* Describe three benefits to using a WebGL library over straight WebGL. **Cross-browser and cross-platform compatibility**. Tight integration with HTML content, including layered compositing, interaction with other HTML elements, and use of the standard HTML event handling mechanisms. Hardware-accelerated 3D graphics for the browser environment.	* Why would we pick 2D vs 3D vs High Definition RP (or Universal Render Pipeline) for our Unity projects? 2D: ● Instantiates wrt a 2D scene (sprites, ortho camera, etc) 3D: ● Instantiates wrt a 3D scene (models, rendering pipeline, etc) 3D with Extras: ● 3D + a post-processing stack High Definition RP: ● Supports shader model 5.0 (DX11 and up) Universal Project Template: ● Universal pipeline where support/optimization a concern
* What is a spritesheet? Why would we use one? Spritesheets are used **to speed up the process of displaying images to the screen**; It is much faster to fetch one image and display only a part of that image than it is to fetch many images and display them.	* In 2D we have sprites instead of models. Describe, in your own words, what this *means* for a Unity project. A sprite is a two-dimensional (2D) bitmap graphic object that can be a static image or animation that is integrated into a larger scene. Sprites are used in games to collectively create a scene. Each sprite is used to represent each object. A "Sprite Sheet" is simply a collection of still images that progress. Once displayed sequentially it creates animation.
* What is a tilemap? Each part of the tilemap is split into separate tiles, for example you might have four different tiles that represent different types of tree, and these tiles are a set size, usually 8x8 or 16x16 pixels. The whole collection of these tiles is called a spritesheet, and the spritesheet will be a set size like 64x64 pixels (and therefore contain 64 tiles).	* What is the purpose of a Sorting Layer? On BG sprite, add a sorting layer! ● Consider it similar to a z-index in webpages ○ Add background, entities, foreground ○ Set Player to entities layer - Sorting Layers are specifically used **to manage the rendering order of 2D objects**, such as sprites, allowing you to display objects in front of other objects, regardless of their physical position in the scene.
* What is the difference between Object Mode and Edit mode? Edit mode is made for modify the geometry of your mesh, like adding vertex, faces, and so on. In object mode you use your mesh as an unique entity, and you can edit it's properties (ie: position, rotation and so on).	* What are the differences between a Rigidbody and a Rigidbody2D? the differences are that **in 2D, objects can only move in the XY plane and can only rotate on an axis perpendicular to that plane**.
* Assume you wanted to wrap a texture around an object. What would you do in Blender to make this happen (at a high level). Go into edit mode, and select all the vertices of sphere. Press U → Sphere Projection. If you've got the UV/Image editor open you should see a very different UV map. UV mapping is a technique used to "wrap" a 2D image texture onto a 3D mesh. Texture	* Let's say we want to spawn a bunch of objects to the screen (say, bullets or enemies or some sort of transient object). What are the high-level steps for doing so? Make a prefab – copy a bunch of times into the scene
* What is the purpose of the 'Mix Shader' in the node editor? This shader is used **to mix two other shader nodes together**. It has inputs for two shaders and a setting to govern the amount each shader contributes to the final result.	* What is the purpose of the .normalized attribute to a Vector3? When normalized, a vector keeps the same direction but its length is 1.0.
* What can you do with Geometry Nodes (multiple demos shown in class)? The geometry node is used to specify how light reflects off the surface. This node is used to change a material's Normal response to lighting conditions.	* What is a ScriptableObject and why would you use it? ScriptableObject is **a serializable Unity class that allows you to store large quantities of shared data independent from script instances**. Using ScriptableObjects makes it easier to manage changes and debugging.
* What is the purpose of a Prefab? **allows fully configured GameObjects to be saved in the Project for reuse**.	* Describe two benefits from using a node-based editor (similar to what we saw in Blender)?
* How do you add physics properties to a GameObject? How do you prevent it from falling through space forever? Add component -> rigid body. Set a ground collider and deselect "is trigger"	* What do we need to do in Unity to get access to the Shader Graph editor? From here you can create either a PBR or Unlit Shader Graph Asset. This will create a Shader Graph Asset in the project. You can double click on the Shader Graph Asset or, with the Shader Graph Asset selected, **select the Open Shader Editor button in the Inspector** to bring up the Shader Graph Window.

* What is one difference between Terrain and a Plane? You need Terrain as it allows you to add vegetation (Trees & Details), make adjustments to the relief like rise land to form mountains and also it gives you ability to paint textures on it. **A plane is a simple game object that can only be scaled on two axes and none of the above can be done with it**	* Assume you created a shader graph and hooked up everything correctly. What do you need to do in the main Unity editor to get your shader graph to be applied to a GameObject? To use Shader Graph you must first create a Shader Graph Asset. In Unity a Shader Graph Asset appears as a normal shader. To create a Shader Graph Asset you **click the create menu in the Project Window and select Shader from the dropdown**. From here you can create either a PBR or Unlit Shader Graph Asset.
* How do you associate a script with Unity? For instance, assume you have a script for moving a player around. What do you have to do to get that C# code to *do* something? 1) Create a script 2) Attach to one (or more) game object(s) – drag script onto player attribute slot 3) Set script attribute properties.	* What is a collider? Why would you use it? Colliders a physics for particles – Use it to make objects not pass through
* Describe the differences between Update, LateUpdate, FixedUpdate FixedUpdate() ● Physics-related updates LateUpdate() ● Called at end of frame ● Unity will: ○ Find all gameobject updates ○ Then call LateUpdates Update() ● Called once per frame ● Animations, AI, other constant updates - ○ Runs as fast as possible (or however you constrain your program)2	* What is the purpose of a tag? Describe an example use case. * Why would you make a Rigidbody Kinematic? Tags **help you identify GameObjects for scripting purposes**. They ensure you don't need to manually add GameObjects to a script's exposed properties using drag and drop, thereby saving time when you are using the same script code in multiple GameObjects. Kinematics **whether physics affects the rigidbody**. If isKinematic is enabled, Forces, collisions or joints will not affect the rigidbody anymore. The rigidbody will be under full control of animation or script control by changing transform. position.
* Describe the differences between Awake and Start Awake() ● Called when GameObject is instantiated/active ○ Still called if component is not 'enabled!' ● Good for initializing variables Start() ● Instantiated and enabled - Runs upon Scene start	* Let's say we add a reference to a GameObject within a script (say, public GameObject player). How do you ensure that your script can interact with that GameObject? Define behaviors ● Attach to Unity objects ● Enable interactivity ● etc. Our steps: 1) Create a script 2) Attach to one (or more) game object(s) 3) Set script attribute properties - Ensure that Preferences → External Tools is set to open a code editor of your choice ● Use an IDE of your choice
* What happens with the following two variables in the Unity Editor (not the code editor): public int speed; [HideInInspector] public int speed2; Speed2 won't show up in inspector and won't be serialized. Non serialized will reset a variable to default on game state change.	* What is the purpose of resetting various GameObjects to origin (i.e., Transform --> Three Dots menu --> Reset)? I'm creating a game where I want a car to pass by over and over again.
* Assume your class name is PlayerController. How would you add another class (say, PlayerControllerSupport) to that file and get Unity to recognize it?	
* What is the difference between Input.GetKey and Input.GetKeyUp? Describe an example. Input.GetKey will **repeatedly** return true while the user holds down the specified key ie firing weapon Input.GetKeyDown will true only **once** when the specified key is pressed. Ie toggle flashlight	