**Oral Exam Prework**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Jillian Jabbes | Team | BrainStew | TL | 2 | Date | 04/13/2025 | Time |  |

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:  The feature I created in BrainStew’s “Techdown” 2D platformer game is the player system, which involved implementing the core playable character mechanics for movement, combat (stats), and physics interactions (player object) with the environment. The movement included the ability for users to send inputs for the character to idle, run, jump, double jump, and be able to utilize further scripts to add powerups or additional combat features like shooting. Lastly applied to the character is sprites, states, and animation tied to the core mechanics.  **------------------------------------------------------------------------------------------------------------------**  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?  One of BrainStew’s first major meetings was deciding on who is responsible for what features, and we all set on one major feature as well as a secondary one whenever help was needed there. So, I ended up organizing my Gantt chart around that plan where it covers mostly my primary feature and for overlaps I’d interact and work with that team member whenever feature overlaps happened to get things working then help out with the secondary feature if required. So in the middle here it states “Work with Weapon Specialist” as the weapon our player uses should be able to interact effectively so I often found myself talking with Bidhi and playtesting with her. In a way this made sense for me as a TL2 too. My secondary feature in case was helping Andrew with bosses, which at this moment we haven’t found much need to besides bouncing ideas.  Personally, I found that my feature up to this point was able to get me to approximately 53 hours when I estimated 57 hours total for everything. I assume that the final stretch of finishing up the game I will definitely reach that. I think my estimation skills aren’t far off, I probably didn’t count all the hours honestly especially when it came to research and figuring out Unity as well as creating graphics. I even taught myself how to use Aseprite, a popular pixel art and animation software.  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.)  (Likely things I’ll point out from the game’s start to finish)   * Main menu: Logo and menu designed by me * Options button: On this page I programmed the script responsible for BC-MODE and the toggle * The script responsible for changing scenes whenever buttons are pressed (not complex, but reused in a lot of places) * Pressing Play…immediately we see the player object the focus * Inputting common movement keys WASD will move the player around, calls my movement scripts, input controllers, ground check, and jump capability – show in the inspector the customizability of many of these values and how they impact movement. * Whenever the player is hit by an enemy, talk about Player Stats script, the player healthbar updating, PTakeDamage script, and how the enemy is sending damage * The lever (not complex, but something quick I made) – was a plan for the level designer to expand on more but this was an MVP basis   Show the C++/C# code that was run. Walk me through the methods called from the time it enters your section of code.  Code showcase found under Assets > src > TL2 > JJ-Scripts and opened on code editor… | /10 |
| **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.)  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: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  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: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Sub Class: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Virtual Function: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Choose a dynamically bound method. What method gets called now?  Change the dynamic type. What method gets called now?  Pick a statically bound method. Which one would be called in each of the two previous cases?  Show me an example of reuse in your code where you violate copyright law.  How does it violate copyright?\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  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.  4. One big or two small, well-chosen patterns.  Small Patterns = {Singleton, Private Class Data}  Which patterns did you choose?  1.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  2.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Why did you choose each pattern? (Justify your use of it).  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? | /4  /3  /3  /4  /4 |