Scripting in Unity

- Scripting allows developers to extend functionality
- Unity uses C#, Javascript, and Boo languages (we will use C#)
- C# syntax is almost identical to what you learned in C++
- Typically scripts apply to a single game object
Scripts as Behavior Components

- Scripts can be added to objects as components
- Scripts are used to create behavior
  - Change size or color
  - Apply intelligence to an object
Activity

- Create a new Unity project
- Add a Cube to the project
- Add a script to the Cube
Variables

{type} variableName = {value};

```csharp
using UnityEngine;
using System.Collections;

public class BasicScripting : MonoBehaviour {

    int number = 2; // Created a integer number
    string hello = "Hello World!"; // Created a String
    double bigNumber = 12314.324234; // Created a Double

    // Use this for initialization
    void Start () {
        Debug.Log(hello);
    }

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

{type} functionName({type} inputVariable){ Do Something }

```csharp
using UnityEngine;
using System.Collections;

public class BasicScripting : MonoBehaviour {

    int number = 2; // Created a integer number

    // Multiplies returns the passed in number multiplied by 2
    int multiplyByTwo(int number)
    {
        return number * 2;
    }

    // Use this for initialization
    void Start ()
    {
        Debug.Log (multiplyByTwo(number));
    }

    // Update is called once per frame
    void Update ()
    {
    }
}
```
using UnityEngine;
using System.Collections;

public class BasicScripting : MonoBehaviour {
    int number = 2; // Created a integer number

    // Multiplies returns the passed in number multiplied by 2
    int multiplyByTwo(int number){
        return number * 2;
    }

    // Use this for initialization
    void Start () {
        int answer = multiplyByTwo (number);
        if (answer > 0) {
            Debug.Log ("Our answer is greater than zero!" );
        } else {
            Debug.Log ("Our answer is less than zero :( ");
        }
    }

    // Update is called once per frame
    void Update () {
    }
}
Loops

- Supports for, while, and do-while loops

```csharp
using UnityEngine;
using System.Collections;

public class BasicScripting : MonoBehaviour {
    int number = 5; // Created a integer number
    // Use this for initialization
    void Start () {
        for (int i = 0; i < number; i++) {
            Debug.Log("Times through the loop" + i);
        }
    }
    // Update is called once per frame
    void Update () {
    }
}
```
Passing in Values Through Inspector

Set a variable to public!
Built in Unity Functions

- Awake()
- Start()
- Update()

- Other functions
  (http://docs.unity3d.com/ScriptReference/MonoBehaviour.html)
Awake Function

- A default function in Unity objects
- Should be treated like a class constructor
- Called once, only once, upon initialization
Start Function

- Called after Awake() upon the first frame if and only if the object is enabled
- Called once and only once
- Called before Update()
Getting a Component

1 using UnityEngine;
2 using System.Collections;
3
4 public class BasicScripting : MonoBehaviour {
5     6     public Color color; // Variable is now setable in Inspector
7     8     // Use this for initialization
9     10    void Start () {
11        Debug.Log (GetComponent<Transform> ().position);
12    } 13    14    // Update is called once per frame
15    16    void Update () {
17    }
Activity

○ Pass the starting color of your cube into your script through the inspector

○ At start, randomly change the color of the light in your scene programmatically (Hint: You may have to adjust the intensity to see the effects)

```csharp
using UnityEngine;
using System.Collections;

public class BasicScripting : MonoBehaviour {
    public Color color; // Variable is now setable in Inspector

    // Use this for initialization
    void Start () {
        GetComponent<Renderer>().material.color = color;
    }

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

- Update() is called every frame when the object is enabled
- This is the most used function in Unity
- Time.deltaTime gives you the amount of time since Update() was called last. Use this for animating!
Translate and Rotate Objects

- The Transform Component of an object holds its Position, Rotation, and Scale
- Use GetComponent<>() to change these values

```csharp
1 using UnityEngine;
2 using System.Collections;
3
4 public class BasicScripting : MonoBehaviour {
5     public Color color; // Variable is now setable in Inspector
6
7     // Use this for initialization
8     void Start () {
9         Debug.Log (GetComponent<Transform> ().position);
10     }
11 }
12
13     // Update is called once per frame
14     void Update () {
15         
16     }
17 }
```
Activity

- Make your cube fly directly up into the sky while spinning

Hints:
- Animating must be done frame by frame to show motion
- Sine and Cosine functions are great ways to oscillate up and down
- You can move a set amount from frame to frame (e.g., 2 meters up) or you can use a rate of movement (e.g., 2 m/s) and Time.deltaTime to figure out how far to move since the last frame
In Class Activity

- Build off your winter wonderland scene from yesterday
- Use scripting to make your snowman move around in a circle

Advanced

- Make the snowman sway as it moves
- Psychedelic light show. Create a bunch of spotlights and have them randomly flashing different colors