Timesheet

|  |  |  |
| --- | --- | --- |
| Date: | Timeframe: | Accomplished |
| 12/12 | 10:30 PM – 1:05 AM | Worked on initial project presentation, as I need something to submit Monday, but I do hope that I make more features before Friday. |
| 12/13 | 11:45 AM – 12:45 PM | Worked on presentation |
| 12/14 | 9:05 – 9:50 AM  2:50 – 3:15 PM  8:10 – 9:00 PM | Removed EnemyManager, shuffled some code around, made enemies freeze when a transition is entered, though I could probably do it better by using enabling/disabling in conjunction to GameManager states. Got more images for presentation, but really wanted to start on UI and definitely the combat.  Made pausing functionality, but for some reason pressing the pause button makes an area of the level super slippery when done during a transition. Also, is affected by the issue (holding downright arrows makes it not register). Actually. It appears a certain x range and a certain y range are slippery (when moving on them, maintain speed until outside range), and I think it correlates to the camera ranges. It appears to be linked to the camera movement, and the black screen bug also showed up. Slipperiness was actually caused by the camera being a child of the player. It actually turns out that children follow parents wherever they go, so if the camera was a child of the player, it’d follow wherever the player went.  <https://learn.unity.com/tutorial/ui-navigation#5e5d380bedbc2a17cb9d00e5>. <https://learn.unity.com/tutorial/ui-navigation-2019-3>. Unity tries automatically making UI flow when interactable elements are added, though you can specify direction the selector goes in Navigation inside the Inspector, though this is from the pre-2019.3 documentation. Except the 2019.3 on thing just listed how to make sub-sets of UI (like an Options drop down), but it didn’t even do that; it seems to have just added a conclusion that’s unnecessary. <https://docs.unity3d.com/ScriptReference/Canvas.html>. Appears that you can have multiple canvases inside each other. It also appears that you can basically create entire projects inside one script (GameObject s = new GameObject(); s.name = “whatever”; s.AddComponent<Thing>();) <https://www.youtube.com/watch?v=wlY5sRewfVQ>. |
| 12/15 | 9:30 – 9:50 AM  10:30 – 11:30 AM  12:20 – 3:00 PM  3:20 – 4:00 PM  7:30 – 9:20 PM  10:00 – 11:00 PM | Couldn’t find anything about switching objects between canvas and scene, though did find stuff for putting 3D objects in the Canvas. <https://forum.unity.com/threads/3d-objects-inside-canvas.266145/>. <https://answers.unity.com/questions/789179/3d-gameobject-anchored-to-the-new-ui-canvas.html#answer-835244>. It appears that you can have multiple UIs, and you can have a UI set to a certain position (WorldSpace render mode).  <https://answers.unity.com/questions/1226961/instantiate-prefab-from-script-inside-a-canvas.html>. <https://github.com/JetBrains/resharper-unity/wiki/Avoid-using-Object.Instantiate-without-%E2%80%9CTransform-Parent%E2%80%9D-parameter-and-using-SetParent-later>. It appears that if you want to set the parent of an object, you can do so using object.transform.SetParent (GameObject.FindGameObjectWithTag("bruh").transform); or you can do it on Instantiation with (GameObject). Instantiate(enemyPrefab, Vector3.zero, Quaternion.identity, GameObject.FindGameObjectWithTag("Canvas").transform); You can’t start Coroutines inside a singleton, as it requires a MonoBehavior, which are found in objects. <https://answers.unity.com/questions/14081/nested-coroutines.html>.  Explored the input issue some more. Holding the shift or ctrl on the side of the used directions doesn’t cause anything bad to happen, but holding the ones on the opposite side do cause one diagonal direction to be blocked. Holding Enter really messes with WASD, and Q does this as well, but only to the WASD keys, and a different diagonal than the shift or Ctrl. Space only affects the Arrows. This computer uses HP Z2 computer with a very wide display and a keyboard, both HP. <https://bugs.mojang.com/browse/MC-123862>. Might possibly be a hardware issue. Messed around with old html games that I made in Video Game design after trying to figure out if the inputs dropping was Unity’s fault or the school hardware’s. Might be a Unity problem, as I checked another computer and that one couldn’t handle downright and a Shift, thought it might’ve had something to do with me using an array for inputs, but actually no. Should look at these later: <https://learn.unity.com/tutorial/textmesh-pro-basics>. <https://learn.unity.com/tutorial/working-with-ui-in-unity>. <https://learn.unity.com/tutorial/ugui-layout-performance-considerations>. <https://learn.unity.com/tutorial/ui-components>. <https://docs.unity3d.com/Manual/class-Camera.html>. <https://answers.unity.com/questions/1387685/how-to-make-a-square-camera-in-2d-game.html>. <https://docs.unity3d.com/Packages/com.unity.2d.pixel-perfect@3.0/manual/index.html>. Pixel Perfect Camera Component is in a separate package from base Unity. <https://blogs.unity3d.com/2019/03/13/2d-pixel-perfect-how-to-set-up-your-unity-project-for-retro-8-bits-games/>. <https://www.youtube.com/watch?v=2qeNu2QApAM>. <https://www.youtube.com/watch?v=5WYvYZnGKKc>. Week 5 has that forum about pixel-perfect cameras. <https://forum.unity.com/threads/pixel-perfect-2d-in-4-3.210497/page-2?_ga=2.227455125.961423169.1608051304-1031676528.1600094299#post-2202561>. That link on the first page from the Week 5 forum. <http://blog.evilwindowdog.com/post/147292156801/a-pixel-perfect-camera-for-unity>. <https://www.youtube.com/watch?v=ryISV_nH8qw>. <https://www.youtube.com/watch?v=CU4YjSZNTnY>.  Actually, I think I should make GameManager a pseudo singleton, as I’ve already run into multiple instances where I want to wait for coroutines, mainly the FadeIn part, and I don’t like having to call an intermediate method everytime for just that one function. This is a bit of an overreaction, as I can simply just have the coroutines be public; additionally, I also don’t know how much would have to be changed to make the GameManager a pseudo-singleton, and I’d rather focus on other elements for now instead of reworking things that’d get me the same result as I have now. <https://answers.unity.com/questions/1227087/how-to-make-an-abstract-pseudo-singleton-class.html>. <http://blog.13pixels.de/2019/unity-and-the-mysterious-singleton/>.  Singletons become worse the more you rely on them, and you can try avoiding them with ScriptableObjects (similar to mono behaviors, but are persistent through scenes and are saved as assets) or the like. <https://www.youtube.com/watch?v=PVOVIxNxxeQ>. <https://www.youtube.com/watch?v=raQ3iHhE_Kk>. Modularity (Systems aren’t directly dependent/hard referenced, Scenes are clean slates aka minimal transient data and objects, every prefab has its functionality in itself, have functionalities relegated to components), Editable (Focus on data/treat it like a machine with inputs and outputs, change the game without having to go into the code every time to find that right value, emergent design/mechanics caused by interactions between many small parts (in reference to modularity)), Debuggable (test components individually, have good debug features and views, don’t fix bugs you can’t understand). ScriptableObjects are good for configurations, Inventories, Enemies that have basic behaviors but different appearances/stats, Audio Collections, and the like. Singletons do have benefits, such as global access, persistent data, and very easy to understand, but it does go downhill once you make bigger projects. Rigid Connections (falls apart when things are out of place), no Polymorphism, hard or impossible to test, one instance may be rough.  Images in UI have anchor points represented by white triangles. Anchor Presets are in the square of the rectangle transform of something like Text. In text, using best fit will resize text to fit the space available. The little O in an object is the pivot and can be moved by the Rectangle tool. Holding shift while using an Anchor Preset will also set the pivot. The Canvas Object can have a CanvasScaler component that automatically scales things based on the resolution. “Scale with screen size”, with a Reference Resolution, will automatically scale UI elements to best fit any resolutions based on the Reference.  Text uses textures that have pixels, and the normal shader creates ugly messes based on how far an empty square is from a used one. TextMesh Pro generate higher quality Text with only a slight decrease in performance, if any. TextMesh Pro can be put into either a 3D scene or your UI.  Layouts have to be replaced everytime they’re modified, which can be expensive like dynamic UI elements not baked into a Nested Canvas system. UIs can have scripts, and Layout Groups can be disabled when changing your layout is no longer necessary. It seems as though Layout Groups are components addable to a GameObject, probably a UI element.  Multiple UI elements can chare the same animation controller. Making custom functions for a button is just like making regular custom functions, down to being attached to a game object. However, they can only accept ints, bools, floats, strings, and Objects. EventSystem objects are automatically created when you make a new Canvas, and it’s useful for making non-pointer navigation. OnClick event happens when you click and release a button. You can add many function calls with a EventTrigger component. Sliders have a listener for OnValueChange, which requires a function to have a matching parameter in order to dynamically edit it. ScrollRects are things that can be moved around in the UI, usually used with masks. ScrollRects, in Elastic and Clamped mode, cannot leave its rectangle, and Elastic has a little bounce when it reaches the edge. |
| 12/16 | 6:00 – 7:20 AM  12:10 – 3:15 PM  10:00 – 11:00 PM | <https://docs.unity3d.com/2019.3/Documentation/ScriptReference/Transform.html>. Coded a little outside of VisualStudio.  <https://answers.unity.com/questions/12301/how-can-i-get-a-parent-gameobject-of-gameobject-us.html>. <https://answers.unity.com/questions/278326/missing-prefab.html>. Unity crashed and now my EnemyHome prefabs are missing.  How to avoid singletons and headaches: reduce need for global managers/global state tracking, Inversion of Control (Dependency Injection (give components what they need to work with), usually done with an Injector that passes around references to stuff). Modulating data: look at the Enemy Prefab, which can be filled out with a ScriptableObject, which can limit extensions and modularity by lumping a bunch of stats together. Scriptables are made with [CreateAssetMenu] access class Name : ScriptableObject{ access type Value //the name is essentially the file name }, then in MonoBehaviors you can have access ScriptableObjectYouMade name; but that might get cumbersome, so you can have a serializable ([Serializable] access class ThingReference{bool UseConstant; type Constant, ScriptableObjectYouMade Bruh; access type Value{get {return UseConstant ? Constant : Bruh.Value} } }).  Unity refuses to open. <https://answers.unity.com/questions/1413291/unity-wont-open-project-1.html>. However, I have tried removing the Temp folder, but it doesn’t work. <https://forum.unity.com/threads/unity-doesnt-open-from-unity-hub.524320/>. <https://support.unity.com/hc/en-us/articles/205698949-I-have-just-updated-installed-a-new-Operating-System-Why-is-my-license-failing-to-activate-?_ga=2.134313209.961423169.1608051304-1031676528.1600094299>. <https://fogbugz.unity3d.com/default.asp?1179270_k32tbfk2b14sml9n&_ga=2.38720427.961423169.1608051304-1031676528.1600094299>. <https://answers.unity.com/questions/1385573/unity-wont-open-any-project-when-i-select-one.html>. <https://answers.unity.com/questions/1305457/unity-editor-wont-open-project.html>. It appears that it can’t load the window layout, which is a cheap reason to not open the project at all. <https://answers.unity.com/questions/757942/how-do-i-fix-failed-to-load-window-layout.html>. <https://answers.unity.com/questions/1712501/stuck-in-failed-to-load-window-layout-loop.html>. <https://forum.unity.com/threads/can-not-launch-unity-due-to-failed-to-load-window-layout.694450/>. Deleting the Library folder’s contents fixed the issue. It appears that Enemy and Player can initiate the collision that causes a transition, but if an Enemy moves into Player after the coroutine happens, then the collision doesn’t happen again unless the player moves via input. Seems interesting: <https://docs.unity3d.com/Packages/com.unity.inputsystem@1.0/manual/ActionBindings.html>. Should make a pause menu to learn about UI navigation: <https://www.youtube.com/watch?v=JivuXdrIHK0&list=PLVMzbML7yE1UGMRb74OoQHSvvOWIIQtT0&index=8&t=0s>. <https://www.youtube.com/watch?v=pbeB9NsaoPs>. <https://www.sitepoint.com/adding-pause-main-menu-and-game-over-screens-in-unity/>. You can (and probably should) put menus inside Objects, though you can use stuff like panels to not have to go through a bunch of hoops. Public static is good if you just want to check some values and not reference a script. Time.timeScale determines how many seconds pass in game every second. UI can be moved with Animations, so should figure out how to bypass simply turning off the Menu Object, enabling and disabling the Interactable stat of the buttons, and have the Animation work. <https://www.youtube.com/watch?v=JeZkctmoBPw>. Animators are activated whenever their Controller is enabled. <https://docs.unity3d.com/2017.4/Documentation/Manual/class-Avatar.html>. <https://forum.unity.com/threads/how-to-disable-hide-ui-elements-without-disabling-them.282100/>. <https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/class-CanvasGroup.html>. <https://forum.unity.com/threads/multiple-animations-for-one-character.777260/>. <https://answers.unity.com/questions/1471995/multiple-animation-controllers.html>. <https://answers.unity.com/questions/1313767/using-one-animation-controller-for-multiple-object.html>. <https://docs.unity3d.com/Manual/animeditor-CreatingANewAnimationClip.html>.  You use the References in place of the ScriptableObjects if that's what you want. RenderTextures are a form of ScriptableObject (Camera just writes values into RenderTexture, UI just reads). PlayerHP is an example (Player changes the Scriptable's Value, UI and other components read the value); Scriptables create a clear divide between what can edit the HP and what reads the HP/Value. Scriptables also let you isolate parts and debug them there.  Event Architecture can modularize systems, let them be reused in other projects, isolate prefabs, optimizable to only check when needed (not constantly checking everything), easily debuggable. UnityEvents are closer to serialized function calls (set up ahead of time what you will do), and they can reduce code (not assuming it's bound to button, etc.), pass in arguments (can also be extended (such as extend UnityEvent<T>), also dynamic (waits for parameter to run code) or static (runs code whenever it's available)), but brings in Rigid bindings, some limited serialization (can take up to 4 arguments, but editor only acknowledges one of 4 basic primitives (i,f,b,s, also Object)), garbage allocation (uses strings, creates garbage, don't call every frame). You can make your own events with Scriptables ( [CreateAssetMenu] access class EventWhatevers : ScriptableObject { access List<GameEventListener> things = new List<GameEventListener>(); access void Raise(){for(i = things.Count-1; //go backwards because you can remove and add listeners){things[i].OnEventRaised();} } void Register/UnregisterListener(GameEventListener bro) //ading and removing listeners), then you make that GameEventListener (public class GameEventListener : MonoBehavior{ EventWhatever ScriptableObjectThing; UnityEvent Response; void OnEnable(){ScObTh.Register(this); } void OnDisable(){opposite of OnEnable()} void OnEventRaised(){ Response.Invoke() }) where a prefab can listen for a specific message and have a specific/reliable response. When you have an event, it will be called and do its event.Raise(), and the Listener will activate when its event is called.  Singleton Managers desire to track all scene objects needed and issue commands, but can run into race conditions (singleton isn't made yet/values not instantiated), can be butted by "Runtime Sets", which can track all objects in a scene, avoid racing, flexible than tags, strings, or singletons. The abstract is (access abstract class RunTimeSet<T>:Scriptable{ List<T> Items = new List<T>(); void Add/Remove(T t){ if(Items.Contains(t)){Items.A/R(t);}} ). Can have multiple lists to track variations like RegularList and FlyingList, can be renderers for special effects (get all of the renderers and put them in). Scriptables reduce the need to know everything about your project.  Enums have to be assigned in the code, difficult to reorder/remove, and can't hold additional data. You can use ScriptableObjects in a fashion like those.  ScriptableObjects can hold entire definitions (Assets), aka you can write functions in them, and they'll only call when you call them, thanks to being referable by prefabs and other Scriptables. Similar to AudioMixer/ AudioMixerGroup. An Inventory may look like a Scriptabel MasterList (Items you can use), Scriptables for said Items, and you can use different Inventories for different scenes. |
| 12/17 | 11:05 – 11:30 AM  12:05 – 3:20 PM  9:05 – 9:35 PM | Your System Scriptable can reference other systems (Like Inventory and Save). Other things I should check out: <https://www.youtube.com/watch?v=NU29QKag8a0>. <https://www.youtube.com/watch?v=6vmRwLYWNRo>. <https://www.youtube.com/watch?v=tGmnZdY5Y-E>. <https://github.com/roboryantron/Unite2017>. ScriptableObjects may or may not be performance intensive.  <https://docs.unity3d.com/Manual/class-Animator.html>. <https://learn.unity.com/tutorial/controlling-animation#5c7f8528edbc2a002053b4df>. <https://learn.unity.com/tutorial/controlling-animation#5c7f8528edbc2a002053b4e2>. Use layers in the Animator window to stack combinable animation clips. Animator.SetLayerWeight(float) changes layer weight. Avatar Masks let you isolate a part of the animation. Sync copies a Layer’s system to it to easily Additive between them. Animator bruh = GetComponent<Animator>(); bruh.SetFloat/Data(“variable”, data); Or, you can use int hashValue = Animator.StringToHash(“Variable”); to replace slow strings. bruh.SetTrigger(hash/”string”) doesn’t need a parameter. AnimatorStateInfo dude = bruh.GetCurrentAnimatorStateInfo(int layer); lets you check the current state/animation you’re in. StringToHash(“Layer Name.State”); gets you the hash of a state. Sub-state machines are good for organizing your animations, and they can also stack. Machines can be linked to each other via Up-Ones and Entry Exits, can transition between multiple on the same layer using barebones parameters (if you Exit one, you probably want to go to the other). Transition arrows are greyed out if they don’t know where a transition could be coming from, so duration and etc. should be done in the sending state. <https://docs.unity3d.com/Manual/AnimationParameters.html>. <https://docs.unity3d.com/Manual/class-Transition.html>. <https://answers.unity.com/questions/1357562/absolute-beginner-play-animation-on-trigger-trigge.html>. <https://www.youtube.com/watch?v=Ll3yujn9GVQ>. <https://www.youtube.com/watch?v=Ll3yujn9GVQ>. <https://www.youtube.com/watch?v=_wxitgdx-UI>. Might want to get that LeanTween thing from the Asset Store, or maybe not. <https://medium.com/game-development-stuff/how-to-create-ui-transitional-animations-with-unity3d-9888e6839181>. Getting NullReferenceException, even though the Debug is returning a non-null value; Object reference not set to an instance of an object. <https://support.unity.com/hc/en-us/articles/206369473-NullReferenceException>. Have to put the enemy on a layer greater than the canvas and background image (make it visible);  I forgot to assign the Texture to the variable in the Inspector.  TileMaps function like different layers. Worked on and finished Presentation. |
| 12/18 | 7:30 – 9:30 AM | Presented and cried.  Switched to a git repository |