly)
            self.game.draw_text("Controls", 15, self.controlsx,
self.controlsy)
            self.draw_cursor()
            self.blit_screen()

    def check_input(self):
        if self.game.BACK_KEY:
            self.game.curr_menu = self.game.main_menu
            self.run_display = False
        elif self.game.UP_KEY or self.game.DOWN_KEY:
            if self.state == 'Volume':
                self.state = 'Controls'
                self.cursor_rect.midtop = (self.controlsx + self.offset,
self.controlsy)
            elif self.state == 'Controls':
                self.state = 'Volume'
                self.cursor_rect.midtop = (self.volx + self.offset,
self.voly)

        elif self.game.START_KEY:
            # TO-DO: Create a Volume Menu and a Controls Menu
            pass

class CreditsMenu(Menu):
    def __init__(self, game):
        Menu.__init__(self, game)

    def display_menu(self):
        self.run_display = True
        while self.run_display:
            self.game.check_events()
            if self.game.START_KEY or self.game.BACK_KEY:
                self.game.curr_menu = self.game.main_menu
                self.run_display = False
            self.game.display.fill(self.game.BLACK)
            self.game.draw_text('Mystery Digits', 50, self.game.DISPLAY_w
/ 2, self.game.DISPLAY_h / 3 - 30)
            self.game.draw_text('Credits',25, self.game.DISPLAY_w / 2,
self.game.DISPLAY_h / 2 - 30)
            self.game.draw_text('Made by Nicolas Capetillo', 20,
self.game.DISPLAY_w / 2, self.game.DISPLAY_h / 2 + 10)
            self.blit_screen()
```