Politechnika Świętokrzyska w Kielcach Wydział Elektrotechniki, Automatyki i Informatyki	
Grafika komputerowa - projekt Informatyka - III rok, Rok akademicki - 2023/2024	
Temat Projektu: **„Piłka nożna głowami 2D”**	Członkowie zespołu: **Krzysztof Łęgowik** **Maksymilian Sowula** **Jakub Szczur** **gr. 3ID12A [Systemy informacyjne]**

Podczas prac nad projektem wykonano zadania zawarte w instrukcjach laboratoryjnych:

W ramach instrukcji laboratoryjnej nr 0:
1. Zapoznano się z kartą przedmiotu w katalogu studiów oraz zasadami zaliczenia zajęć.
2. Utworzono zespół do tworzenia projektu.
3. Wybrano technologię tworzenia silnika: SFML oraz zapoznano się z jej dokumentacją.

W ramach instrukcji laboratoryjnej nr 1 zaimplementowano w projekcie:
1. Utworzono projekt szkieletu silnika 2D
2. Zaimplementowano klasę Engine, którą posiada funkcjonalności:
   - zparametryzowano tryb graficzny jako wybór rozdzielczości
   - zparametryzowano liczbę klatek animacji na sekundę,
   - uruchomiono obsługę myszy/klawiatury
   - zaimplementowano czasomierz w głównej pętli
   - zaimplementowano obsługę czyszczenia ekranu do bitmapy
   - dodano obsługę błędów z możliwością wyświetlania i logowania do pliku komunikatów diagnostycznych,
   - zaimplementowano zamknięcie gry

W ramach instrukcji laboratoryjnej nr 2 zaimplementowano w projekcie:
1. Wielokrotne buforowanie wykorzystywane przez klasę Engine
2. Klasę PrimitiveRenderer, która ma funkcjonalność rysowania prymitywów
3. Rozszerzono funkcjonalność klasy PrimitiveRenderer o metodę rysującą odcinek, która wykorzystuje algorytm przyrostowy
4. Klasę Point2D reprezentującą współrzędne punktu w przestrzeni 2D
5. Klasę LineSegment reprezentującą odcinek.
6. Rozszerzono funkcjonalność klasy PrimitiveRenderer o metodę (lub kilka metod) umożliwiającą narysowanie linii łamanej otwartej lub zamkniętej

W ramach instrukcji laboratoryjnej nr 3 zaimplementowano w projekcie:
1. Rozszerzono funkcjonalność klasy PrimitiveRenderer o kolejną metodę rysującą okrąg, która wykorzystuje algorytm rysowania okręgu w oparciu o 4-krotną symetrię.
2. Rozszerz funkcjonalność klasy PrimitiveRenderer o kolejną metodę rysującą elipsę, która wykorzystuje algorytm rysowania elipsy w oparciu o 4-krotną symetrię.
3, Rozszerzono funkcjonalność klasy PrimitiveRenderer o metodę umożliwiającą narysowanie dowolnego wielokąta (łamanej zwyczajnej zamkniętej)
4. Rozszerzono klasę PrimitiveRenderer o możliwość rysowania prymitywów wypełnionych kolorem
5. Rozszerzono funkcjonalność klasy PrimitiveRenderer o metody umożliwiające wypełnienie obszaru kolorem przez spójność. Jedna metoda implementuje algorytm border fill, a druga flood fill
6. Rozszerzono funkcjonalność klasy PrimitiveRenderer o metodę umożliwiającą narysowanie dowolnego wielokąta wypełnionego kolorem. Zastosowano algorytm wypełniania obszaru kolorem z kontrolą parzystości

W ramach instrukcji laboratoryjnej nr 4 zaimplementowano w projekcie:
1. Hierarchię klas dla różnych obiektów silnika gry
   - Klasa znajdująca się najwyżej w hierarchii ma nazwę GameObject
   - Klasa bazowa dla wszystkich obiektów gry, które mogą być w pewien sposób aktualizowane, jest klasą UpdatableObject. Implementuje metodę Update()
   - Klasa bazowa dla wszystkich obiektów gry, które mogą zostać narysowane na ekranie, jest klasą DrawableObject. Implementuje metodę draw()
   - Klasa bazowa dla wszystkich obiektów gry, które mogą podlegać przekształceniom geometrycznym, jest klasą TransformableObject. Jej interfejs zawiera abstrakcyjne metody translate(), rotate() i scale().
   - Klasa bazowa dla wszystkich obiektów gry, które mogą zostać narysowane z użyciem prymitywów, jest klasą ShapeObject
2. Rozszerzono funkcjonalność klasy Point2D i klasy LineSegment o możliwość wykonywania na tych prymitywach przekształceń geometrycznych opisanych w instrukcji.
3. Zaimplementowano klasę Player reprezentującą gracza. Umożliwia sterowanie graczem z użyciem klawiatury i/lub myszy

W ramach instrukcji laboratoryjnej nr 5 zaimplementowano w projekcie:
1. Klasę BitmapHandler, obsługującą podstawową funkcjonalność związaną z bitmapami
2. Rozszerzono opracowaną na zajęciach hierarchię klas silnika gry
   - Klasą bazową dla wszystkich obiektów gry, które mogą być w pewien sposób animowane, nazwano AnimatedObject. Jej interfejs implementuje metodę animate()
   - Klasą bazową dla wszystkich obiektów gry, które mogą zostać narysowane z użyciem bitmap, nazwano BitmapObject. Klasa ta dziedziczy po DrawableObject. Uwzględnij że obiekt gry składa się z więcej niż jednej bitmapy. Zaimplementowano dla wirtualnej metody draw().

- Klasą bazową dla wszystkich obiektów gry, które mogą zostać narysowane z użyciem bitmap, i które mogą być animowane na zasadzie wykorzystania sprite'ów, nazwano SpriteObject. Klasa ta dziedziczy po BitmapObject i AnimatedObject.

3. Zmodyfikowano klasę Player, tak aby gracz był renderowany a jego ruch animowany z użyciem bitmap (sprite'ów). W tym celu wprowadzono dziedziczenie tej klasy po SpriteObject. Stworzono sprite'y

<u>W ramach instrukcji laboratoryjnej nr 6 zaimplementowano w projekcie:</u>

1. Zaimplementowano demo technologiczne prezentujące funkcjonalności opracowanego silnika gry 2D, które umożliwiają przetestowanie poszczególnych jego funkcji, automatycznie i przy interakcji użytkownika.
2. Wykonano dokumentację techniczną rezultatów pracy z pierwszej części semestru, w skład której wchodzą
   - Sprawozdanie w formie PDF
   - Dokumentacja kodu źródłowego opracowana z wykorzystaniem narzędzia Doxygen
3. Wysłano całość rezultatów pracy na Platforme Moodle w aktywności SILNIK 2D w terminie wskazanym przez prowadzącego.