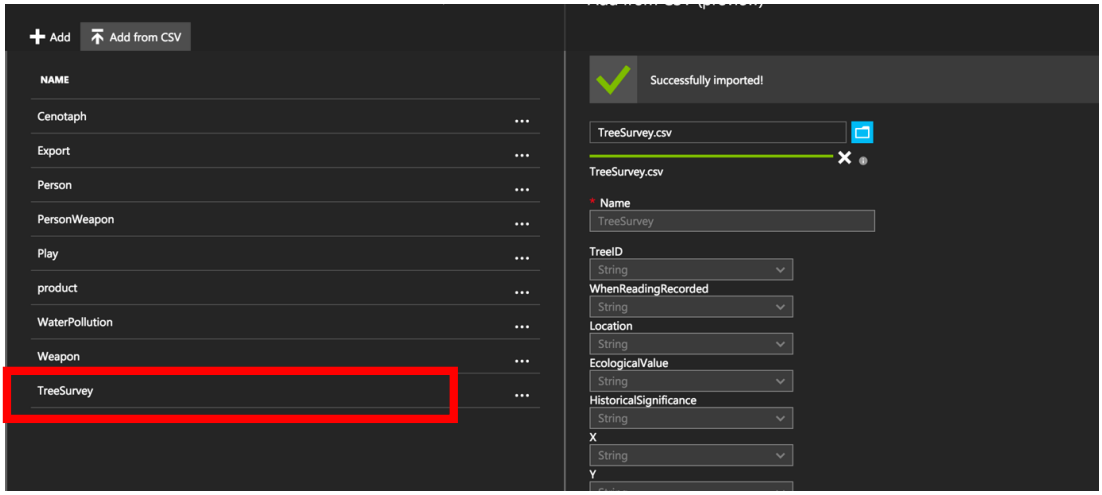
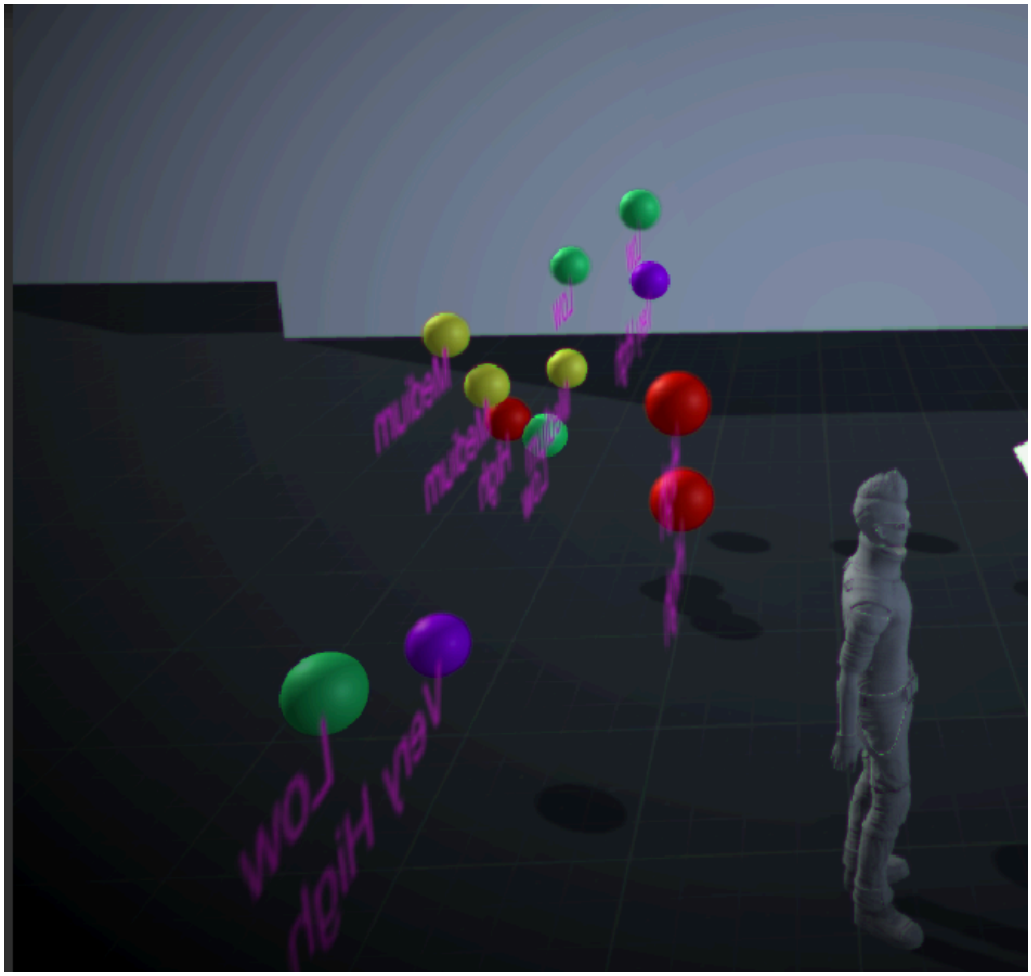
Lab Test 2

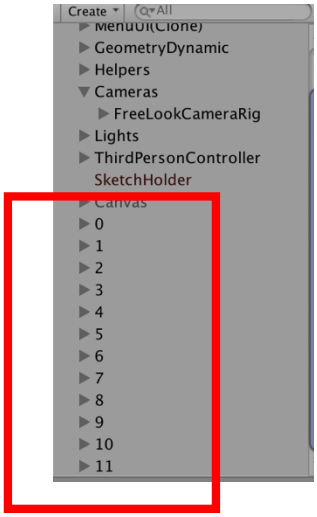
By: Chahat Chawla (ccha504, 8492142) Github link:
https://github.com/chahatchawla/labtest2ccha504

1) Load TreeSurvey.xlsx to Azure



2) Plot into 3D space using any object type – Choose any 3 dimensions





As you can see in these image, 12 spheres are plotted into the 3D Space using the **x y z coordinates** from the TreeSurvey data.

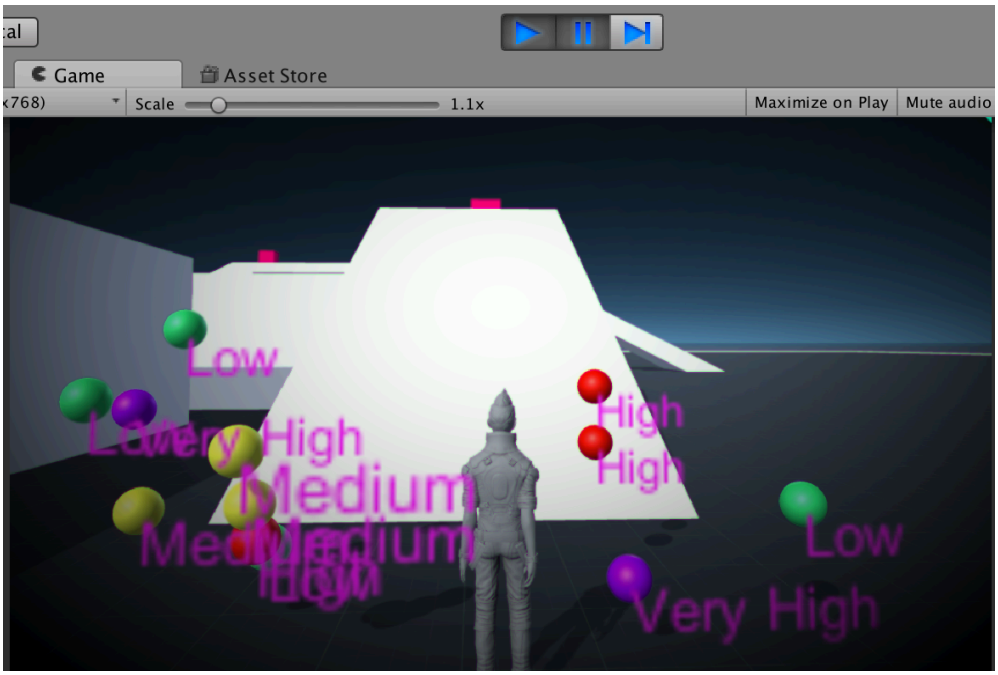
To do this:

1) x, y, z data from table is converted into floats
2) using the coordinate, the SphereWithText prefab is instantiated in the 3D space

```
void Start () {
    //Reguest.GET can be called passing in your ODATA url as a string in the form:
    //http://{Your Site Name}.azurewebsites.net/tables/{Your Table Name}?zumo-api-version=2.0.0
    //The response produce is a JSON string
    jsonResponse = Request.GET(_WebsiteURL);

    //Just in case something went wrong with the request we check the reponse and exit if there is no response.
    if (string.IsNullOrEmpty(jsonResponse))
    {
        return;
    }

    //We can now deserialize into an array of objects - in this case the class we created. The deserializer is smart enough to
    data = JsonReader.Deserialize<TreeSurvey[]>(jsonResponse);
    int i = 0;

    //We can now loop through the array of objects and access each object individually
    foreach (TreeSurvey reading in data)
    {
        //convert the string readings to floats - as the data given is in floats
        float x = float.Parse (reading.X);
        float y = float.Parse (reading.Y);
        float z = float.Parse (reading.Z);

        //Plot the prefab "SphereWithText" on the3D space using the reading data
        GameObject newSphere = (GameObject)Instantiate(myPrefab, new Vector3(x, y, z), Quaternion.identity);
        newSphere.name = i.ToString();

        //Label on each point with their Ecological Value
        newSphere.GetComponentInChildren<TextMesh>().text = reading.EcologicalValue;

        //Different physical (color) attribute depending on Ecological Value
        if (reading.EcologicalValue == "Very High") {
            newSphere.GetComponent<Renderer> ().material = material2;
        } else if (reading.EcologicalValue == "High") {
            newSphere.GetComponent<Renderer> ().material = material3;
        } else if (reading.EcologicalValue == "Medium") {
            newSphere.GetComponent<Renderer> ().material = material4;
        }else {
            newSphere.GetComponent<Renderer> ().material = material5;
        }
```

3) Put label on each point

The label chosen is the **Ecological Value** of the reading as it is an important piece of information.
Depending on whether the ecological value of the reading is very high, high, medium or low, the sphere's colour changes (purple = very high, red = high, yellow = medium, green = low). This supports the the labelling affect.




```
void Start () {
    //Reguest.GET can be called passing in your ODATA url as a string in the form:
    //http://{Your Site Name}.azurewebsites.net/tables/{Your Table Name}?zumo-api-version=2.0.0
    //The response produce is a JSON string
    jsonResponse = Request.GET(_WebsiteURL);

    //Just in case something went wrong with the request we check the reponse and exit if there is no response.
    if (string.IsNullOrEmpty(jsonResponse))
    {
        return;
    }

    //We can now deserialize into an array of objects - in this case the class we created. The deserializer is smart enough to
    data = JsonReader.Deserialize<TreeSurvey[]>(jsonResponse);
    int i = 0;

    //We can now loop through the array of objects and access each object individually
    foreach (TreeSurvey reading in data)
    {
        //convert the string readings to floats - as the data given is in floats
        float x = float.Parse (reading.X);
        float y = float.Parse (reading.Y);
        float z = float.Parse (reading.Z);

        //Plot the prefab "SphereWithText" on the3D space using the reading data
        GameObject newSphere = (GameObject)Instantiate(myPrefab, new Vector3(x, y, z), Quaternion.identity);
        newSphere.name = i.ToString();

        //Label on each point with their Ecological Value
        newSphere.GetComponentInChildren<TextMesh>().text = reading.EcologicalValue;

        //Different physical (color) attribute depending on Ecological Value
        if (reading.EcologicalValue == "Very High") {
            newSphere.GetComponent<Renderer> ().material = material2;
        } else if (reading.EcologicalValue == "High") {
            newSphere.GetComponent<Renderer> ().material = material3;
        } else if (reading.EcologicalValue == "Medium") {
            newSphere.GetComponent<Renderer> ().material = material4;
        }else {
            newSphere.GetComponent<Renderer> ().material = material5;
        }
```

4) Select Object (Using raycast)

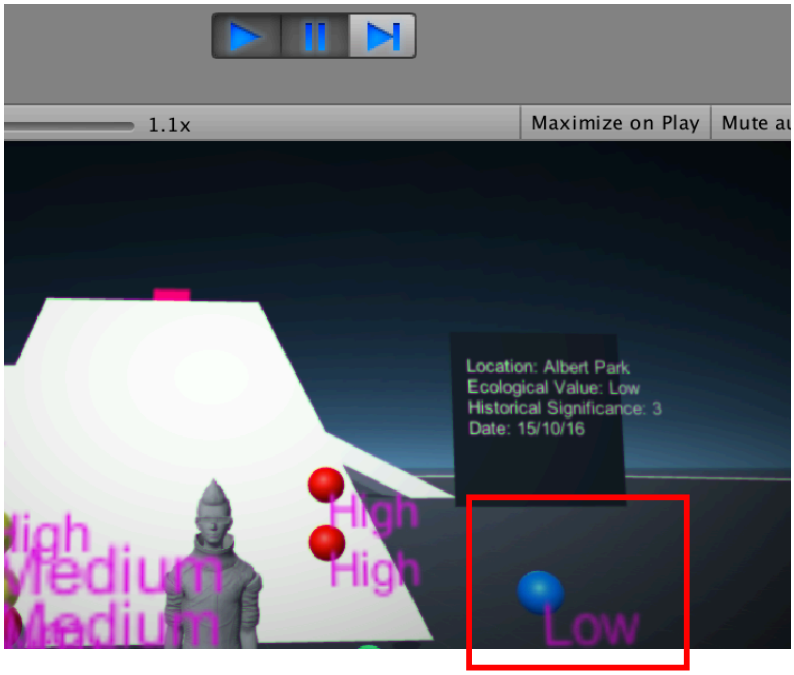
Raycasting was used to select an object. The following screenshot shows the code for it.

```
// Update is called once per frame
void Update () {

    //Get Mouse position
    RaycastHit hitInfo = new RaycastHit ();
    Ray ray = Camera.main.ScreenPointToRay (Input.mousePosition);

    //If left mouse button clicked
    if (Input.GetMouseButtonDown (0)) {
        if (Physics.Raycast (ray, out hitInfo)) {

            //If the object selected was a sphere
            if (hitInfo.collider.tag == "Sphere")

                //Change its color to blue
                hitInfo.collider.gameObject.GetComponent<Renderer> ().material = material1;

                //Get index of sphere object in the tree survey data
                int index = int.Parse (hitInfo.collider.gameObject.name);

                //Create a new panel that shows the point data
                GameObject newPanel = (GameObject)Instantiate (cube,
                new Vector3 (hitInfo.point.x, hitInfo.point.y + 1.0f, hitInfo.point.z), Quaternion.identity);
                newPanel.GetComponentInChildren<TextMesh> ().text = "Location: " + data [index].Location
                + "\n" + "Ecological Value: " + data [index].EcologicalValue
                + "\n" + "Historical Significance: " + data [index].HistoricalSignificance
                +"\n" + "Date: " + data [index].WhenReadingRecorded;
            }

            //If the object selected was a cube/panel
            if (hitInfo.collider.tag == "Cube") {
                //Destroy the panel
                Destroy (hitInfo.collider.gameObject);
            }
        }
    }
```

5) When Object is selected – change visible attribute

So when a sphere (single reading point) is selected/clicked on then the selected sphere turns **blue**. The following screenshot shows the code for it.

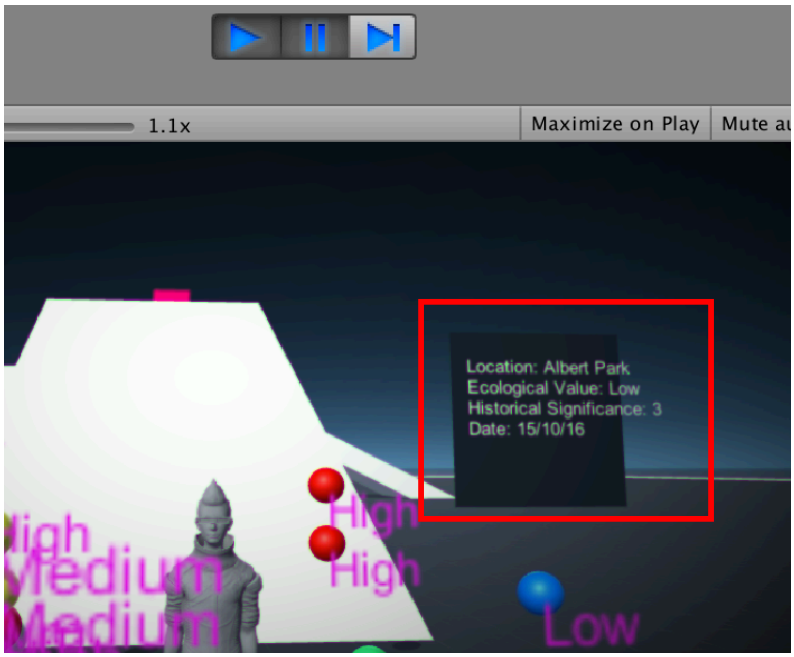
```
// Update is called once per frame
void Update () {


    //Get Mouse position
    RaycastHit hitInfo = new RaycastHit ();
    Ray ray = Camera.main.ScreenPointToRay (Input.mousePosition);

    //If left mouse button clicked
    if (Input.GetMouseButtonDown (0)) {
        if (Physics.Raycast (ray, out hitInfo)) {

            //If the object selected was a sphere
            if (hitInfo.collider.tag == "Sphere") {

                //Change its color to blue
                hitInfo.collider.gameObject.GetComponent<Renderer> ().material = material1;

                //Get index of sphere object in the tree survey data
                int index = int.Parse (hitInfo.collider.gameObject.name);

                //Create a new panel that shows the point data
                GameObject newPanel = (GameObject)Instantiate (cube,
                                        new Vector3 (hitInfo.point.x, hitInfo.point.y + 1.0f, hitInfo.point.z), Quaternion.identity);
                newPanel.GetComponentInChildren<TextMesh> ().text = "Location: " + data [index].Location
                + "\n" + "Ecological Value: " + data [index].EcologicalValue
                + "\n" + "Historical Significance: " + data [index].HistoricalSignificance
                + "\n" + "Date: " + data [index].WhenReadingRecorded;
            }

            //If the object selected was a cube/panel
            if (hitInfo.collider.tag == "Cube") {
                //Destroy the panel
                Destroy (hitInfo.collider.gameObject);
            }
        }
    }
}
```



6) When Object is selected – display panel with point data

So when a sphere (single reading point) is selected/clicked on then a panel is instantiated and the text on the panel is updated to show the data related to the selected point entry. The following screenshot shows the code for it.

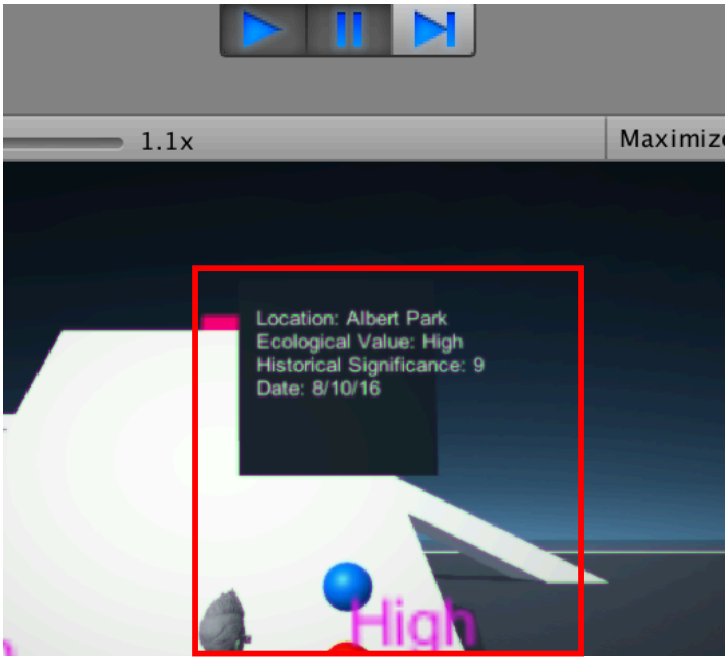
```
// Update is called once per frame
void Update () {


    //Get Mouse position
    RaycastHit hitInfo = new RaycastHit ();
    Ray ray = Camera.main.ScreenPointToRay (Input.mousePosition);

    //If left mouse button clicked
    if (Input.GetMouseButtonDown (0)) {
        if (Physics.Raycast (ray, out hitInfo)) {

            //If the object selected was a sphere
            if (hitInfo.collider.tag == "Sphere") {

                //Change its color to blue
                hitInfo.collider.gameObject.GetComponent<Renderer> ().material = material1;

                //Get index of sphere object in the tree survey data
                int index = int.Parse (hitInfo.collider.gameObject.name);

                //Create a new panel that shows the point data
                GameObject newPanel = (GameObject)Instantiate (cube,
                                      new Vector3 (hitInfo.point.x, hitInfo.point.y + 1.0f, hitInfo.point.z), Quaternion.identity);
                newPanel.GetComponentInChildren<TextMesh> ().text = "Location: " + data [index].Location
                + "\n" + "Ecological Value: " + data [index].EcologicalValue
                + "\n" + "Historical Significance: " + data [index].HistoricalSignificance
                + "\n" + "Date: " + data [index].WhenReadingRecorded;
            }

            //If the object selected was a cube/panel
            if (hitInfo.collider.tag == "Cube") {
                //Destroy the panel
                Destroy (hitInfo.collider.gameObject);
            }
        }
    }
}
```



As seen in this screenshot, the panel is instantiated 1.0f above (y) the sphere that was selected.

The panel shows the following point data: Location, Ecological Value, Historical Significance and Date Recorded.

Another example of the panel:



The panel can also be closed (destroyed) by clicking on the panel itself. The following screenshot shows the code for it.

```
// Update is called once per frame
void Update () {


    //Get Mouse position
    RaycastHit hitInfo = new RaycastHit ();
    Ray ray = Camera.main.ScreenPointToRay (Input.mousePosition);

    //If left mouse button clicked
    if (Input.GetMouseButtonDown (0)) {
        if (Physics.Raycast (ray, out hitInfo)) {

            //If the object selected was a sphere
            if (hitInfo.collider.tag == "Sphere") {

                //Change its color to blue
                hitInfo.collider.gameObject.GetComponent<Renderer> ().material = material1;

                //Get index of sphere object in the tree survey data
                int index = int.Parse (hitInfo.collider.gameObject.name);

                //Create a new panel that shows the point data
                GameObject newPanel = (GameObject)Instantiate (cube,
                                                new Vector3 (hitInfo.point.x, hitInfo.point.y + 1.0f, hitInfo.point.z), Quaternion.identity);
                newPanel.GetComponentInChildren<TextMesh> ().text = "Location: " + data [index].Location
                + "\n" + "Ecological Value: " + data [index].EcologicalValue
                + "\n" + "Historical Significance: " + data [index].HistoricalSignificance
                + "\n" + "Date: " + data [index].WhenReadingRecorded;
            }

            //If the object selected was a cube/panel
            if (hitInfo.collider.tag == "Cube") {
                //Destroy the panel
                Destroy (hitInfo.collider.gameObject);
            }
        }
    }
}
```

7) Place Unity project in Github:

https://github.com/chahatchawla/labtest2ccha504