CS437

**FINAL REPORT**

CLEAR



# 

# Contributions

# Patrick Collins:

* Worked to create a containerized development environment (with codenvy IDE, noVNC, SFML) so that all group members could easily work outside the lab
* came up with the idea of Clear; and of everything (block types, generator, modes) that ended up making it in the final version besides the office-context humor (which Ivan thought of.) (Of course, owing credit to the games that directly inspired it.)
* coordinated early group meetings.
* gave input and guidance on software architecture and development.
* added makefile - handles all the cmake stuff and cleanup etc.
* refactored the entire codebase multiple times; in addition to the times described below.
* fully revamped all of the block classes; redoing all their functions and inheritance.
* fully revamped the game board's representation, changing it from a 2D array of ints into a 2D array of pointers to instances of the block classes.
* redid everything to standardize the coordinate system. we'd been mixing [row][col] and [col][row], as well as x/y and row/col terms; in inconsistent and error-causing ways.
* implemented most of the game’s logic. (predicates etc)
* fully refactored all of the screens and menus
* standardized formatting for the whole repo
* created the entire generator from scratch; with all supporting logic
* trimmed down inefficiencies and get rid of magic strings etc.
* added a bunch of functions (with new functionality; not just de-duplicating) to simplify the code
* Implemented the functions that hint relies on.
* split lots of existing functions into a bunch of smaller functions that could be reused in more places
* added functions to convert between positions on the board and pixels on the screen, to simplify GameView.
* added functions to check whether blocks can be removed; which will be critical for the generator. (involves checking if modifying other blocks on the board affects removability)
* created the import/export functions for boards and blocks
* fixed a bunch of bugs

# Ivan Echevarria:

* Rewrote design document (expand related games, player composition, UI storyboards, level design; add and write overview, changelog, team contributions, status)
* Made GameState class and separated game logic from GameView class
* Thought of the idea for office-style humor
* Moved SFML shapes from Clear.cpp to GameView
* Moved keyboard input handling from GameState to GameView
* Refactored all classes
* Expanded board from 1-D to 2-D/Rewrote all early loops
* Updated block selection behavior
* Simplified collision and selection logic
* Added mouse interaction
* Added shadows, selection highlighting and path highlighting
* Created GameState
* Created BoardState
* Added selection highlighting in main menu
* Added title animation
* Add comments
* Make levels
* Fixed a bunch of bugs

# Yosef Ejigu:

* Created Clearboard and Rotating\_Block classes
* Added attributes to Block and Normal\_Block classes
* Contributed to the separation of GameView and GameState
* Fixed graphics overlay issue in GameView
* Fixed issues with the clearboard addition and removal
* Expanded board from 1-D to 2-D/Rewrote all early loops with Ivan
* Found the fonts that were used in the game
* Built in the screen handling system
* Created the Clear animation from scratch
* Removed the hard coded string and placed them in a definitions header file
* Added the hint system into the game
* Created the transition as well as the menu screen classes
* Designed an alternative main menu, although it was eventually scrapped

# Yiyang Zhao:

* Created basic framework of Clear (GameView, Normal\_Block)
* Made functional collision detection
* Made mouse input and keyboard input functional
* Created repository
* Contributed to expansion of the board from 1-D to 2-D
* Rewrote all early loops with Ivan
* fixed the cmake discrepancy between github and local
* Create prototype of generator
* added a bunch of functions to simplify the code
* Add menu with Yosef
* Add start & continue button
* Add mouse interaction on menu
* Add level menu
* Connect the GameScreen to GameState
* Design some levels
* Add custom mode in main menu and connect to BoardGenerator
* Recording demo video
* Remove a brunch magic numbers
* Fixed a bunch of bugs

# 

# Challenges and Solutions

During the last period, the most challenge of us was designing the generator. We added the custom mode in our game, in this mode, players are able to set the board size by themselves and select count of each type of block in the board. Based on these conditions, a board should be generated. We must guarantee the puzzle is solvable which means we should take consideration into all combinations of all types of blocks and tabs.

The next challenge was hint of the game, we added the hint button in the game. If players have no idea about which block could be moved. They can get the hint from the button. What a challenge here is that we should find out which block can be moved out of the board directly or potentially in current board state. The block which can be potentially removed means that the movement path is not blocked and the tab can be unlocked by rotating itself or adjacent blocks. If a block satisfy all these conditions, the pointer of the block would be returned by hint function.

Another challenge was separating the GameState from the GameScreen. In our previous versions, all parameters regarding to the gamestate located in GameScreen class. This architecture was no clear for developer. Developer would be prone to misuse or ignore such parameters and also render some unpredictable errors. In order to solve this, we created a new class called GameState to save such parameters. That make us easily to get the current state of the game.

Level designing would also be the challenge we faced. There are lots of elements that we need to be considered, such as the different combinations of block types, the difficulty, interestingness. Most important is that we should ensure the puzzle in each level is solvable. Designing a level is really time consuming, we take lots of time for that.

SegmentFault and pointer usage are always the huge challenges for us throughout this project. We searched the solution by Google, checked the code and traced the error file by file, line by line. We actually gained lots of knowledge by debugging and our programming (c++) ability would be obviously improved.