Name	Kenny Kline	Team	Nautilus	TL	1	Date	Nov 9	Time	3:30p

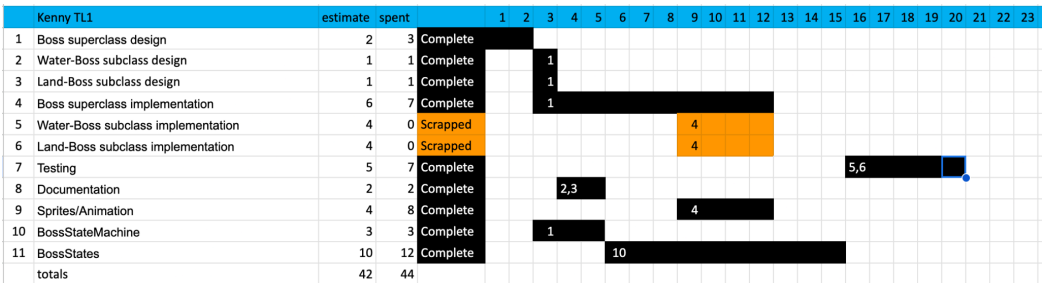
Fill in the underlined areas (and the boxes above), now but don't write on the remainder of this form.

Contribution: Briefly describe what your feature(s) is/are:

/10

My feature was the boss in the game.

Walk me through your Gantt chart. How long did this take? How long did you estimate it would take? What did you learn about your skill as an estimator?



	Kenny TL1	estimate	spent		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	Boss superclass design	2	3	Complete																							
2	Water-Boss subclass design	1	1	Complete			1																				
3	Land-Boss subclass design	1	1	Complete			1																				
4	Boss superclass implementation	6	7	Complete			1																				
5	Water-Boss subclass implementation	4	0	Scrapped									4														
6	Land-Boss subclass implementation	4	0	Scrapped									4														
7	Testing	5	7	Complete																5,6							
8	Documentation	2	2	Complete				2,3																			
9	Sprites/Animation	4	8	Complete									4														
10	BossStateMachine	3	3	Complete			1																				
11	BossStates	10	12	Complete						10																	
	totals	42	44																								

I estimated that this project would take 42 hours, and my final gantt chart has 44 hours of time on it. This would make it seem like I am pretty good at estimation, except that I ended up having to scrap a feature. I had initially planned on having different variants of the boss, which would be in different environments. However, other parts of the project ran longer than expected, so I did not have time to implement that.

Run your game and point out places where your code is called and run. (I will cycle through asking you this question and the next one until you either run out of interesting things to talk about or it is clear that you have made an above average contribution.)
Show the C++/C# code that was run. Walk me through the methods called from the time it enters your section of code.

Video Link: https://www.youtube.com/watch?v=6LpcKoTc3ME

Technical:
Walk me through your test plan. Give an example where a test case later found a bug in your code by things a teammate added later. (Or explain why you chose a test case specifically because you wanted to ensure that a teammate would know if they broke your code.)

/4

I added a test case which confirms that when the player shoots a projectile at the boss, the boss detects the bullet and takes damage, and plays the damage animation. If the player or the bullets were modified in a breaking way (ie, the tag on the collider of the bullets was changed) then this test would fail, and we would know to get the two sections of code back in sync.

/3

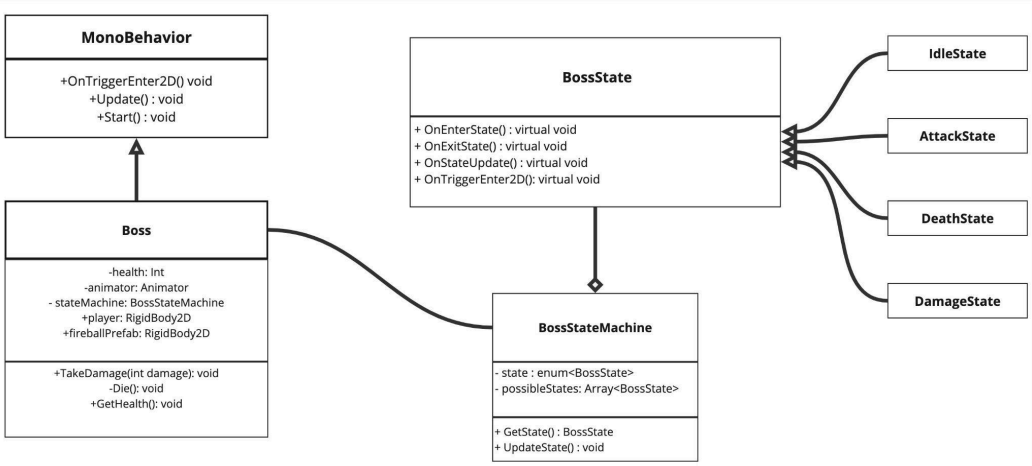
Pick a Prefab you have created that is documented well in a separate readme file. (I will point to several places in your code documentation and ask) What question where you trying to answer here? Who do you anticipate would be asking that question? What other questions might this person need the answers to? Prefab Name: Boss Documentation: https://github.com/Spovis/ShipwreckProtocol/blob/master/Assets/TeamMembers/Kenny/Prefabs/boss_unity_asset_store.html	/3
Show me a class in your code where there could be either static or dynamic binding. Write some mock code on this paper showing how you would set the static type and dynamic type of a variable. Super Class: BossState Sub Class: IdleState Virtual Function: BossStateMachine initialization Choose a dynamically bound method. What method gets called now? The definition of OnEnterState which is defined in IdleState is defined. Change the dynamic type. What method gets called now? If the dynamic type is BossState rather than IdleState, then the parent definition is called. Pick a statically bound method. Which one would be called in each of the two previous cases? The parent's definition of OnExitState is called in both cases.	/4
Show me an example of reuse in your code where you violate copyright law. How does it violate copyright? In my Fireball prefab, I use a sprite which is the fireball from Super Mario Bros. What did you have to do to integrate it with the code you wrote? What are the legal implications if you market your code with the re-used portion? Use fair use argue that you can use this anyway. To integrate it, I just downloaded the sprite from the internet and added it to the Fireball object in Unity. If I were to try to sell this game or the prefab, it could be seen as copyright violation since that sprite was created by another private company and is owned by them. If I were to be sued, I would argue that this is not copyright violation because 1) Amount - the fireball sprite is a tiny part of the Super Mario Bros game, and it is by no means a defining aspect of the game (as opposed to something like the sprite for Mario, which would be much more significant). 2) Effect on the potential market - Just because the fireball sprite is in my game, that will not make anyone feel like they have already seen Mario Bros or negatively impact the originality of either game.	/4
4. One big or two small, well-chosen patterns. Small Patterns = {Singleton, Private Class Data} Which patterns did you choose?	

1. State Machine

Why did you choose each pattern? (Justify your use of it).

I chose the state machine pattern because it is well suited to a situation where you want to transition a certain object between different states in a clean way. For the boss, it can be in several different states, and it will need to perform different actions as it switches between them (ie decrease health, trigger animations). This seemed like a use case that reflects the benefits of the state machine.

Draw the class diagram for your pattern(s).



Would something else have worked as well or better than this pattern? When would be a bad time to use this pattern?
I think a state machine pattern was the right choice for this situation. It worked well, and in my opinion, the code is pretty clear and easy to read.
A state machine would not be well suited to something like a data pipeline, where you are relying heavily on many reads/writes. Because each operation has an extra layer of abstraction around it, that would make both the readability difficult. It could also slow down the execution speed, since you have extra data objects and function calls.