Please read the *entire* Tetris handout before starting these questions. It will give you a better grasp of the project, which will aid you in thinking about its design and the questions below. Also, if you have trouble with these questions, we highly encourage you to attend the Help Session and bring any questions you may have.

These questions should be completed before you begin coding Tetris, as they are intended to help guide you toward an appropriate design. Please remember that this is worth approximately 20% of the grade for Tetris, which itself is a considerable percentage of your final grade. So make sure your answers are correct, and post a question to Piazza or ask a TA on hours if you need clarification on any points.

**Instructions**

**IMPORTANT NOTE:** This design check will be graded interactively, like DoodleJump's.

During the meeting, we expect you to have answers prepared **ON PAPER** (with preference to a printed typed response) for all of the questions listed on the next sheet. **There will be a 10% deduction for not bringing the answers to the design questions in on paper.**

You should 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 the design check.

We are expecting a higher quality design than you are used to. Please don't be scared; as long as you think through your design for Tetris very carefully you'll do fine. You may consult with a TA before the design check if you need help. Any extra work you put in now with design will pay off later when coding the project.

The design check will focus on the material that you have prepared in advance. **Do not come to the design check without an initial plan of attack.**

Again, you MUST bring a printed out version of your answers to these DQs to the design check!
How to Sign up

**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.

We were very lenient for DoodleJump design check sign-ups due to technical difficulties, but that WILL NOT be the case for Tetris. **If you fail to sign up for design checks by Sunday, Nov. 8th 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 the way they did for DoodleJump.

**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.

**There are two spreadsheets, which will be emailed out to you.** 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:** Do NOT try to overwrite another person’s name with your own. If you try to do so, you won’t be successful. The official schedule won’t change. Google Sheets also has edit history, and if we find out you tried to overwrite another student’s name, you will be penalized.
Design Check Questions:

1. What data structure will you use to represent the board? What type of object will this data structure hold? Given a piece’s coordinates, how will you figure out where you place its squares within your data structure?

2. How will you check to see if a move is valid (i.e. that the current piece is not trying to move off the board or onto an already fallen piece)? Will the board keep track of this? Or the piece? Or the individual components of the piece?

3. Give pseudocode for the algorithm you will use to clear completed lines, you should pseudocode both detecting filled lines and removing them. This algorithm must work on both a graphical level ("hiding" the cleared blocks, and moving the blocks of the other lines downward) and a logical one (updating the game board with the new arrangement of squares from the current piece). If you are unclear on what pseudocode is, refer back to lecture slides, the pseudocode lab or the DoodleJump DQs.

4. 
   a. Create a simple logical (not graphical) containment diagram for your program. You must use UMLet for this or you won’t receive credit for this question!
   You do not need to indicate graphical containment. All classes in your program should be modeled in your diagram.
   
   b. Create a simple inheritance diagram for your program, showing which classes extend which other classes, and which interfaces (if any) they implement. You must use UMLet for this or you won’t receive credit for this question!
   *PLEASE* ask on Piazza if you do not understand what we want in these diagrams.

   Note: You must have a diagram for BOTH containment AND inheritance.

5. Give a detailed description of each of the major classes in your program. Think carefully about all the classes you will need! This description should include:
   
   ● Its name
   ● A short description of what this class does
   ● A list of its instance variables
   ● A list of methods this class has, and a very short (1 sentence) description of what each does

   This is only a partial list of questions a TA may ask you.