```
#Simon Memory Game
import tkinter
from tkinter import StringVar, ACTIVE, NORMAL, DISABLED
import random

#Define window
root = tkinter.Tk()
root.title('Simon Memory Game')
root.iconbitmap('simon.ico')
root.geometry('400x400')
root.resizable(0,0)

#Define fonts and colors
game_font1 = ('Arial', 12)
game_font2 = ('Arial', 8)
white = "#c6cbcd"
white_light = "#fbfcfc"
magenta = "#90189e"
magenta_light = "#f802f9"
cyan = "#078384"
cyan_light = "#00fafa"
yellow = "#9ba00f"
yellow_light = "#f7f801"
root_color = "#2eb4c6"
game_color = "#f6f7f8"
root.config(bg=root_color)

#SEt global variables for the game
time = 500
score = 0
game_sequence = []
player_sequence = []

#Define functions
def pick_sequence():
    """Pick the next value in the sequence.  Do not allow for repeated values."""
    while True:
        value = random.randint(1,4)
        #Sequence is size 0, so take the value regardless
        if len(game_sequence) == 0:
            game_sequence.append(value)
            break
        #make sure the current value is not the same as the last value in the sequence
        elif value != game_sequence[-1]:
            game_sequence.append(value)
            break

    #Now that the value is added to the sequence, play the sequence
    play_sequence()


def play_sequence():
    """Play the entire sequence for a given round by animating the buttons"""
    #Change button label
    change_label("Playing!")

    #Without delay, all buttons will animate at the same time.  The delay adds the
    'time' variable to each .after()
    delay = 0
    for value in game_sequence:
        if value == 1:
            root.after(delay, lambda:animate(white_button))
        elif value == 2:
            root.after(delay, lambda:animate(magenta_button))
        elif value == 3:
            root.after(delay, lambda:animate(cyan_button))
        elif value == 4:
```

```
67              root.after(delay, lambda:animate(yellow_button))
68
69          #Increment delay for next iteration of loop
70          delay += time
71
72
73  def animate(button):
74      """Animate a given button by chaning its color"""
75      button.config(state=ACTIVE)
76      root.after(time, lambda:button.config(state=NORMAL))
77
78
79  def change_label(message):
80      """Update the start button text and color to let the player know their status."""
81      start_button.config(text=message)
82
83      if message == "Wrong!":
84          start_button.config(bg='red')
85      else:
86          start_button.config(bg=game_color)
87
88
89  def set_difficulty():
90      """Use radio buttons to set difficulty.  Difficulty affects time between button
        'flashes'"""
91      global time
92
93      #Change the time (difficulty) based off the value of the radio buttons
94      if difficulty.get() == 'Easy':
95          time = 1000
96      elif difficulty.get() == 'Medium':
97          time = 500
98      else:
99          time = 200
100
101
102 def press(value):
103     """Simulate pressing a button for player."""
104     #Add the players press to players sequence
105     player_sequence.append(value)
106
107     #If the current 'round' is over, check to see if the player enter the correct
        sequence of button presses
108     if len(player_sequence) == len(game_sequence):
109         check_round()
110
111
112 def check_round():
113     """Determine if the player entered the correct sequence of button presses for a
        round."""
114     global player_sequence
115     global game_sequence
116     global score
117
118     #The player is correct, so change the label, update score, wait, then start the
        next round.
119     if player_sequence == game_sequence:
120         change_label("Correct!")
121         score += len(player_sequence) + int(1000/time)
122         root.after(500, pick_sequence)
123     #The player is incorrect so change label, update score, disable buttons, and reset
        for new game.
124     else:
125         change_label('Wrong!')
126         score = 0
127         disable()
128         #The game is over, so wipe the game sequence
```

```
129             game_sequence = []
130             #Wait 2 seconds then change the message
131             root.after(2000, lambda:change_label("New Game"))
132 
133         #Regardless of win or lose for a round, wipe the players sequence
134         player_sequence = []
135 
136         #Update score label
137         score_label.config(text="Score: " + str(score))
138 
139 
140     def disable():
141         """Dislabe all buttons so they can't accidentally be pressed."""
142         white_button.config(state=DISABLED)
143         magenta_button.config(state=DISABLED)
144         cyan_button.config(state=DISABLED)
145         yellow_button.config(state=DISABLED)
146 
147 
148     def enable():
149         """Enable all buttons to start the game and pick the first value in the sequence."""
150         white_button.config(state=NORMAL)
151         magenta_button.config(state=NORMAL)
152         cyan_button.config(state=NORMAL)
153         yellow_button.config(state=NORMAL)
154 
155         #Pick a value!
156         pick_sequence()
157 
158 
159     #Define Layout
160     #Create frames
161     info_frame = tkinter.Frame(root, bg=root_color)
162     game_frame = tkinter.LabelFrame(root, bg=game_color)
163     info_frame.pack(pady=(10, 20))
164     game_frame.pack()
165 
166     #Layout for the info frame
167     start_button = tkinter.Button(info_frame, text="New Game", font=game_font1,
        bg=game_color, command=enable)
168     score_label = tkinter.Label(info_frame, text="Score: " + str(score), font=game_font1,
        bg=root_color)
169     start_button.grid(row=0, column=0, padx=20, ipadx=30)
170     score_label.grid(row=0, column=1)
171 
172     #Layout for the game frame
173     #Make the game buttons
174     white_button = tkinter.Button(game_frame, bg=white, activebackground=white_light,
        borderwidth=3, state=DISABLED, command=lambda:press(1))
175     magenta_button = tkinter.Button(game_frame, bg=magenta, activebackground=magenta_light,
        borderwidth=3, state=DISABLED, command=lambda:press(2))
176     cyan_button = tkinter.Button(game_frame, bg=cyan, activebackground=cyan_light,
        borderwidth=3, state=DISABLED, command=lambda:press(3))
177     yellow_button = tkinter.Button(game_frame, bg=yellow, activebackground=yellow_light,
        borderwidth=3, state=DISABLED, command=lambda:press(4))
178 
179     white_button.grid(row=0, column=0, columnspan=2, padx=10, pady=10, ipadx=60, ipady=50)
180     magenta_button.grid(row=0, column=2, columnspan=2, padx=10, pady=10, ipadx=60, ipady=50)
181     cyan_button.grid(row=1, column=0, columnspan=2, padx=10, pady=10, ipadx=60, ipady=50)
182     yellow_button.grid(row=1, column=2, columnspan=2, padx=10, pady=10, ipadx=60, ipady=50)
183 
184     #Make radio buttons for difficulty
185     difficulty = StringVar()
186     difficulty.set('Medium')
187     tkinter.Label(game_frame, text='Difficulty:', font=game_font2,
        bg=game_color).grid(row=2, column=0)
188     tkinter.Radiobutton(game_frame, text="Easy", variable=difficulty, value="Easy",
```

```
    font=game_font2, bg=game_color, command=set_difficulty).grid(row=2, column=1)
189 tkinter.Radiobutton(game_frame, text="Medium", variable=difficulty, value="Medium",
    font=game_font2, bg=game_color, command=set_difficulty).grid(row=2, column=2)
190 tkinter.Radiobutton(game_frame, text="Hard", variable=difficulty, value="Hard",
    font=game_font2, bg=game_color, command=set_difficulty).grid(row=2, column=3)
191
192 #Run root window's main loop
193 root.mainloop()
```