

## Dokumentacja projektu - gra w statki

Autor: Piotr Michta

### Opis:

Projekt obejmuje implementację gry w statki w języku Python z użyciem biblioteki pygame.

Projekt składa się z czterech plików:

battleships.py - plik zawiera implementację menu głównego oraz uruchamia plik game.py

game.py - implementacja właściwej gry zawierająca główną pętlę gry oraz wiążąca elementy z pliku game\_components.py we właściwe komponenty gry

game\_components.py - plik zawiera implementację komponentów gry - obiekt reprezentujący kafelek planszy (klasa Tile) oraz całą planszę (klasa Board), wraz ze wszystkimi metodami

oponent.py - plik zawierający implementację przeciwnika komputerowego oraz metody jego oddziaływania z komponentami gry

### Mechanizmy:

Klasa Tile - każdy kafelek rysuje siebie oraz jest świadomy swojego stanu (odblokowany/zablokowany na dodawanie statku, czy jest lub nie jest statkiem, czy jest trafiony czy nie). W zależności od stanu w którym się znajduje rysuje siebie w odpowiedni sposób - znając swoje kolory, nie są to jednak wszystkie możliwości kolorowania kafelka - gdy jest częścią zatopionego statku lub gdy znajduje się na nim kursor myszy jego pokolorowanie na odpowiedni zostaje zlecone przez klasę Board oraz w trakcie stawiania statków kolorowanie na odpowiedni kolor zleca metoda odpowiedzialna za stawianie statków.

Klasa Board - zawiera w sobie dwuwymiarową tablicę kafelków (instancji klasy Tile) reprezentującą pola gry. Zawiera także listę statków znajdujących się na danej planszy. Gdy pole nie jest zatopionym statkiem, kafelki rysują się same natomiast kolorowanie tych będących częścią zatopionego statku zleca klasa Board. Klasa ta posiada również metodę wywoływaną przy uderzeniu w planszę i odpowiednio modyfikuje stany kafelków.

Stawianie statków gracza na planszy - funkcja place\_one\_ship w pliku game.py odpowiada za postawienie jednego statku gracza na planszy. Funkcja oblicza maksymalne pole na którym gracz może położyć odpowiednio duży statek w pozycji horyzontalnej lub wertykalnej oraz sprawdza przy każdej zmianie pozycji myszy czy pola na których gracz chce położyć statek nie są zablokowane. Po położeniu statku, przy pomocy funkcji place\_ships statek jest dodawany do listy statków odpowiedniej planszy oraz dodaje każdy kafelek do instancji klasy Board osobno za pomocą metody place\_ship\_tile, aby odpowiednio zmienić stan przekazanego kafelka na kafelek statku oraz sąsiednich kafelków na zablokowane do położenia statku.

Generowanie statków przeciwnika - funkcja generate\_ship generuje statek o zadanej długości na podanej jako parametr planszy. Funkcja losuje pozycję umieszczenia statku

oraz położenie wertykalne lub horyzontalne, a następnie sprawdza dostępność kafelków wchodzących w skład potencjalnego nowego statku. Dodanie statku do planszy jest analogiczne jak dla statków gracza.

#### Instrukcja:

Aby rozpocząć grę należy wcisnąć guzik "PLAY" w menu głównym, a następnie za pomocą myszy umieścić swoje statki przy pomocy myszy na planszy po lewej stronie. Statki można obracać za pomocą klawisza R, a żeby go umieścić należy nacisnąć lewy przycisk myszy. Po umieszczeniu wszystkich statków na planszy - jeden o rozmiarze 4, dwa o rozmiarze 3, trzy o rozmiarze 2 i cztery o rozmiarze 1 można oddawać strzały w statki przeciwnika na planszy po prawej stronie za pomocą myszy. Aby oddać strzał należy nacisnąć lewym przyciskiem myszy na wybrane pole. W przypadku pudła kolor kratki będzie czarny, w przypadku trafienia statku czerwony. Kolor jasnoczerwony oznacza że tylko część statku została zatopiona, natomiast kolor ciemnoczerwony oznacza całkowite zatopienie statku. Zwycięzcą jest gracz, który pierwszy zatopi wszystkie statki przeciwnika.

#### Referencje do materiałów wykorzystanych:

<https://www.pygame.org/docs/> - strona pygame zawierająca poradniki jak korzystać z biblioteki

<https://github.com/pygame/pygame> - repozytorium z kodem biblioteki

<https://www.pygame.org/wiki/tutorials> - poradniki, które polecają twórcy biblioteki

#### Inne użyte poradniki:

[http://www.raspberrypi-geek.com/Archive/2014/05/Pygame-modules-for-interactive-program-s/\(offset\)/2](http://www.raspberrypi-geek.com/Archive/2014/05/Pygame-modules-for-interactive-program-s/(offset)/2)

<https://nerdparadise.com/programming/pygame/part3>