

Podstawy grafiki komputerwej 1 silnik do gier 2d

Rafał Grot, Kamil Gunia, Piotr Górski

November 21, 2023

Contents

| | | |
|----------|--|----------|
| 1 | Wymagania systemowe | 2 |
| 2 | Użycie bilblioteki engine | 2 |
| 3 | Wybrane klasy wykorzystywanie przez silnik. | 2 |
| 3.1 | GameObject | 2 |
| 3.2 | Drawable | 2 |
| 3.3 | AnimatedObject | 2 |
| 3.4 | AnimatedSpriteSheet | 2 |
| 3.4.1 | Command | 2 |
| 3.4.2 | Przykład | 3 |
| 3.5 | UpdateableObject | 3 |
| 4 | Przykład użycia silnika | 3 |

1 Wymagania systemowe

- SFML
- cmake

2 Użycie biblioteki engine

Aby mieć dostęp do silnika gry należy wykorzystać plik nagłówkowy `engine.hpp`.

3 Wybrane klasy wykorzystywane przez silnik.

3.1 GameObject

GameObject to klasa reprezentująca obiekt w grze.

3.2 Drawable

Drawable jest to abstrakcyjna klasa bazowa wszystkich klas, które można narysować.

3.3 AnimatedObject

Jest to klasa zawierająca metodę wirtualną `animate()`, należy ją przeciążyć w celu realizacji animacji.

3.4 AnimatedSpriteSheet

Klasa reprezentująca obiekty, które są animowane przy użyciu bitmap. Dziedziczy po AnimatedObject.

- Obiekty tej klasy wczytują informacje o animacji z plików.
- Konstruktor przyjmuje ścieżkę do katalogu, który musi zawierać plik `config.txt`

Metadane animacji opisane w pliku konfiguracyjnym mają następujący format:

COMMAND
ARGS

Bezpośrednio za linią z `COMMAND` musi znajdować się linia z argumentami oddzielanym spacją. Linie które mają być traktowane jako komentarze zaczynają się znakiem "#".

3.4.1 Command

Prawidłowe opcje `COMMAND` to:

SPRITESHEET Posiada jeden parametr – ścieżkę relatywną od katalogu, w którym znajduje się plik konfiguracyjny, do pliku zawierającego tablice spritów.

ANIMATION Oznacza ładowanie animacji, kolejne wywołania oznaczają nowe typy animacji. Nie przyjmuje parametrów.

FRAME Zawiera informacje o pojedynczej klatce animacji. Przyjmuje parametry oznaczające kolejno

- pozycje x lewego górnego rogu sprita.
- pozycje y lewego górnego rogu sprita.

- rozmiar w osi x sprita.
- rozmiar w osi y sprita.
- czas trwania klatki

3.4.2 Przykład

```
#Comment lines start with '#'
#SPRITESHEET should be followed with with relative path in the next line
#(from config file directory) to the spritesheet

#COMMENTS CAN NOT be in between INFO
#example:
#SPRITESHEET
##SOMECOMMENT
#filename.png
#
#is not allowed

#<path>
SPRITESHEET
spritesheet.png

#Indicates new animation
ANIMATION
#FRAME represents frame of animation
#it is followed by line containing
#<pos x> <pos y> <size x> <size y> <duration>
FRAME
0 0 100 100 0.5
FRAME
100 0 100 100 0 0.5
```

3.5 UpdateableObject

Obiekty których stan logiczny się zmienia powinny przeciążać wirtualną metodę `update` klasy `UpdateableObject`. Silnik nie wywołuje tej metody, należy zadbać aby była ona wywoływana np. przy użyciu własnej pętli gry.

4 Przykład użycia silnika

Program wyświetlający animację.

```
#include "SFML/Graphics/Color.hpp"
#include "animatedSpriteSheet.hpp"
#include "engine.hpp"
#include <iostream>

namespace G {
std::string basePath = "resources/";
}; // namespace G
```

```
AnimatedSpriteSheet animation(G::basePath + "animation");

int main() {
    Engine::getInstance().setMaxFps(3).setResolution({1000, 1000}).buildWindow();

    animation.setPosition({300, 300});
    animation.setColor(sf::Color::Cyan);

    Engine::getInstance().add(&animation);

    Engine::getInstance().loop();
}
```