CS329e – game dev

Objected-oriented props in games

* Objects (obj) = a black box that you interact with through methods and its state are called properties
* Objects: webpage, tables, forms, buttons, links, img
* Events = triggers javascript to run
* Declare javascript functions in the head of html

html input + holding variable for java script:

<form>

<input name = “input\_name” type = text>

<input name = “input\_name\_2” type = button value = “Show me” onClick = “myFunction()”>

</form>

**WAMP WILL SOMETIMES NOT CLEAR ITS CACHE AND RELY ON IT FOR CODE-Y BITS, CNTRL + F5 TO CLEAR**

**Java Script syntax:**

For loop construction

For (var i = [start], i < 10, i++){

Blah blah blah

};

For([calling arbitrary var],[condition to continue],[increment])

Iterating through array

arr=[1,2,3,4]

for(var i = 1, i<arr.length,i++){

each run of for loop works on individual element of array

arr.length isn’t a method so arr.length() is wrong

};

Regex

Match vs test

Test = returns a Boolean to tell if the regex is detected in test string

Match = returns the regex string, and index of first character

Able to add **modifiers** at the end of the regex expression

[Rubular.com](http://rubular.com/) = test simple regex on test strings + gives a little reference guide to all the regex

[Regex101.com](https://regex101.com/) = more complex regex testing + gives list of modifiers

This: effectively the python self tag when you are creating objects but more freely used

Still most used in constructor functions(create objects)

New: keyword that initializes the new object(var name = new human([contructor func]))

**Escape character: \ or just use different types of quotes [“ ‘ “ or ‘ “ ‘]**

Turnary operators:

Var num = [Boolean statement] ? [value if bool is true]:[value if bool is false]

Prototype func:

Its for adding methods/properties to overall obj without altering the original constructor func

3 functions go into establishing any type of game state:

Preload: mostly to just load up any art assets that will be used

Create: setting the initial values for variables in game state + which funcs used

Update: updates the info in frame of the game(about 60 times per second [60fps])

**Event Listener(for keyboard input):** (captlization matters) game.input.keyboard.addKey(….).onDown.add([func],[listenerContext],[priority],[arguments for the function]);

…. is the keycode for the key you are using

**Keycodes can be determined via**: Phaser.Keyboard.[key on keyboard]

Numbers need to be typed out and capitalized 1 = ONE

Even listeners are local to .js file they are located in

Functions are global is they are labeled like

function [funcname]([args]){

…..

};

**History of games**

1976 - 1978 = birth of 8 bit consoles:

ROM cartridges + microprocessors were born

Atari bought by the warner corporation 🡪 Atari 2600 produced($200)

Atari + Apple released PC’s

Nintendo releases Othello(1st Nintendo game)

1979 – 1981 = The Golden Age

Asteriods(Atari) Pac-man(Namco/Midway) Frogger(Konomi/Sega) Donky Kong(Nintendo)

Tanks(Atari/US army)

1982 - 1983 = The videogame crash

Focus on hype + influx of bad games (the standout game of this = E.T the extra terristral)

Riase of PC computer(commodore 24)

1985 – 1988 = Birth of home console market

1985: Nintendo releases NES with Zelda(8 – bit console)

1989 = Gameboy relased + tetris included with

16 – bit Golden Age

1989 = Sega released the Gensis(16-bit console) + Sonic

1991 = Nintendo releases Super NES(16-bit)

32-64 Bit Gaming: Modern Age