Deceptively Simple.

Insanely Fun.

My goal with this doc is to get the kids thinking about what is fun in a game, and what the games they love have in common especially when it comes to mobile games.

MOBILE GAME DESIGN sounds like a pretty easy and sometimes scary word, but what does it mean to us as developers? Mobile games are a lot like old school game boy games. You have a small screen, and not exactly a super computer. Mobile games need to be

* Able to run on a small device
* Use the screen space properly
* Addictive
* Simple yet fun

Lots of games use the idea of never reaching a game over. The player wants to keep on keep on keep on playing. They do not have a set end to the game. We also want to include things like social interaction, and the ability to multi task and compete.

Below is a list of VERY popular games for mobile and computer. Can you tell me what they all have in common?

My goal for the next part is the kids to discuss multiplayer and easy to jump into games. The games also use their screen space appropriately. The games were all games they told me they liked. I suggest asking the students their favorite mobile games and editing this to have those in the list. Any connection is a good connection and good design doesn’t just apply to mobile.

* Call of duty
* World of Warcraft
* Clash of Clans
* Hearthstone
* Minecraft pocket edition
* Geometry dash

Write the list below:



We are going to now watch an awesome video about why games are addictive. Write blow some game design philosophies of the developers. I had them watch a game theory on what makes games addictive. The link is <https://www.youtube.com/watch?v=_BTGgCEFuQw>. It has some...moments.. That you could fast forward through ,but it has a lot of science behind it and he uses REALLY good examples for the students to discuss.

Look at the picture on the front page of the hearthstone app on the phone. What does the app do well based on what we discussed?

I love hearthstone……I wanted to make sure they noticed that the game had some but not tooo many animations and used screen space well. I also discussed that it had multiplayer and collectables.

Code TIME!!!

I used this for both the intro course and the UNITY course. Both classes need to discuss game design philosophy, but unity they need to talk about c#. You can delete this for the younger TOTS or you can use it to teach some basic coding.I have worked on several unity games so I know for a fact the code below is really nice to have.

Unity Game Engine is an open source AWESOME engine. You can make epic games for mobile apps and they are easily portable. We will be using c# to create some epic games. We will be reading and interpreting code to help you become an epic programmer. Below is some code we are going to talk about in a second. First let's talk about the basics.

Declaring Variables

int i, j, k;  
char c, ch;  
float f, salary;  
double d;

Initializing variables!

int d = 3, f = 5; /\* initializing d and f. \*/  
byte z = 22; /\* initializes z. \*/  
double pi = 3.14159; /\* declares an approximation of pi. \*/  
char x = 'x'; /\* the variable x has the value 'x'. \*/

Types of variables

using UnityEngine;  
using System.Collections;  
  
public class VariablesAndFunctions : MonoBehaviour  
{   
 int myInt = 5;  
   
   
 void Start ()  
 {  
 myInt = MultiplyByTwo(myInt);  
 Debug.Log (myInt);  
 }  
   
   
 int MultiplyByTwo (int number)  
 {  
 int ret;  
 ret = number \* 2;  
 return ret;  
 }

//end of code

Unity game engine

using UnityEngine;  
using System.Collections;  
  
public class IfStatements : MonoBehaviour  
{  
 float coffeeTemperature = 85.0f;  
 float hotLimitTemperature = 70.0f;  
 float coldLimitTemperature = 40.0f;  
   
  
 void Update ()  
 {  
 if([Input.GetKeyDown](http://docs.unity3d.com/Documentation/ScriptReference/Input.GetKeyDown.html)([KeyCode.Space](http://docs.unity3d.com/Documentation/ScriptReference/KeyCode.Space.html)))  
 TemperatureTest();  
   
 coffeeTemperature -= [Time.deltaTime](http://docs.unity3d.com/Documentation/ScriptReference/Time-deltaTime.html) \* 5f;  
 }  
   
   
 void TemperatureTest ()  
 {  
 // If the coffee's temperature is greater than the hottest drinking temperature...  
 if(coffeeTemperature > hotLimitTemperature)  
 {  
 // ... do this.  
 print("Coffee is too hot.");  
 }  
 // If it isn't, but the coffee temperature is less than the coldest drinking temperature...  
 else if(coffeeTemperature < coldLimitTemperature)  
 {  
 // ... do this.  
 print("Coffee is too cold.");  
 }  
 // If it is neither of those then...  
 else  
 {  
 // ... do this.  
 print("Coffee is just right.");  
 }  
 }  
}

// start of onCollisionEnter

using UnityEngine;  
using System.Collections;

public class ExampleClass : [MonoBehaviour](https://docs.unity3d.com/ScriptReference/MonoBehaviour.html) {  
 [AudioSource](https://docs.unity3d.com/ScriptReference/AudioSource.html) audio;  
   
 void Start() {  
 audio = GetComponent<AudioSource>();  
 }  
   
 void On[Collision](https://docs.unity3d.com/ScriptReference/Collision.html)Enter([Collision](https://docs.unity3d.com/ScriptReference/Collision.html) collision) {  
 foreach ([ContactPoint](https://docs.unity3d.com/ScriptReference/ContactPoint.html) contact in collision.contacts) {  
  [Debug.DrawRay](https://docs.unity3d.com/ScriptReference/Debug.DrawRay.html)(contact.point, contact.normal, [Color.white](https://docs.unity3d.com/ScriptReference/Color-white.html));  
 }  
 if (collision.relativeVelocity.magnitude > 2)  
 audio.Play();  
   
 }  
}

using UnityEngine;  
using System.Collections;

public class MyGameClass : [MonoBehaviour](https://docs.unity3d.com/ScriptReference/MonoBehaviour.html) {

void MyGameMethod() {  
 // [Message](https://docs.unity3d.com/ScriptReference/VersionControl.Message.html) with a link to an object.  
  [Debug.Log](https://docs.unity3d.com/ScriptReference/Debug.Log.html) ("Hello", gameObject);

// [Message](https://docs.unity3d.com/ScriptReference/VersionControl.Message.html) using rich text.  
  [Debug.Log](https://docs.unity3d.com/ScriptReference/Debug.Log.html)("<color=red>Fatal error:</color> [AssetBundle](https://docs.unity3d.com/ScriptReference/AssetBundle.html) not found");  
 }  
}

NOTES! Add any notes you want below

It is really good to provide some code space for kids. They can plan out their apps or just scribble down some ideas. Getting them to take a step back is an awesome way to help them troubleshoot and problem solve