CS246 - Assignment 5 Plan

Team members: Alvin Tran (a2tran), Guoyao Feng (g2feng)

Plan:

During the brainstorming phase of this project, it was decided that the project would employ the model-view-controller architecture. In this architecture, the project is designed to allow a user to use a controller to interact with the data and behaviour of the main application (the Quadris game), and provide the user with the means of viewing the progress on the application graphically. Thus, much of the planning for this project will be based on how to design it to fit this architecture, and who will be responsible for various sections of the architecture.

A session will be held to decide what classes are to exist in the different layers of the program's architecture. For example, it may be decided that a class would be used to represent any kind of block in the program, that there will be child classes of that block class representing the specific block types, such as the L-block, the I-block, Z-block, etc, and that these classes would be implemented at the game-logic layer of the code. Coming up with and organizing the placement of these classes will involve the creation of an initial UML diagram of the program. This task is expected to take around one or two days, finishing on November 24, roughly.

Once the initial design of the program has been thought up, documented, and laid out on a UML diagram, the actual programming of the project will be split up based on the different layers of the program. To start off, this project will first aim to satisfy its minimum requirements, which is to create a game of Quadris that generates its blocks in a fixed sequential manner (level 0) and displays on a textual grid on the command line. Since the project will employ the model-view-controller architecture, members of this group must handle with a model layer, a view layer, and a controller layer. Thus, Alvin Tran will handle the game logic (model) layer, while Guoyao will handle the view and controller layer.

For the initial programming phase of this project, a member of the group will start by creating the header files that follow the design of UML diagrams for the project. Hopefully, this task will be finished by the first due date. At the same time, the other member would create a set of basic test cases to run their portion of the code under once the code is done. This is to ensure the members have an idea on how the finished program is supposed to behave like before finishing the code, as creating such cases would encourage each member to figure out such details about the program's output in order to write proper test cases. Once this is done, the members will write the actual code for the declarations made in the header files. It is believed that the visual display components of the application will be the first target to implement since it will be of great help for debugging underlying logic. Considering the workload from another programming course, the development of each layer is expected to take around three to

four days after the due day, meaning it is expected to be done around November 29.

As the different layers will be handled by different people, they will be tested separately from each other to ensure that, alone, they work. This will likely require the person handling the layer to create a substitute layer to help the layer of focus run (for example, if the person is working on the graphical portion of the program, the code for a board similar to the one used for this project may be created to test if the graphics code will render that board correctly). As such, this portion of the project will likely last one day for each person's portion of the project, indicating that it will be accomplished on November 30.

Once all the individual layers have been created and tested, they will all be combined and tested once again to see that they can be used together. In order to do this effectively, a make file that combines the individual make files created during the previous phases will be created to help compile the code for the different layers of the project together. This part of the project will involve another session of testing and possible redesigning or recoding, which is expected to take another one to two days, giving it a finishing date of December 2.

When it has been decided that the project is satisfactory to this point, the project will be expanded to include more levels of Quadris (primarily, it will now employ a random number generator), and will include a graphical user interface. Adding these features is expected to take another one to two days, thus giving it an expected finishing date of December 3. Recombining the whole project (possibly updating the make file in the process), testing it as a whole, and fixing any bugs up to this point is expected to take an additional day after that, signaling the end of the primary project on December 4, if it is deemed satisfactory at this point.

Once the main project is done, another brainstorming session is to take place with regards to extra features to the project, if time permits. The development of any extra features, if any, is undecided at this point, so there is no set timeline for the different components of such development.