

Projekt programistyczny						
Rok akademicki	Termin	Rodzaj studiów	Kierunek	Prowadzący	Grupa	Sekcja
2014/2015	Wtorek	SSI	INF	dr inż. Arkadiusz Biernacki	GKiO3	1
	12:45 - 15:00					



Karta projektu

Danmaku Shooter

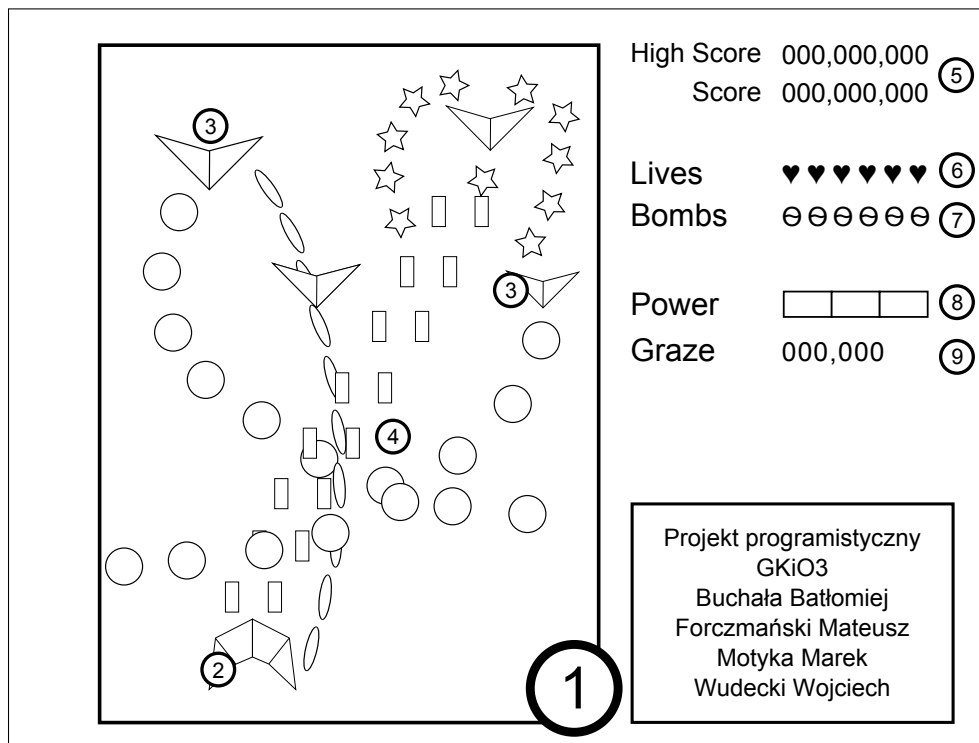
Skład sekcji:

Buchała	Bartłomiej
Forczmański	Mateusz
Motyka	Marek
Wudecki	Wojciech

Krótki opis aplikacji

Shoot' em up (w skrócie zwany shmup) jest gatunkiem gier akcji wywodzącym się w prostej linii od gier typu *Space Invaders* lub *River Raid*. Kontrolowana przez gracza postać (np. statek) w pojedynkę stawia czoło przeciwnikom, niszcząc ich za pomocą wystrzeliwanych pocisków, jednocześnie unikając ich ataków. Podgatunek shmupów, zwany *danmaku* (z jap. *ściana pocisków* lub *piekło pocisków*) kładzie większy nacisk na omijanie wrogich ataków, niż na ofensywie. Przykładowymi danmaku są np. *Ikaruga* czy większość gier z uniwersum *Touhou Project*.

Gra powstawać będzie jako projekt łączony z przedmiotów Projekt Programistyczny i Grafika Komputerowa.



1. Ekran gry właściwej. W jej obrębie znajduje się gracz, pociski oraz wszyscy wrogowie.
2. Grywalna postać, porusza się po ekranie gry, unikając pocisków oraz strzelając do wrogów.
3. Wrogowie, których należy pokonać.
4. Chmura pocisków. Jak widać na rysunku, nie wchodzi ze sobą w żadną interakcję, każdy leci swoim wyznaczonym torem. Sprajty wrogów są niewrażliwe na swoje pociski, nie występuje *friendly fire*.
5. Liczba zdobytych punktów oraz porównywanie ich z największym wynikiem.
6. Liczba pozostałych żyć. W trakcie gry można zdobywać kolejne. Utrata wszystkich kończy grę.
7. Liczba pozostałych bomb. Każda wykorzystana bomba zapewnia kilkusekundową odporność na wrogie pociski oraz umożliwia pojedynczy silniejszy atak. Można je zdobyć w trakcie gry.
8. Pasek mocy, napęlnia się w trakcie gry wraz ze zdobytymi punktami. Utrata życia skutkuje zmniejszeniem paska o 1 segment.
9. Liczba "otarć", czyli uniknięć bardzo blisko pociskowi. Aby umożliwić większe wyzwanie, ostateczny wynik pomnożony jest przez licznik Graze.

Założenia projektowe

1. Narzędzie do modelowania UML

Enterprise Architect

Jest to narzędzie, który poznaliśmy semestr wcześniej na projekcie z przedmiotu Inżynieria Oprogramowania. Ponieważ ww. przedmiot kładł bardzo duży nacisk na modelowanie diagramów, większość z nas jest z nim dobrze zaznajomiona. Przejrzystość tego programu może być niezwykle przydatna podczas tworzenia np. diagramu przypadków użycia. Równolegle wykorzystujemy EA do stworzenia diagramów UML dla projektu z przedmiotu Bazy Danych II.

2. System kontroli wersji

GitHub

Jest to bardzo powszechnie stosowane narzędzie wśród deweloperów pracujących w grupach. Jego zaletami jest m.in. klient aplikacji na system operacyjny Windows, łatwy do przyswojenia i intuicyjny interfejs i łatwość w zarządzaniu starszymi wersjami. Podobnie jak w przypadku Enterprise Architecta, większość z nas miała już wcześniej styczność z GitHubem i korzysta z niego od pewnego czasu.

3. Narzędzia pracy grupowej

Facebook/Skype/Gmail

W celu ułatwienia komunikacji wykorzystane zostaną popularne sieci społecznościowe oraz komunikatory. Ułatwi nam to wspólną współpracę jak i dzielenie się pomysłami oraz dokonanymi zmianami.

4. Środowisko programistyczne

Microsoft Visual Studio 2012

Wybraliśmy to środowisko ze względu na przyjazny interfejs, który nawet w najtrudniejszych sytuacjach potrafi okazać się bardzo pomocny. Pracujemy na nim od dłuższego czasu co ułatwi nam szybki dostęp do potrzebnych narzędzi. Etap debugowania posiada wiele możliwości kontroli naszego programu, co na pewno pozwoli uniknąć kilku czasochłonnych błędów. Ponieważ czasami pojawiają się problemy w kompilacji związane z różnymi wersjami VS, wspólnie wybraliśmy jedną wersję środowiska w celu usunięcia tych przeszkód.

5. Język programowania

C++ / C#

Głównym kryterium wyboru tych języków był wysoki poziom integracji z platformą programistyczną .NET oraz środowiskiem programistycznym Microsoft Visual Studio. Dzięki temu możemy uniknąć problemów podczas łączenia poszczególnych komponentów naszego projektu. Dodatkowo, język C++ poznaliśmy wcześniej na przedmiocie Programowanie Komputerów, a także używaliśmy go na innych laboratoriach - posiadamy pewne doświadczenie w tworzeniu aplikacji z jego użyciem. Język C# jest nieco bardziej zaawansowane i na jego użycie zdecydujemy się wtedy, gdy opanujemy go na odpowiednim poziomie.

6. Biblioteka graficzna

DirectX / OpenGL

7. Platforma programistyczna

.NET Framework

W tym przypadku ponownie zadecydowała integracja z innymi produktami od Microsoftu (VS, język C#) oraz fakt, że firma parę miesięcy temu udostępniła platformę na zasadach Open Source.