

Я реализовал класс Game, который отвечает за все взаимодействия с пользователем в игре Морской бой. Он инициализирует, проводит и заканчивает игру.

Класс содержит поля:

- player_name_1 - имя первого игрока,
- player_name_2 - имя второго игрока,
- log - информация о последних действиях игроков.

Реализованы следующие методы:

- game_enter() - инициализирует игру, просит пользователя ввести размер поля, пока пользователь не введёт корректное число.
- calculate_ships(int fieldSize, int& remember_ship_sells) - просчитывает рекомендуемое количество кораблей для расстановки на поле размером fieldSize. Алгоритм этого метода описан ниже.
- check_status(Player player1, Player player2) - проверяет, не выполнено ли условие победы.
- game_end(Player player1, Player player2) - завершает игру, выводит победителя.
- info_players(Player player1, Player player2) - возвращает информацию о том, кто играет.
- output_inf(Player player1, Player player2) - выводит корабли игроков. Если игра с компьютером, то оба. При игре игрок против игрока, выводится только поле противника, чтобы нельзя было подглядеть расстановку кораблей.
- process_steps(Player& player1, Player& player2, bool computer, bool only_comp) - описывает как происходит ход. Параметр computer указывает на то, что игрок является компьютером, а параметр only_comp показывает, что игра находится в режиме компьютер против компьютера.
- output_finish_info(Player player1, Player player2) - вывод информации после боя.
- cycle_play(Player& player1, Player& player2, bool computer1, bool computer2) - главный цикл игры, запускающий process_steps для игроков, до выполнения условия окончания игры. В конце выводит информацию об игроках(output_finish_info) и победителя(game_end). Параметры computer1 и computer2 передают информацию о том, является ли тот или иной игрок компьютером.

Практически в каждый метод передаются как параметры игроки Player player1, Player player2.

Внутри класса также прописаны вспомогательные методы для работы с консолью(clearPartOfConsole, SetCursor) и списком событий(save_log, displayLog, scroll_log).

Алгоритм для вычисления количества кораблей (в методе calculate_ships) работает следующим образом. Для начала находится количество клеток, которые должны быть заполнены кораблями, было решено взять 20% от всего количества. Метод последовательно расставляет корабли, пока для них есть место. Он начинает заполнять с одно палубных. Каждый раз, когда количество n-палубных становится меньше на 2 чем количество (n-1)-палубных, программа пытается вставить ещё один n-палубный.

Пример:

Поле: 13x13. Всего количество клеток: 169. Значит клеток для кораблей: 33.

	1-палуб.	2-палуб.	3-палуб.	4-палуб.	5-палуб.	6-палуб.
Шаг 1	1	0	0	0	0	0
Шаг 2	2	0	0	0	0	0
Шаг 3	2	1	0	0	0	0
Шаг 4	3	1	0	0	0	0
Шаг 5	3	2	0	0	0	0
Шаг 6	3	2	1	0	0	0
Шаг 7	4	2	1	0	0	0
Шаг 8	4	3	1	0	0	0
Шаг 9	4	3	2	0	0	0
Шаг 10	4	3	2	1	0	0
Шаг 11	5	3	2	1	0	0
Шаг 12	5	4	2	1	0	0
Шаг 13	5	4	3	1	0	0
Шаг 14	5	4	3	2	0	0
Шаг 15	6	4	3	2	0	0
Шаг 16	6	5	3	2	0	0