Antoni Siółko

Maksym Shvets

Dokumentacja

Gra: Monkey Jump

1. Wybieranie tematu:  
   Po rozpoczęciu projektu od razu padło kilka pomysłów na temat popularnych gier telefonicznych takich jak Tetris, 2048, coś w stylu „Need for Speed” oraz jakaś 3d gra w labiryncie. W końcu doszliśmy do wniosku, że byłoby ciekawie poeksperymentować z grą „Monkey Jump”, ale, która polega na tym, małpa rusza się w górę w nieskończoność skacząc na platformach. Dla początku skorzystaliśmy z tutorialu, w którym funkcjonalność kodu była maksymalnie prosta, ale w niektórych miejscach pusta, co musieliśmy dodać dla pełnego funkcjonowania naszej gry. Też dodaliśmy monstra którego trzeba starać się ominąć, a jeżeli się da to można na niego trafić z góry i dostać dodatkowe punkty. Jeżeli spotkacie się w innym miejscu, niestety, musicie zacząć grę ponownie.

Wykorzystywana biblioteka:  
Na początku chcieliśmy użyć omawianej na ćwiczeniu biblioteki „Allegro 5”, ale znaleźliśmy bardziej rozpowszechnioną opcję, które jest używana w wielu projektach. Jest to biblioteka SFML(*Simple and Fast Multimedia Liberary*). Poniżej zamieszczamy krótki tutorial, jak ta biblioteka musi się dołączać do Visual Studio (używaliśmy go do napisania kodu więc polecamy najpierw go pobrać w wersji Community, też do tego jest link - <https://visualstudio.microsoft.com/vs/> )

1. Tutorial

Wejdź na stronę <https://www.sfml-dev.org/download/sfml/2.5.1/> i pobierz wersję odpowiadającą twojemu VS. Jeśli używasz programu Visual Studio 2015 lub nowszego, możesz śmiało użyć wersji SFML skompilowanej dla Visual C++ 15 (VS 2017), która jest nadal kompatybilna z nowszymi wersjami zestawu narzędzi.

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Wypakuj w dowolnym miejscu folder SFML. Otwórz projekt w Visual Studio.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Kliknij prawym przyciskiem myszy na swoją aplikację i wybierz właściwości.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Otwórz zakładkę C/C++ i wybierz Dodatkowe katalogi plików nagłówkowych

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, numer

Opis wygenerowany automatycznie

Wybierz edytuj

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, wyświetlacz

Opis wygenerowany automatycznie

Następnie wklej ścieżkę folderu “include” z wypakowanego SFML

Obraz zawierający tekst, oprogramowanie, Oprogramowanie multimedialne, zrzut ekranu

Opis wygenerowany automatycznie

Wzakładce Preprocesor edytuj pole “Definicje preprocesora”

Obraz zawierający tekst, zrzut ekranu, wyświetlacz, oprogramowanie

Opis wygenerowany automatycznie

W wyznaczonym polu należy dopisać: SFML\_STATIC

Teraz otwieramy zakładkę Konsolidator, a w niej Ogólne i wyszukujemy Dodatkowe katalogi biblioteki.

Obraz zawierający tekst, zrzut ekranu, wyświetlacz, oprogramowanie

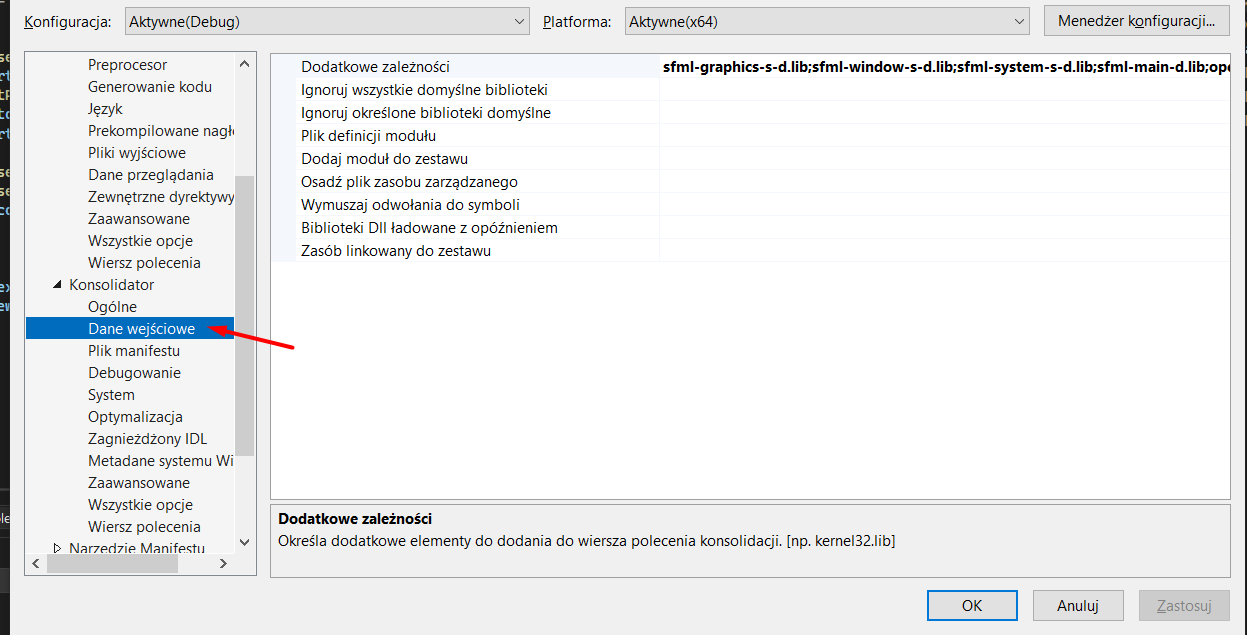
Opis wygenerowany automatycznie

W wyznaczonym polu wklejamy ścieżkę do folderu lib z wypakowanego SFML

Obraz zawierający tekst, zrzut ekranu, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Następnie otwieramy zakładkę Dane Wejściowe w Konsolidatorze



Wklejamy w polu Dodatkowe zależności następujące rzeczy:

sfml-graphics-s-d.lib sfml-window-s-d.lib sfml-system-s-d.lib sfml-main-d.lib opengl32.lib freetype.lib winmm.lib gdi32.lib

Kolejność, która została podana ma znaczenie.

To wszystko. Teraz możemy zacząć grać. Miłej zabawy!

Jeżeli tutorial ten nie jest zrozumiały lub nie wystarcza wizualnej treści możecie obejrzeć wideo na YouTube, które nam pomogło prawidłowo załadować wszystkie pliki. Poniżej zostawiam link:  
<https://www.youtube.com/watch?v=VWWSc2nqrEA&t=171s&ab_channel=Zipped>