

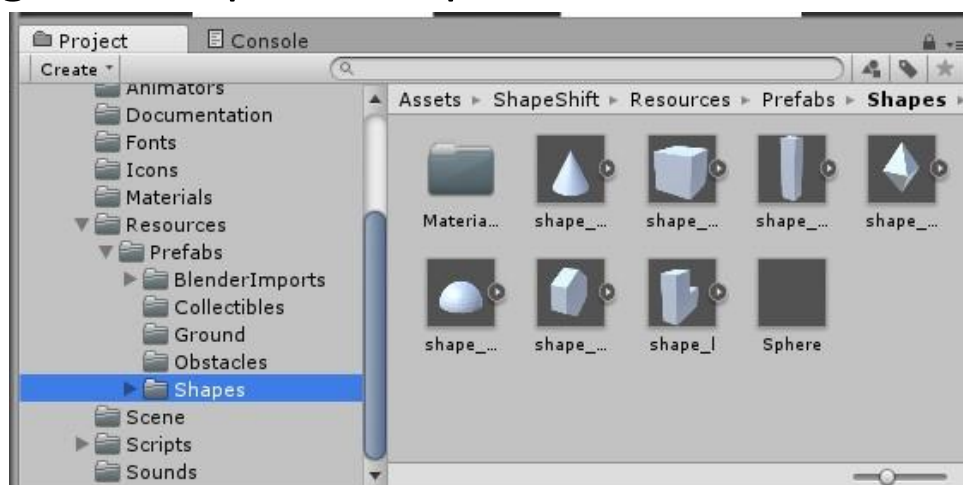
**SHAPE SHIFT**  
**DOCUMENTATION**  
**FOR**  
**UNITY ASSETS STORE**

# ADDING NEW SHAPES AND OBSTACLES

1) Drag and drop the obstacle in the this folder



2) Drag and drop the shape in the this folder



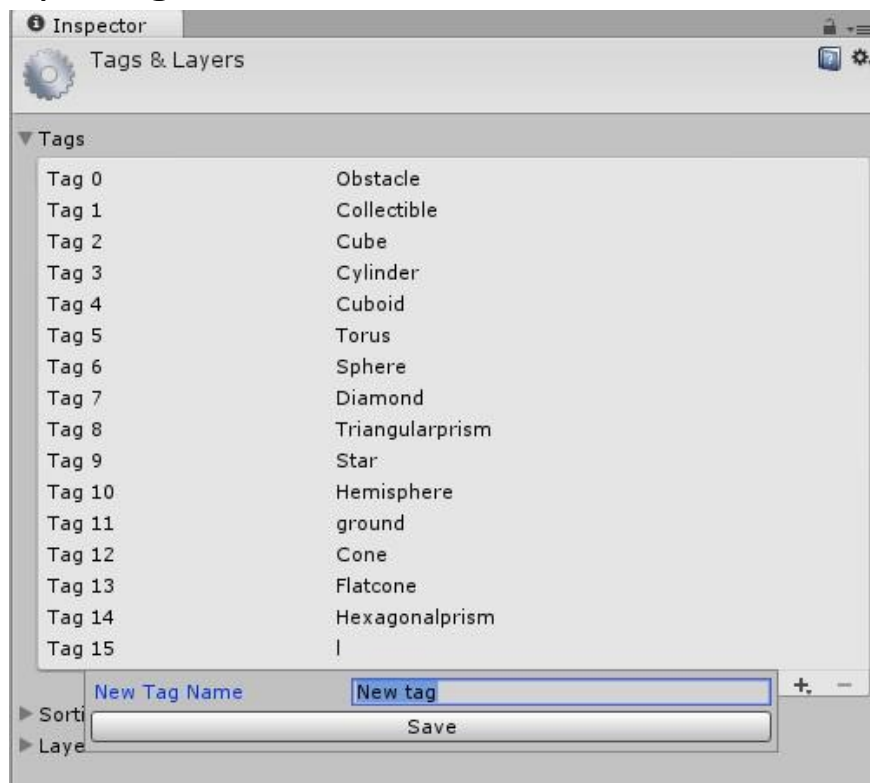
### 3)add the shape name

This is the location oof the script below :

Shape Shift\Assets\ShapeShift\Scripts\Properties\Shapes

```
public enum shape
{
    Cube, Sphere, Triangularprism, 1, Cone, Cuboid , Torus , Diamond , Hexagonalprism, Hemisphere// add new shape name here
}
```

### 4)add shape tag



## 5)add the commented lines of code

- prefabs/Obstacles/ add the name of the obstacle placed in the folder in “1” above .
- prefabs/Shapes/ add the name of the shape placed in the folder in “2” above

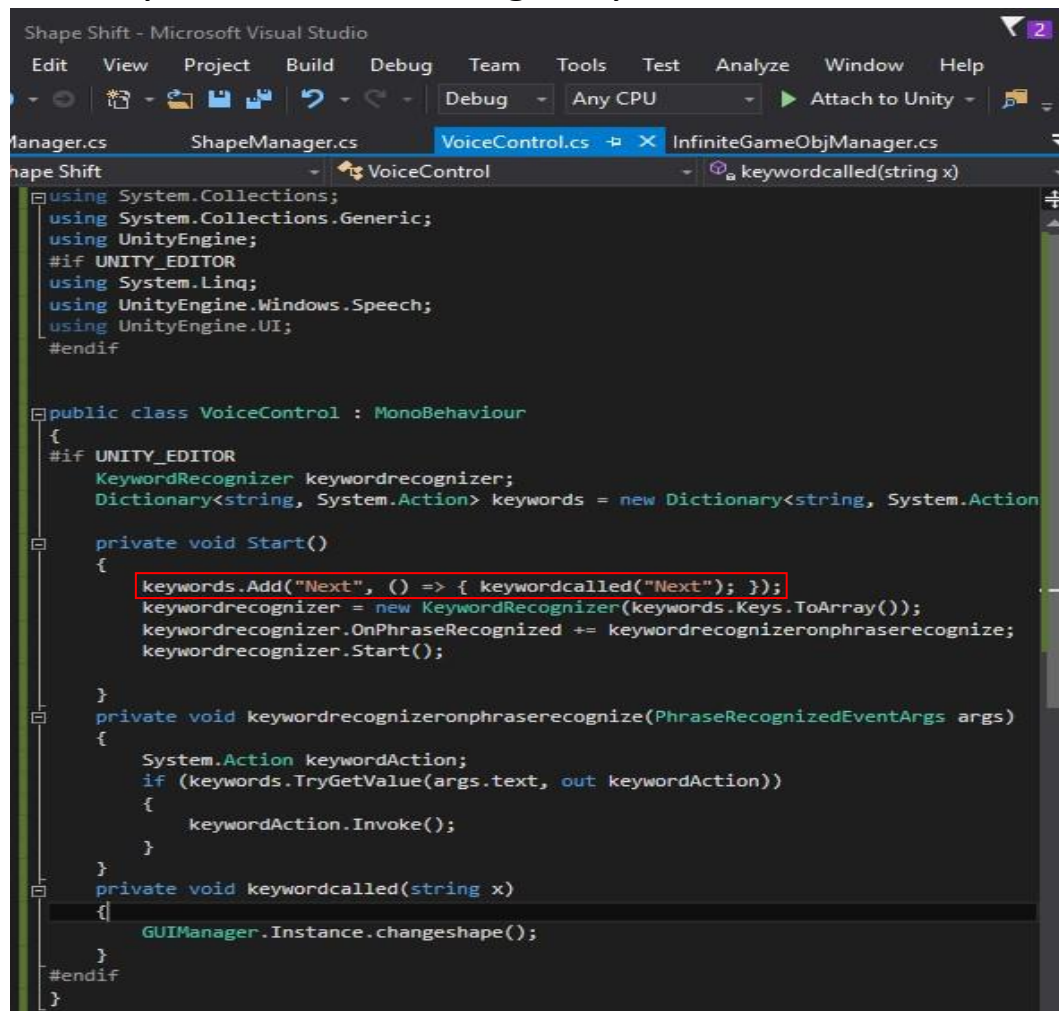
NB: Look our for name spaces.

- Select the shape type in shape.**shape name**

```
}
public IEnumerator GameObjPropertiesLoad()
{
    Ground.Add(new Ground(Resources.Load<GameObject>("Prefabs/Ground/StartingBlock"))));
    Collectables.Add(new Collectables(Resources.Load<GameObject>("Prefabs/Collectibles/Collectible"))));
    Obstacles.Add(new Obstacles(Resources.Load<GameObject>("Prefabs/Obstacles/obstacle_cube"), false, shape.Cube));
    Obstacles.Add(new Obstacles(Resources.Load<GameObject>("Prefabs/Obstacles/obstacle_cuboid"), false, shape.Cuboid));
    Obstacles.Add(new Obstacles(Resources.Load<GameObject>("Prefabs/Obstacles/obstacle_diamond"), false, shape.Diamond));
    Obstacles.Add(new Obstacles(Resources.Load<GameObject>("Prefabs/Obstacles/obstacle_1"), false, shape.1));
    Obstacles.Add(new Obstacles(Resources.Load<GameObject>("Prefabs/Obstacles/obstacle_cone"), false, shape.Cone));
    Obstacles.Add(new Obstacles(Resources.Load<GameObject>("Prefabs/Obstacles/obstacle_hemisphere"), false, shape.Hemisphere));
    Obstacles.Add(new Obstacles(Resources.Load<GameObject>("Prefabs/Obstacles/obstacle_sphere"), false, shape.Sphere));
    Obstacles.Add(new Obstacles(Resources.Load<GameObject>("Prefabs/Obstacles/obstacle_hexagonalprism"), false, shape.Hexagonalprism));
    // Obstacles.Add(new Obstacles(Resources.Load<GameObject>("Prefabs/Obstacles/shape name"), false, shape.shape name));    Add this line code and place shape name
    shapes.Add(new Shapes(Resources.Load<GameObject>("Prefabs/Shapes/shape_cube"), shape.Cube, false));
    shapes.Add(new Shapes(Resources.Load<GameObject>("Prefabs/Shapes/shape_cuboid"), shape.Cuboid, false));
    shapes.Add(new Shapes(Resources.Load<GameObject>("Prefabs/Shapes/shape_diamond"), shape.Diamond, false));
    shapes.Add(new Shapes(Resources.Load<GameObject>("Prefabs/Shapes/shape_1"), shape.1, false));
    shapes.Add(new Shapes(Resources.Load<GameObject>("Prefabs/Shapes/shape_cone"), shape.Cone, false));
    shapes.Add(new Shapes(Resources.Load<GameObject>("Prefabs/Shapes/shape_hemisphere"), shape.Hemisphere, false));
    shapes.Add(new Shapes(Resources.Load<GameObject>("Prefabs/Shapes/Sphere"), shape.Sphere, false));
    shapes.Add(new Shapes(Resources.Load<GameObject>("Prefabs/Shapes/shape_hexagonalprism"), shape.Hexagonalprism, false));
    //shapes.Add(new Shapes(Resources.Load<GameObject>("Prefabs/Shapes/shape name"), shape.shape name, false));    Add this line code and place shape name
    GUIManager.Instance.UpdateBestScore(ScoreManager.Instance.bestScore);
    GUIManager.Instance.UpdateCollectible(ScoreManager.Instance.collectible);
    InfiniteGameObjManager.Instance.InstantiateGameObjects();
    yield return new WaitForSeconds(1);
    GameManager.Instance.SoundtoggleOnLoad(GameManager.Instance.soundindex);
    yield return null;
}
```

# VOICE CONTROL

- Voice control will only work in unity editor.
- Say “Next” to change shape.
- Look for keyword.add to change keyword.



```
Shape Shift - Microsoft Visual Studio
Edit View Project Build Debug Team Tools Test Analyze Window Help
Manager.cs ShapeManager.cs VoiceControl.cs InfiniteGameManager.cs
Shape Shift VoiceControl keywordcalled(string x)
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
#if UNITY_EDITOR
using System.Linq;
using UnityEngine.Windows.Speech;
using UnityEngine.UI;
#endif

public class VoiceControl : MonoBehaviour
{
    #if UNITY_EDITOR
    KeywordRecognizer keywordrecognizer;
    Dictionary<string, System.Action> keywords = new Dictionary<string, System.Action>();

    private void Start()
    {
        keywords.Add("Next", () => { keywordcalled("Next"); });
        keywordrecognizer = new KeywordRecognizer(keywords.Keys.ToArray());
        keywordrecognizer.OnPhraseRecognized += keywordrecognizeronphraserecognize;
        keywordrecognizer.Start();
    }

    private void keywordrecognizeronphraserecognize(PhraseRecognizedEventArgs args)
    {
        System.Action keywordAction;
        if (keywords.TryGetValue(args.text, out keywordAction))
        {
            keywordAction.Invoke();
        }
    }

    private void keywordcalled(string x)
    {
        GUIManager.Instance.changeshape();
    }
    #endif
}
```

- 
- NB: voice recognition is really slow if anyone has a better script please let me know.

# Adjust spawn frequency

Script listed below is infinite obj manager

```
private void SpawnObstacle(Vector3 pos, int dir)
{
    float distance = Vector3.Distance(lastObstacleSpawnPosition[lastObstacleSpawnPosition.Count - 1], new Vector3(pos.x, pos.y + 5.41f, pos.z));
    float y = Random.Range(5.0f, 20.0f);
    if (distance > y)
    {
        int x = inGameObstacleSpawnIndex[Random.Range(0, inGameObstacleSpawnIndex.Count)];
        obstacle = (GameObject)Instantiate(SaveLoad.Instance.Obstacles[x].obstacleObj, SaveLoad.Instance.Obstacles[x].obstacleObj.transform.position = new Vector3(pos.x, pos.y + 5.5f, pos.z), Quaternion.identity);
        gameObjSpawned.Add(obstacle);
        lastObstacleSpawnPosition.Add(new Vector3(pos.x, pos.y + 5.41f, pos.z));
        obstacle.name = "Obstacle" + "[" + x + "]";
        obstacle.tag = SaveLoad.Instance.Obstacles[x].type.ToString(); // remove
        GameObject.Color.Instance.obstacleColor(obstacle);
        if (dir == 1)
        {
            obstacle.gameObject.transform.rotation = Quaternion.Euler(0, 90, 0);
        }
        else if (dir == 0)
        {
            obstacle.gameObject.transform.rotation = Quaternion.Euler(0, 0, 0);
        }
    }
}

private void SpawnCollectible(Vector3 pos)
{
    float distance = Vector3.Distance(lastCollectibleSpawnPosition[lastCollectibleSpawnPosition.Count - 1], new Vector3(pos.x, pos.y + 5.5f, pos.z));
    float y = Random.Range(30.0f, 200.0f);
    if (distance > y)
    {
        GameObject collectible = (GameObject)Instantiate(SaveLoad.Instance.Collectables[0].CollectablesObj, SaveLoad.Instance.Collectables[0].CollectablesObj.transform.position = new Vector3(pos.x, pos.y + 5.5f, pos.z), Quaternion.identity);
        gameObjSpawned.Add(collectible);
        collectible.transform.rotation = Quaternion.Euler(90, 0, 0);
        collectible.tag = "Collectible";
        collectible.name = "Collectible";
        lastCollectibleSpawnPosition.Add(collectible.transform.position);
    }
}

public void ActiveSound(int x, bool add)
```