

# Raport Programowanie Obiektowe

## Gra Break the Brick

### Wstęp:

Głównym założeniem projektowym jest zapoznanie się z podejściem obiektowym w programowaniu, czyli np. tworzenie klas, metod oraz obiektów, polimorfizm. Przydatne okaże się również dziedziczenie.

Projekt będzie realizował grę typu Arkanoid o nazwie „Break the Brick”. Do realizacji projektu posłużono się narzędziem VS 2019, projekt będzie pisany w języku C++, a interfejs graficzny stworzony zostanie za pomocą zewnętrznej biblioteki do grafiki 2D którą jest SFML.

### 1. Opis projektu:

Tworzony projekt realizuje prosty schemat gry Arkanoid. Na początku wyświetli się proste menu ułatwiające poruszanie się po grze (wyświetlona zostanie instrukcja obsługi, otwarcie lub zamknięcie okna).

Głównym zadaniem gry jest pozbycie się wszystkich klocków obecnych na planszy, odbijając kulę od paletki, którą porusza się za pomocą klawiszy PRAWO i LEWO na klawiaturze. Kulka ze stałą prędkością porusza się po okienku, wykrywając kolizje ze ścianami oraz bloczkami (wtedy je niszczy). Klasy dziedziczą z klasy sf::Drawable.

Stworzono szkielet większości klas i zaczęto rozważać poszczególne relacje między nimi.

### 2. Struktura programu

Główne klasy programu (do wszystkich obiektów użyto biblioteki SFML):

✚ **Class Kulka** – odpowiedzialna za ruch i implementację kulki. Nadanie jej kształtu okręgu oraz prędkości. Za kolizję kulki ze ścianami okna gry.

✚ **Class Kloc** – odpowiedzialna za wygenerowanie klocka, nadanie mu kształtu prostokąta. Zostały zastosowane metody do zniszczenia klocków. Później w funkcji main() zostaną wygenerowane rzędy i kolumny klocków.

✚ **Class Paletka** – wygenerowanie paletki, nadanie jej kształtu prostokąta, nadanie mu wymiarów oraz prędkości.

Dodatkowo stworzony został plik nagłówkowy **Scena.h**, dzięki któremu wyświetli się menu.

✚ Plik, w którym mieści się funkcja `main()` nosi nazwę **brick\_and\_ball.cpp**. Tutaj zostały wypisane funkcje odpowiedzialne za kolizje między poszczególnymi obiektami, aby później wywołać je w `main()`. W samej funkcji `main()` dzięki bibliotece graficznej SFML zostało stworzone okno gry wraz z liczbą klatek na sekundę (60). Warto dodać, że stworzone zostało tło imitujące niebo, dzięki zaimplementowaniu `Sprite`. Okno gry można powiększać i pomniejszać oraz nim poruszać dzięki poleceniu „`okno.pollEvent(wydarzenie)`”. Zastosowano pętlę `if`, aby móc wyjść z gry (dwie różne możliwości wyjścia). Wywołano obiekty, umożliwiono poruszanie się im. Opisano iterator odpowiedzialny za zniszczone klocki.

W klasie `Kloc` została wprowadzona metoda wirualna, odpowiedzialna za rysowanie klocka.

```
virtual void draw(sf::RenderTarget& target, sf::RenderStates state) const;
```

### 3. Krótką instrukcja użytkownika:

Po uruchomieniu programu wyświetli się menu, informujące o możliwych opcjach dalszych działań. Jeśli wciśniemy liczbę „2”, pojawi się instrukcja użytkownika. Po wciśnięciu „3”, program zakończy się. Gdy wciśnięta zostanie liczba „1”, uruchomi się właściwa gra. Kulka zawsze będzie lecieć ze środka. Aby poruszać paletką

- w prawo - należy wcisnąć prawą strzałkę na klawiaturze
- lewo - należy wcisnąć lewą strzałkę na klawiaturze

Trzeba tak kierować paletkę, aby uderzyła w nią piłka i rozbiła wszystkie klocki obecne na ekranie. Okno gry można powiększać i pomniejszać oraz nim poruszać. Aby wyjść z gry, należy wcisnąć klawisz `Esc` lub przycisk `X` w okienku. Wtedy w konsoli pojawi się napis mówiący, na którym levelu zakończyło się grę. Poniżej zrzut ekranu działającej gry.



