Target 9 Game

ALEXANDROS ATTESHI

Functions Breakdown

Cosmetic and Informative Functions:

- void line()
- void header(const char[])
- void print_grid()
- void display_grid(int, int)
- void game_information(int, int)
- void options(stack<Move>&, stack<Move>&)

User input Functions:

- bool question(const char[])
- void set_target()
- int difficulty_set()
- int options_choice()
- void move_choice(int&, int&)

Simple fact check Functions:

- bool int_check(int, int, int)
- bool win_check()

Algorithmic Functions:

- void initialize_grid(int)
- void undo_grid(int, int)
- void redo_grid(int, int)
- void choice_implementation(int, int&, stack<Move>&, stack<Move>&);

Function void initialize_grid(int)

This function sets all numbers in the grid to a preconfigured number

Then

It generates two random numbers between 0 and greed size, using a time seed for row and column

Then

It passes the row and column generated to undo_grid function

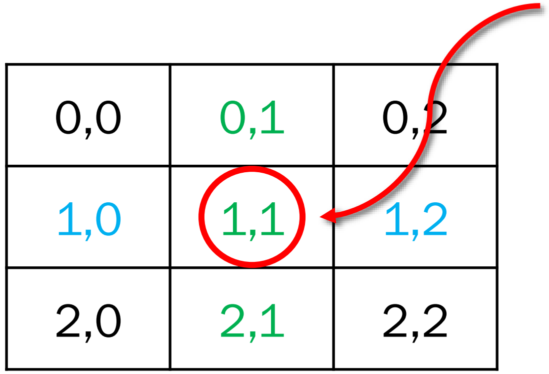
This is done as many times as the difficulty

Function void undo_grid(int, int)

The purpose of this function is to decrement by 1 all the numbers in the same row and column.

To decrement by one, I took the number and added 9 and then to get the remainder I did a modulus 10 and replaced the original with the answer.

Example: I have number 9 and I want it to become 0.
9+9 = 18
18/10= 1.8
The remainder is 8



0,0	0,1	0,2
1,0	1,1	1,2
2,0	2,1	2,2

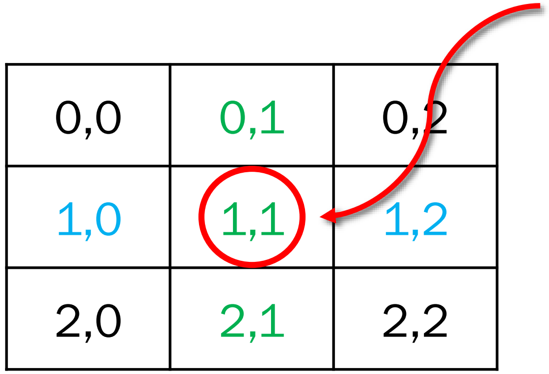
Now we need to **change the numbers**,
if our number is the middle one,
we change this one first and then using two for loops one for the row and another for the column change the rest,
and inside the for loops we implement an if statement to make sure we don't change the middle one in this case, each time we pass

Function void redo_grid(int, int)

The purpose of this function is to increment by 1 all the numbers in the same row and column.

To increment by one, I took the number and added 1 and then to get the remainder I did a modulus 10 and replaced the original with the answer.

Example: I have number 9 and I want it to become 0.
9+1 = 10
10/10= 1
The remainder is 0



0,0	0,1	0,2
1,0	1,1	1,2
2,0	2,1	2,2

Now we need to **change the numbers**,
if our number is the middle one,
we change this one first and then using two for loops one for the row and another for the column change the rest,
and inside the for loops we implement an if statement to make sure we don't change the middle one in this case, each time we pass

Function void choice_implementation(int choice, int& counter, stack<Move>& undoStack, stack<Move>& redoStack)

This is the function is the brain of the program, its job is to relegate tasks to the other functions, we can split it into 3 main tasks (play, redo and undo)

Play:

- Ask the user to enter his choice for row and column through function move_choice
- Passes the row and column to function undo_grid in order to increment by 1
- Save the row and column to undoStack
- Empty's the redoStack
- Increments the counter for the moves by 1

Undo:

- Cals the undo_grid Function with last row and column entry from the undoStack
- Save the row and column to redoStack
- Remove's last entry from undoStack
- Decrement's counter by 1

Redo:

- Cals the redo_grid Function with last row and column entry from the redoStack
- Save the row and column to undoStack
- Remove's last entry from redoStack
- Increment's counter by 1