

Zastosowane rozwiązania

Do realizacji projektu został użyty język C. Jest to podyktowane:

- * dobrą jego znajomością przez cały zespół
- * zapewnieniem dobrej wydajności dla bardziej wymagających operacji
- * faktem, że później wykorzystywane biblioteki są również napisane w C, tak więc brakiem przeszkód w postaci potrzeby wykorzystania innego języka

Jako środowisko programistyczne zostało wybrane Visual Studio 2022 ze względu na zapewniane przez nie bardzo przydatne opcje takie jak między innymi integracja GitHubem, system Rozwiązań (Solutions) oraz unikalne opcje konfiguracyjne. Ponadto również jest narzędziem z którym zespół był obeznany.

Kluczowym w stworzeniu projektu było użycie biblioteki raylib, która jest jedyną zależnością zewnętrzną użytą w kodzie gry.

Biblioteka raylib[1] służy do zarządzania warstwą graficzną projektu. W projekcie korzystano między innymi z zapewnianych przez nią funkcjonalności:

- * Tworzenia Okienek
- * Rysowania Prostych Kształtów, Tekstur oraz Tekstu
- * Wczytywania plików takich jak: pliki tekstowe, tekstury, czcionki oraz muzyka
- * Odtwarzanie muzyki
- * Obliczania prostych zależności matematycznych, jak na przykład odległość dwóch punktów

Wybór tej biblioteki został umotywowany dobrą jej znajomością przez głównego programistę, a także jej relatywne nieskomplikowanie co pozwoliło na szybką adaptację zespołu do tejże technologii.

Biblioteka CUnit[2] jest biblioteką którą wykorzystaliśmy do stworzenia testów jednostkowych. Sama biblioteka jest porzucona przez oryginalnych twórców, tak więc skorzystaliśmy z wersji której rozwój został kontynuowany przez społeczność.

Wybór tej biblioteki był umotywowany jej prostotą oraz znajomością przez głównego testera.

Algorytmy warte wspomnienia

Do renderowania tekstur postaci w poprawnej kolejności użyto algorytmu sortowania przez wstawianie. Zdecydowano się na ten algorytm, ze względu na fakt iż przez większość czasu sortuje on elementy już posortowane co jest jego optymistycznym przypadkiem dającym złożoność liniową. Przy gwałtownych ruchach kamery, gdy położenie tekstur względem kamery się zmienia złożoność może sięgać $O(n^2)$, jednak jest to statystycznie rzadki przypadek. Warto nadmienić, że po każdym sortowaniu nowa kolejność jest zapamiętywana, co właśnie umożliwia opisaną wyżej złożoność liniową.

[1] raylib.com

[2] <https://gitlab.com/cunity/cunit>