

All teams communications are being done through Discord.

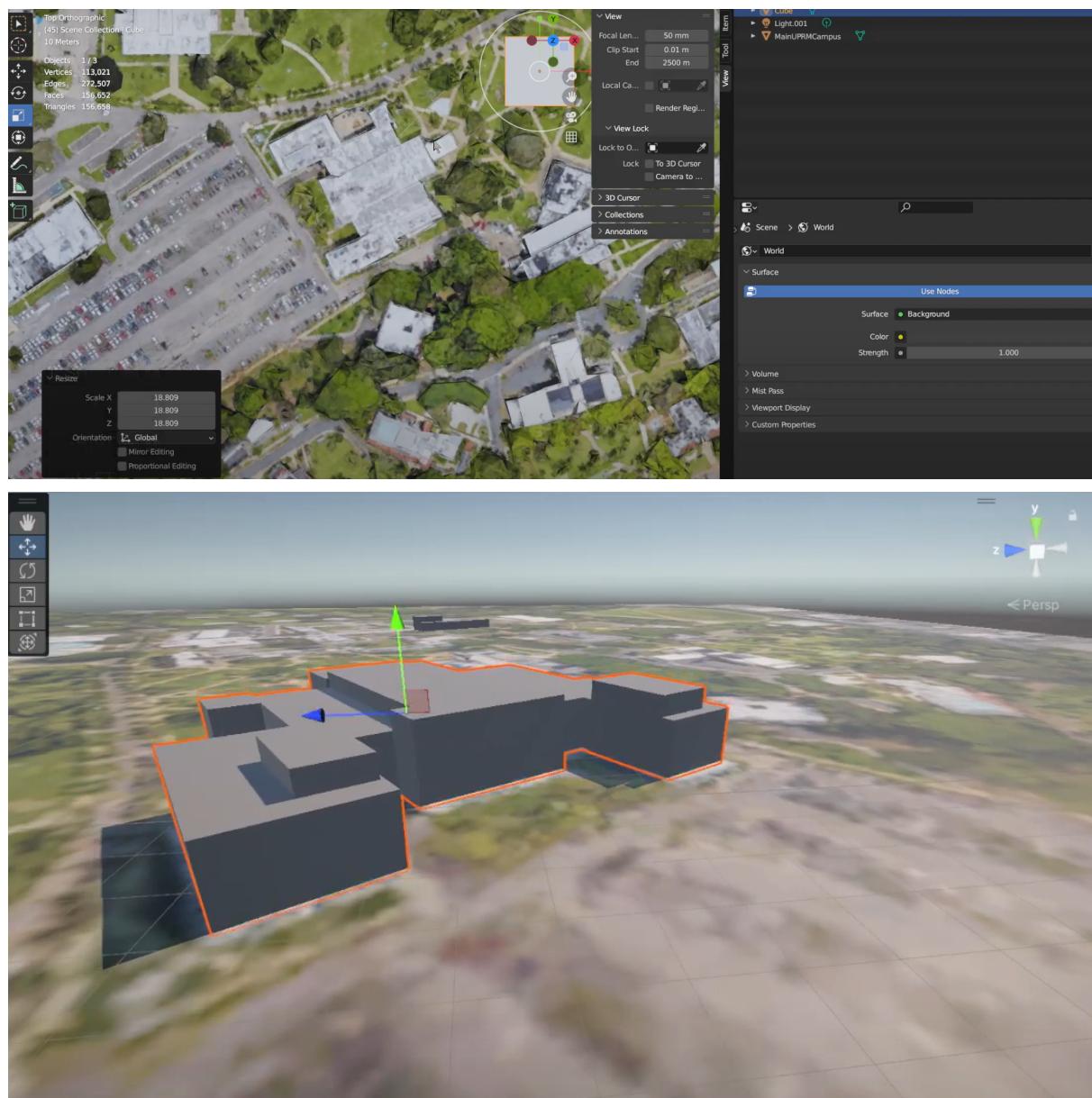
## Imagining (2D/3D)

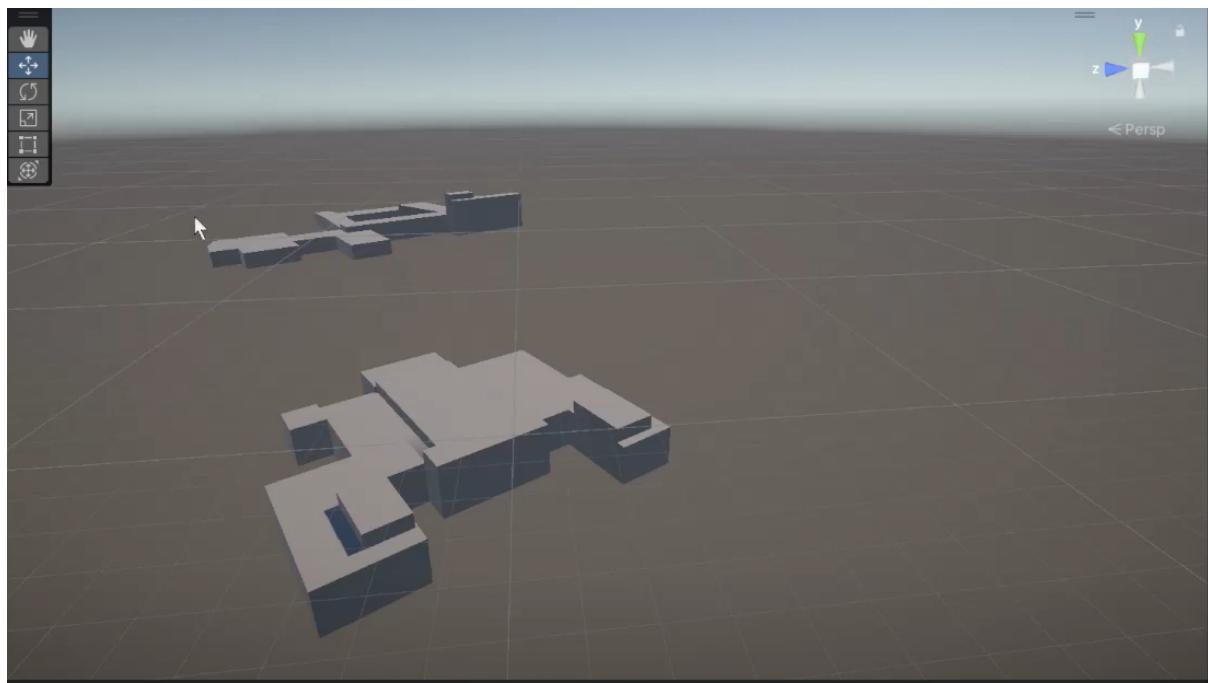
### Members:

- Jan R. Badillo Rivera
- Natalia Zoe Pérez Montalvo
- Victoria M Miranda Osuna
- Alejandro Behar Gutierrez

### Progress:

The 3-D/2-D team successfully started the model of the map that will be used when 3-D is needed. They decided on what platforms to use and focus on. Also helped in the type of file that should be sent for other members to work with it.





### **Progress:**

The 3-D/2-D team successfully started the model of the map that will be used when 3-D is needed. They decided on what platforms to use and focus on. Also helped in the type of file that should be sent for other members to work with it.

### **Frontend:**

#### **Members:**

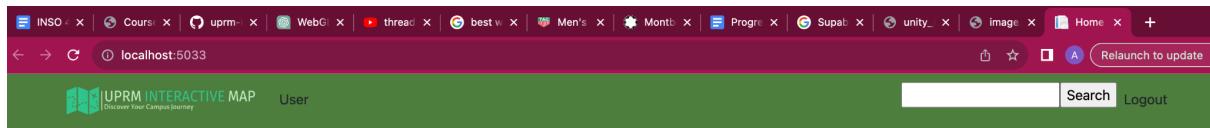
- Alexander Y Ibarrondo-Rios
- Irsaris Perez-Rodriguez
- Oniel A Plaza-Perez
- Orlando G Mercado-Tellado
- Carolina Z Rodriguez-Maldonado
- Edjoel Colon-Nogueras

### **Progress:**

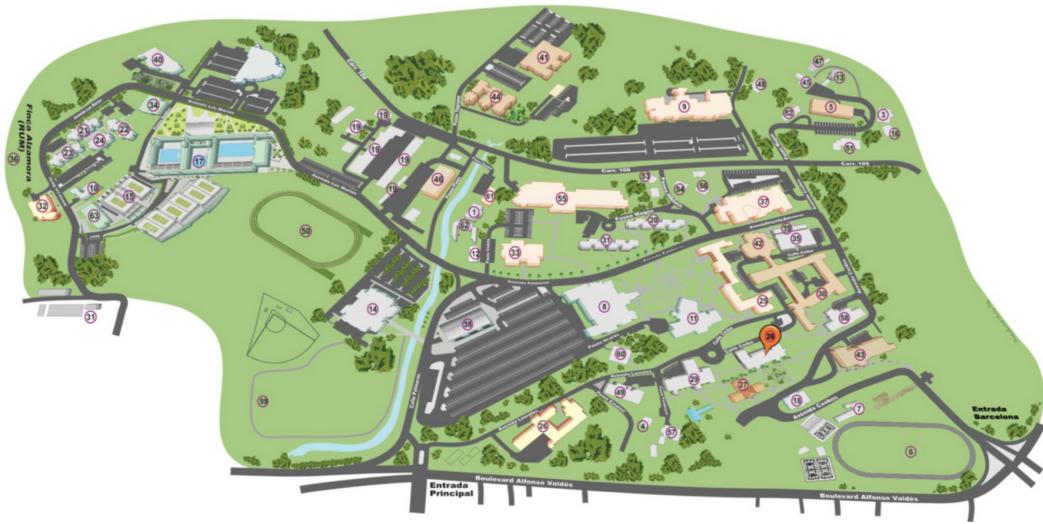
The frontend team successfully started development on the views (htmlcs) files for the Interactive Map. They worked on some wireframes and began implementation of such wireframes. Example of the wireframes:



### Frontend Inprogress of UPRM Static Map:



### Welcome to UPRM Interactive Map



### Backend:

#### Members:

- Joshua R Santiago-Ibarra (Team Lead)
- Mark A Toro-Blas
- John Trinidad

- Mateo I Muniz-Velazquez
- Benjamin Diaz-Espada
- Agustin Barrero
- Misael Moctezuma
- Sebastian A Granda-Marulanda

### Progress:

The Backend team successfully implemented Supabase in their development environment. They were able to create and test their doings using boiler plates for authentication while the front-end team developed the views for the Interactive Map. Below some code snippets:

```
● ● ●

1 // if session in cookie exist, return Unity env
2 if (!session) {
3     // if session in cookie doesn't exist, return Authentication
4     return <Auth supabaseClient={supabase} appearance={{ theme:
    ThemeSupa }} />;
5 } else {
6     return (
7     <>
8     /* if Unity not Loaded, show % Load complition */
9     {isLoaded === false &&
10      <div className="loading-overlay">
11          <p>Loading... ({loadingPercentage}%)</p>
12      </div>
13  }
14  /* Component for overLay UI */
15  <Overlay/>
16  /* Componet with Unity on background */
17  <Unity
18      unityProvider={unityProvider}
19      devicePixelRatio={devicePixelRatio}
20      style={{ visibility: isLoaded ? "visible" : "hidden", width:
    "100%", height: "100%", top: "0", left: "0", position: "absolute" }}
21  />
```

```
● ● ●
1  const { unityProvider, loadingProgression, isLoaded } =
2    useUnityContext({
3      loaderUrl: "./assets/Builds.loader.js",
4      dataUrl: "./assets/Builds.data.br",
5      frameworkUrl: "./assets/Builds.framework.js.br",
6      codeUrl: "./assets/Builds.wasm.br",
7    });
8    useEffect(() => {
9      supabase.auth.getSession().then(({ data: { session } }) => {
10        setSession(session);
11      });
12    const {
13      data: { subscription },
14    } = supabase.auth.onAuthStateChange((_event, session) => {
15      setSession(session);
16    });

```

```
● ● ●
1  const { unityProvider, loadingProgression, isLoaded } =
2    useUnityContext({
3      loaderUrl: "./assets/Builds.loader.js",
4      dataUrl: "./assets/Builds.data.br",
5      frameworkUrl: "./assets/Builds.framework.js.br",
6      codeUrl: "./assets/Builds.wasm.br",
7    });
8    useEffect(() => {
9      supabase.auth.getSession().then(({ data: { session } }) => {
10        setSession(session);
11      });
12    const {
13      data: { subscription },
14    } = supabase.auth.onAuthStateChange((_event, session) => {
15      setSession(session);
16    });

```