Przygoda z wokselowym silnikiem gry. Część 1

Na rynku można znaleźć bardzo dużo zaawansowanych silników gier, które pozwalają na szybkie stworzenie gry bez zbytniego zastanawiania się: „jak to działa pod spodem?". Czasami twórcy gier decydują się na stworzenie własnego silnika, ponieważ te dostępne na rynku nie nadają się do gry, jaką sobie wymarzyli. Często też napisanie własnego silnika pozwala lepiej doszlifować finalny produkt poprzez rezygnację z komponentów, które są dla nas zbędne. Ogólnodostępne silniki posiadają ich wiele, ponieważ muszą objąć swoimi możliwościami jak najwięcej gier, które potencjalni developerzy będą tworzyć za ich pomocą. Jednym z takich przypadków, gdzie warto spróbować napisać własny silnik gry, może być gra opierająca się na wokselach. A na pewno będzie to świetny pretekst, by spróbować napisać własny silnik dla celów edukacyjnych.

CZYM ZAJMIEMY SIĘ W ARTYKULE

O tym, jak napisać cały silnik od podstaw, można by napisać całą książkę – nawet jeżeli jest tak prosty jak nasz, dlatego w tym artykule skupimy się na kilku rzeczach, w szczególności tych oscylujących wokół wokseli.

Listingi zawarte w artykule są napisane w C++, podobnie jak cały silnik, dlatego znajomość tego języka może być bardzo przydatna, jednak nie niezbędna do zrozumienia algorytmów i rozwiązań, które zastosowałem.

Przy czytaniu kodu silnika gry przydatna będzie znajomość podstawowych operacji matematycznych stosowanych w grafice 3D – nie będzie tego dużo, jednak niektóre elementy kodu bez tej wiedzy mogą wydać się niezrozumiałe.

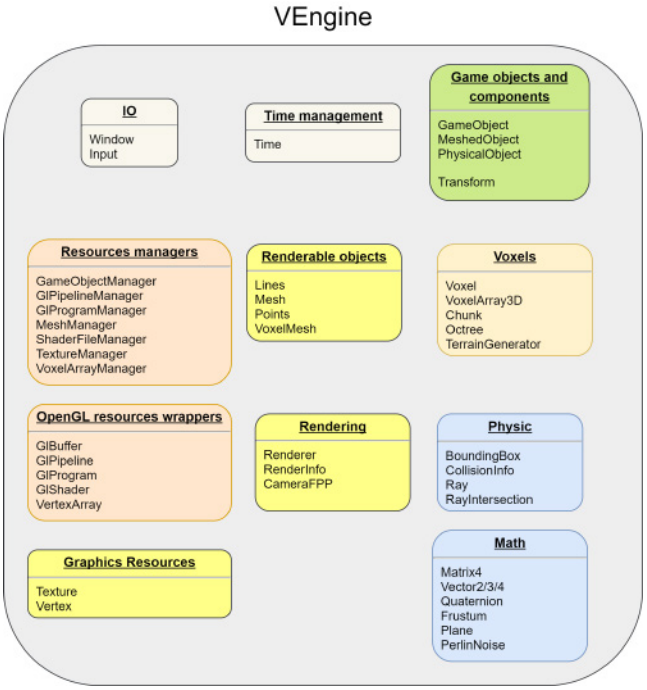
Nie będziemy się zagłębiać we wszystkie zagadnienia związane z grafiką 3D – podejmiemy jedynie najciekawsze tematy, a bardziej podstawową wiedzę postaram się chociaż nieco przybliżyć czytelnikom, którzy swojej przygody z grafiką 3D jeszcze nie zaczęli. Przygotowałem kilka klas pomocniczych, opakowujących API OpenGL, dlatego też, drogi Czytelniku, jego znajomość nie jest wymagana do zrozumienia treści artykułu, jednak pojawią się pewne nawiązania do samego OpenGLa.

Artykuł będzie podzielony na dwie części, przy czym w pierwszej skupimy się na wyświetlaniu wokseli, w drugiej natomiast na elementach silnika gry oraz prostej fizyce.

Na początku przedstawię komponenty, z jakich składa się mój silnik, następnie zastanowimy się nad wokselem – co to takiego jest i jakie są związane z nim problemy, stworzymy optymalną siatkę wierzchołków na podstawie tablicy wokseli, zastanowimy się, jak nałożyć tekstury na nasze woksele, wygenerujemy prosty teren złożony z wokseli, zrobimy sobie bardzo krótki wstęp do potoku graficznego oraz języka GLSL (OpenGL Shading Language), aż w końcu napiszemy proste shadery, które będą potrafiły wyświetlać nasze woksele. Cały kod silnika można znaleźć w repozytorium GitHub, do którego link znajduje się na końcu artykułu. Zachęcam do zapoznania się z nim osobom bardziej zainteresowanym tematem. No to czas start!

BUDOWA SILNIKA

Listingi, które zaprezentuję poniżej, byłyby bardzo ciężkie do analizy w oderwaniu od kontekstu. Dlatego zanim przedstawię jakiś kod, który będzie w większości implementacją pewnych komponentów silnika wokselowego, zobaczmy, z czego on się składa. Roboczo nazwałem go VEngine (od Voxel Engine). Najważniejsze komponenty zostały przedstawione na Rysunku 1.



Rysunek 1. Najważniejsze elementy silnika wokselowego VEngine

Menedżery zasobów

Silnik, na którym zostanie w przyszłości zbudowana gra, musi posiadać bazę danych dla różnych zasobów – czy to tekstur, shaderów,