Meeting #7

10-21-21

Michael Lee

Deliverables

- Research 10-20 coordinate system to use for 3D brain
- Improvements on the 3D brain

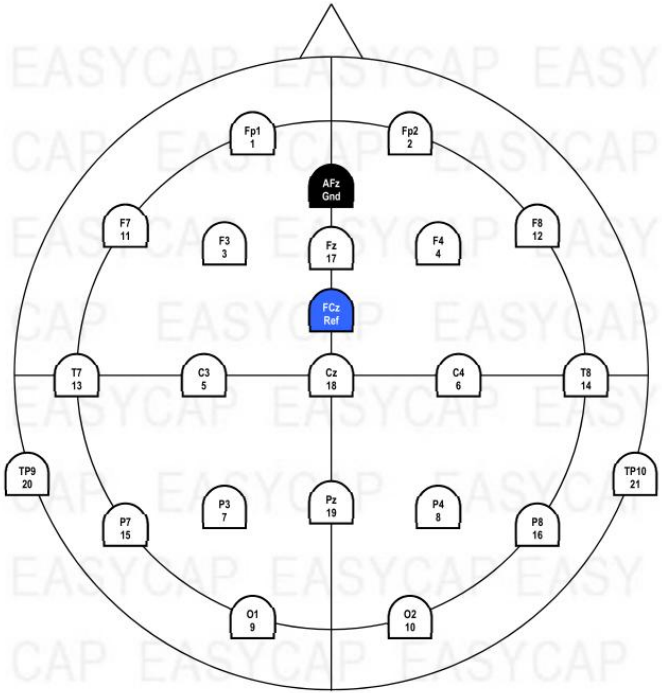
Methodology and Learnings

- 10-10 and 10-20 layouts
- EasyCap Electrode Layouts
- Pandas Dataframe
- Callbacks

Methodology and Learnings

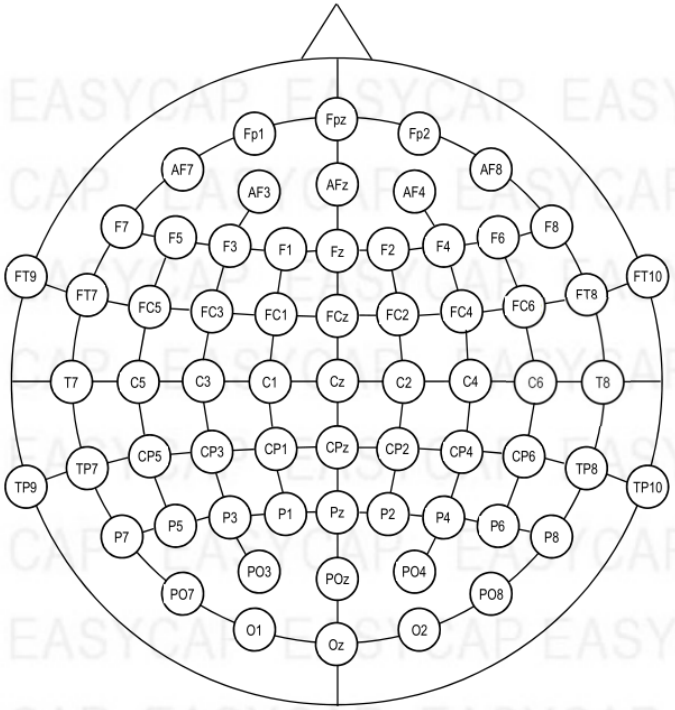
21 Channels: Classic 10/20-System: 19 electrodes plus TP9/TP10 (near mastoids) plus Ref plus Gnd

M25

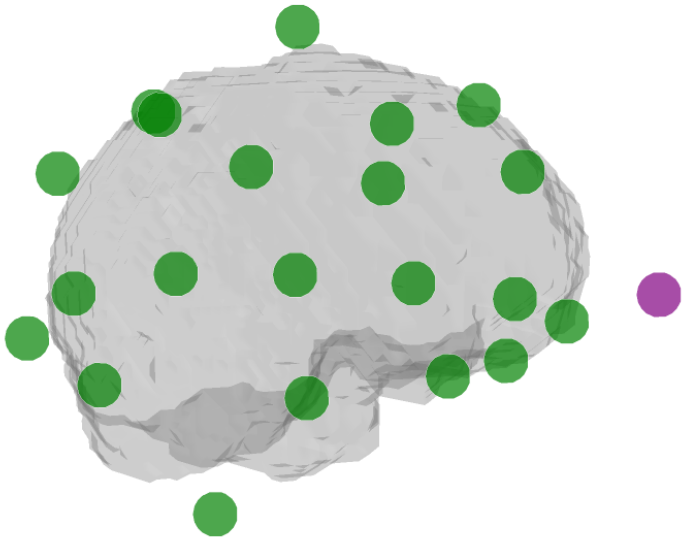


64 Channels: Full 10% System covering the 10/20 area

M64

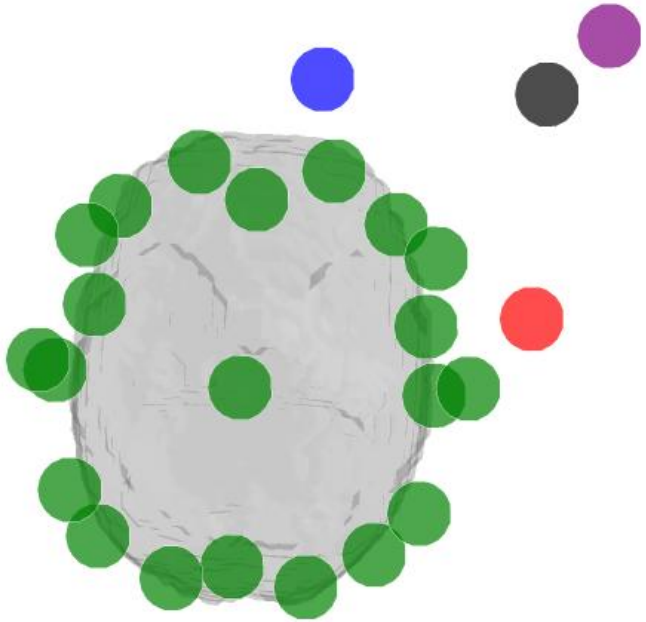


Results

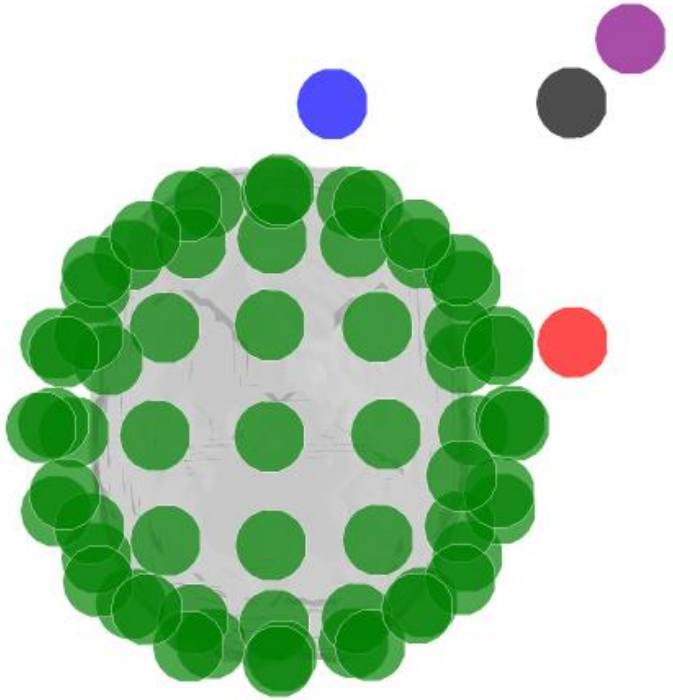


trace 1
trace 2
trace 3
trace 4
Fp1
Fp2
Fz
F3
F4
F7
F8
Cz
C3
C4
Pz
P3
P4
O1
O2
T7
T8
P7
P8
FT9
FT10

Results



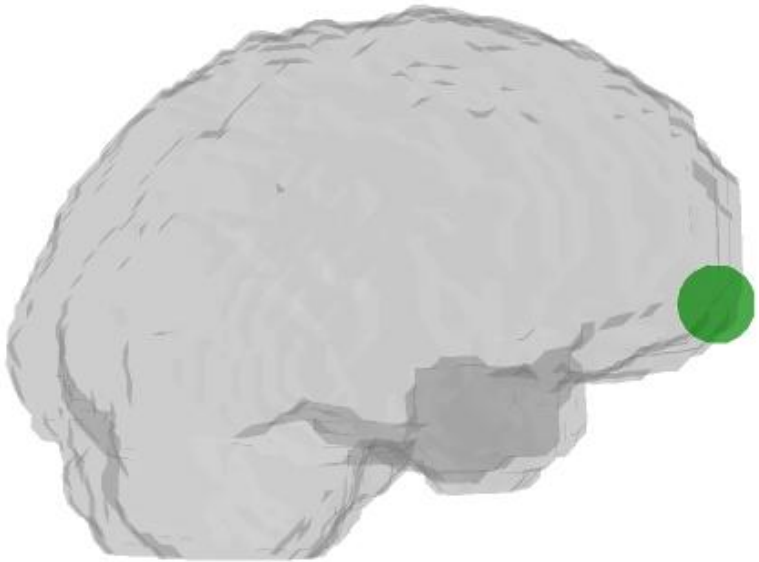
10-20 Layout



10-10 Layout

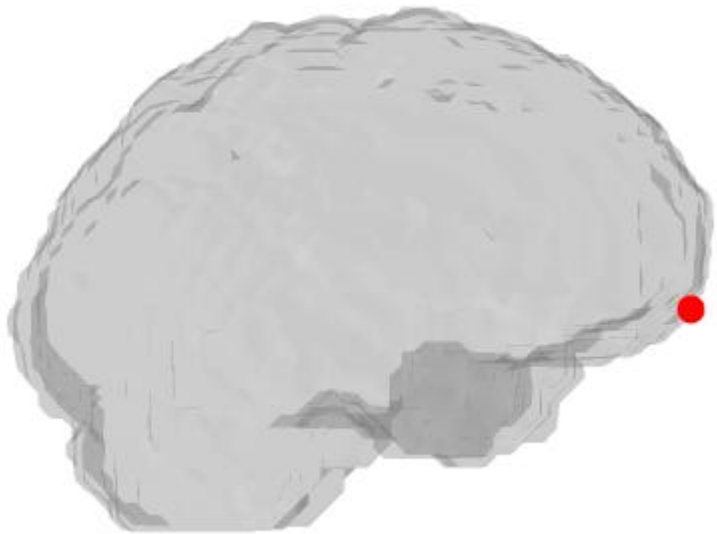
Results - FigureWidget

```
fig = go.Figure(data=data_brain)  # from plotly.graph objects module
fig.update_layout(**default_3d_layout)
fig2 = go.FigureWidget(fig)  # create widget to add components

...
xoffset = 84.5385 + 35
yoffset = 84.9812 + 45
zoffset = 42.0882 + 25

...

myrow = df.loc[df["Name"] == "Fp2"]
myrowx = myrow.iloc[0]['x']
myrowy = myrow.iloc[0]['y']
myrowz = myrow.iloc[0]['z']

...

fig2.add_scatter3d(x=[myrowy + xoffset],
                   y=[-myrowx + yoffset],
                   z=[myrowz + zoffset],
                   marker_size=[50, 50, 50],
                   marker=dict(color='green'),
                   name=myrow.iloc[0]['Name'],
                   uid='test')
```



Results – add_trace()

```
# First number is to shift the negative values to start at 0
# Second number is to place the plot on the model
xoffset = 84.5385 + 35
yoffset = 84.9812 + 45
zoffset = 42.0882 + 25

...

myrow = df.loc[df["Name"] == "Fp2"]
myrowx = myrow.iloc[0]['x']
myrowy = myrow.iloc[0]['y']
myrowz = myrow.iloc[0]['z']

fig = go.Figure(data=data_brain)  # from plotly.graph objects module
fig.update_layout(**default_3d_layout)

# Alternate way to make a scatter 3d
fig.add_trace(go.Scatter3d(
    x=[myrowy + xoffset],
    y=[-myrowx + yoffset],
    z=[myrowz + zoffset],
    marker=dict(color='red'),
    name=myrow.iloc[0]['Name'])
)
```



Results

```
app = dash.Dash(__name__)

app.layout = html.Div([
    dcc.Graph(
        id="image-display-graph-3d",
        # figure=fig,
        config=dict(displayModeBar=False)
    ),
    dcc.Dropdown(
        id='dropdown1-for-graph',
        options=[
            {'label': 'Fp1', 'value': 'Fp1'},
            {'label': 'Fp2', 'value': 'Fp2'}
        ],
        value='Fp1'
    )
])

# Updates graph when the dropdown is changed
@app.callback(
    dash.dependencies.Output('image-display-graph-3d', 'figure'),
    dash.dependencies.Input('dropdown1-for-graph', 'value')
)
def update_graph(selected_value):
    fig = make_default_3d_fig(data_brain, df, selected_value)
    return fig

app.run_server(debug=True)
```

```
myrow = df.loc[df["Name"] == callback_value]
print(myrow)
myrowx = myrow.iloc[0]['x']
myrowy = myrow.iloc[0]['y']
myrowz = myrow.iloc[0]['z']
```

```
Empty DataFrame
Columns: [Name, x, y, z]
Index: []
```