Part Two Code Walk Guide

Student Guide

Walk us through your design (2 minutes)

Start with the demo. Ask them to demonstrate the functions of their program. It may take some time to load 10 players into the system, so consider asking the students to preload 9 player up-front before the presentations begin.

Walk us through your code (3 minutes)

Switch to the code. If they have any new diagrams they wish to share or want to bring up their part 1 diagrams, they can at this point as well. Encourage them to focus on how the Model, View, and Controller components are set up, and how the Controller reacts to actions taken by the user.

Answer some design questions from the instructor (5 minutes)

Ask the students the questions from the question section, allowing them to go through their code and rerun their program to answer them.

Discuss the feedback you get from us (3 minutes)

Leave a few minutes at the end for the student to assess their project part 2, as well as their final thoughts on the project overall. If they need some prompting, ask them about any limitations they noticed, anything they'd do differently if they were to start the project over from scratch, or any further development they would do on the project if given more time.

Rubric

Description of Code (25 points)

The clarity of their explanation of the technical details of the code and their understanding of it. They need to demonstrate that they know what their code does and how it works, uncertainty or hesitation about what is happening or failing to talk about large chunks of functionality is a penalty here.

Quality of Explanation (25 points)

How much they can talk about and explain their design decisions. If they only describe the code as it is, that's not enough, they should talk about how the code fulfills the requirements and why they decided to use one thing over another. You don't have to agree with their design decisions, so long as they make some sense and there was some intent.

Testing (25 points)

Instead of unit tests, this concerns how clear and intuitive their GUI is to use, and how well they explain the GUI during their demo. They can lose some marks if the GUI is hard to understand even if they do a good job explaining it during the demo, and can also lose marks with a good GUI they don't explain or show off much.

Code Walk Reflection (25 points)

This mostly comes from the question-and-answer section and the last few minutes where they discuss their thoughts about the design overall, whether they recognize its limitations, and what they would do differently if they were to start over. The more honest and critical they can be about the state of their current code (which might even be "my code is in a great place!" if it really is) and the more retrospective they can be about the whole project, the better.

Questions

You may not have time to get through all of these questions, so pick whichever ones you think will cover areas that were not covered in the student's presentation. We should be trying to help them reach all four targets in the rubric. If they've already fully answered a question before you get to it, you can skip it.

-**When a user puts in a student whose age is over 10 years old, where is the error thrown?** Ideally, the error starts with the Model (hopefully throwing an exception) and gets passed to the Controller which then tells the View to tell the User. If the View does the age-check internally without engaging the Model or Controller, that violates Model-View-Controller separation. Remember, they don't lose marks here for having a View which handles errors, it's about whether they understand what's going on and can answer your question about it.

-**Can a player be removed once the team and lineup is created? What happens to the lineup if a player is removed from it? What happens to the team if it's pushed below 10 players?** Any answer is fine here, they may decide to not allow players on the lineup to be removed or a player to be removed if the team drops to less than 10, or allow it and have the system update the lineup or make the team unavailable. What they don't want is for the lineup to show just six players or still show the removed player, or for the old team roster to stay up on screen. Again though, they lose points here for their explanation and not the design error itself, if they think to show you these issues or can explain them readily then that's good, if they don't understand what's going on that's bad!

-**How does the Controller access the Model? Does the View access the Model anywhere?** We won't be able to confirm whether the View actually does access the Model or not unless we have some time to examine the code in more detail, but they should be able to confidently say that it doesn't (or admit that it does). If they aren't sure or don't show you any evidence, that's an issue. As for the Controller, ideally it only accesses the Model through one class (maybe a Team, maybe something called Model). If the Controller instances Player objects that it hands to the Model or Team, that does count as the Controller interacting with more than one Model class, and it's an issue if they don't recognize that.

Tips

-**It's important to keep track of time.** If a presenter goes over 15 minutes you should ask them to wrap up. If they really need a few more minutes you can offer to let them explain a little more after everyone else has done their presentations, but only if they need just another couple minutes.

-**Make sure to make notes and put some draft marks down during each presentation.** You can go back and revise those marks afterwards, but you need to at least record your reaction in the moment to help you reassess.

-**You need to be able to provide at least a few bits of feedback to justify the final marks**, so consider which area or areas on the rubric they could improve on and cite at least one concrete example of something they could have done there (for example, if they lost marks on Reflection you might suggest they should identify some potential weaknesses and limitations or talk about what they would have done differently if starting over).

-**Due to the group setting, some students may take cues from those who presented before them** in terms of what they highlight or how they explain their work. This is fine, even positive if a student gives their peers good ideas for what to talk about, though be ready to intervene if people get the wrong idea about what they need to spend their presentation time on. If a student who has already presented hears something they suddenly feel they should have covered they may want to comment on it as well, this may be allowed but try to discourage people from doing this too much so the presentations don't break down into a general discussion.