

Uniwersytet Śląski  
Instytut Informatyki

Paulina Jarosz  
Dawid Masłowski

## **Hidden Dragon – Ukryty Smok**

Projekt zaliczeniowy z programowania warstwy wizualnej gry.

# 1. Wstęp

Celem projektu jest pokazanie możliwości środowiska OpenGL, szczególnie poprzez dynamiczne ruchy bohatera oraz obiektów przemieszczających się po mapie. Ponadto napisano optymalne rozwiązanie kolizji gracza z przedmiotami, które nie obciąża sprzętu. Tym samym próbowano udowodnić jak wiele można stworzyć tworząc w wyżej wspomnianym środowisku.

Projekt był wymyślany i pisany od podstaw. Wiele linijek kodu było ciągle testowanych i poprawianych na bieżąco. Dzięki temu widać było zmiany w działaniu gry i można było od razu zareagować i wyeliminować wszelkie błędy jakie pojawiały się w kodzie.

Gra, która została napisana w OpenGL jako projekt zaliczeniowy jest grą zręcznościową, która polega na zbieraniu jabłek lecąc smokiem.

Postarano się, aby wszystkie detale były jak najbardziej dopracowane, wliczając w to model bohatera oraz przedmiotów do zbierania, tło i starannie dobraną muzykę oraz dźwięki, dzięki którym można jeszcze bardziej wczuć się w sielankowy klimat gry.

## **2. Teoria i algorytmy**

Napisano funkcję sprawdzającą kolizję bohatera z przedmiotami, które zbiera. Korzysta ona z aktualnej pozycji bohatera oraz przedmiotu i przelicza czy możliwa była kolizja czy też nie przy każdym takcie timer'a. Jest to funkcja, która działa cały czas, ale dzięki pewnym wprowadzonym ograniczeniom najważniejsze obliczenia są wykonywane tylko przez kilka taktów. Oznacza to, że funkcja nie jest dużym obciążeniem dla sprzętu i nie spowalnia jego działania przez dużą ilość obliczeń.

### 3. Opis programu

Program został napisany w języku C++ oraz GLSL. Korzystano z bibliotek:

- 1) OpenGL, GLEW oraz GLUT – biblioteki podstawowe, których użyto między innymi do utworzenia okna z grą, ustawienia kolorów, renderowania obiektów czy też zarządzania animacją przez funkcję timer
- 2) ASSIMP – dodatkowa biblioteka, używana na potrzeby wczytywania do gry modeli, które były tworzone w aplikacji zewnętrznej (na przykład w 3DSMax czy Maya)
- 3) SOIL – dodatkowa biblioteka użyta do wczytywania tekstur do modeli, znacznie ułatwia ten proces i zapewnia obsługę wielu podstawowych rozszerzeń plików graficznych
- 4) GLM – dodatkowa biblioteka, zawierająca wszystkie operacje matematyczne, jest niezbędna do poprawnego działania gry (między innymi do obsługi kolizji, przemieszczania się)
- 5) irrKlang – dodatkowa biblioteka, służy do obsługi muzyki, pozwala odtwarzać dźwięki w wielu formatach (między innymi .mp3 czy .wav), bardzo łatwa w użyciu
- 6) string, fstream, sstream, iostream, vector – dodatkowe biblioteki z języka C++, dzięki nim możliwe do użycia stały się zmienne typu string, vector, ale także można było wywoływać dane na konsolę, co znacznie ułatwiło testowanie gry

#### 3.1. Możliwości programu

Program umożliwia zmianę poziomów gry w zależności od stanu bohatera, bezproblemowo sprawdza kolizje, umożliwia zbieranie przedmiotów, a także ma zaimplementowaną funkcję śmierci bohatera i końca gry.

### **3.2. Opis programu**

Gra jest bardzo przyjazna dla użytkownika i wykorzystuje tylko kilka klawiszy z klawiatury:

- klawisz 'ESC' – służy do wyjścia z gry
- klawisz 'Spacja' – służy do uruchomienia gry (na ekranie startowym) oraz restartu gry po przegranej
- klawisz 'W' – służy do zmiany ścieżki poruszania się na wyższą
- klawisz 'S' – służy do zmiany ścieżki poruszania się na niższą

Gdy bohater umrze rozgrywka kończy się, ekran gry zmienia się na ekran końcowy a wynik gracza wyświetlany jest na konsoli.