

Mateusz Kurcoń

Grupa: P3

SEA BATTLE

Dokumentacja projektu

Programowanie w języku C++

Spis treści

1. Wstęp	3
2. Zasady gry w statki	3
3. Instrukcja obsługi gry.....	4
4. Opis kodu.....	6

1. Wstęp

Gra strategiczno-planszowa dla dwóch osób. Wynaleziona przez Clifforda Von Vicklera na początku XX wieku, a opatentowana przez Milton Bradley Company w 1943 roku. Gra jest znana w wielu krajach pod nazwą: gra w Statki lub Salvo. W polskiej literaturze można się też spotkać z nazwą: wojna morska lub bitwa morska.





Źródło: Wikipedia

Gra została stworzona w języku C++ przy użyciu biblioteki graficznej SFML. Jest odwzorowanie rozgrywki statków na kartce papieru. Dodatkowo można rozegrać wojnę okrętów przeciwko komputerowi.

2. Zasady gry w statki

Do gry w statki potrzebne są dwie plansze składające się z 10 wierszy, i 10 kolumn. Celem gry jest jak najszybsze „zatopienie” okrętów przeciwnika.

Na jednej planszy gracz rozstawia swoje statki, a na drugiej przeciwnik. Statki mogą być ustawione w pionie i poziomie, nie mogą stykać się ze sobą bokami, ani rogami, i nie mogą być prostokątne (w kształcie litery L). Na planszy rozstawia się domyślnie 10 statków (możliwość zmiany ilości statków za pomocą zakładki „Settings”):

- Jeden 4-masztowiec 
- Dwa 3-masztowce 
- Trzy 2-masztowce 
- Cztery 1-masztowce 

Rozgrywka

Oddawanie strzałów odbywa się naprzemiennie: najpierw jeden z graczy oddaje strzał, jeśli uda się mu trafić, może kontynuować ruch do momentu, aż chybi. Kolej oddawania strzału gracza jest symbolizowana poprzez strzałkę. Zniszczenie statku ma miejsce, kiedy wszystkie pola symbolizujące okręt zostaną trafione przez danego gracza. Moment zatopienia statku jest sygnalizowany otoczeniem danego statku symbolami z kotwicą. Pudło oznaczane jest symbolem kotwicy, strzał w statek jest oznaczony x w czerwonej obwódce. W pola, w które został oddany strzał, nie można ponownie strzelić.

Znaczenie symboli:



- oznacza pole na planszy



- oznacza pudło, jest jednocześnie otoczką zniszczonego statku



- oznacza część statku



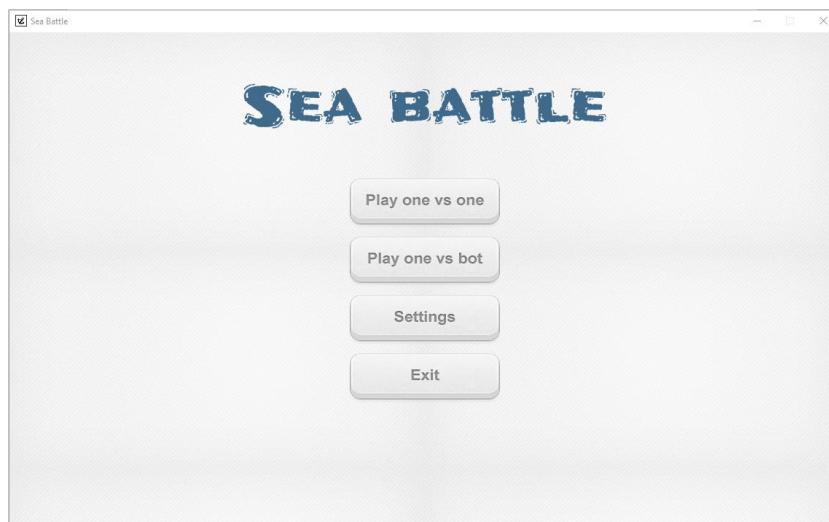
- wskazuje na którą planszę można oddać strzał



- oznacza trafienie części statku

3. Instrukcja obsługi gry

Ekran startowy gry:



Opis przycisków:

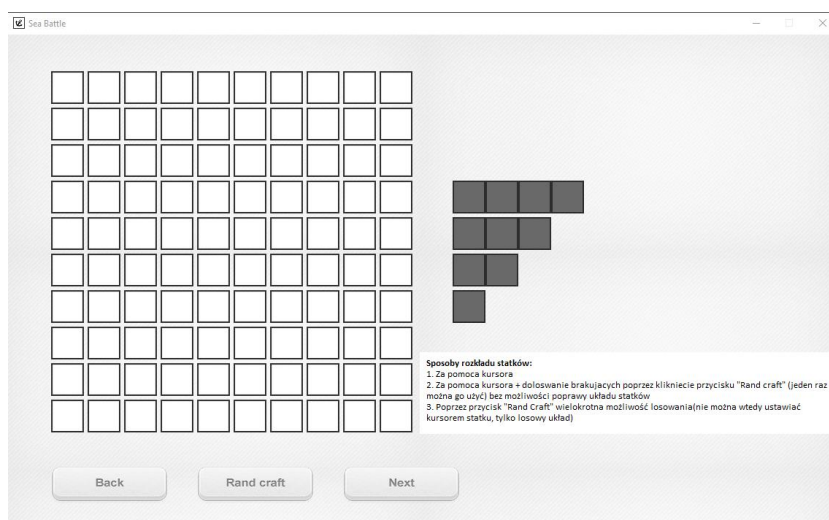
„Play one vs one” – tryb gry, umożliwiający rozgrywkę z znajomym

„Play one vs bot” – tryb gry, umożliwiający rozgrywkę z komputerem

„Settings” – ustawienia gry(konfiguracja ilości statków, otaczanie statków)

„Exit” – wyłączenie gry

Rozstawienie statków na planszy:



„Back” – powrót do ekranu startowego gry

„Rand craft” – losowe rozstawienie statków

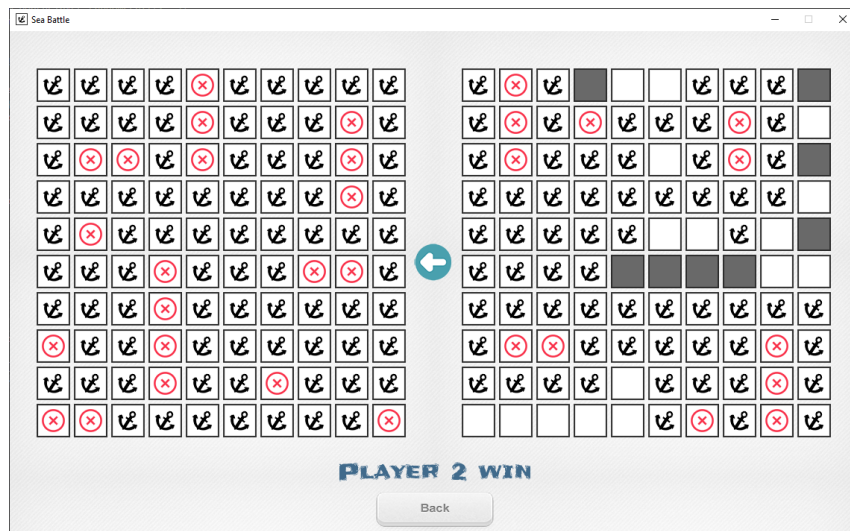
„Next” – przejście do ekranu dalej

Statki można ułożyć na planszy na dwa sposoby:

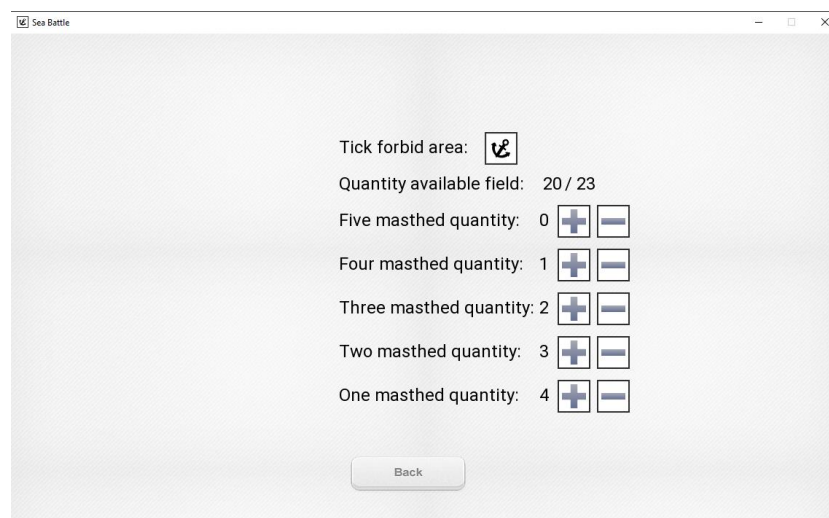
1. Używając przycisku „Rand Craft”, który losowo rozkłada na planszy statki, po użyciu tej funkcji nie ma możliwości przestawienia statku przez gracza.
2. **Lewym przyciskiem myszy** klikamy w wybrany statek, zaznaczamy go i poprzez przenoszenie kursora, można go ułożyć na planszy w wybranym miejscu.
Orientację statku pionową/poziomą zmienia się poprzez kliknięcie **prawego przycisku myszy**.
3. Ustawienie kilku wybranych statków na planszy i dołosowanie pozostałych za pomocą przycisku „Rand craft”

Koniec rozgrywki:

Gdy zostaną zniszczone wszystkie statki danego przeciwnika, pod planszami jest wyświetlana informacja, który z graczy wygrał oraz wyświetlane są wszystkie niezniszczone statki wygranego gracza.



Ustawienia



Za pomocą przycisku „+” lub „-” możemy zwiększyć/zmniejszyć ilość danego statku (domyślnie jest ustawione 10 statków jak w przypadku standardowych zasad gry).

4. Opis kodu

Nazwa	Opis
<i>GridField</i>	Bazowa abstrakcyjna klasa służąca do tworzenia obiektów posiadających: współrzędne, szerokość, wysokość, texture, detekcja kliknięcia myszą.
<i>Field</i>	Klasa dziedzicząca po klasie GridField. Jest to klasa obsługująca pojedyncze pole na planszy(Board)
<i>Button</i>	Klasa dziedzicząca po klasie GridField. Obsługa przycisków.
<i>Board</i>	Bazowa abstrakcyjna klasa dostarczająca niezbędne elementy do utworzenia planszy
<i>BoardPlayer</i>	Klasa dziedzicząca po klasie Board. Obsługa planszy gracza
<i>BoardAi</i>	Klasa dziedzicząca po klasie Board. Zawiera mechanikę gry komputera
<i>Craft</i>	Klasa reprezentująca statek
<i>CraftType</i>	Enum zawierający typy pól: zeroMasthed, oneMasthed, twoMasthed, threeMasthed, fourMasthed, forbid
<i>Direction</i>	Enum zawierający możliwe kierunki rozstawienia statków: N, S, E, W
<i>Orientation</i>	Enum zawierający oientację statku: Vertical/Horizontal
<i>GameMode</i>	Enum zawierający tryby gry: OneVsOne, OneVsAi
<i>GameSettings</i>	Klasa zawierająca tymczasowo wybrane plansze i tryb gry
<i>Game</i>	Główna klasa programu zawierająca metody startowe i główną pętlę gry
<i>IScreen</i>	Abstrakcyjna klasa do tworzenia nowych widoków, zawiera metody wirtualne: handleInput, update, render
<i>StartScreen</i>	Interfejs ekranu głównego gry
<i>SelectCraftWindow</i>	Interfejs ekranu wyboru statków
<i>GameScreen</i>	Interfejs obsługi rozgrywki
<i>Settings</i>	Klasa przechowująca ustawienia gry, ilość statków, otaczanie pola
<i>SettingsScreen</i>	Interfejs ekranu ustawień
<i>CheckboxBtn</i>	Klasa pochodna klasy GridField, reprezentuje przycisk checkbox
<i>IncDecBtn</i>	Klasa tworząca komponent 2 przycisków do zwiększania/zmniejszania licznika
<i>Text</i>	Klasa pomocnicza do wyświetlania tekstu

GridField:

	GridField() Domyślny konstruktor
virtual void	setCoordinate(int x, int y) ustawienie współrzędnych obiektu
std::vector<int>	getCoordinate() const Pobranie aktualnych współrzędnych
virtual int	getCoordinateX() const Pobranie aktualnej współrzędnej x
virtual int	getCoordinateY() const Pobranie aktualnej współrzędnej y
void	setWidth(float width) ustawienie szerokości obiektu
float	getWidth() pobranie szerokości obiektu
void	setHeight(float height) ustawienie wysokości obiektu
float	getHeight() pobranie szerokości obiektu
bool	onClick(float mouseX, float mouseY) detekcja kliknięcia na obiekt, zwraca true jeżeli kliknięto, false w przeciwnym wypadku
void	loadTexture(const std::string& location) wczytywanie textury obiektu
void	setSprite(const sf::Texture& texture) ustawienie sprite obiektu na podstawie tekstury
sf::Sprite	getSprite() const zwraca sprite obiektu

Field:

	Field() Domyślny konstruktor
void	setCoordinate(int x, int y) override ustawienie współrzędnych obiektu
int	getCoordinateX() const override Pobranie aktualnej współrzędnej x
int	getCoordinateY() const override Pobranie aktualnej współrzędnej y
void	setType(const CraftType type) ustawienie typu pola
void	setChecked(bool checked) ustawienie, czy dany obiekt został kliknięty true/false
bool	getChecked() const zwraca informację o zaznaczeniu pola, true/false
void	setSpaceBetweenField(int spaceBetweenField) ustawia margines pomiędzy polami
int	getSpaceBetweenField() const zwraca jaki jest margines pomiędzy polami
bool	hitCraft() const sprawdza, czy kliknięte pole na planszy jest częścią statku, jeżeli tak zwraca true

void	rotateField(const sf::Texture& texture) metoda zmieniająca statek z pozycji poziomej na pionową i odwrotnie
Orientation	getPosition() const zwraca aktualną orientację pola, Vertical/Horizontal

Board:

	Board() Domyślny konstruktor
void	resetBoard() ustawienie wyczyszczenie i ustawienie domyślnych ustawień planszy
void	setDimensionX(int dimensionX) ustawienie wymiarów osi x planszy(wymiary tablicy dwuwymiarowej)
int	getDimensionX() const Pobranie wymiaru x tablicy dwuwymiarowej
void	setDimensionY(int dimensionY) ustawienie wymiarów osi y planszy(wymiaru tablicy dwuwymiarowej)
int	getDimensionY() const Pobranie wymiaru y tablicy dwuwymiarowej
std::array<std::array<Field, 10>, 10>	getFieldTab() const zwraca tablice 2 wymiarową z polami planszy
void	addCraft(const Craft& craft) dodanie statku do tablicy
std::vector<Craft>	getCraftTab() const pobranie tablicy z statkami
void	updateCraftTab(int index, const Craft& craft) zaaktualizowanie tablicy z statkami
void	setClickedField(const Field& clickedField) ustawienie aktualnie klikniętego pola na planszy w tymczasowej zmiennej
Field	getClickedField() const pobranie pola, które zostało kliknięte
void	updateTabEl(const Field& field) zaaktualizowanie jednego element tablicy
void	renderBoard(sf::RenderWindow& mWindow, bool showCraft=false, bool showForbidArea=false) const wyrenderowanie planszy na ekranie
void	randomCraft(int type, int quantity) wylosowanie wolnych miejsc dla danego typu statku i ilości na planszy, ustawienie na planszy statków
bool	checkCraftIsDestroyed(const Field& field) sprawdzenie, czy dany statek został zniszczony oraz zaznaczenie na mapie obszaru zabronionego
void	tickForbidArea(const Craft& craft) zaznaczenie obszaru otaczającego statek
void	checkNSDirection(int rowIndex, int colIndex, int type, bool& allowCraft, int direction)

	Sprawdzenie, czy można ustawić statek w pionie w podanym kierunku N lub S
void	checkVertical(int rowIndex, int colIndex, int type, bool& allowCraft, Direction& allowedDirection) sprawdzenie, czy można rozstawić statek w kierunku N lub S
void	checkWEDirection(int rowIndex, int colIndex, int type, bool& allowCraft, int direction) sprawdzenie, czy można ustawić statek w poziomie w podanym kierunku W lub S
void	setCraftOnMap(const Direction& allowedDirection, Craft& craftModel, int craftIndex) ustawienie statku na planszy
void	handleInput(const sf::RenderWindow& window) obsługa kliknięć w plansze
int	getBoardStats() const zwraca ilość zniszczonych elementów statku na planszy
void	delCraft(int craftIndex) usunięcie statku z planszy
void	setCraftTab(const sf::Texture& fourMTexture, const sf::Texture& threeMTexture, const sf::Texture& twoMTexture, const sf::Texture& oneMTexture) Ustawienie tablicy z statkami o podanych teksturach statków

BoardAi:

	BoardAi() Domyślny konstruktor
Field	action(std::array<std::array<Field, 10>, 10> fieldTab, std::vector<Craft> craftTab) logika gry komputera, strzela w pole planszy i zwraca to pole
Field	getAvailableField(const std::array<std::array<Field, 10>, 10>& fieldTab) zwraca dostępne pola do zaznaczenia na planszy
void	delForbidAvailableField(const std::vector<Field>& forbidArea) usunięcie z tablicy dostępnych pól do wyboru, elementów które otaczają statek
int	getIndexAvailableField(int coordinateX, int coordinateY) zwraca index do pola z tablicy dostępnych pól na podstawie współrzędnych
void	clearHitCraftTab() wyczyszczenie tablicy z elementami częściowo zniszczonego statka

BoardPlayer:

void	tickField(Field& field) zaznaczenie pola na planszy
-------------	---

Craft:

void	Craft() Domyślny konstruktor
std::vector<Field>	getArea() const pobranie pól statku
std::vector<Field>	getForbidArea() const pobranie obszaru otaczającego statek
void	addField(const Field& field) dodanie elementu statku do obszaru
void	addForbidArea(const Field& field) dodanie pola do tablicy z obszarem otaczającym statek
void	setOrientation(const Orientation& orientation) ustawienie orientacji statku Vertical/Horizontal
Orientation	getOrientation() const pobranie orientacji statku
void	setCraftType(CraftType craftType) ustawienie typu statku
CraftType	getCraftType() const pobranie typu statku
int	getCraftTypeNumber(CraftType craftType) const zwraca w postaci int typ statku
void	setQuantityDestroyedEl() const ustawienie ilości zniszczonych elementów statku
int	getQuantityDestroyedEl() const pobranie ilości zniszczonych elementów statku
void	destroyEl() zwiększenie o jeden ilości zniszczonych elementów statku
Bool	checkStateCraft() const zwraca true, jeżeli statek został zniszczony

IScreen:

void	setBackground(const std::string& location) ustawia tło okna
void	setMousePosition(const sf::RenderWindow& window) ustawia aktualne położenie kursora
virtual void	void handleInput(sf::RenderWindow& window, const sf::Event& event) metoda wirtualna do przechwytywania zdarzeń myszy/klawiatury
virtual void	update(sf::Time deltaTime) metoda wirtualna do aktualizowania stanu danego okna
virtual void	void render(sf::RenderWindow& window) metoda wirtualna renderująca dane elementy na oknie

CheckboxBtn:

	CheckboxBtn() Domyślny konstruktor
void	setChecked(bool isChecked) zaznaczenie przycisku checkboxa
bool	getChecked() const zwraca stan przycisku true/false
void	checkIsChecked() metoda sprawdza, czy checkbox jest zaznaczony/odznaczony i podmienia textury
void	updateCoordinate() aktualizuje pozycję komponentu graficznego

Settings:

bool	getShowForbidArea() const; metoda zwraca, czy w trakcie rozgrywki po zniszczeniu statku ma zaznaczyć otoczkę w okół statku
void	setShowForbidArea(bool showForbidArea) metoda ustawia, czy statki po zniszczeniu mają zostać otoczone
int	getQuantityFiveMasthed() const metoda zwraca ilość 5-masztowców
int	getQuantityFourMasthed() const metoda zwraca ilość 4-masztowców
int	getQuantityThreeMasthed() const metoda zwraca ilość 3-masztowców
int	getQuantityTwoMasthed() const metoda zwraca ilość 2-masztowców
int	getQuantityOneMasthed() const metoda zwraca ilość 1-masztowców
int	getCurrentCraftField() const zwraca ilość obecnie zajętych pól na planszy
void	incDecQuantityCraft(CraftType craftType, int number) metoda zwiększająca/zmniejszająca ilość podanego statku o podaną ilość
int	getMaxOneMasthed() const metoda zwraca ile można maksymalnie postawić 1-masztowców na mapie
int	getMaxFiveMasthed() const metoda zwraca ile można maksymalnie postawić 5-masztowców na mapie
int	getQuantityCraft() const metoda zwracająca ilość statków