Комп’ютерний практикум № 05

Асоціативні контейнери

Рівень “Високий”

Хід роботи

**Завдання 1**

*З використанням SFML розробити базовий ігровий процес для гри в стилі ​Tower Defense​. Даний практикум виконувати на базі програми, розробленої в рамках роботи над комп’ютерним практикумом № 04.*

*Удосконалити код попереднього практикуму таким чином, щоб керування пам’яттю відбувалось засобами розумних покажчиків.*

Приклад:

vector<vector<cell>> field;

vector<enemy> enemies;

vector<tower> towers;

**Завдання 2**

*Всі клітинки карти та об’єкти гри з практикуму № 04 відмальовувати за допомогою растрових зображень.*

**Завдання 3**

*Реалізувати ​програмний кеш всіх зображень, які використовуються в грі, у вигляді глобального об’єкта, реалізованого за допомогою шаблону проектування ​Singleton​.*

*Кеш зберігати за допомогою асоціативного контейнера, а при зберіганні та передачі зображень між об’єктами програми використовувати принципи RAII (зокрема, розумні покажчики).*

**Texture\_pool.h**

#pragma once

#include "header.h"

class Texture\_pool {

public:

map<int, sf::Texture> textures;

static Texture\_pool& Instance(){

static Texture\_pool s;

return s;

}

private:

Texture\_pool();

//~Texture\_pool();

Texture\_pool(Texture\_pool const&) = delete;

Texture\_pool& operator= (Texture\_pool const&) = delete;

};

**Texture\_pool.cpp**

#include "Texture\_pool.h"

Texture\_pool::Texture\_pool(){

sf::Texture texture;

texture.loadFromFile("Textures/castle.jpg");

textures.insert(pair<int, sf::Texture>(INT(cell\_type::BASE), texture));

texture.loadFromFile("Textures/road.jpg");

textures.insert(pair<int, sf::Texture>(INT(cell\_type::ROAD), texture));

texture.loadFromFile("Textures/tower(0).jpg");

textures.insert(pair<int, sf::Texture>(INT(cell\_type::TOWER), texture));

texture.loadFromFile("Textures/grass.jpg");

textures.insert(pair<int, sf::Texture>(INT(cell\_type::NOT\_ROAD), texture));

texture.loadFromFile("Textures/enemy\_spawn.jpg");

textures.insert(pair<int, sf::Texture>(INT(cell\_type::ENEMY\_SPAWN), texture));

texture.loadFromFile("Textures/best\_tower.jpg");

textures.insert(pair<int, sf::Texture>(INT(cell\_type::BEST\_TOWER), texture));

texture.loadFromFile("Textures/enemy(1)t.jpg");

textures.insert(pair<int, sf::Texture>(INT(cell\_type::ENEMY1), texture));

texture.loadFromFile("Textures/enemy(2)t.jpg");

textures.insert(pair<int, sf::Texture>(INT(cell\_type::ENEMY2), texture));

texture.loadFromFile("Textures/track(1)t.jpg");

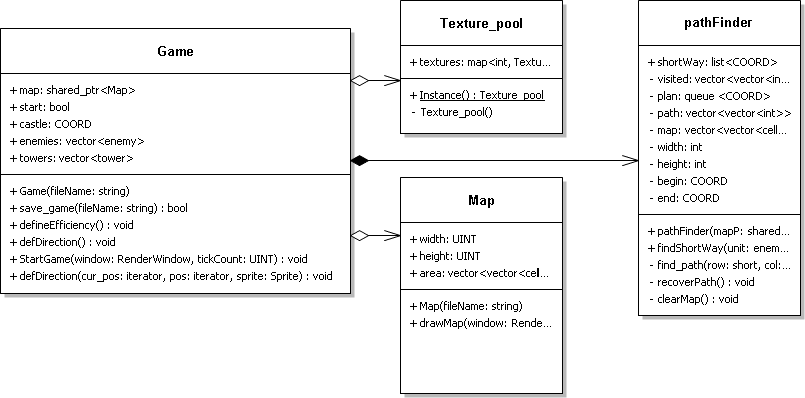
textures.insert(pair<int, sf::Texture>(INT(cell\_type::FOOTPRINT1), texture));

texture.loadFromFile("Textures/track(2)t.jpg");

textures.insert(pair<int, sf::Texture>(INT(cell\_type::FOOTPRINT2), texture));

}}

**UML діаграма класів**



|  |
| --- |
| **Makefile**  all: solve  solve: main.o CreateMap.o FindPath.o Texture\_Pool.o          g++ main.o CreateMap.o FindPath.o Texture\_Pool.o -o main -L ".. /SFML/lib" -lsfml-graphics-d -lsfml-window-d -lsfml-system-d  main.o: main.cpp          g++ -c main.cpp -I"../SFML/include"  CreateMap.o: CreateMap.cpp          g++ -c CreateMap.cpp -I "../ SFML/include"  FindPath.o: FindPath.cpp          g++ -c FindPath.cpp -I"../SFML/include"  Texture\_Pool.o: Texture\_Pool.cpp          g++ -c Texture\_Pool.cpp -I"../SFML/include"  clean:      del -rf \*.o \*.exe |

**Висновок:** в рамках роботи над комп’ютерним практикумом я практикувала використання асоціативних контейнерів. Створення класу Одиночка і його використання, а також отримала навички відображення UML-діаграми проекту.