

# **Laboratorium Programowania Komputerów 4**

## **Opis klas projektu Snake**

Autor : Michał Wieczorek

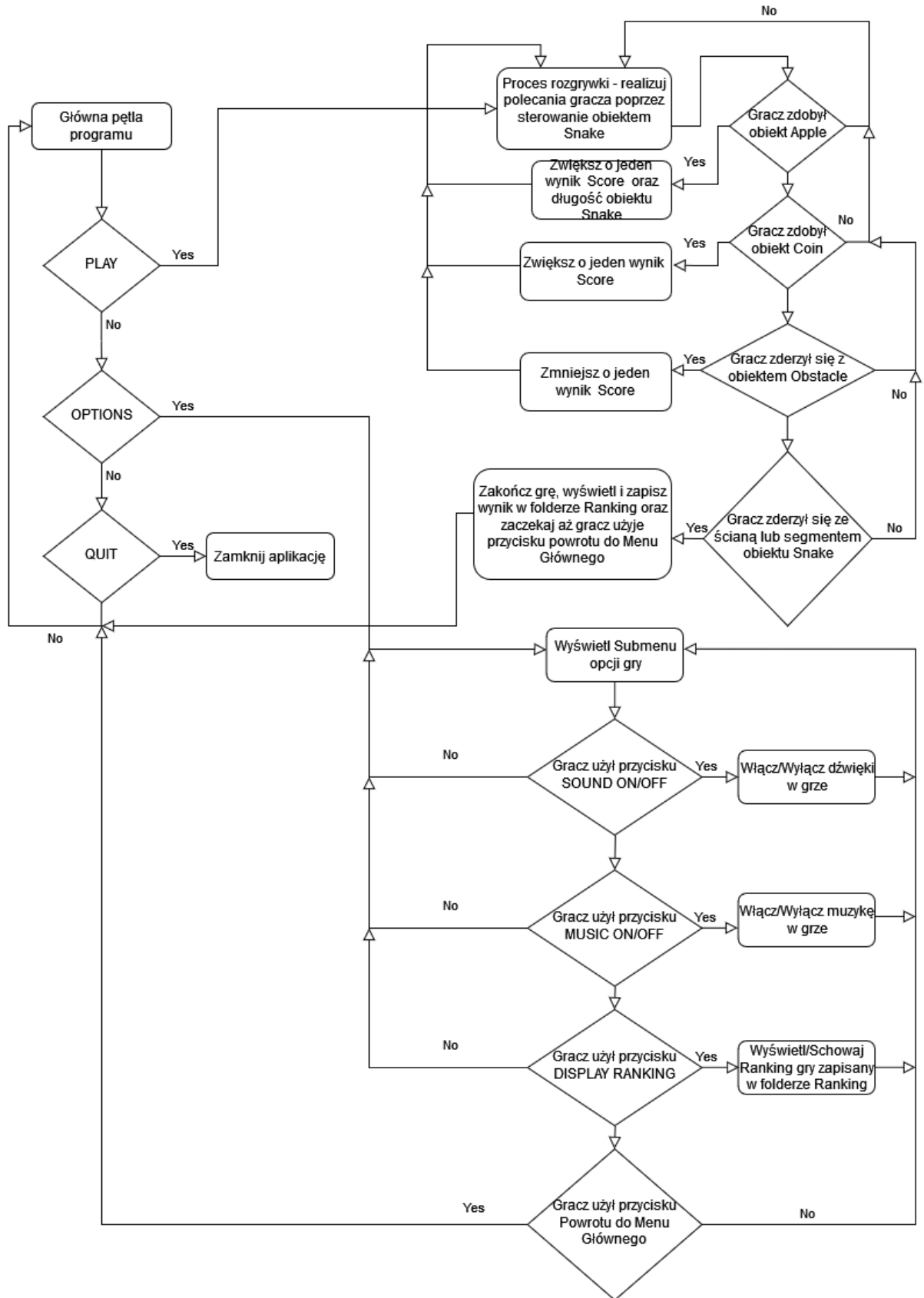
Grupa dziekańska : 2

Sekcja : 4

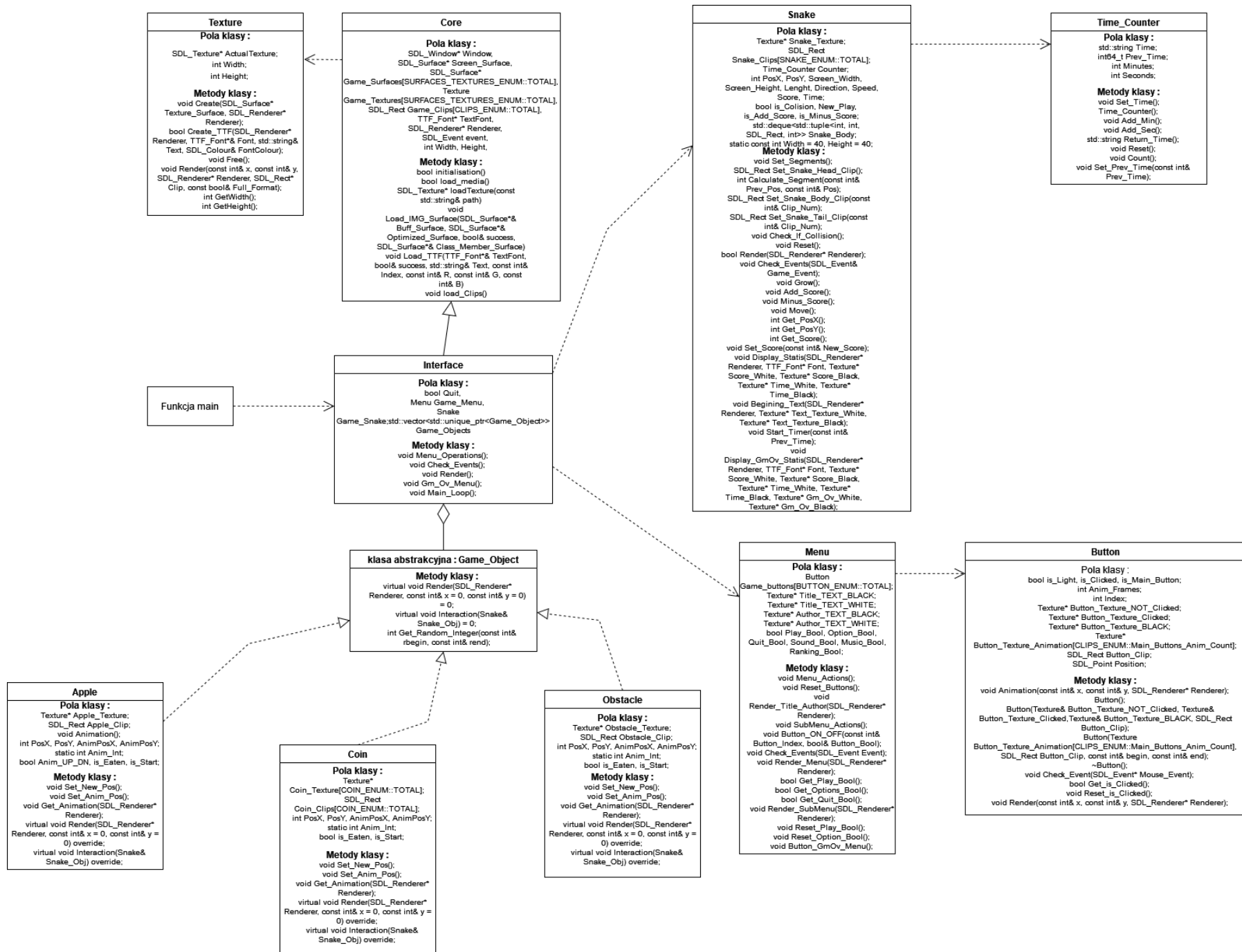
Prowadzący : dr. Inż. Krzysztof Taborek

Data : 10.05.2023 r.

## 1. Ogólny schemat blokowy aplikacji



## 2. Diagram klas (na następnej stronie) – utworzony zgodnie ze standardem UML



## Opis klas :

- **Core** - klasa odpowiedzialna za przechowywanie i obsługę wszystkich plików zewnętrznych programu takich jak pliki graficzne (.PNG) lub dźwiękowe (na ten moment jeszcze nie zaimplementowane). Inicjalizują również wszystkie biblioteki zewnętrzne SDL tj. SDL2, SDL\_ttf, SDL\_mixer, SDL\_image.
- **Interface** – jest to jedyna klasa, której instancja jest tworzona w funkcji main. Cały program funkcjonuje na jej bazie. Wszystkie inne klasy są z nią powiązane w większy lub mniejszy sposób zgodnie z wzorcem projektowym Kompozyt.
- **Texture** – klasa tworząca i obsługująca tekstury gry. Za jej pośrednictwem renderowany jest każdy obiekt w grze.
- **Snake** – obiekt gry, którym gracz steruje strzałkami w momencie gdy gra jest uruchomiona (po włączeniu gry przyciskiem „PLAY”). Jest najbardziej rozbudowanym obiektem w programie. Za pośrednictwem poszczególnych metod spełnia założenia gry tzn. :
  - w momencie dotknięcia głową swojego segmentu lub ściany (końca ekranu) kończy grę.
  - w przypadku zjedzenia/zdobycia obiektu Apple zwiększa licznik punktów Score oraz swoją długość (dodaje dodatkowy segment).
  - w przypadku zjedzenia/zdobycia obiektu Coin zwiększa licznik punktów Score.
  - jeśli „uderzy” głową obiekt Obstacle (przeszkoda) odejmuje punkt od licznika Score.Klasa obsługuje licznik Score oraz Time bieżącej rozgrywki. Jest ściśle powiązana z klasą Time\_Counter, dzięki której jest w stanie mierzyć czas rozgrywki.
- **Time\_Counter** – klasa odpowiedzialna za mierzenie czasu podczas rozgrywki oraz udostępnienie jej wyniku w klasie **Snake** w postaci zmiennej std::string.
- **Menu** – klasa, której zadaniem jest obsługa menu głównego oraz submenu opcji gry. Jest powiązana z klasą **Button** poprzez przechowywanie oraz zarządzaniem instancjami tych obiektów.
- **Button** – klasa powiązana z klasą **Menu**. Jest implementacją przycisków w grze. Zapewnia ich kilka rodzajów tj. przycisku z menu głównego (niebieskie posiadające animację z kilkudziesięciu tekstur), przyciski z submenu opcji (posiadające prostą animację opartą na różnych kolorach czcionek TTF oraz napisów).
- **Game\_Object** – jest to klasa abstrakcyjna, której metodą są przesłanianie przez klasy **Apple**, **Coin** oraz **Obstacle**. Została utworzona zgodnie z ideą wzorca projektowego Fabryka Abstrakcyjna. Posiada również metodę Get\_Random\_Integer(...) używaną przez klasy podrzędne do tworzenia animacji oraz renderowania w losowych miejscach na mapie w przypadku kontaktu z obiektem **Snake**.
- **Apple** – klasa reprezentująca jabłko w grze, które zapewnia wydłużenie obiektu Snake, sterowanego przez gracza, o jeden segment oraz zwiększenie o jeden licznika punktów Score w przypadku kontaktu z obiektem **Snake**. Klasa posiada animację w przypadku kontaktu z obiektem Snake i renderuje się w nowym, losowym miejscu na mapie.
- **Coin** – klasa reprezentująca monetę w grze, które zapewnia zwiększenie o jeden licznika punktów Score w przypadku kontaktu z obiektem **Snake**. Klasa posiada animację w przypadku kontaktu z obiektem Snake i renderuje się w nowym, losowym miejscu na mapie.
- **Obstacle** – klasa reprezentująca przeszkodę (skrzynkę) w grze. W przypadku kontaktu z obiektem **Snake** licznik Score jest dekrementowany o jeden. Klasa posiada animację w przypadku kontaktu z obiektem Snake i renderuje się w nowym, losowym miejscu na mapie.

**\*Aktualnie program jest w trakcie pisania. Niektóre klasy zostaną rozszerzone o nowe funkcjonalności lub mogą zostać zmodyfikowane**

### **3. Opis zagadnień użytych zagadnień omawianych na laboratorium**

- Moduły – w programie zamiast dyrektywy `#include` użyto wyłącznie modułów. Zamiast plików nagłówkowych `.h` utworzono pliki z rozszerzeniem `.ixx`, w których klasy i wszystkie inne komponenty są eksportowane i importowane.
- Biblioteka `Filesystem` – biblioteka zostanie użyta w celu utworzenia folderu „Ranking” w plikach gry, którego celem będzie przechowywanie informacji na temat wyników gier (najprawdopodobniej będą to pliki tekstowe z rozszerzeniem `.txt`) oraz obsługi tych plików. Na obecny moment komponent nie został jeszcze zaimplementowany w programie.

