

Miłosz Kowalczyk

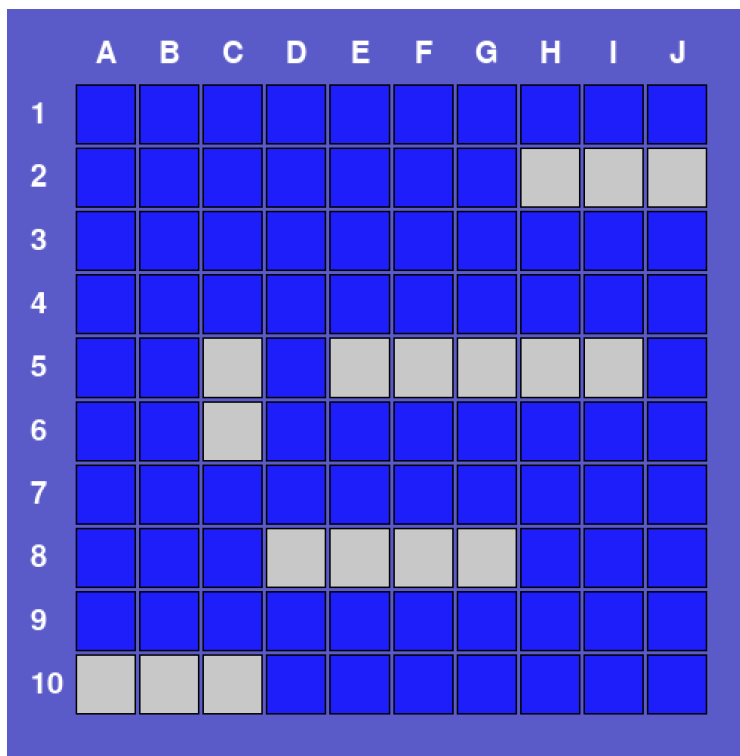
Statki

Statki to klasyczna gra polegająca na trafienie pięciu statków o różnych rozmiarach. Są one rozmieszczone losowo na planszy i poprzez zgadywanie trzeba zatopić je wszystkie aby wygrać. Ta wersja gry jest jednoosobowa i kończy się kiedy trafimy wszystkie statki ustawione przez komputer.

Opiera się ona na wersji z 1990 roku, która posiada pięć statków:

Numer statku	Nazwa statku	Rozmiar statku
1	Carrier	5
2	Battleship	4
3	Cruiser	3
4	Submarine	3
5	Destroyer	2

Tabela 1. Statki obecne w grze.



Rysunek 1. Wygląd gry z widocznymi statkami.

Jak widać na rysunku powyżej, statki nie mogą do siebie przylegać. Między nimi musi znajdować się bufor co najmniej jednej kratki. Każda kratka jest podpisana literami oraz liczbami. W tradycyjnej, papierowej wersji, jest to wykorzystywane, aby powiedzieć, w które pole strzelamy. Tutaj jednak pole, które chcemy trafić wybieramy poprzez kliknięcie kursorem myszki.

Uruchomienie gry

Do uruchomienia gry potrzebujemy zainstalowany język Python oraz bibliotekę Pygame.

Aby uruchomić grę, potrzebujemy w terminalu przejść do folderu zawierającego wszystkie pliki gry a następnie wpisać komendę:

```
python statki.py
```

Struktura plików:

Battleships/

-- statki.py	# Główna logika gry
-- settings.py	# Konfiguracja gry
-- ship.py	# Klasa statku
-- button.py	# Klasa przycisku

Omówienie kodu:

Statki.py – główna klasa gry

`__init__()`

Tworząc nową grę inicjalizujemy pygame, pobieramy ustawienia z pliku settings oraz wywołujemy funkcję `reset_game()`, która tworzy nam planszę z rozmieszczonymi statkami. Plansza ma rozmiary 10x10 i składa się z 4 wartości:

0 – woda

1 – statek

2 – trafiony statek

3- pudło

`reset_game()`

Tworzy okno gry, ustawia w nim kolory, tworzy nową pustą planszę, a następnie ustawia na niej statki. Resetuje liczbę prób trafień oraz ustawia warunek końca gry na fałsz.

`place_ships()`

Zajmuje się rozmieszczeniem pięciu statków, które będą przechowywane w liście `ships`. Dla każdego z pięciu statków rozpoczyna pętlę, która zakończy się tylko gdy statkowi zostanie przypisane jakieś miejsce na tablicy. Zapewnia to zmienna `placed`. Losowane są współrzędne `x` oraz `y` jak i orientacja statku – H dla poziomej, V dla pionowej. W zależności od rozmiaru w liście `ship_cells` zapisywane są współrzędne wszystkich pól zajętych przez statek. Sprawdzane są dwa warunki: pierwszy – czy statek wychodzi poza granicę planszy, drugi – czy statek koliduje z jakimś innym statkiem lub buforem wokół niego. Jeśli tak, to zaczynamy pętlę od nowa i losujemy nowe położenie statku. Jeśli nie, to zmieniamy pola planszy, które statek zajmuje na 1 oraz tworzymy obiekt statku, który dodajemy do listy `ships`. Po rozmieszczeniu wszystkich statków zwracamy listę `ships`.

`check_collision(ship_cells)`

Funkcja sprawdza, czy statek, który chcemy stworzyć koliduje z jakimś innym statkiem, lub jego otoczeniem. Najpierw sprawdzamy, czy którakolwiek, z komórek, które chcemy przypisać nowemu statkowi należy już do jakiegoś innego okrętu. Jeśli tak, to zwracamy, prawdę, że występuje kolizja. Jeśli nie, przechodzimy do sprawdzania kolizji otoczenia. Używając zagnieżdżonych pętli sprawdzamy, wszystkie 8 pól dookoła danej lokalizacji. Jeśli znajdują się one poza obszarem planszy, to je ignorujemy, ale jeśli nie to sprawdzamy czy nachodzą na jakiś

statek. Jeśli tak to zwracamy prawdę. Jeśli zakończymy sprawdzanie dla wszystkich komórek okrętu, bez wykrycia kolizji, to zwracamy fałsz.

`check_victory()`

Sprawdza czy wszystkie statki zostały zatopione

`show_start_screen()`

Wyświetla napis „Start Game” oraz przycisk, który umożliwia jej rozpoczęcie. Czeki aż przycisk zostanie kliknięty.

`show_victory_screen()`

Wyświetla wiadomość o wygranej, a także o ilości błędnych trafień oraz przycisk pozwalający na ponowne zagranie. Czeki aż przycisk zostanie kliknięty i jeśli jest, to resetuje grę.

`check_events()`

Sprawdza, czy zostało kliknięte zamknięcie okienka i jeśli tak, to kończy grę. Sprawdza też każde kliknięcie myszką i przesyła je do funkcji `check_guess` aby ta sprawdziła, czy trafiono w statek, czy nie.

`check_guess(x, y)`

Bierze koordynaty kliknięcia myszki i odpowiednio przesuwa je według przesunięcia ekranu gry. Następnie sprawdza czy kliknięcie znajdowało się w obszarze planszy. Jeśli tak, to sprawdzamy jakie miejsce zostało kliknięte. Jeśli została trafiona woda, zwiększamy liczbę prób o 1 i zmieniamy komórkę na pudło. Jeśli został trafiony statek, to zmieniamy komórkę na trafioną. Sprawdzamy czy został trafiony statek, przez iterowanie przez listę statków do momentu kiedy trafimy (lub nie) na miejsce zajmowane przez jakiś okręt.

`draw_grid()`

Rysujemy litery i cyfry otaczające planszę, a później planszę ze wszystkimi komórkami oraz odstępami między nimi. Kolor komórki wybiera funkcja `get_cell_color`.

`get_cell_color(cell_value)`

Bierze wartość która znajduje się na danym położeniu tabeli i odpowiednio zwraca kolory.

0 – woda	niebieski
1 – statek	niebieski
2 – trafiony statek	czerwony
3- pudło	biały

`update_screen()`

Rysuje ekran, planszę i wyświetla liczbę strzałów nietrafionych.

`show_attempts()`

Wyświetla tekst z liczbą nietrafionych strzałów.

`run_game()`

Zaczyna od ekranu startowego. Po kliknięciu startu rozpoczyna pętlę gry, gdzie sprawdza zdarzenia i odpowiednio aktualizuje zawartość ekranu. W momencie gdy zostanie spełniony warunek zwycięstwa, z lekkim opóźnieniem pokazuje ekran wygranej.

Klasa Button:

Klasa reprezentująca przycisk.

`__init__(screen, x, y, width, height, text, color, hover_color, text_color)`

Ustawia wszystkie swoje ustawienia, kolory, rozmiar, tekst.

`draw()`

Rysuje się na ekranie.

`is_clicked(event)`

Sprawdza czy został kliknięty.

Klasa Settings

Zawiera rozmiar ekranu, planszy, komórek, marginesy, kolory, które będą używane.

Klasa Ship

Klasa reprezentująca pojedynczy statek.

`__init__(x, y, size, direction)`

Ustawia lokalizację statku, jego rozmiar i kierunek (pionowy i poziomy)

`check_hit(col, row)`

Sprawdza czy przekazane koordynaty to położenie jakiegś z jego komórek. Jeśli nie to zwraca fałsz, jeśli tak to ustawia tę komórkę na trafioną i zwraca prawdę.

`is_sunk()`

Sprawdza czy wszystkie komórki statku zostały zatopione.