Final Project Design Questions

Design Check Signups: Until 12/1 at 11:59PM
Design Checks: December 2 – December 5
Help Session: November 19th, 2:30 PM (in class)

Before writing even one line of code, you should design your final project. This process will require you to break down and outline your program into classes, design your data structure(s), clarify the major functionality of your program, and pseudocode important methods. After designing your program, you will find that writing the program is a much simpler process.

If you do not sign up for your design check on time or you do not show up for your design check then you will get a 0% on your design check. If you cannot make any of the available times you must let us know ahead of time. Exceptions will be made for illness and other extenuating circumstances. “I forgot” is not an excuse.

Note: If you are doing an Indy project, you should follow the instructions on this handout as well as the instructions on the Indy handout.

How To Approach This Assignment

We highly, highly, highly recommend attending the In-Depth Help Sessions for your final project during class time on November 19th.

Additionally, you should read and reread the final project handouts thoroughly. Be sure you understand all of the functionality you are supposed to. You should be able to code directly from your answers to these questions.

IMPORTANT NOTE:
This design check will be graded interactively! We expect good written work. You should prepare your answers as if they would be turned in and graded, and bring them in ON PAPER. But this time you will be explaining your thought process to us during a 25-minute meeting. There will be a 10% deduction for not bringing the answers to the design questions in on paper.

During the meeting, we expect you to have answers prepared for all of the questions listed on the next sheet. We will ask you to explain in detail a subset of the questions below and provide feedback on these questions.

We are requiring you to physically bring in the following documents for the design check:
1. UML diagrams for containment / inheritance
   - Include ALL classes (including support classes and JavaFX components)
   - Write out signatures of major methods (as in the design lab) either in your diagram or in a separate list.
2. Pseudocode for the important algorithms in your project, as described by question 3

You should also be able to answer any other questions a TA may have about your planned design. You can take notes away from the design check. Your grade will be emailed to you after your check.

The design check will focus on the material you have prepared in advance. **Do not come to the design check without thoroughly thinking through your answers.** If you have questions, you should visit TA hours prior to your check.

Please note that failure to complete the design check during the period of **December 2nd to 5th** (Indy: 11/30 - 12/1) will result in a **0% on your design check**, which is worth 17% of your final project grade.

**How to Sign Up for a Design Check**

**Note:** You are fully responsible for all of the following information. Make sure you read the instructions thoroughly before signing up. The TAs are not responsible for any mistakes you make as a result of not understanding this information.

If you fail to sign up for a design check by Tuesday, Dec. 1st at 11:59pm, you will lose **20% of your DQ grade**.

**Sign-ups are first come first serve**, so if you wait too long to sign up and cannot make any of the remaining available times, TAs will **NOT** schedule design checks outside of the scheduled times.

**Head TAs will NOT be accepting emails regarding rescheduling design check sign-ups** unless there is a valid medical excuse. In other words, the policies of design checks resemble that of a midterm. If you miss your design check, you will not get credit and cannot reschedule.

The instructions for how to sign up are identical to how you signed up for Tetris design checks:

For each Final Project (except Indy), there are two spreadsheets, which you can access below. The **Unconfirmed Sign-up sheet** is where you can actually sign up. The **Official Schedule sheet** is what the TAs will use to call students for design checks. After you sign up using the Unconfirmed Sign-up sheet, make sure you check the Official Schedule spreadsheet to ensure you've actually signed up successfully.

**Before you sign up, please read the following carefully.** Once you successfully sign up for a design check (i.e. your name appears on the Official Schedule sheet), you will **NOT** be able to
sign up again. **Removing your name from the Unconfirmed Sign-up sheet will NOT remove it from the official sheet.** So, once you sign up, that is your spot for good; you cannot reschedule and you cannot sign up again.

**To be very clear:** Please sign up for a design check on the Unconfirmed Sign-up sheet. That is how you sign up for design checks, but your meeting isn't confirmed until it appears on the Official Schedule as well. In other words, the Unconfirmed Sign-up sheet has no significance with regards to the Design Check Schedule, it is simply a way to get onto the Official Schedule sheet. If you’re confused about this, email the TA list or post on Piazza for clarification.

REMEMBER: Write down the time you sign up for. You will get a 0% for missing your design check.

**Note:** You can only sign up for a design check for one project (unless you’re doing Indy, in which case you should sign up for both Indy and your backup project). Be sure you are writing your name down in the correct time slot for the correct project, because once you write it down and it appears on the Official Schedule, you will not be able to move it from that cell.

**Sketchy**
- [Unconfirmed Sign-up Sheet](#)
- [Official Schedule](#)

**Pacman**
- [Unconfirmed Sign-up Sheet](#)
- [Official Schedule](#)

**Othello**
- [Unconfirmed Sign-up Sheet](#)
- [Official Schedule](#)

**Indy**
- [Sign-up Sheet](#)

*Instructions to sign up for Indy are on the spreadsheet.*

*Remember:* you must also email your proposal to the Indy TAs by **Tuesday 8pm**.
Design Check Questions:

1. What data structure(s) will you use in this assignment? What will these data structures hold? How will you use them (be specific)?

2. Provide pseudocode for important methods. Make sure you explain the methods specified below.

   **Note:** Do not just regurgitate the pseudocode we give you in the handout! Think critically and write detailed pseudocode.

   - **Sketchy:** You should provide extremely detailed pseudocode for undo/redo, resizing a shape, and writing and reading a `CurvedLine` to and from a file. Each line of your pseudocode should have an almost 1:1 correspondence with a line of Java. Again, be specific!
   - **Othello:** You should provide pseudocode for your MiniMax AI algorithm and for checking for a “sandwich” (i.e., when there is a valid move). The vast majority of your work in prepared for your design checks should be understanding these two algorithms!
   - **Pacman:** You should provide pseudocode for the ghost’s breadth-first-search target-finding algorithm. Your pseudocode should translate almost line by line to Java, so be sure to include *all your variable initializations* and *correct types*. Write your helper methods in your pseudocode.
   - **Indy:** You should provide pseudocode for the major algorithm in your project. If you are still developing the specifics, it is allowed to be at a higher level than typical for CS15 Pseudocode.

3. Create a complete inheritance diagram. Include any interfaces you will write or use. **At this point it goes without saying - use UMLet!**

4. Create a complete containment diagram that resembles the ones we created in the Design Lab. If you prefer—these major methods and instance variables like you did in Question 5 of the Tetris DQs. **At this point it goes without saying - use UMLet!**

5. Write out a sample run of your program; explain what the user is doing and how the program reacts. This should be very specific (see examples below).

   Include information for each step about your program’s flow of control. Is it completely user-driven, completely automated, or a mix of the two? How will the flow of control be managed (be specific)?
Example of not specific enough: "The user clicks the button and the pieces flip over."
Example of specific enough: “The user clicks the ‘MoveButton’ which triggers an EventHandler that then prompts the user or AI to make a move”

Make sure you have a good understanding of the major components of your project!