



MEDIEVAL PIXELS

PROJEKT

Projekt uważam za udany, zdołałem stworzyć 6 poziomów z dwiema innowacjami m.in. drzwi i klucz potrzebny do ich otworzenia oraz eliksir, który zwiększa skok bohatera. Program działa bez zarzutów i wszelkie błędy zostały wyeliminowane.



HARMONOGRAM

Początkowo wszystko działało zgodnie z wcześniejszym planem, aż do czasu pojawienia się problemu z kolizją bohatera względem platform, problem był na tyle duży, że męczyłem się z nim 4 tygodnie i ostatecznie musiałem zmienić kod by wszystko działało jak powinno. Trochę czasu zajęło też tworzenie grafik poziomów oraz animacji postaci. Finalnie gra jest grywalna, w wersji alfa, w wolnym czasie mam zamiar dodać kolejne poziomy oraz kolejne innowacje.

POSTĘPY

MENU

Stworzenie menu oraz
dodanie funkcjonalnych
przycisków

LEVEL 1

Stworzenie pierwszego
poziomu, aby dodawać
i sprawdzać kolejne
funkcje

RUCH BOHATERA

Funkcja, która pozwala
na poruszanie się
postaci po wciśnięciu
przycisków

RUCH WROGÓW

Funkcja, która pozwala
wrogom poruszanie się

KOLIZJA

Funkcja wykrywająca
kolizje bohatera
z wrogami oraz
platformami

INNOWACJE

Rzeczy ubarwiające
rozgrywkę przez co
staje się ona
ciekawsza

POSTĘPY

BŁĘDY

Szukanie wszelkich
błędów w rozgrywce
i ich naprawianie

GRAFIKA

Stworzenie wszystkich
grafik potrzebnych
w rozgrywce

WERSJA ALFA

Finalizacja wersji alfa gry
bez błędów, nadająca
się do gry

OCZEKIWANY WYGLĄD PROJEKTU

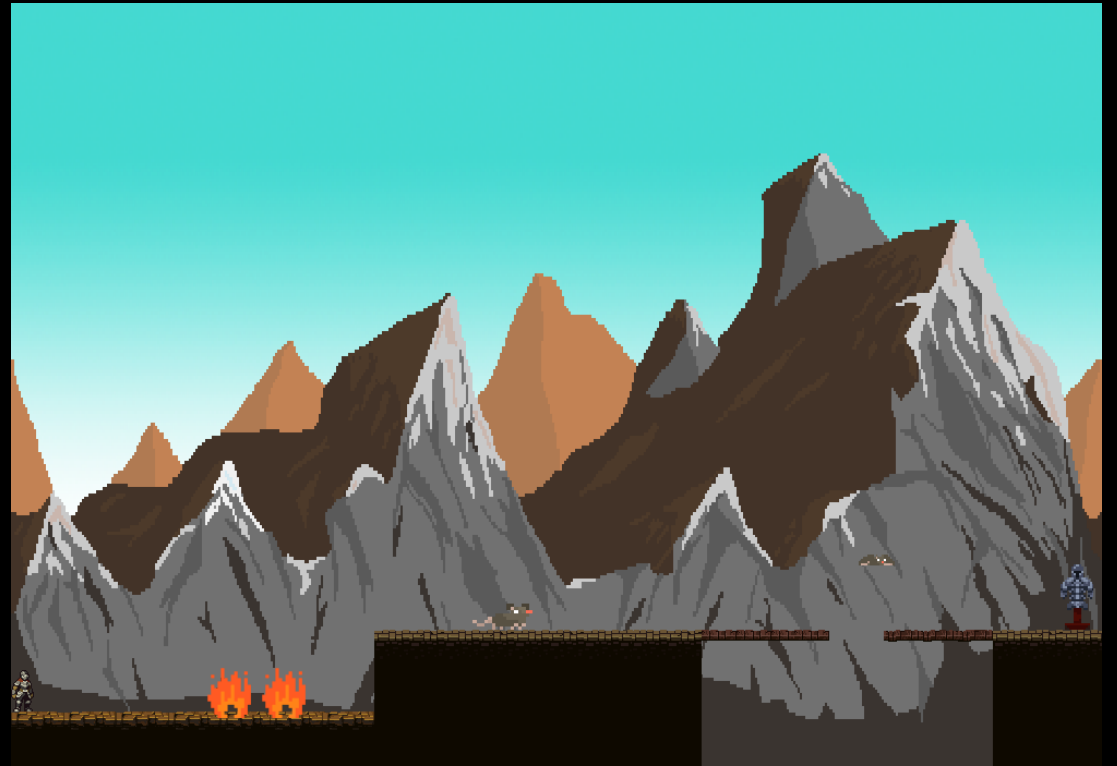
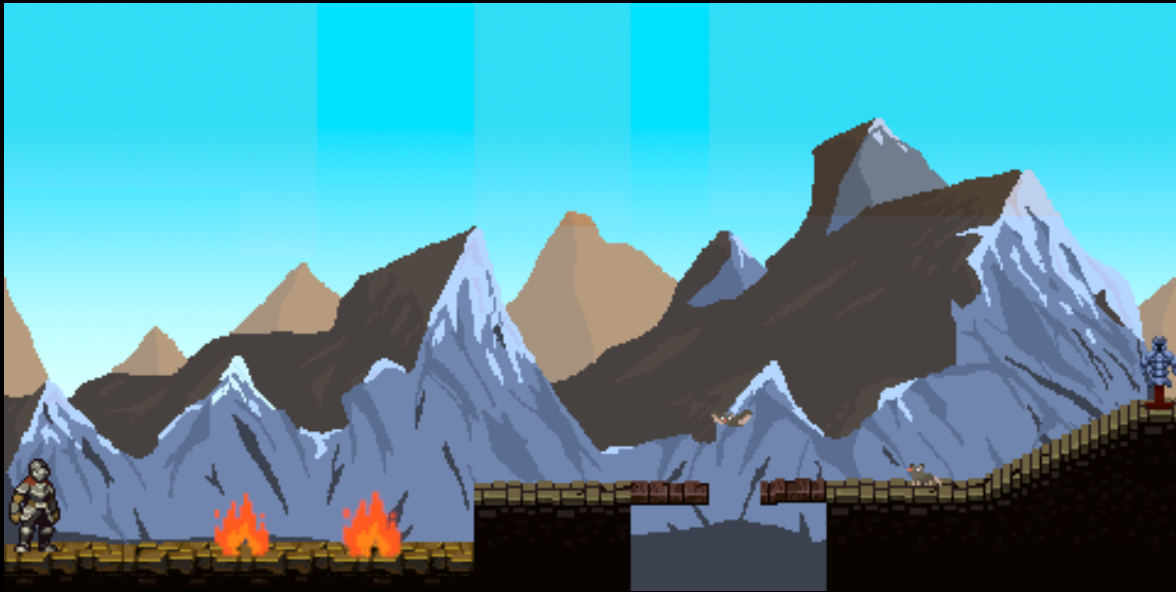


Poziom przedstawia bohatera oraz przeszkody w postaci płomieni, szczura i nietoperza. Po prawej stronie widać „flagę”, która jest stojakiem ze zbroją do której bohater ma dotrzeć.

RZECZYWISTY WYGLĄD PROJEKTU



PORÓWNANIE



Przez wcześniejsze posiadanie grafik udało się odwzorować wizualizację, lecz zmianie uległ rozmiar bohatera, stał się on mniejszy, co pomaga w stworzeniu bardziej dynamicznej rozgrywki.

DEBUGGOWANIE

Proces debuggowania najczęściej polegał na uruchomieniu gry i testowaniu co się stanie po wykonaniu danej akcji, testowałem czy postać nie pojawi się pod teksturami lub czy nie będzie przez nie przenikała, debuggowałem również przez wypisywanie potrzebnych mi wartości do konsoli i jak się one zmieniają po zmodyfikowaniu kodu.

PROPOZYCJA PROWADZĄCEGO

DOUBLE JUMP

Program sprawdza czy bohater ma dostępny drugi skok i czy bohater już skoczył, dostępny jest on gdy postać zaczyna spadać po wykonanym skoku. Jego wysokość jest mniejsza, aby nie dawał on aż takiej korzyści.

JĘZYK I TECHNOLOGIE

- Język programowania – Python
- Biblioteka – Pygame
- Środowisko – Visual Studio Code





DZIĘKUJE ZA UWAGĘ !

Paweł Szydłowski