***1/30 Unity intro***

File>create project

create game objects from Create Tab under Hierarchy

NAVIGATION- perspective mode

Hold right click + WASD (front-left-down-middle)

Project contains four folders

Assets:

-within this folder makes a “scenes” folder, “materials” etc, stay organized w subfolders

Library

ProjectSettings

Temp

Project window in unity (on bottom)

basic commands

right-click>create>folder

right-click>create>material (color)

Assets> Create> Material > Color

To apply a material to multiple game objects:

1.shift select multiply objects

2.go to inspector window, and scroll to materials drop down section

3. Choose “element”

Game Tab allows you to look through the camera

Assets > import package >terrain assets

assets > import package >skybox

To MAKE YOUR GAME PLAYABLE:

file>build settings…

select web player (easiest)

click switch platform

Add scene to the build:

“click add current”

Click build

Save in project (make a folder called BUILD – usually multiple builds)

\*If the .html file File>build settings

Check “offline deployment” to distribute an offline version (generates .js JAVACRIPT files)

***CLASS 2- CODING IN UNITY***

HOW TO USE GITHUB

http://www.youtube.com/watch?v=Hs\_Z99nOKM8

Go to github.com

go to upper right plus button create new repo

git is a source control / backup

commit - snapshot (to save work)

repo - hosts all the commits

local ---push---> remote <---push---

you ---pull---> github

sourcetree (<=== DOWNLOAD THIS)

clone/new

create new repository

... button

go to unity folder

reporitory type: git

folder

.gitignore

select a file in the library/ folder (left click in working copy) right click and click ignore

select ignore everything beneath library

ignore everything in temp/

***SUBMITTING YOUR HOMEWORL (GIT/SOURCETREE)***

github click on plus button to create new repo

name it, public

create repository

copy https url

SOURCETREE:

select all in bottom window, with question marks, what sourcetree is detecting as new

Click ADD

COMMIT (make note of what you’re committing)

Repository > Add remote

name: origin

url: from github

Copy paste url link from github of repository

PUSH

check the box

click okay

Copy paste links from wiki page (third down on right menu)

***HOW TO USE MONODEVELOP IN UNITY*** (CODE IS AT BOTTOM OF THIS DOC)

coding using monodevelop:

unity create folder scripts

create c#

double click after naming

Gen coding vocabulary:

functions = verb (process)

variables = nouns (data)

class = container

“class Mom

age;

CallRobert();

syntax=grammar

save and go back to unity

click console tab

click main camera

attach code to the main camera (drag and drop c#script asset onto main camera)

YOUR HOMEWORK:

- expand your text world! your text world should have at least 6 rooms / nodes... remember a "room" can be a moment in time, or a phase in a branching conversation, etc.

- model a background for your text world using Maya somehow -- really, I just want you to practice using Maya... maybe make a lovely statue to put in the back?

- read 10PRINT, ch. 10 and 15!

PAINT COLORS ONTO YOUR MODEL THAT YOU MADE FOR HOMEWORK!...

- using Maya's "3D Paint" tool... figure it out!

the code for monodevelop (click on the )

using UnityEngine;

using System.Collections;

public class helloworld : MonoBehaviour {

string currentRoom = "lobby";

// Use this for initialization

void Start () {

// if you see"//" that's a comment

// instructions have to go inside functions

Debug.Log ("Hello World");

}

// Update is called once per frame

void Update () {

GetComponent<TextMesh>().text = "Hola Mundo";

if (currentRoom == "lobby") {

GetComponent<TextMesh> ().text += "\nYou are in the lobby. \"Hey,\" says the guard.";

GetComponent<TextMesh> ().text += "\nPress [F] to go to 511.";

if (Input.GetKeyDown (KeyCode.F)) {

currentRoom = "room511";

}

} else if (currentRoom == "room511") {

GetComponent<TextMesh> ().text += "\nYou are at Building Worlds.";

GetComponent<TextMesh> ().text += "\nPress [G] to go to my room.";

if (Input.GetKeyDown (KeyCode.G)) {

currentRoom = "myroom";

}

} else if (currentRoom == "myroom") {

GetComponent<TextMesh> ().text += "\nsssssssssssss";

}

}

}

===================================

GENERAL QUICKSTART / TIPS FOR USING THIS TOOL AND GETTING THE RESULTS INTO UNITY

1) make sure you've selected an object to paint! you also may need to do "Automatic Mapping" of UVs on your model... see [http://download.autodesk.com/global/docs/maya2014/en\_us/index.html?url=files/Mapping\_UVs\_Automatic\_UV\_mapping.htm,topicNumber=d30e220675](http://download.autodesk.com/global/docs/maya2014/en_us/index.html?url=files/Mapping_UVs_Automatic_UV_mapping.htm,topicNumber=d30e220675" \t "_blank)

2) go to the Rendering tab (in the Shelf, which is the row of all those icons) and then \*DOUBLE-CLICK\* on the last icon, the 3D Paint Tool.

3) before you can start painting, you must assign a texture... in tool settings sidebar, go to "File Textures" section, make sure Attribute is set to "Color", then click "Assign/Edit Textures"... you'll want at least 512 x 512, save as .PNG

4) now you can start painting, play with brush settings at the top of the tool settings sidebar, play with colors

5) to save your texture, go to "File Textures" section of tool settings sidebar and click "Save Texture"... look in the lower-right of Maya interface, and it'll print the path of the file... this is where you need to go to get the .PNG!

6) copy and paste the .PNG from there somewhere into your \UnityProject\Assets\ folder, import everything as usual

7) edit the Unity material, and point the texture slot to the .PNG... everything should work now

- for full reference, Maya's documentation is pretty good: [http://download.autodesk.com/global/docs/maya2014/en\_us/files/3D\_Paint\_Tool.htm](http://download.autodesk.com/global/docs/maya2014/en_us/files/3D_Paint_Tool.htm" \t "_blank)

***Class 4 PHYSICS INTRO UNITY***

Maya review

POLYGONS MODULE

All objects in Maya scene

Mesh>combine

to paint as one object

Project UV MAP

--Create UVS >Automatic mapping

RENDERING MODULE

double-click 3D module

Assign/edit texture

Paint using flood tool and brush

Save texture

Save scene as a .ma no .mb (maya ascii)

***Creating scripts to move gameObjects***

Unity

create > C#

name it

Double click to open monoDevelop

void Update () {  
        GetComponent<Transform> ().position += new Vector3 (0f, 1f, 0f);  
    }

Getcomponent = access component

<Tranform> = component (module “verb”)

+= adds to existing value

new Vector 3 (new always comes with Vector 3)

(01,1f,0f) 🡪 f = “float”

\*\*shortcut for GetComponent<Transform>().

transform.position:::

void Update () {  
        transform.position += new Vector3 (0f, 1f, 0f);  
    }

- - -- - - - -

***UNITY’S PHYSICS SYSTEM***

add component to gameObject

physics > ***rigid body***

Add directional light

in inspector

Shadow type > hard shadow

- - - - - -

***Model collider meshes in MAYA to run physics simulations on odd shaped objects***

Add component>physics>mesh collider

(low polygon mesh made in Maya)

In inspector

match mesh (project list of assets)

\*Two mesh colliders do not interact (one attached to object, and one to floor in gravity example)

Check/enable “convex” on mesh collider inspector option of falling object to make them interact

\*Most preferable to not use a mesh collider at all, just use box collider

- - - - -

RIGID BODY settings

drag- bowling ball has none, feather has a lot

constraints

-freeze position

-freeze rotation

\*\*\*You can only set pivots in maya\*\*\*\*

\_ \_ \_ \_

MAKE IT BOUNCE – (after applying rigid body)

Assets>import package >import physics material

inspector menu

Box collider>material >physic material

\_ \_ \_

(must also have rigid body)

Component> physics > constant force

(makes something shoot in a direction)

Homework

make rubegoldberg

Rendering successful”

Make sure to point camera to see all of game, and change clipping planes far to 10000

edit>settings>quality

shadow distance up

build out often because physics is variable, what you see in unity is not necessarily what you get

3dsky.org convert from 3D max to .ma’s

***CLASS 5***

Unity shaders

make a material invisible turn the mesh renderer off

assets > create > material

Chapter 10 of 10print concepts:

single line of code repeats to make a pattern

code goes by 10s to allow for adding new code without offsetting all following code (references to specific lines are not fucked)

Chapter 15

manipulating code makes different patterns, visual

same idea for sound synthesis

Chapter 25

REM PORTS to other Platform

porting from one platform to another – act of translation

multiple languages so each affords us different opportunities and means of creative with varying degrees of difficulty

Perverse programming language

languages: Malboge (looks like sdfsG%”$&Hfgn), Path (looks like ; [[[[ /// ,,,)

Shakespeare programming language (print = Speak your mind!)

review:

vector = group of numbers

(0,1,3) – example of a 3d vector

(5,6) – a 2d vector

IN code: vector3 (x,y,z)

Vectors -can represent a position

-but also direction, line segments, rays

ex: (2,2) is description for a point on a graph, and a line from origin to that point

void Update () {

GetComponent<Transform>().position += new Vector3 (0f,1f,0f);

//shorthand for same thing as above:

transform.Translate (new Vector3 (0f,1f,0f));

}

}

***Code can be utilized with physical forces in unity***

Make object a rigid body to prepare for physics applications

Create C# script

make void Update “void FixedUpdate”

void FixedUpdate () {  
        GetComponent<Rigidbody> ().AddForce (new Vector3 (0f,10f,0f));  
  
        *//f means it's a float*  
        *//. let's us access any members of the thing before it, ie access rigidbody*  
        *//shortcut :: rigidbody.Addforce (new Vector3 (0f,10f,0f));*  
    }  
}

This is awesome because this action will allow collision detection with other objects

**How do you make an object move on it’s local axis, not just continually up based on the global axis???????**

GetComponent<Rigidbody> ().AddForce (GetComponent<Transform> ().up);

*//shortcut :: rigidbody.Addforce (transform.up);*

If object falls at this point, then the gravity is overpowering your force, so you need to up the ante:

**SCALAR**

* for component-wise scaling

(0,1,0) x 10f

(0,10,0)

rigidbody.Addforce (transform.up \*10f);

play with these:

rigidbody.Addforce (transform.up \*10f, ForceMode.Acceleration);

rigidbody.Addforce (transform.up \*10f, ForceMode.Impulse);

rigidbody.Addforce (transform.up \*10f, ForceMode.VelocityChange);

**INPUTS**

if, then statements

-add keycode

    void FixedUpdate () {  
  
        if (Input.GetKey (KeyCode.Space)) {  
                        rigidbody.AddForce (transform.up \* 20f, ForceMode.VelocityChange);  
                }

open up constraints of rigidbody

and freeze rotation, this will **keep your object upright**

-this will make your key codes always accurate to the motion of the object

HOMEWORK for week 6:

Read Gamefeel chapter 1 (yang will email pdf)

Play game – pale machine

bo.en.info/URLpalemachine.html

trail renderer- tongue

applying force and twerk, rigid body attached to a string (car keys)

-make some sort of interactive toy like pale machine

-have at least two scenes::

-try to put in music??!??

**SCENES**

change to the next scene after 60 seconds

-make a new script called “scene change”

    void Start () {  
      
    }  
      
    *// Update is called once per frame*  
    void Update () {  
    *//change the scene after 60 seconds*  
        if (Time.time > 60f) {  
            Application.LoadLevel ("dancingBaby");          
            Application.LoadLevel (1); *//load the scene #1 in "File >>BuildSettings"*  
        }  
        *//dancingBaby is name of next scene file*  
            *//Time.time a float keeping track of time,*  
  
    if (Input.GetKey (KeyCode.R)) {  
            Application.LoadLevel (1); *//load the scene #1 in "File >>BuildSettings"*  
        }  
  
    }  
}

Back in unity, apply script to a null object (you can make this object invisible by unchecking mesh renderer)