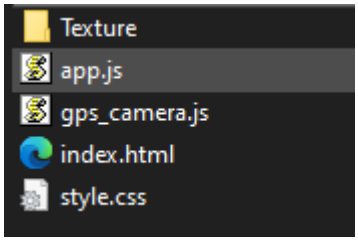
Salt Strong AR + GPS

Necessary documents



important data

1.- in case of changing the style or position

in case of changing the style, position of the div (class = buttonCloseLabel or class = buttonUrl) consider the following function

```
/* no move */
.buttonCloseLabel
{
  position:absolute;
  top: 0px;
  transform-origin: center center;
  left: calc(50% - 15px);
  margin-top: -45px;
  width: 30px;
  height: 30px;
  font-size: 24px;
  background-image: url('Texture/close.png');
  background-repeat: no-repeat;
  background-position: center;
  background-size: contain;
  background-color: rgba(255, 255, 255, 0.0);

}
```

```css
/* no move */
.buttonUrl
{
  margin: 4px 2px;
  padding: 15px 32px;
  border-radius: 10px;
  font-size: 16px;

  background-color: #4482c3; /* blue */
  border: none;
  color: white;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  cursor: pointer;
}
```

function: gpsMain.updatePolygonsTxt()
This part of the code helps detect if the crosshair is in the area of the close button or url button, when changing position or size, the values should be adjusted

```
/** Btn close Button*/
/**calculate the position of the button div */
let btn= elem.querySelector('#closeButton');
let top = y - btn.offsetTop;
let left = x + btn.offsetLeft;
if (top -45< gpsMain.halfScreenSize.height+35 && top-45 >gpsMain.halfScreenSize.height+10 ) //-45 margin-top
{
    if (left< gpsMain.halfScreenSize.width-5&& left> gpsMain.halfScreenSize.width-35 )
    {
        /**if crosshair is inside button area and clicked, activate function */
        if(gpsMain.touch)
        {
            btn.onclick()
        }
    }
}
```

```
/**btn url */
/**calculate the position of the button div */
btn= elem.querySelector('#btnUrl');
top = y + btn.offsetTop;
left = x + btn.offsetLeft+5;
// console.log (left)
if (top<gpsMain.halfScreenSize.height&& top>gpsMain.halfScreenSize.height-btn.offsetHeight)
{
    if (left<gpsMain.halfScreenSize.width && left>gpsMain.halfScreenSize.width- btn.offsetWidth)
    {
        if(gpsMain.touch)
        {
            /**if crosshair is inside button area and clicked, activate function */
            btn.onclick()
        }
    }
}
```

2.- Sensores (compass Heading)

The accuracy of the compass angle is different from each smartphone's sensors.

```
_onDeviceOrientation: function (event) {
    if (event.webkitCompassHeading !== undefined) {
        if (event.webkitCompassAccuracy < 50) {
            gpsMain.heading = event.webkitCompassHeading;
        } else {
            console.warn('webkitCompassAccuracy is event.webkitCompassAccuracy');
            console.log ('webkitCompassAccuracy is event.webkitCompassAccuracy')
        }
    } else if (event.alpha !== null) {
        if (event.absolute === true || event.absolute === undefined) {
            gpsMain.heading = gpsMain._computeCompassHeading(event.alpha, event.beta, event.gamma);
        } else {
            console.warn('event.absolute === false');
        }
    } else {
        console.warn('event.alpha === null');
    }
},
    /**
 * Get device orientation event name, depends on browser implementation.
 * @returns {string} event name
 */
_getDeviceOrientationEventName: function () {
    if ('ondeviceorientationabsolute' in window) {
        var eventName = 'deviceorientationabsolute'
    } else if ('ondeviceorientation' in window) {
        var eventName = 'deviceorientation'
    } else {
        var eventName = ''
        console.error('Compass not supported')
    }

    return eventName
},
```

3.- create new 3d objects

add as children to object (pivotePoligono)

4.- API hierarchy


```
{
  "status": "success",
  "result": [
    {
      "id": 6,
      "state_id": 10,
      "Name": "00 Nick's Test Spot",
      "Comments": "Do not remove.",
      "lat": 27.4867955520334,
      "lng": -82.4036759454727,
      "color": "#FF0000",
      "html": "Do not remove.",
      "url": "https:\/\/www.saltstrong.com\/",
      "PolygonCoords": "[[{\"lat\":27.487546,\"lng\":-82.403544},{\"lat\":27.487448,\"lng\":-82.403607},
{\"lat\":27.487396,\"lng\":-82.403543},{\"lat\":27.487524,\"lng\":-82.403451},{\"lat\":27.487574,\"lng\":-82.403525}],
[{\"lat\":27.487303,\"lng\":-82.403706},{\"lat\":27.487251,\"lng\":-82.403662},{\"lat\":27.487128,\"lng\":-82.403745},
{\"lat\":27.487169,\"lng\":-82.403785}],[{\"lat\":27.486968,\"lng\":-82.404506},{\"lat\":27.486818,\"lng\":-82.404492},
{\"lat\":27.486645,\"lng\":-82.404522},{\"lat\":27.486494,\"lng\":-82.404699},{\"lat\":27.486388,\"lng\":-82.404972},
{\"lat\":27.486425,\"lng\":-82.405214},{\"lat\":27.48655,\"lng\":-82.405128},{\"lat\":27.486629,\"lng\":-82.404877},
{\"lat\":27.486711,\"lng\":-82.404709}]]",
      "distance": 0.011670617428373074
    },
    {
      "id": 859,
      "state_id": 10,
      "Name": "000 Nick Dev Test",
      "Comments": "You found a Discount Code <BR>\r\nUse: XYZ123 for an additional <BR>10% off!",
```


Result (array) >

PolygonCoords (Group, array)>

Polygon (array)>

vertex (json{lat: , lgn: })