

Meeting #6

10-14-21

Michael Lee

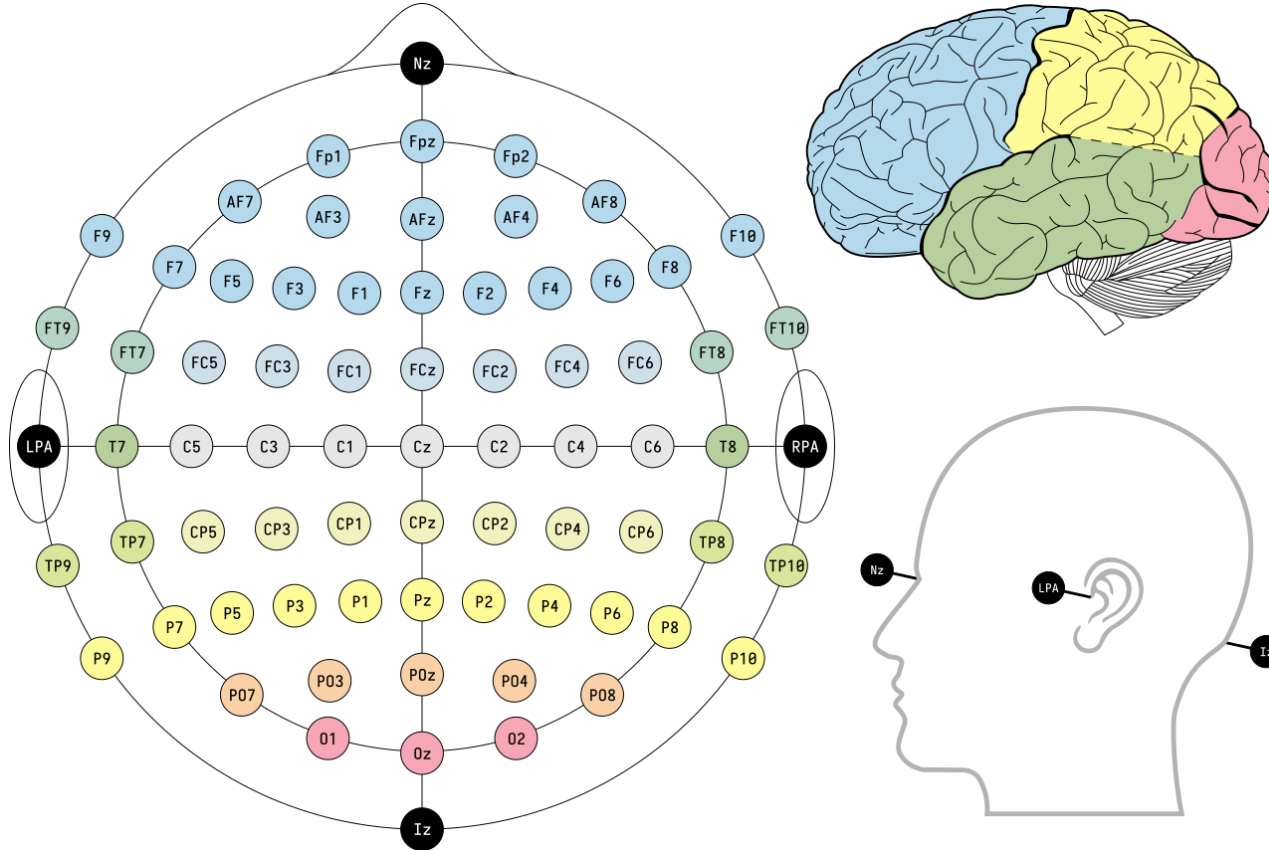
Deliverables

- Display 3D brain with coordinates

Methodology and Learnings

- Plotly FigureWidget
- Pandas DataFrame
- 10-10 system

Methodology and Learnings

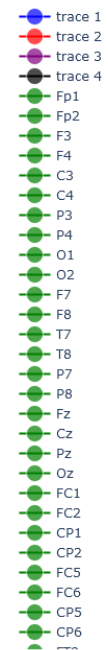
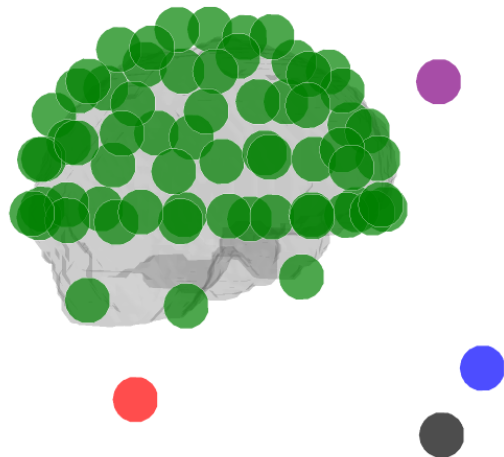


Results

```
47 def make_default_3d_fig(data_brain, df):
48     # adapted from plotly dash
49     default_3d_layout = dict(
50         scene=dict(
51             yaxis=dict(visible=False, showticklabels=False, showgrid=False, ticks=""),
52             xaxis=dict(visible=False),
53             zaxis=dict(visible=False),
54             camera=dict(
55                 up=dict(x=0, y=0, z=1),
56                 center=dict(x=0, y=0, z=0),
57                 eye=dict(x=1.25, y=1.25, z=1.25),
58             ),
59         ),
60         height=800,
61     )
62
63     fig = go.Figure(data=data_brain) # from plotly.graph objects module
64     fig.update_layout(**default_3d_layout)
65     fig2 = go.FigureWidget(fig) # create widget to add components
66
67     # add scatter with 3d coordinates
68     fig2.add_scatter3d(x=[100], y=[0], z=[0], marker_size=[50, 50, 50], marker=dict(color='blue'))
69     fig2.add_scatter3d(x=[0], y=[100], z=[0], marker_size=[50, 50, 50], marker=dict(color='red'))
70     fig2.add_scatter3d(x=[0], y=[0], z=[100], marker_size=[50, 50, 50], marker=dict(color='purple'))
71     fig2.add_scatter3d(x=[0], y=[0], z=[0], marker_size=[50, 50, 50], marker=dict(color='black'))
72
73     # First number is to shift the negative values to start at 0
74     # Second number is to place the plot on the model
75     xoffset = 84.5385 + 35
76     yoffset = 84.9812 + 45
77     zoffset = 42.0882 + 25
78
79     # Adds the dataframe data to the model
80     # Flips over y=x and also flips over the x-axis
81     for row in df.iteruples(index=False):
82         fig2.add_scatter3d(x=[row[2] + xoffset], y=[-row[1] + yoffset], z=[row[3] + zoffset], marker_size=[50, 50, 50],
83                             marker=dict(color='green'),
84                             name=row[0])
85
86     return fig2
```

```
89 # main function to run app
90 if __name__ == "__main__":
91     # first 3 lines adapted from Github:
92     # https://github.com/plotly/dash-sample-apps/tree/main/apps/dash-3d-image-partitioning
93     img = image.load_img("assets/BraTS19_2013_10_1_flair.nii") # read in nii file from assets folder in same directory
94     img = img.get_fdata().transpose(2, 0, 1)[::-1].astype("float")
95     img = img_as_ubyte((img - img.min()) / (img.max() - img.min()))
96
97     # get the file of the cartesian coordinates
98     filepath = 'assets/EEG01_chanlocs_cartesian.txt'
99     df = pd.read_csv(filepath, delim_whitespace=True, names=['Name', 'x', 'y', 'z'])
100
101     data_brain = create_brain_data(img)
102     fig = make_default_3d_fig(data_brain, df)
103     # print(fig)
104
105     app = dash.Dash(__name__)
106
107     app.layout = html.Div(
108         dcc.Graph(
109             id="image-display-graph-3d",
110             figure=fig,
111             config=dict(displayModeBar=False, ),
112         )
113     )
114
115     app.run_server(debug=True)
```

Results



Results

