

1/6 Sky Wars – dokumentacja techniczna

Sky Wars 1.0

Dokumentacja techniczna

Autorzy:

Przemysław Dobrowolski, PCC

Paweł Gorczyński, PCC

Michał Harasimowicz, PIS

Maciej Jurkowski, PCC

Piotr KomisarSKI, PCC

Spis treści

Architektura aplikacji.....	3
Fizyka.....	3
Siła ciągu.....	3
Siła nośna.....	3
Siła oporu.....	4
Sztuczna Inteligencja	4
Dźwięk.....	5
System sterowania.....	5

Architektura aplikacji

Aplikacja podzielona jest na niezależne moduły. Każdy z modułów ma spełniać swoje zadanie tak jakby nie było żadnych innych modułów. W danej chwili może być aktywny tylko jeden moduł. W trakcie aktywności moduł otrzymuje wszystkie komunikaty wejścia od użytkownika a także dostaje wszystkie zdarzenia synchronizacji. Można wyróżnić moduły takie jak: MenuModule, SceneModule, AboutModule, LobbyModule, SettingsModule. Zarządzaniem modułami zajmuje się rdzenny moduł AppCore który odpowiedzialny jest za zainicjowanie aplikacji i wszystkich modułów a następnie uaktualnienie pierwszego z modułów. Istnieje możliwość przesyłania poleceń z jednego modułu do innego. Realizowane to jest przez odwołanie się do AppCore i zapytanie o konkretny moduł (wyszukiwany po nazwie). Jeśli odnajdziemy taki moduł i jest on aktywny to możemy na nim wykonać interesujące nas polecenie.

Fizyka

Do realizacji fizyki wykorzystaliśmy bibliotekę OgreOde, która udostępnia podstawowe metody i obiekty ODE w środowisku Ogre.

Samolot reprezentowany jest przez 7 obiektów Body o przypisanej geometrii Box-a. Wymiary, masy i pozycje wczytywane są z pliku i są właściwe dla modelu samolotu, tak samo jak wszystkie współczynniki potrzebne do obliczeń. Samolot konstruowany geometrycznie jest tak by środek ciężkości wypadł w początku, lokalnego dla całego samolotu, układu współrzędnych. Dla każdego obiektu Body wyliczany jest na podstawie wcześniej wspomnianych danych tensor bezwładności względem środka ciężkości samolotu i podawany w trakcie tworzenia reprezentacji fizycznej elementu do obiektu BoxMass, który zostaje dopiero dołączony do Body, a owe Body geometrii.

Tak stworzone elementu samolotu (lewe i prawe skrzydło, lewy i prawy ster wysokości, orczyk, kadłub, silnik) łączone są 'Joint-ami' sztywnymi (FixedJoint).

Fizyczny model samolotu połączony jest z graficzną reprezentacją, którą jest mesh samolotu – obiekt Ogre::Entity. Dodatkowo w OgreOde::World (reprezentacji fizycznego świata z biblioteki OgreOde) włączona jest grawitacja oddziałująca na wszystkie Body, dlatego też w dalszej części tekstu nie będę jawnie wspominał o dodawaniu czy wyliczaniu sił ciężkości.

Co klatkę uruchamiana jest metoda AirplanePhysics::Update w której wyliczane są wartości sił działających na poszczególne części modelu, a następnie są one dostawiane do owych części. OgreOde gwarantuje, iż w każdej klatce siły dostawione w poprzedniej zostaną wyzerowane.

Siła ciągu.

Na początek zacznijmy od siły ciągu. Silniki odznaczają się właściwymi sobie mocami. Moc silnika przemnożona przez współczynnik wydajności śmigła daje nam moc ciągu, która jest równa sile ciągu pomnożonej przez prędkość. Zaś współczynnik wydajności śmigła przybliżany jest przez wielomian 3 rzędu postaci: $aJ + bJ^3 = \eta$, gdzie J to prędkość samolotu przez iloczyn średnicy śmigła i ilość jego obrotów na sec. To wszystko daje nam następujący wzór na siłę ciągu:

$$F_t = \frac{P_e}{nd} \left(a + b \frac{v^2}{n^2 d^2} \right)$$

Ta siła musi zostać jeszcze przemnożona przez współczynnik spadku mocy

$$\phi = \frac{\rho - C}{1 - C} \rho_0$$

, gdzie C to stała spadku mocy wynikłego z wysokości, niezależna od

mechanicznych aspektów silników (wynosi 0.12), zaś ρ_0 i ρ to gęstość powietrza i gęstość powietrza na poziomie morza.

Siła nośna.

Wyrażona jest wzorem $F_L = \frac{1}{2} C_L \rho v^2 A$, gdzie A to pole powierzchni płata, v – prędkość

natarcia powietrza na skrzydło, w naszym przypadku przybliżyliśmy to prędkością samolotu w kierunku jego dzioba, ρ to gęstość powietrza w którym porusza się samolot, zaś C_L to współczynnik siły nośnej wyliczany na podstawie kąta natarcia skrzydeł (kąt pomiędzy prędkością całkowitą samolotu (kierunkiem poruszania), a kierunkiem jego dzioba), wychylenia „flap” i wysokości na której porusza się samolot. Dla kątów natarcia z przedziału właściwego dla danego modelu samolotu (-maxKatNatarcia, maxKatNatarcia) współczynnik jest przybliżany prostą rosnącą, dla pozostałych kątów malejącą (współczynniki również są pobierane z pliku), dodatkowo dla wychylonych flapów dodawane są odpowiednie stałe (wczytywane z pliku). Jeżeli samolot ponadto znajduje się na wysokości mniejszej jak rozpiętość skrzydeł do współczynnika C_L dodawana jest kolejna stała przybliżająca efekt wirów powstających nad ziemią pomagających przy starcie i nabieraniu wysokości.

Siła oporu.

Siła oporu wyraża się tym samym wzorem co siła nośna z tą tylko różnicą iż zamiast współczynnika C_L , mamy C_D czyli współczynnik oporu, a prędkość to prędkość w kierunku prostopadłym do powierzchni dla której liczymy siłę. Współczynnik oporu jest stałą, zatem opór zależy wyłącznie od zmiennej v i gęstości powietrza (która zależy od wysokości).

Poza zwykłym oporem występuje jeszcze opór indukowany, którego współczynnik wyrażony jest

wzorem: $C_{Di} = \frac{C_L^2}{\pi A_R e}$, gdzie $A_R = \frac{s^2}{A}$, s – rozpiętość płata, A – pole powierzchni płata, e –

stała właściwa dla modelu samolotu z przedziału (0,1] (dla Cesny e = 0.77, dla Spitfire e = 1.0).

Dla poprawnego liczenia oporów do każdej pobranej z body prędkości liniowej zrzutowanej na odpowiedni kierunek pobierana jest dodatkowo prędkość kątowa i z niej wyliczana prędkość liniowa dla punktu w którym ma zostać przyłożona siła. Opory są liczone dla obu tych prędkości. Dodatkowo dla oporów skrzydeł czy stateczników zastosowano 'wzmocnienie' by osiągnąć szybsze wyhamowanie, aby samolot za łatwo nie zaczynał niekontrolowane wirować. Nie jest to zgodne do końca z fizycznym modelem ale ułatwia sterowanie i umożliwiała lepsze testowanie oporów.

Wszystkie wyliczone siły są wyliczane w układzie lokalnym i tak też dostawiane przy pomocy odpowiednich metod udostępnianych przez OgreOde. Opory wyliczane są dla każdej części, dla każdego kierunku i dostawiane w środkach ciężkości. Ciąg jest liczony raz i dostawiany do kadłuba, siły nośne są liczone dla skrzydeł i stateczników. Sterowanie lotkami i pozostałymi sterami polega na dodaniu siły nośnej wytworzonej przez ten element pomnożonej przez określony współczynnik proporcjonalności i wychylenie steru, lotki (które jest z przedziału [-1.0, 1.0]). Wyliczone siły są dostawiane w środkach skrzydeł, sterów. Dodatkowo dla usprawnienia sterowania kierunkiem odpowiednio proporcjonalna siła wynikła z wychylenia steru jest przykładana do przedniej części kadłuba (tak by momenty pędu wynikające z obu tych sił były jednakowe). Rozwiązanie to

5/6 Sky Wars – dokumentacja techniczna

zastosowaliśmy żeby usprawnić sterowanie samolotem i zwiększyć znaczenie prędkości kątowej dla skrętów pitch, yaw, które dalej są wykorzystywane do oporów.

Powierzchnie sterów, a co za tym idzie siły nośne zostały tak dobrane by samolot znajdował się we względnie stabilnym stanie (momenty w lokalnym zerze całego samolotu równoważyły się).

Sztuczna Inteligencja

System sztucznej inteligencji jest zrealizowany jako integralna część projektu. Boty sterują tym samym interfejsem co człowiek. Głównym i jedynym celem systemu sztucznej inteligencji jest podjęcie przez bota decyzji. Przez decyzję rozumiemy zestaw wszystkich argumentów do wszystkich funkcji w interfejsie sterowania samolotem.

Jest zastosowany filtr, który decyduje o tym, które obiekty na scenie bot „widzi”. Iterujemy po wszystkich obiektach, na scenie filtrując te, które bot może zobaczyć i tą listę przekazujemy botowi.

Boty mają pewną pamięć, tzn. w przypadku zobaczenia jakiegoś obiektu pamiętają ostatnią informację o nim, oraz czas pozyskania tej ostatniej informacji.

W przypadku, gdy jakiś obiekt wyjdzie z pola widzenia bota, wtedy przez jakiś czas ekstrapolujemy jego położenie. Po tym czasie boty przestaje się interesować tym obiektem.

Sztuczna inteligencja ma dwie warstwy.

Pierwsza warstwa dokonuje wyboru punktu, oraz kierunku, wyznaczając położenie, w którym bot by się chciał znaleźć. Dokonuje tego na podstawie przeglądu sytuacji oraz dostępnych mu położzeń i prędkości innych obiektów.

Jest kilka kryteriów, które decydują o tym wyborze. Każde kryterium ocenia jakiś wybór i przypisuje mu jakąś wartość liczbową.

Następnie całościowa ocena wyboru jest liczona jako suma ważona poszczególnych ocen.

Wagi przypisane poszczególnym kryteriom opisują „charakter” bota. Można na przykład zmieniać jego nastawienie na bardziej agresywne jedynie zmieniając te wagi.

Druga warstwa przyjmuje jako argument punkt oraz kierunek, w którym bot by się chciał znaleźć. Decyduje ona o wychyleniu drążka bazując na obecnym położeniu, kierunku oraz prędkości. Dodatkowo zadaniem drugiej warstwy jest ominięcie ewentualnych przeszkód i wzniesień.

Efekt jest taki, że w przypadku podążania za jakimś obiektem tor lotu bota jest nieco falowy. Wynika to z tego, że chcąc zmienić kierunek na pożądanym bot wychyla drążek w tą stronę, aczkolwiek działanie bota ma pewną bezwładność, więc w momencie osiągnięcia pożądanego kierunku lotu drążek jest jeszcze minimalnie wychylony, samolot nieco zbacza z określonego kierunku, i bot musi wychylić drążek w drugą stronę.

Jednak zostawiliśmy takie zachowanie uznając że nie ma ono większego wpływu na podążanie za jakimś obiektem, a jest nawet pożądanym w warunkach bojowych.

Dźwięk

W grze dźwięk realizowany jest przy pomocy biblioteki OpenAL.

System sterowania

Do obsługi systemu sterowania wykorzystaliśmy dołączoną do OgreSDK bibliotekę Object Oriented Input System (OIS), która pełni funkcję warstwy abstrakcji m.in. nad Direct Input.

Gra obsługuje 3 podstawowe rodzaje urządzeń wejściowych:

- mysz (wykorzystywana do obsługi menu oraz obrotu free-lookowej kamery),
- klawiatura jako urządzenie sterowania samolotem w trakcie gry,
- joystick (lub też kierownicę) jak wyżej.

Domyślne sterowanie wygląda następująco:

- strzałki lewo - prawo – zmiana wychylenia samolotu,
- strzałki góra - dół – przepustnica,
- klawisze a - d – skręt,
- klawisze w - s – wznoszenie/opadanie,
- spacja – strzał,
- klawisz c – zmiana kamery,
- klawisz ~ - wyświetlenie konsoli,
- klawisz ESC – wyjście do menu.

Dodatkowo zaimplementowana została obsługa technologii Force Feedback, w postaci dwóch efektów:

- drgania oscylacyjne przy strzelaniu z karabinu,
- automatyczne centrowanie kierownicy po wychyleniu.