DIRECTED RESEARCH REPORT

Submitted by:
Akash Mohan Das
Graduate Student
Department of Electrical Engineering
Viterbi School of Engineering
USC

Under the Guidance of :
Dr. David Krum
Associate Director
ICT Mixed Reality Lab

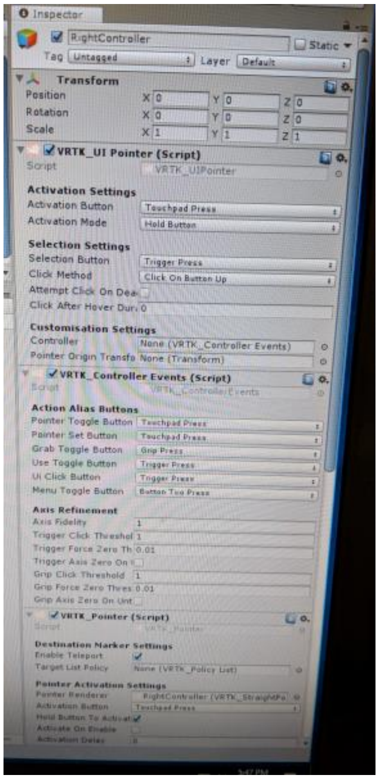
Abstract :

The research was aimed at visualizing the data in the VR environment. The stats of population and GDP of 50 states of USA was taken in the CSV format. The graphs for which were plotted and visualized in the Unity using VRTK.

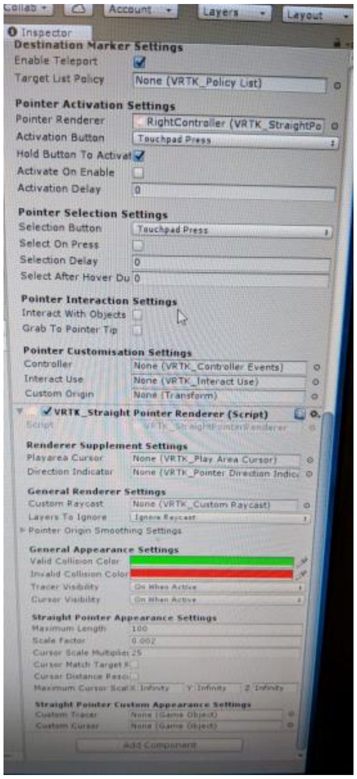
Work:

Based on work done in the first phase i.e. ray casting on an object and plotting iris data, the two works were concatenated to plot population vs states and GDP vs states. VRTK was used to ray cast from the controller on buttons 'Population' and 'GDP' which upon trigger press plotted the respective graphs. A canvas was setup for the buttons and the 'on click' component of button was used to setup a function to follow on clicking the button. The function here was 'plot graph' for respective buttons which takes the string as argument that specifies the file from where CSV data is read from.

The Hierarchy and Inspector looked as follows :

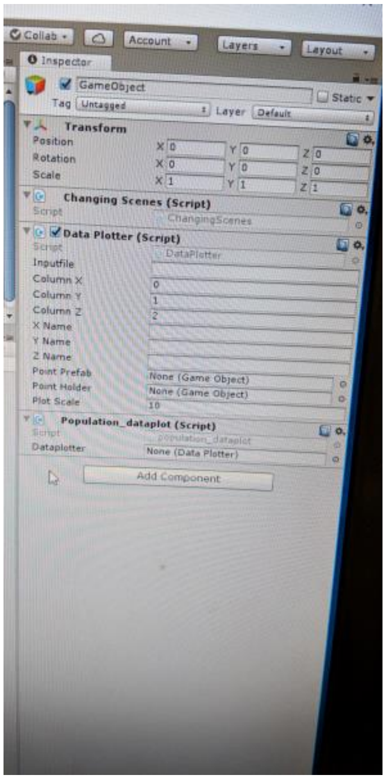


☑ RightController			
Tag	Untagged	Layer	Default
Transform			
Position	X 0	Y 0	Z 0
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1
☑ **VRTK_UI Pointer (Script)**			
Script	VRTK_UIPointer		
Activation Settings			
Activation Button	Touchpad Press		
Activation Mode	Hold Button		
Selection Settings			
Selection Button	Trigger Press		
Click Method	Click On Button Up		
Attempt Click On Dea	□		
Click After Hover Dur	0		
Customisation Settings			
Controller	None (VRTK_Controller Events)		
Pointer Origin Transfo	None (Transform)		
☑ **VRTK_Controller Events (Script)**			
Script	VRTK_ControllerEvents		
Action Alias Buttons			
Pointer Toggle Button	Touchpad Press		
Pointer Set Button	Touchpad Press		
Grab Toggle Button	Grip Press		
Use Toggle Button	Trigger Press		
Ui Click Button	Trigger Press		
Menu Toggle Button	Button Two Press		
Axis Refinement			
Axis Fidelity	1		
Trigger Click Threshol	1		
Trigger Force Zero Th	0.01		
Trigger Axis Zero On	□		
Grip Click Threshold	1		
Grip Force Zero Thres	0.01		
Grip Axis Zero On Unt	□		
☑ **VRTK_Pointer (Script)**			
Script	VRTK_Pointer		
Destination Marker Settings			
Enable Teleport	☑		
Target List Policy	None (VRTK_Policy List)		
Pointer Activation Settings			
Pointer Renderer	RightController (VRTK_StraightPo		
Activation Button	Touchpad Press		
Hold Button To Activat	☑		
Activate On Enable	□		
Activation Delay	0		



Destination Marker Settings	
Enable Teleport	☑
Target List Policy	None (VRTK_Policy List)
Pointer Activation Settings	
Pointer Renderer	RightController (VRTK_StraightPo
Activation Button	Touchpad Press
Hold Button To Activat	☑
Activate On Enable	□
Activation Delay	0
Pointer Selection Settings	
Selection Button	Touchpad Press
Select On Press	□
Selection Delay	0
Select After Hover Du	0
Pointer Interaction Settings	
Interact With Objects	□
Grab To Pointer Tip	□
Pointer Customisation Settings	
Controller	None (VRTK_Controller Events)
Interact Use	None (VRTK_Interact Use)
Custom Origin	None (Transform)
☑ **VRTK_Straight Pointer Renderer (Script)**	
Script	VRTK_StraightPointerRenderer
Renderer Supplement Settings	
Playarea Cursor	None (VRTK_Play Area Cursor)
Direction Indicator	None (VRTK_Pointer Direction Indic
General Renderer Settings	
Custom Raycast	None (VRTK_Custom Raycast)
Layers To Ignore	Ignore Raycast
Pointer Origin Smoothing Settings	
General Appearance Settings	
Valid Collision Color	🟩
Invalid Collision Color	🟥
Tracer Visibility	On When Active
Cursor Visibility	On When Active
Straight Pointer Appearance Settings	
Maximum Length	100
Scale Factor	0.002
Cursor Scale Multiplier	25
Cursor Match Target R	□
Cursor Distance Resc	□
Maximum Cursor Scal	X Infinity Y Infinity Z Infinity
Straight Pointer Custom Appearance Settings	
Custom Tracer	None (Game Object)
Custom Cursor	None (Game Object)

Fig.1 and fig.2 Components of Right and Left Controller



Inspector

☑ GameObject ☐ Static

Tag Untagged Layer Default

Transform

Position X 0 Y 0 Z 0

Rotation X 0 Y 0 Z 0

Scale X 1 Y 1 Z 1

Changing Scenes (Script)

Script ChangingScenes

☑ Data Plotter (Script)

Script DataPlotter

Inputfile

Column X 0

Column Y 1

Column Z 2

X Name

Y Name

Z Name

Point Prefab None (Game Object)

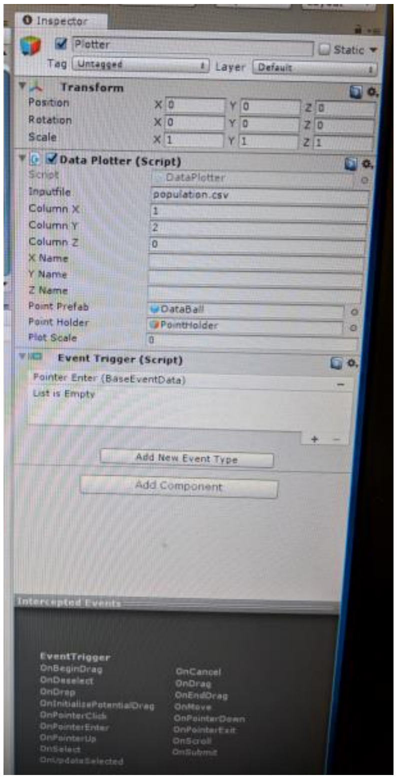
Point Holder None (Game Object)

Plot Scale 10

Population_dataplot (Script)

Script population_dataplot

Dataplotter None (Data Plotter)



Inspector

☑ Plotter ☐ Static

Tag Untagged Layer Default

Transform

Position X 0 Y 0 Z 0

Rotation X 0 Y 0 Z 0

Scale X 1 Y 1 Z 1

☑ Data Plotter (Script)

Script DataPlotter

Inputfile population.csv

Column X 1

Column Y 2

Column Z 0

X Name

Y Name

Z Name

Point Prefab DataBall

Point Holder PointHolder

Plot Scale 0

Event Trigger (Script)

Pointer Enter (BaseEventData)

List is Empty

Intercepted Events

EventTrigger

OnBeginDrag OnCancel

OnDeselect OnDrag

OnDrop OnEndDrag

OnInitializePotentialDrag OnMove

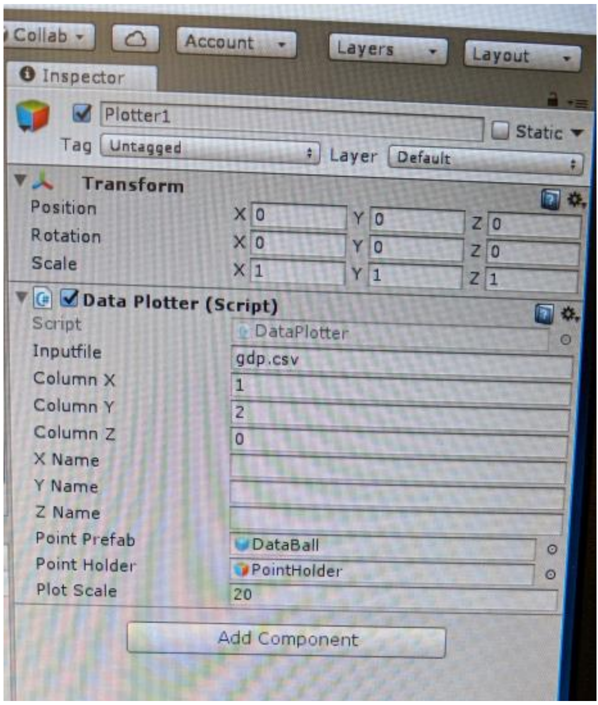
OnPointerClick OnPointerDown

OnPointerEnter OnPointerExit

OnPointerUp OnScroll

OnSelect OnSubmit

OnUpdateSelected



Inspector

☑ Plotter1 ☐ Static

Tag Untagged Layer Default

Transform

Position X 0 Y 0 Z 0

Rotation X 0 Y 0 Z 0

Scale X 1 Y 1 Z 1

☑ Data Plotter (Script)

Script DataPlotter

Inputfile gdp.csv

Column X 1

Column Y 2

Column Z 0

X Name

Y Name

Z Name

Point Prefab DataBall

Point Holder PointHolder

Plot Scale 20

Fig3 : Components of an empty Game object that contains Plotter. Fig4: Components of Plotter(Population). Fig.5: Components o f Plooter1(GDP)

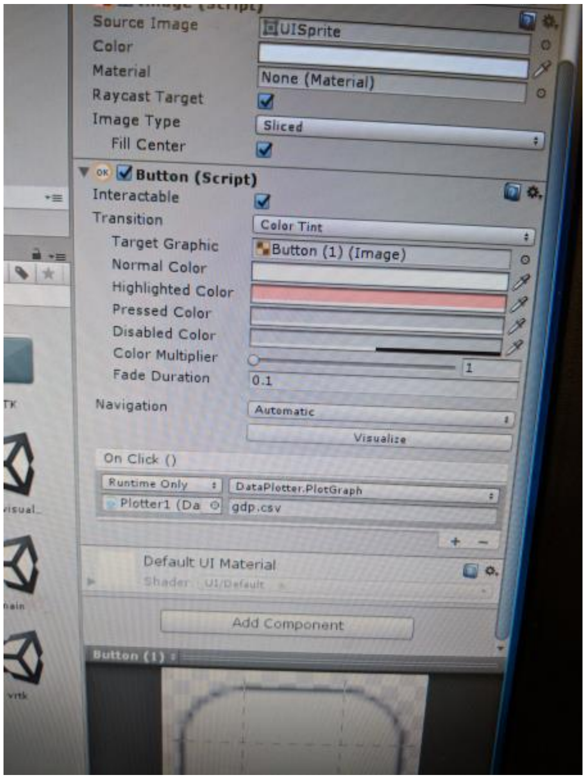


Image (Script)

Source Image UISprite

Color

Material None (Material)

Raycast Target ☑

Image Type Sliced

Fill Center ☑

☑ Button (Script)

Interactable ☑

Transition Color Tint

Target Graphic Button (1) (Image)

Normal Color

Highlighted Color

Pressed Color

Disabled Color

Color Multiplier 1

Fade Duration 0.1

Navigation Automatic

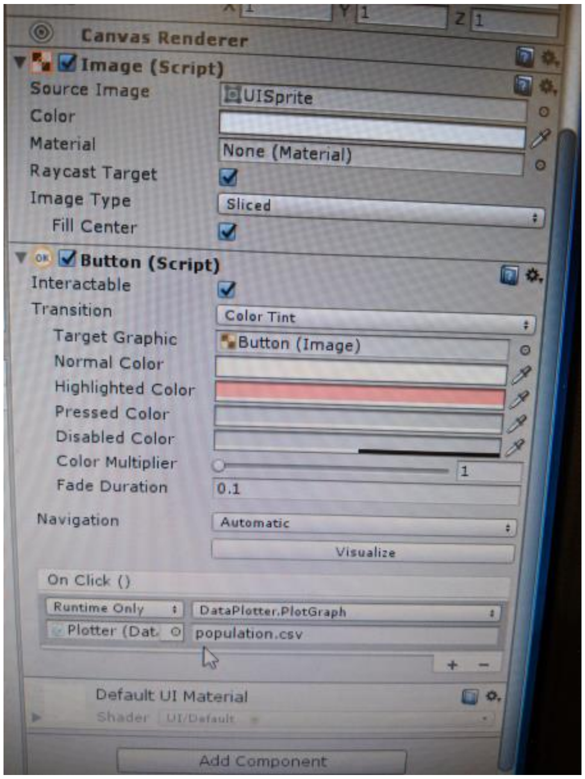
On Click ()

Runtime Only DataPlotter.PlotGraph

Plotter1 (Da gdp.csv

Default UI Material

Shader UI/Default



Canvas Renderer

☑ Image (Script)

Source Image UISprite

Color

Material None (Material)

Raycast Target ☑

Image Type Sliced

Fill Center ☑

☑ Button (Script)

Interactable ☑

Transition Color Tint

Target Graphic Button (Image)

Normal Color

Highlighted Color

Pressed Color

Disabled Color

Color Multiplier 1

Fade Duration 0.1

Navigation Automatic

On Click ()

Runtime Only DataPlotter.PlotGraph

Plotter (Dat population.csv

Default UI Material

Shader UI/Default

Fig. 6: Components of Button (GDP)

Fi g.7 : Components of Button(Population)

Code for Plot graph :

```
using System.Collections;
using System.Collections.Generic;
```

Code for Plot graph :

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System;

public class DataPlotter : MonoBehaviour {
public string inputfile;
    private List<Dictionary<string, object>> pointList;
    public int columnX = 0;
    public int columnY = 1;
    public int columnZ = 2;
    public string xName;
    public string yName;
    public string zName;
    public GameObject PointPrefab;
    public GameObject PointHolder;
    public float plotScale;
    Color color;
float xMax;
    float yMax;
    float zMax;
    float xMin;
    float yMin;
    float zMin;
Vector3[] dataPoints;
// Use this for initialization
    public void Start () {
        pointList = CSVReader.Read(inputfile);
//Debug.Log(pointList);
        List<string> columnList = new List<string>(pointList[1].Keys);
        //Debug.Log("There are " + columnList.Count + " columns in CSV");
//foreach (string key in columnList)
        //    Debug.Log(" Column name is" + key);
        xName = columnList[columnX];
        yName = columnList[columnY];
        //zName = columnList[columnZ];
        //xMax = findMaxVal(xName);
        //yMax = findMaxVal(yName);
        //zMax = findMaxVal(zName);
        //xMin = findMinVal(xName);
        //yMin = findMinVal(yName);
        //zMin = findMinVal(zName);
dataPoints = new Vector3[50];
        PointPrefab.SetActive(false);
GameObject DataPoint;
//First Instantiate All Points
        for(int i=0; i<50; i++)
        {
            DataPoint = Instantiate(PointPrefab);
            DataPoint.transform.parent = PointHolder.transform;
            dataPoints[i] = DataPoint.transform.position;
        }
}
private float findMaxVal(string columnName)
    {
        float maxVal = Convert.ToSingle(pointList[0][columnName]);
for( var i=0; i<pointList.Count; i++)
        {
            if(maxVal < Convert.ToSingle(pointList[i][columnName]))
            {
                maxVal = Convert.ToSingle(pointList[i][columnName]);
}
        }
        return maxVal;
    }
private float findMinVal(string columnName)
    {
        float minVal = Convert.ToSingle(pointList[0][columnName]);
for (var i = 0; i < pointList.Count; i++)
        {
            if (minVal > Convert.ToSingle(pointList[i][columnName]))
            {
                minVal = Convert.ToSingle(pointList[i][columnName]);
}
        }
        return minVal;
    }
public void PlotGraph(string filename)
    {
pointList = CSVReader.Read(inputfile);
for (var i = 0; i < pointList.Count; i++)
        {
            float x = System.Convert.ToSingle(pointList[i][xName]);
            float y = System.Convert.ToSingle(pointList[i][yName]);
            float z = 20.37f;
            //float z = (System.Convert.ToSingle(pointList[i][zName]) - zMin) / (zMax - zMin);
dataPoints[i] = new Vector3(x, y, z);
        }
int j = 0;
        foreach (Transform child in PointHolder.transform)
        {
            child.transform.position = dataPoints[j]  ;
            child.gameObject.GetComponent<Renderer>().material.color = new Color(dataPoints[j].x, dataPoints[j].y, dataPoints[j].z, 1.0f);
            child.gameObject.SetActive(true);
            j++;
            }
            // UpdatePlot();
//GameObject dataPoint = Instantiate(PointPrefab, new Vector3(x, y, z) * plotScale, Quaternion.identity);
            //dataPoint.transform.parent = PointHolder.transform;
            //dataPoint.SetActive(true);
            //string dataPointName = pointList[i][xName] + " " + pointList[i][yName] + " " + pointList[i][zName];
            //dataPoint.transform.name = dataPointName;
            //dataPoint.GetComponent<Renderer>().material.color = new Color(x, y, z, 1.0f);
            //if (String.Compare(System.Convert.ToString(pointList[i][5]), "setosa", true))
            //{
            //    color = new Color(1, 0, 0, 1.0f);
            //}
            //else if(String.Compare(System.Convert.ToString(pointList[i][5]), "versicolor",true))
            //{
            //    color = new Color(0, 1, 0, 1.0f);
            //}
            //else
            //{
            //    color = new Color(0, 0, 1, 1.0f);
            //}
            //dataPoint.GetComponent<Renderer>().material.color = color;
}
public void UpdatePlot()
    {
        int i = 0;
        foreach(Transform child in PointHolder.transform)
        {
            child.transform.position = dataPoints[i] * plotScale;
            child.gameObject.GetComponent<Renderer>().material.color = new Color(dataPoints[i].x, dataPoints[i].y, dataPoints[i].z, 1.0f);
            child.gameObject.SetActive(true);
            i++;
        }
    }
}
```

Outputs:

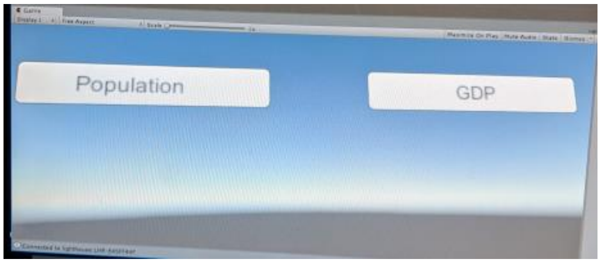


Fig. 8: Initial setup

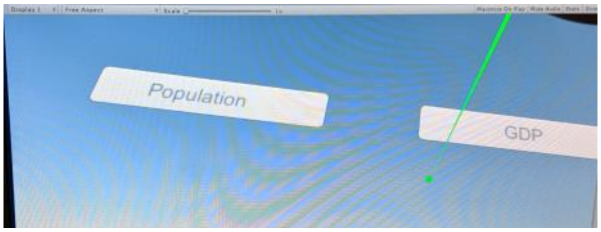


Fig 9: On Ray cast

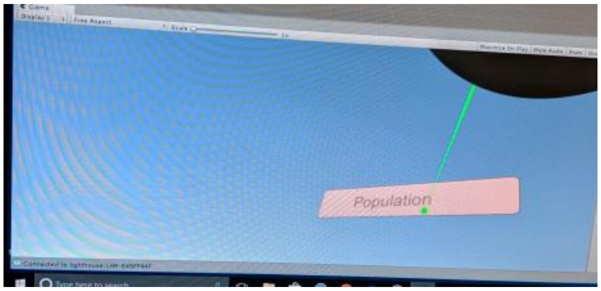


Fig 10: Selection of Population button

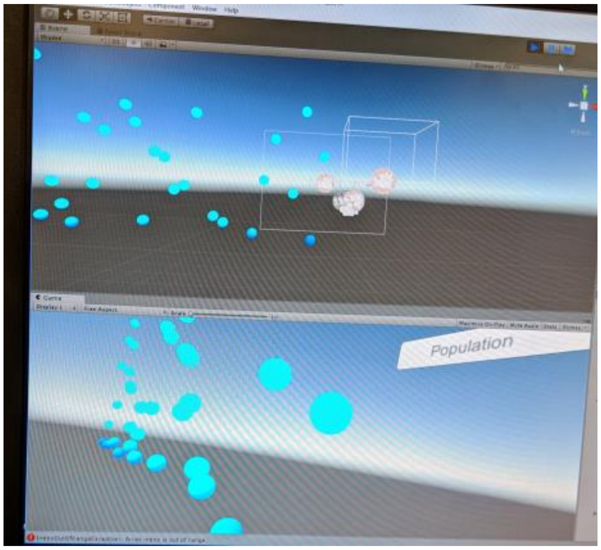


Fig 11: Plot of Population of 50 states

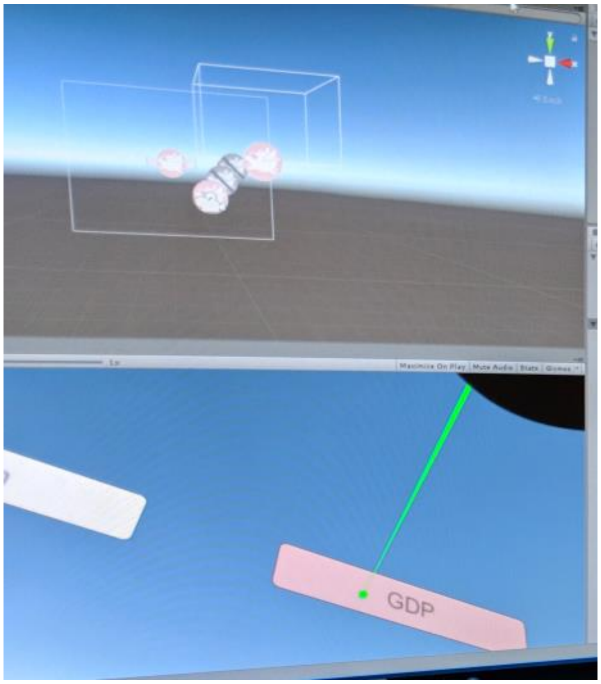


Fig 12: Selecting GDP button

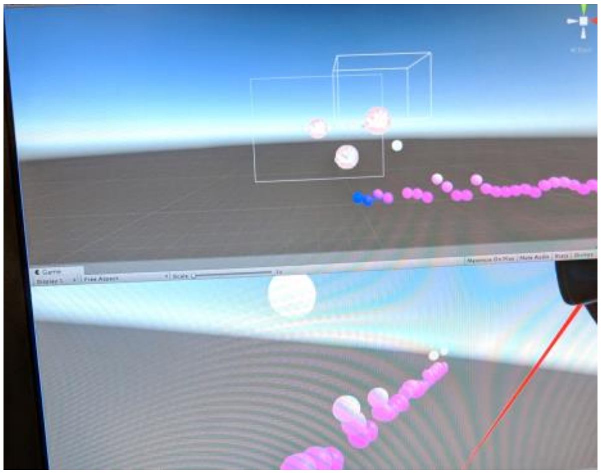


Fig 13: Plot of GDP of 50 states

More to do: I had planned to ray cast on each data ball and highlight the data point with corresponding Population / GDP valu e and name of the State. Due to time constraints, couldn't finish it up.