Politechnika Śląska

Wydział Automatyki, Elektroniki i Informatyki

PROGRAMOWANIE KOMPUTERÓW 4

SFML_RPG

Autor: Jakub Lachman
Prowadzący: Dr. Anna Gorawska
Rok Akademicki: 2020/2021
Rodzaj Studiów: SSI
Semestr: 4
Termin Laboratoriów: Poniedziałek 9:00-10:30 Wtorek 8:00-9:30
Sekcja: 11

1. Treść Zadania

Napisać za pomocą języka C++ oraz biblioteczki graficznej SFML prostą grę turową, opartą na walce z nieprzerwanie pojawiącymi się przeciwnikami.

2. Analiza Zadania

Głównym celem projektu jest poznanie podstaw biblioteki graficznej SFML oraz przećwiczenie nowych tematów poznanych w czwartym semestrze PK

2.1 Ogólna Idea Pisania Programu

Program został napisany przede wszystkim w taki sposób, aby był jak najbardziej „modularny”. Dzięki temu można relatywnie łatwo dodawać nowych przeciwników, umiejętności oraz wydarzenia.

2.2 Klasy

Tworząc program, utworzone zostały poniższe klasy:

Clickable - Opisująca obiekty „klikalne”
Button i Unitframe - Dziedziczące z Clickable, opisują one poszczególnie przycisk jak i „ramkę” na jednostkę

Entity- Opisująca szkielet działania jednostek, czy to gracza czy to przeciwnika
Player i Enemy - Dziedziczące z Entity, opisują dokładniej sposób działania danej jednostki

Event - Będąca szkieletem to działania „wydarzeń”
GenDMGUpgrade - Przykładowe wydazenie, dziedziczy z Event

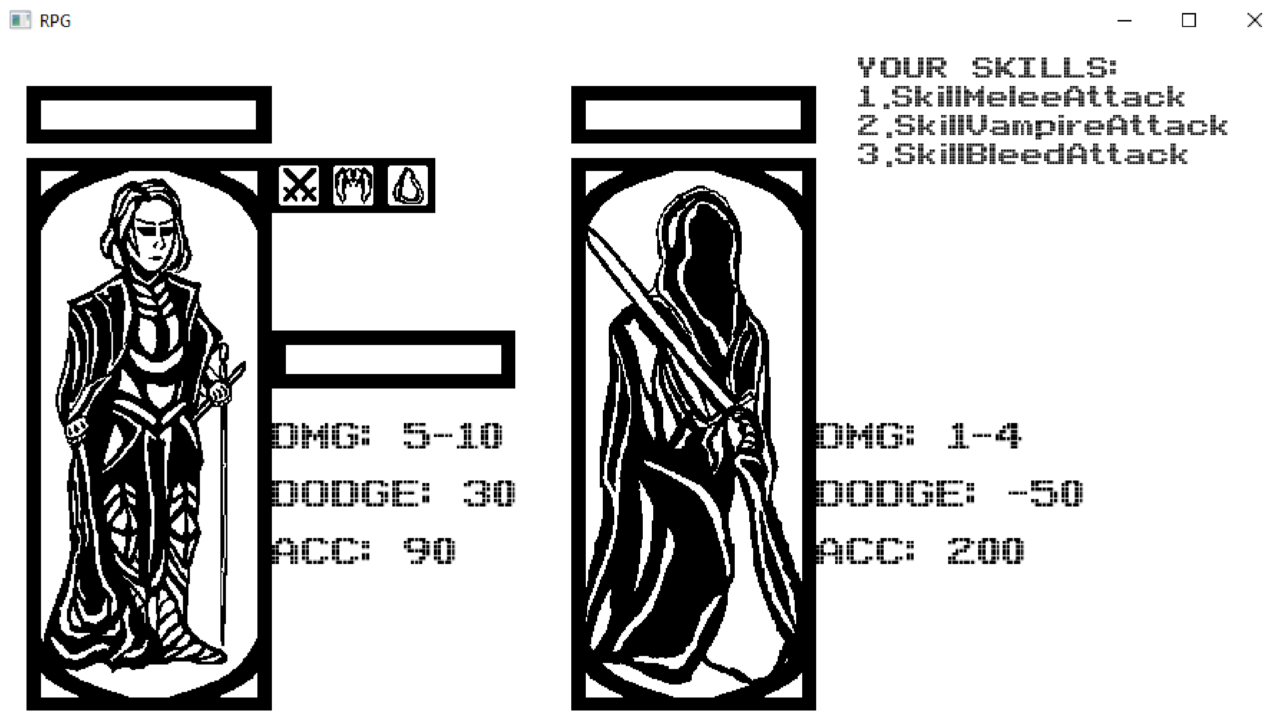
EventInstance - Przetrważającą wydarzenie obecnie mające miejsce na ekran gry.
FightInstance - Przetrważającą walkę obecnie mającą miejsce na ekran gry.

Skill - Opisująca szkielet działania umiejętności
BleedAttack, MeleeAttack i VampireAttack - Opisujące przykładowe umiejętności

HealtBar - Opisuje działanie pasku zdrowia
Random - Służy do generowania liczb losowych

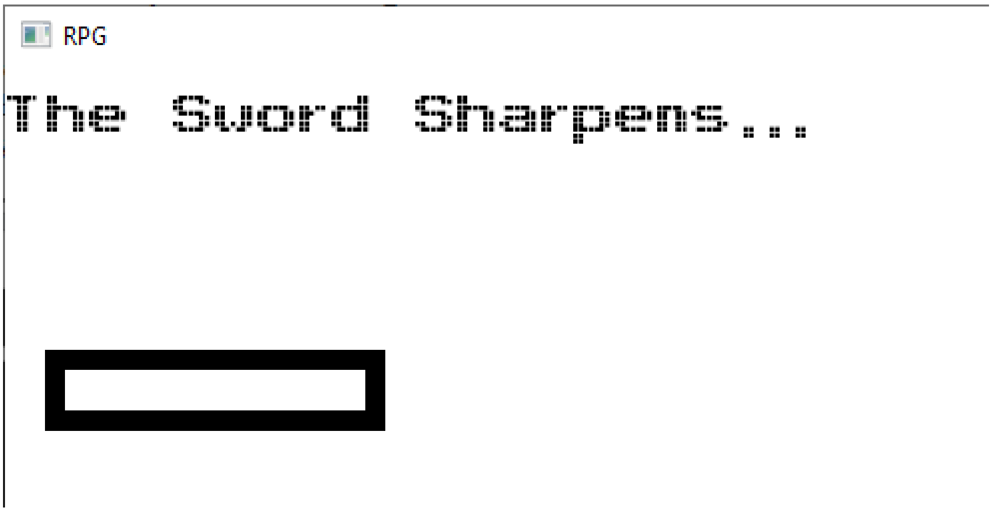
2.3 Krótki Opis Działania Programu

Program toczy się w nieskończonej pętli, do czasu utraty przez gracza wszystkich punktów zdrowia lub zamknięcia okna programu. Gra składa się z dwóch naprzemiennie pojawiących się ekranów: wydarzeń i walki.



```
YOUR SKILLS:
1.SkillMeleeAttack
2.SkillVampireAttack
3.SkillBleedAttack

DMG: 5-10
DODGE: 30
ACC: 90

DMG: 1-4
DODGE: -50
ACC: 200
```

Ekran Walki



```
The Sword Sharpens...
```

Ekran Wydarzenia

3. Specyfikacja Wewnętrzna

Program wymaga zainstalowanej biblioteki SFML 2.5 aby móc skutecznie działać.

4. Specyfikacja zewnętrzna

Program został utworzony zgodnie z paradygmatem strukturalnym. Został on podzielony na pliki:

Clickable.h / Clickable.ccp
Button.h / Button.ccp
Unitframe.h / Unitframe.cpp

Entity.h / Entity.cpp
Enemy.h /Enemy.cpp
Player.h / Player.cpp
Stats.h
Status.h

Event.h / Event.cpp
GenDMGUpgrade.h / GenDMGUpgrade.cpp

Skill.h / Skill.cpp
VampireAttack.h / VampireAttack.cpp
MeleeAttack.h / MeleeAttack.cpp
BleedAttack.h / BleedAttack.cpp
enums.h

Random.h / Random.cpp

Healthbar.h / Healthbar.Cpp

FightInstance.h / Eventintstance.cpp
EventInstance.h / EventInstance.cpp

main.cpp

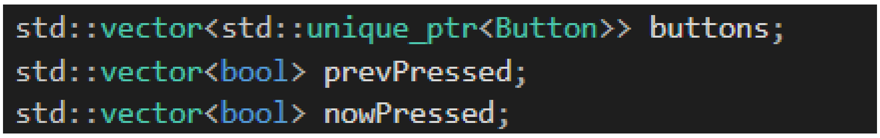
4. Struktura Programu oraz Opis Typów i Funkcji

Szczegółowy opis jest zawarty w pliku „Dokumentacja”

5. Wykorzystanie Nowości Poznanych na zajęciach

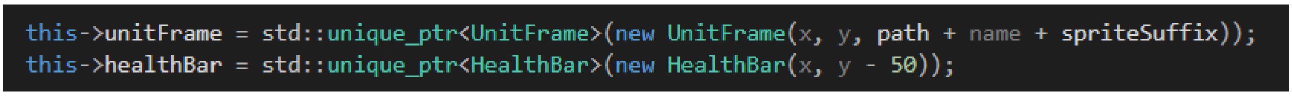
W projekcie wykorzystano cztery nowo poznane tematy:

- Kontenery STL



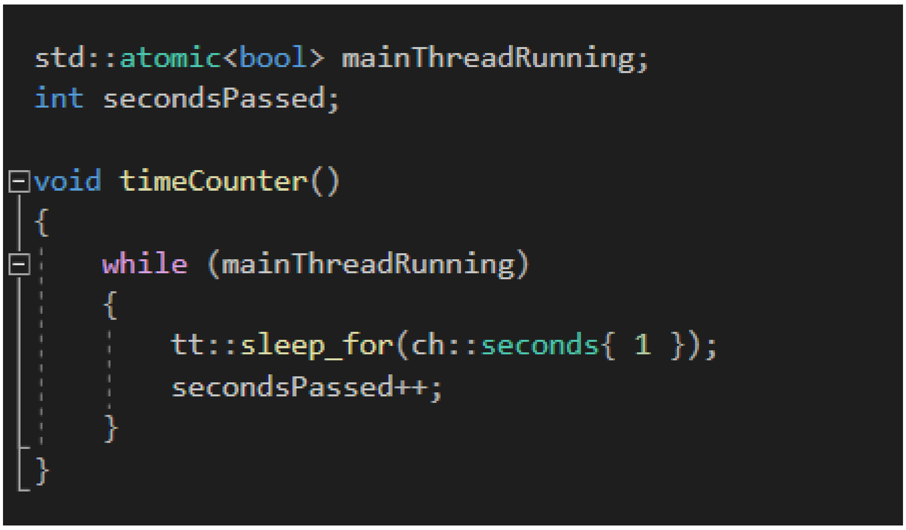
```
std::vector<std::unique_ptr<Button>> buttons;
std::vector<bool> prevPressed;
std::vector<bool> nowPressed;
```

- Inteligetne Wskaźniki

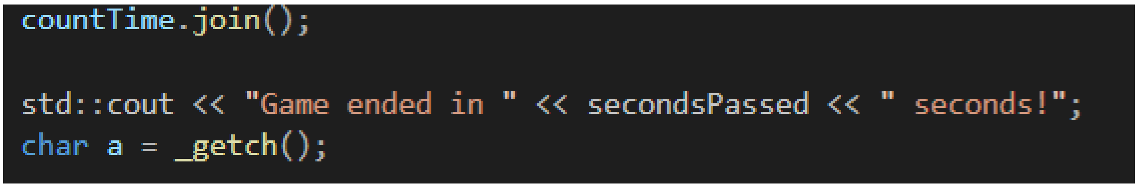


```
this->unitFrame = std::unique_ptr<UnitFrame>(new UnitFrame(x, y, path + name + spriteSuffix));
this->healthBar = std::unique_ptr<HealthBar>(new HealthBar(x, y - 50));
```

- Wątki

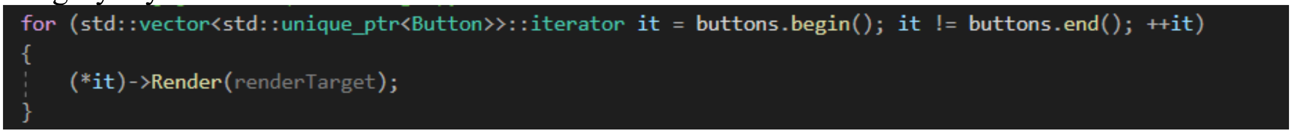


```
std::atomic<bool> mainThreadRunning;
int secondsPassed;

void timeCounter()
{
    while (mainThreadRunning)
    {
        tt::sleep_for(ch::seconds{ 1 });
        secondsPassed++;
    }
}
```



```
countTime.join();

std::cout << "Game ended in " << secondsPassed << " seconds!";
char a = _getch();
```

- Algorytmy STL



```
for (std::vector<std::unique_ptr<Button>>::iterator it = buttons.begin(); it != buttons.end(); ++it)
{
    (*it)->Render(renderTarget);
}
```

6. Testowanie

Program został przetestowany na każdy mozliwy sposób. Nie zawiera wycieków pamięci.

6. Wnioski

Program okazał się ciekawym sposobem na wykorzystanie czasu w samym sobie, a ponadto pozwolił poznać bibliotekę SFML jak i przećwiczyć tematy poznane na zajęciach.