

CS 246 Assignment 5 Plan

Quadris Implementation Plan

This document outlines the roles, responsibilities and tasks set for each member of cs246_037, namely Harris Rasheed (h2rashee) and Amlesh Jayakumar (a3jayaku). It will outline the deadlines and milestones that each member will strive to achieve.

Plan of Implementation

The tasks in this project revolve around four of the major submission items.

- System UML Diagram
- Plan of Attack Document
- Quadris System Code
- Design Document

The Plan of Attack Document will be designed and discussed by both team members and targets and goals set for the duration of the two weeks on Sunday 20th November 2011.

Plan of Attack

The table below outlines the task, its responsibilities, and the team member responsible for its initiative (not the sole responsibility for the whole task) as well as the deadline date.

Task	Description	Team Lead	Deadline
UML Diagram Discussion	Talking about the architecture, required classes, fields and relationships in the system	Harris Rasheed / Amlesh Jayakumar	Friday 18 th November 2011 00:00
UML Diagram Design & Agreement	Drawing and design of the discussed UML diagram. Agreement from both partners of the feasibility and suitability of the system blueprint	Harris Rasheed	Sunday 20 th November 2011 22:00
Analysis of Time Constraints and Modularity of the Classes	Discussion and estimates on the demands of each module on each team member as well as the modularity and ease of implementing new features for bonus marks as well	Amlesh Jayakumar	Monday 21 st November 2011 23:00
Plan of Attack Document Write-up	Write-up of the Plan of Attack document for goals and targets	Harris Rasheed	Tuesday 22 nd November 2011 23:00

Quadris: Board, Block, Cell Header File	Typing up and write up of the main header files as well as pulling the necessary headers and files from previous assignments (Trie, PRNG)	Amlesh Jayakumar	Wednesday 23 rd November 2011 00:00
Quadris Code: board.cc, block.cc	Implementation of the functions outlined in the UML diagram for the Block and Board class	Amlesh Jayakumar	Wednesday 23 rd November 2011 17:00
Quadris Code: parser.cc, main.cc	Implementation of the functions outlined in the UML diagram for the Parser class and the Main function's file	Harris Rasheed	Wednesday 23 rd November 2011 17:00
Quadris Code: Block Rotation & Collision Prevention	Implementation of the rotation function for the block at any point on the board and prevention of collisions with other pieces and the edges of the board	Amlesh Jayakumar	Thursday 24 th November 2011 00:00
Quadris Code: Block Movement	Implementation of the block's movement around the board with the built-in functions. Recognition of commands having no effect e.g. moving a block left already at the leftmost wall of the grid	Harris Rasheed	Thursday 24 th November 2011 00:00
Quadris Code: Block Entry Collision	Implementation of a new block's entry into the grid yielding an immediate collision resulting in game over response to the user	Amlesh Jayakumar	Thursday 24 th November 2011 00:00
UML Re-Evaluation	Re-evaluation on the feasibility of the current prototype and if it will cater to the needs of the remaining bonus and extra features	Harris Rasheed	Thursday 24 th November 2011 00:00
Implementation of Level 0, a Memory Check on the Program's Allocation	Implementation of Level 0 and reading the blocks from sequence.txt as opposed to being based on probabilities for each level	Harris Rasheed	Thursday 24 th November 2011 00:00
Graphics Tetris Board Window	Design and set-up of the main window to host the grid upon which Quadris will be played, the level, current score and high score would be displayed	Amlesh Jayakumar	Friday 25 th November 2011 00:00

Graphics Tetris Next Block Window	Design and set-up of the side window showing the next block which will be placed on the Quadris grid is displayed	Amlesh Jayakumar	Friday 25 th November 2011 00:00
Graphics Methods draw(), undraw() Implementation	Implementation of the draw functions and its calls in the appropriate places to construct and maintain the graphics mode of Quadris	Amlesh Jayakumar	Friday 25 th November 2011 00:00
Valgrind Check	Re-check memory de-allocation with Valgrind if there are any further memory leaks independent of Xwindow's faulty allocation	Harris Rasheed	Friday 25 th November 2011 02:00
Make File Creation	Create the make file with the relationships for each class and object built-in in order to appropriately re-compile code and header files	Harris Rasheed	Friday 25 th November 2011 12:00
Bonus Features	Implementation of the extra features for bonus marks as outlined at the end of this document	Amlesh Jayakumar	Friday 2 nd December 2011 00:00
Code Documentation	Review of the code to see that it is sufficiently documented and explained as well as an evaluation of the simplicity of the code	Harris Rasheed / Amlesh Jayakumar	Saturday 3 rd December 2011 14:00
Design Document	Discussion and write-up of the design document which outlines the architecture and ideas behind classes and methods in the code and how the code behind operates to make the graphics a seamless application	Harris Rasheed	Saturday 3 rd December 2011 14:00
Final Testing Plan	A final in-depth test plan execution on the Quadris program graphical & text mode as well as the text mode on its own; this will include a memory check	Harris Rasheed / Amlesh Jayakumar	Sunday 4 th December 2011 17:00

Note: Team lead of a task indicates that the person is in charge for the initiative for the project. Both team members will concurrently carry out the tasks outlined above in an attempt to co-operate and have a mutual understanding of all areas of code. This is possible because of the living arrangements of the two team members.

Goals of the Project

Statement of Intent

Both team members agree that the minimum target of implementing the assignment's basic requirements of a complete implementation should and can be met within the given time span. This includes but is not limited to the implementation of four different levels, a text and graphics option for the game, the command interpreter abilities to read a command given a unique beginning of the command and a command multiplier.

Extra Features

In addition to this, we plan to implement extra features for bonus marks. This includes:

- a renaming command feature; If the user deems that the chosen names for each command is too complicated or misleading, he/she may rename the command to any string solely comprised of alphabets.
- a feature that will check whether a given command is ambiguous; it will inform the user that they have input a command prefix that isn't completely unique because more than one matching command was found. This will guide the user to use the appropriate command.
- minimal redrawing of the cells on the board (graphical window) to reduce the amount of visual change in the grid after every move. Only the cell blocks that change in colour are redrawn making the program faster and efficient in the use of drawing algorithms.
- a cheat mode; This will involve the user entering the command *cheats* which enables cheats mode and permits additional commands that go against the spirit of the game. Cheats that can be used include:
 - a command that destroys (removes) a row given the row number relative to the bottom of the grid; this, of course, does not grant the user any points but helps them destroy a row that prevents them from potentially clearing a row below it
 - a command to clear the screen in an attempt to restart the game but however retaining the current score; this will typically be used by the user when they have made a mistake and need to reset the board without causing a game over
- a hint feature; This will provide the user with the option of being able to get help from the program for the next possible best move that they could take in the game.
- a help feature; This will describe the actions of each command (similar to the *man* command in bash). It will also aid the user in the case where an ambiguous prefix is input.
- a music feature; This plays the classic Tetris music on loop for the duration of the game.