Make a python script using the nicegui package. I will provide the “backend” code that I have at the end of the message. I want the screen to be populated by buttons. At the top I want a row with two big buttons based on the teams in the set teams. When one of the buttons is pressed the function select\_team(t) is to be called with that team’s name as t. Then the team buttons should disappear.

Underneath I want buttons from the events set, and when one of those is pressed I want the buttons to disappear, select\_event(e) with that event as e. if e is a key in the dictionary events\_and\_their\_subevents then e’s values should appear as new buttons, if one of those is pressed then is called select\_subevent(e, se) where se is the button’s subevent. If e I not a key in events\_and\_their\_subevents don’t make any new buttons.

Underneath I want to display an image of a field (<https://upload.wikimedia.org/wikipedia/commons/thumb/2/2e/Bandy_pitch_metric.svg/2560px-Bandy_pitch_metric.svg.png>) pressing it should save the coordinates (x, y) and call the function set\_coordinates(x, y) and then remove the image.

Finally I want the numbers n in the set players to appear as buttons at the bottom of the screen and pressing a button should call set\_player(n). Then the buttons.

When one of each type of button is pressed then all buttons should be regenerated and the procedure should be started over.

The backend code is as follows:

teams = {'iks', 'bol'}

events = {'skott', 'frislag', 'bolltapp', 'närkamp', 'hörna', 'inslag', 'utkast',

    'avslag', 'mål', 'utvisning', 'stop', 'passning', 'friläge', 'straff',

    'offside', 'rensning', 'timeout', 'boll', 'brytning', 'skottyp', '40', 'kontring'}

events\_and\_their\_subevents = {'skott': {'utanför', 'räddning', 'täckt'},

                                'skottyp': {'friställande', 'inlägg', 'utifrån', 'dribbling', 'centralt', 'fast', 'retur'},

                                'bolltapp': {'tappad', 'passförsök'},

                                'passning' : {'straffområde', 'lång', 'farlig'},

                                'mål': {'spelmål', 'hörnmål', 'straffmål', 'frislagsmål'},

                                'utvisning': {'5', '10'}

                                }

players = {1, 2, 5, 6, 7, 10, 13, 23, 89}

def select\_team(t):

    pass

def select\_event(e):

    pass

def select\_subevent(se):

    pass

def select\_coordinates(x, y):

    pass

def select\_player(n):

    pass

Svar:

import nicegui as ng

teams = {'iks', 'bol'}

events = {'skott', 'frislag', 'bolltapp', 'närkamp', 'hörna', 'inslag', 'utkast',

          'avslag', 'mål', 'utvisning', 'stop', 'passning', 'friläge', 'straff',

          'offside', 'rensning', 'timeout', 'boll', 'brytning', 'skottyp', '40', 'kontring'}

events\_and\_their\_subevents = {

    'skott': {'utanför', 'räddning', 'täckt'},

    'skottyp': {'friställande', 'inlägg', 'utifrån', 'dribbling', 'centralt', 'fast', 'retur'},

    'bolltapp': {'tappad', 'passförsök'},

    'passning': {'straffområde', 'lång', 'farlig'},

    'mål': {'spelmål', 'hörnmål', 'straffmål', 'frislagsmål'},

    'utvisning': {'5', '10'}

}

players = {1, 2, 5, 6, 7, 10, 13, 23, 89}

# Backend functions

def select\_team(t):

    # Your implementation here

    print(f"Selected team: {t}")

def select\_event(e):

    # Your implementation here

    print(f"Selected event: {e}")

def select\_subevent(e, se):

    # Your implementation here

    print(f"Selected event: {e}, Subevent: {se}")

def select\_coordinates(x, y):

    # Your implementation here

    print(f"Selected coordinates: ({x}, {y})")

def select\_player(n):

    # Your implementation here

    print(f"Selected player: {n}")

def generate\_team\_buttons():

    buttons = [ng.ui.button(team, on\_click=lambda t=team: select\_team(t)) for team in teams]

    return buttons

def generate\_event\_buttons():

    buttons = [ng.ui.button(event, on\_click=lambda e=event: select\_event(e)) for event in events]

    return buttons

def generate\_subevent\_buttons(event):

    if event in events\_and\_their\_subevents:

        buttons = [ng.ui.button(subevent, on\_click=lambda se=subevent: select\_subevent(event, se))

                   for subevent in events\_and\_their\_subevents[event]]

        return buttons

    return []

def generate\_player\_buttons():

    buttons = [ng.ui.button(str(player), on\_click=lambda n=player: select\_player(n)) for player in players]

    return buttons

def main\_window():

    team\_buttons = generate\_team\_buttons()

    while True:

        #ng.app(title="Sports App", width=800, height=600, header=False)

        with ng.ui.row():

            for button in team\_buttons:

                ng.ui.box(width="50%")(button)

        selected\_team = ng.await\_selection()

        team\_buttons = []

        event\_buttons = generate\_event\_buttons()

        with ng.ui.row():

            for button in event\_buttons:

                ng.ui.box(width="50%")(button)

        selected\_event = ng.await\_selection()

        event\_buttons = []

        subevent\_buttons = generate\_subevent\_buttons(selected\_event)

        with ng.ui.row():

            for button in subevent\_buttons:

                pass

                #ng.ui.box(width="50%")(button)

        ng.ui.button("Field Image", on\_click=lambda: select\_coordinates(0, 0))

        player\_buttons = generate\_player\_buttons()

        with ng.ui.row():

            for button in player\_buttons:

                pass#ng.ui.box(width="50%")(button)

        selected\_coordinates = ng.await\_selection()

        player\_buttons = []

if \_\_name\_\_ == "\_\_main\_\_":

    main\_window()