

Politechnika Śląska w Gliwicach
Wydział Informatyki, Elektroniki i Informatyki



Programowanie Komputerów

Arkanoid

autor	Sławomir Krzykała
prowadzący	dr inz. Tomasz Moroń
rok akademicki	2019/2020
kierunek	teleinformatyka
rodzaj studiów	SSI
semestr	3
grupa	1

Spis treści

1. Opis projektu	3
1.1. Cel projektu	3
1.2. Analiza problemu projektu	3
2. Klasy	3
2.1. Diagram klas przestrzeni nazw game – logika gry	3
2.2. Opis klas przestrzeni nazw game – logika gry	4
2.3. Diagram klas przestrzeni nazw display - wyświetlanie	5
2.4. Opis klas przestrzeni nazw display – wyświetlanie	5
3. Specyfikacja zewnętrzna	5
3.1. Opis działania, instrukcja	5
3.2. Wygląd planszy poszczególnych poziomów	7
3.2.1. Poziom 1	7
3.2.2. Poziom 2	7
3.2.3. Poziom 3	8
3.2.4. Poziom 4	8
3.2.5. Poziom 5	9
3.2.6. Poziom 6	9
4. Dokumentacja wewnętrzna	10
5. Podsumowanie	10

1. Opis projektu

1.1. Cel projektu

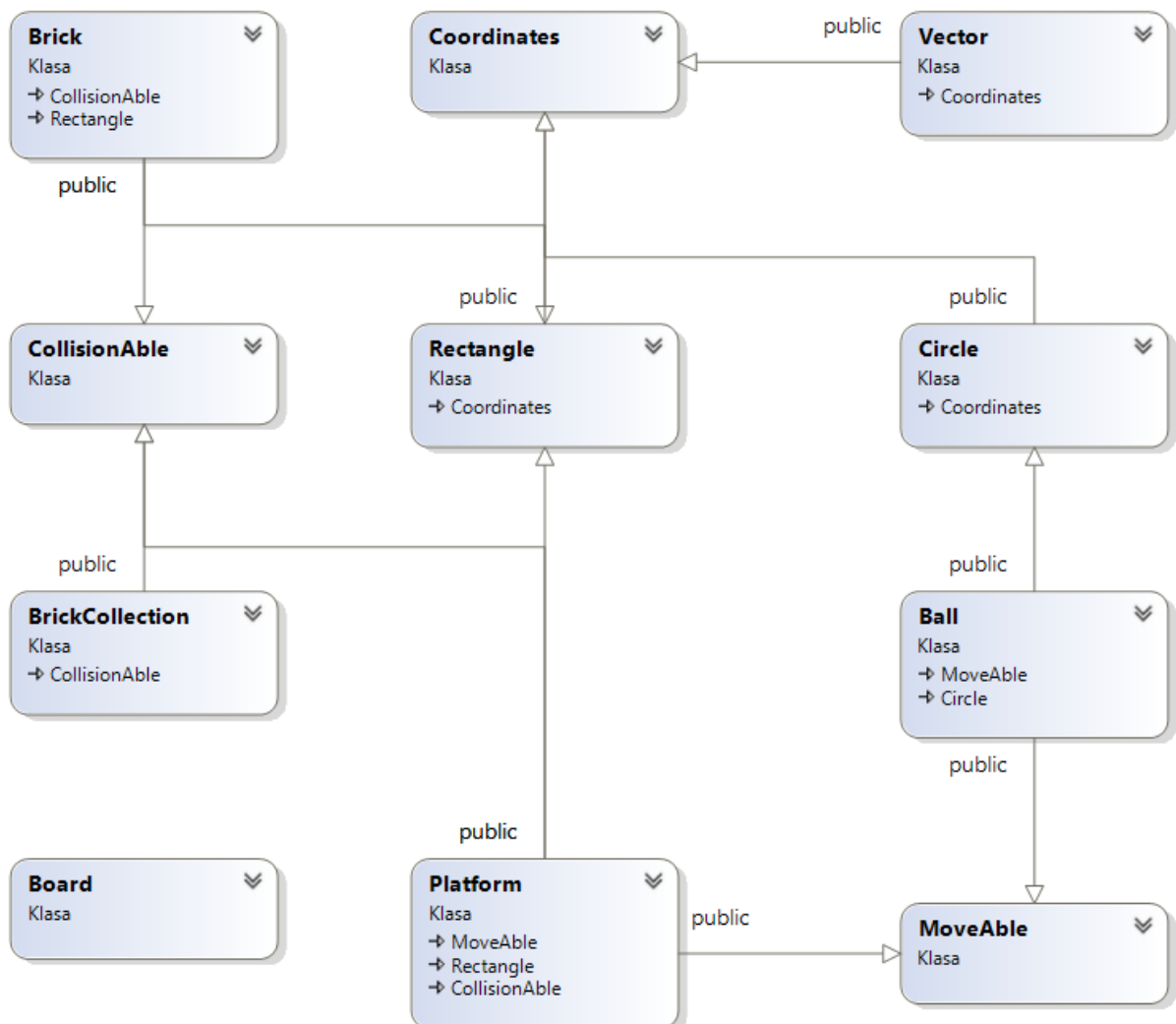
Celem projektu było stworzenie gry typu Arkanoid, uwzględniając wytyczne. Opis jest dołączony do sprawozdania w postaci prezentacji multimedialnej. Do stworzenia gry została wykorzystana biblioteka SFML.

1.2. Analiza problemu projektu

Aplikacja symulująca swobodny ruch piłki podczas którego następują zderzenia (odbicia) z klockami, platformą sterowaną przez gracza za pomocą klawiatury, oraz brzegami okna aplikacji (lewego, prawego oraz górnego). Jeżeli nieodbita za pomocą platformy piłka przemieści się do dolnego brzegu okna aplikacji gracz traci życie w grze. Jeżeli piłka odbita piłka uderzy w klocek, klocek znika a gracz dostaje punkty. Ułożenie klocków może być dowolne.

2. Klasy

2.1. Diagram klas przestrzeni nazw game – logika gry



2.2. Opis klas przestrzeni nazw game – logika gry

Coordinates reprezentuje punkt na płaszczyźnie/planszy. Pośrednio dziedziczą ją wszystkie klasy reprezentujące elementy gry (piłka, klocki, platforma) jako ich środek. Posiada właściwość x, y.

Vector reprezentuje kierunek przemieszczania się obiektu, oraz kierunek odbicia przy kolizji obiektów.

Rectangle reprezentuje prostokąt, dziedziczą ją klasy reprezentujące platformę oraz klocek. Posiada właściwości width – szerokość; height – wysokość.

Circle reprezentuje okrąg, dziedziczy ją klasa reprezentująca piłkę. (W przyszłości można dodać spadające bonusy w kształcie koła). Posiada właściwość r – promień.

Brick reprezentuje pojedynczy klocek. Posiada właściwość color – kolor klocka.

BrickCollection reprezentuje kolekcję wszystkich klocków. Posiada właściwość listę collection – lista klocków.

Platform reprezentuje platformę.

Ball reprezentuje piłkę. Posiada właściwość damage – obrażenia.

MoveAble klasa abstrakcyjna (posiada metodę czysto wirtualną), reprezentująca możliwość przemieszczania się obiektu. Dziedziczona przez Platform oraz Ball. Posiada właściwości direction – kierunek przemieszczania; speed – szybkość.

CollisionAble klasa abstrakcyjna (posiada metodę czysto wirtualną), reprezentująca możliwość kolizji z piłką. Dziedziczona przez Platform oraz Brick/BrickCollection.

Board reprezentuje całą planszę rozgrywki, posiada właściwości game_ball, default_ball – piłka w grze/domyślna; game_platform, default_platform – platforma w grze/domyślna; brick_collection - kolekcja klocków; begin_board, end_board – początek/koniec planszy; clock - zegar.

Player reprezentuje gracza, posiada właściwości level – poziom; score – punkty; life – życia; name – nazwa; fingerPrint – identyfikator; version – wersja.

2.3. Diagram klas przestrzeni nazw display - wyświetlanie



2.4. Opis klas przestrzeni nazw display – wyświetlanie

Board reprezentuje wyświetlaną planszę gry, posiada właściwości platform, ball, bricksCollection które są stałymi referencjami do stałych obiektów które im odpowiadają w klasie game::board; bricks_visible – mapa prostokątów z SFML reprezentująca wyświetlane klocki; ball_visible, platform_visible - wyświetlana piłka/platforma.

Background reprezentuje tło planszy wczytywane z pliku graficznego.
Dziedziczy po sf::Sprite z SFML.

TextLabel reprezentuje wyniki/statystyki wyświetlane po lewej u góry podczas gry.

TextCenter reprezentuje tekst wyświetlany na środku ekranu – instrukcja, tekst zakończenia gry.

3. Specyfikacja zewnętrzna

3.1. Opis działania, instrukcja

Po uruchomieniu aplikacji wyświetla się instrukcja:

*„Odbijaj piłkę aby zbić wszystkie klocki.
Z każdym następnym poziomem piłka przyspiesza!*

Sterowanie:

[spacja] – start

[strzałki]/[A]/[D] - ruch platformy

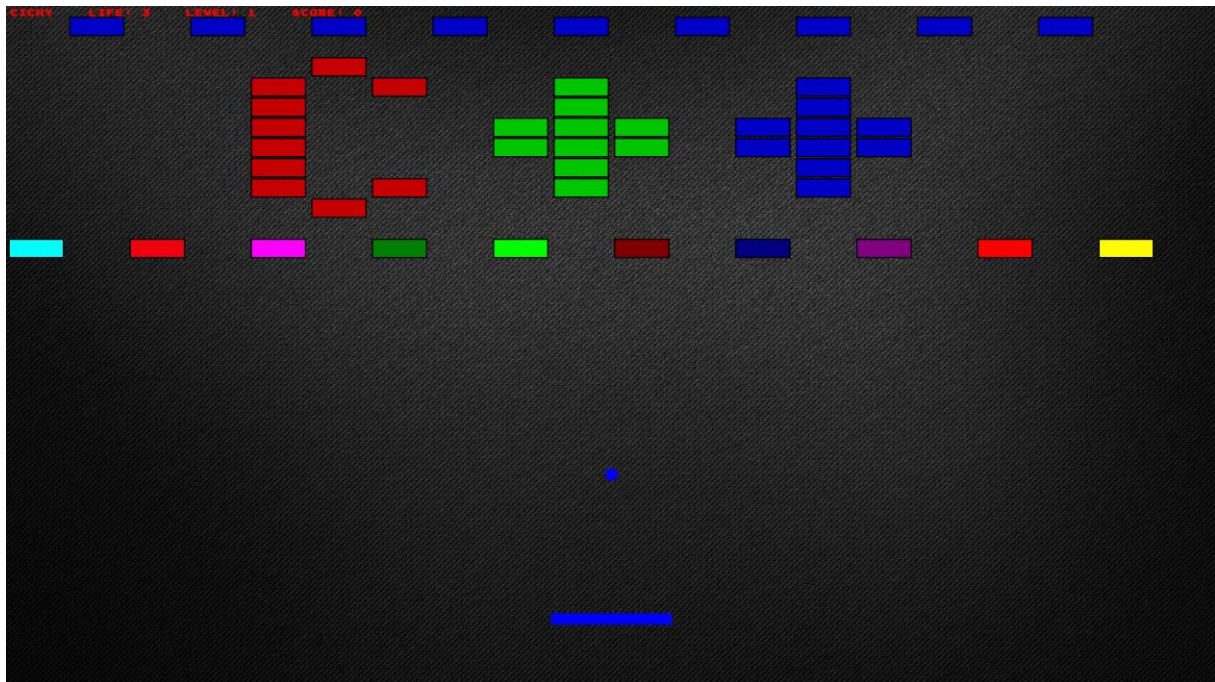
[H] – pomoc

[P] – pauza

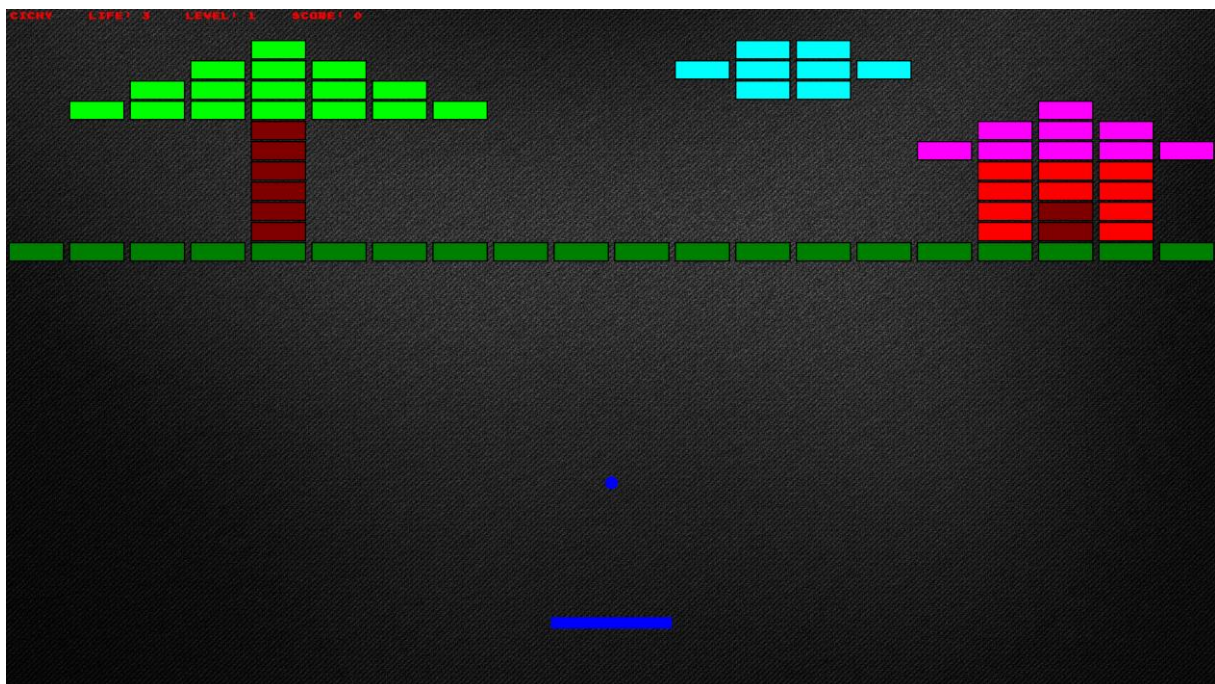
[esc] – wyjście”

3.2. Wygląd planszy poszczególnych poziomów

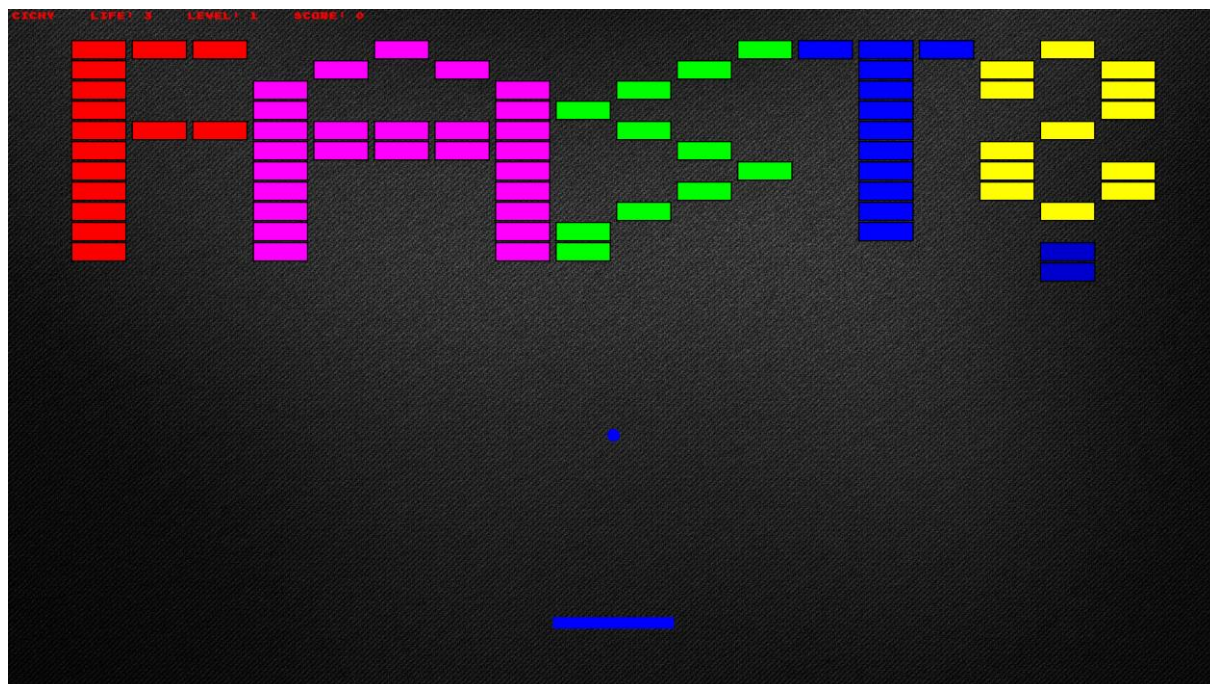
3.2.1. Poziom 1



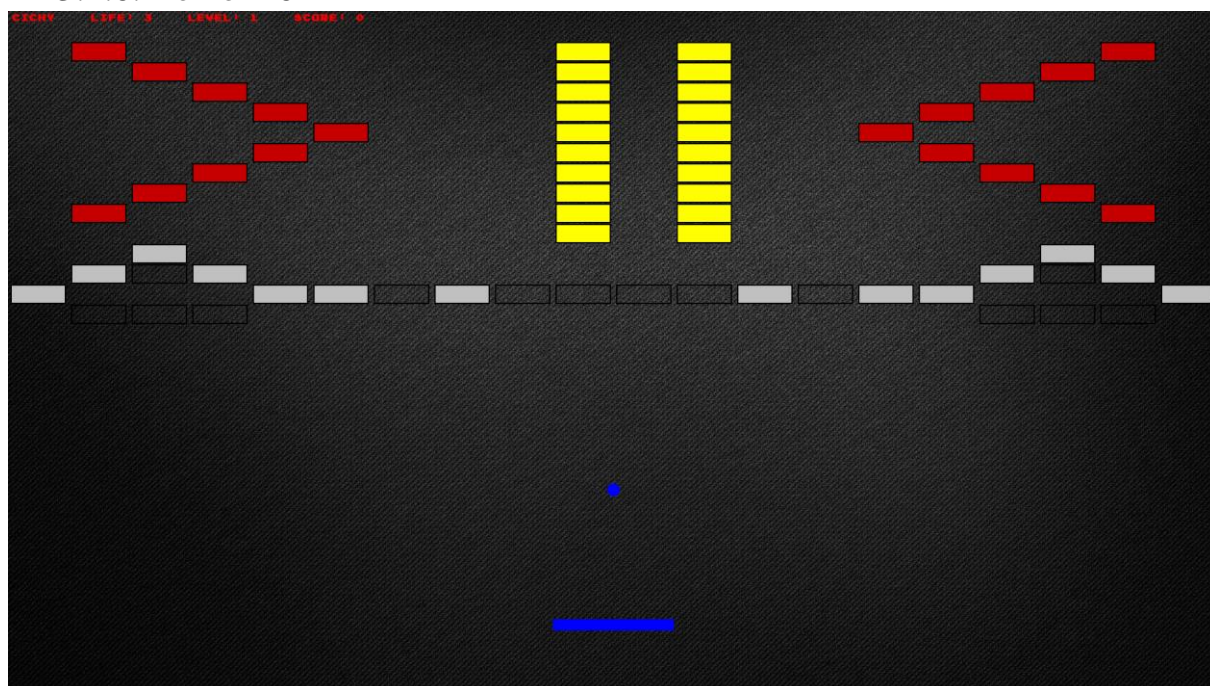
3.2.2. Poziom 2



3.2.5. Poziom 5



3.2.6. Poziom 6



4.Dokumentacja wewnętrzna

Dokumentacja wewnętrzna została wygenerowana za pomocą narzędzia Doxygen i jest dołączona do sprawozdania.

5.Podsumowanie

Dzięki projektowi mogłem praktycznie wykorzystać poznane elementy języka C++. Poznałem również bibliotekę SFML, na podstawie której wykonałem elementy programu odpowiedzialne za interfejs użytkownika. Biblioteka SFML po zapoznaniu się z dokumentacją wydaje się być bardzo przyjazna oraz dosyć łatwa w użyciu.