

Konspekt projektu zaliczeniowego
z przedmiotu "Zaawansowane Programowanie
Obiektowe i Funkcyjne"

Eryk Dobrosz
Patryk Fijałkowski
Marcin Gackowski
Artur Haczek

5 grudnia 2018

1 Temat

Chcemy stworzyć grę typu multiplayer arena shooter z widokiem topdown. Gracze będą łączyć się z serwerem i pojedynkować się na arenie. Gra będzie nosić nazwę **BLASTBACK**.



BLAST
BACK

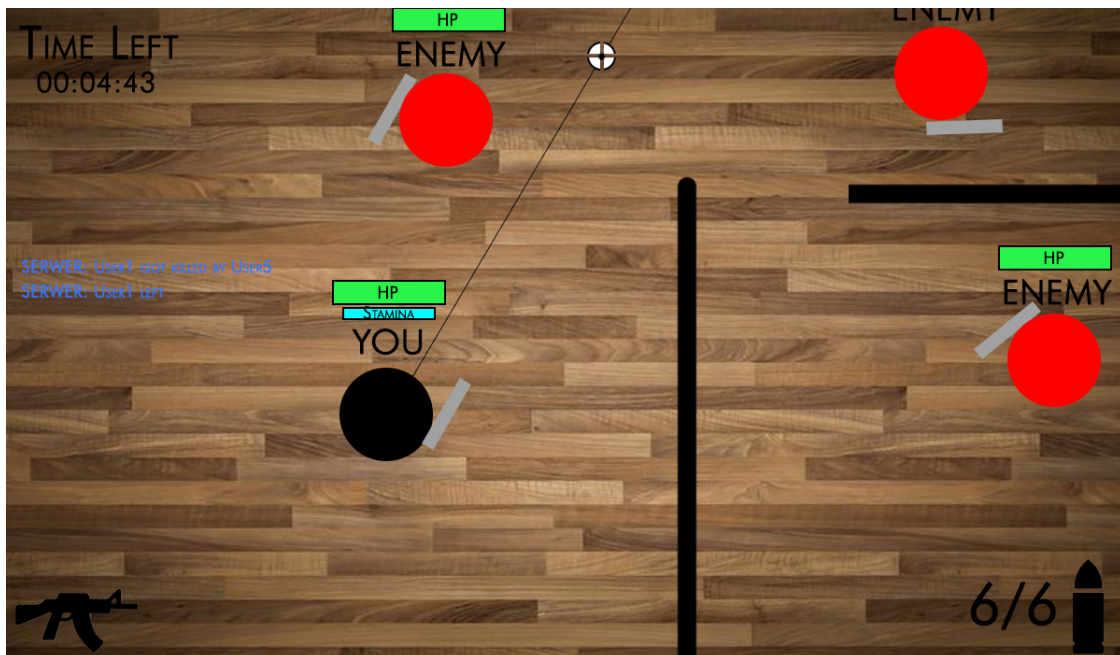
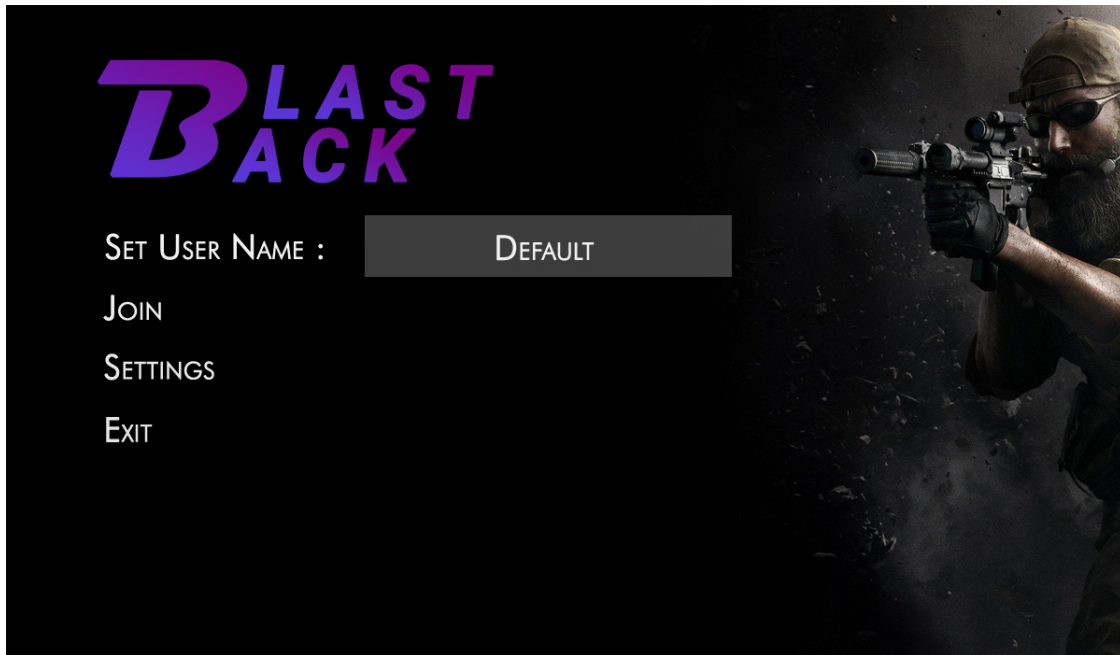
2 Technologie

Zdecydowaliśmy się na użycie technologii wymienionych poniżej:

- Kontrola wersji - **GitHub**.
Adres do repozytorium: <https://github.com/gsrmeen/blastback>.
- Planowanie zadań - **HacknPlan**.
Hacknplan usprawnia organizowanie zadań w podejściu agile'owym pozwalając na tworzenie sprintów, przydzielanie osób do zadań i dzieląc je zależnie od stopnia wykonania na wzór metodologii kanban.
- Tworzenie aplikacji graficznej - **jMonkeyEngine**.
jMonkeyEngine to silnik graficzny przeznaczony do tworzenia gier zaprogramowany całkowicie z użyciem Javy. Odciąża on programistę w znacznym stopniu ułatwiając rendering i dając dostęp do bibliotek umożliwiających m.in: symulowanie fizyki, generowanie terenu oraz tworzenie UI.
- Aspekt sieciowy - **SpiderMonkey**.
Jest to jedna z bibliotek dostępnych dla jMonkeyEngine, która dostarcza bazowych funkcjonalności potrzebnych przy tworzeniu gier typu multiplayer. Pozwala na stworzenie aplikacji typu serwer, aplikacji typu klient oraz odpowiednich wiadomości, które będą wykorzystywane w komunikacji pomiędzy tymi aplikacjami.

3 Mockups

Poniżej szkice podstawowego UI oraz widoku gry.



4 MoSCoW

Aby jasno ustalić priorytety w rozwoju naszej gry, wykorzystaliśmy technikę MoSCoW.¹

1. MUST have

- Komunikacja z serwerem
Gracze używają aplikacji klienckich by łączyć się z aplikacją serwerową. Projekt składa się z dwóch oddzielnych aplikacji serwera i klienta. Do komunikacji między tymi aplikacjami będziemy wykorzystywać protokół UDP ze względu na jego szybkość oraz to, że w przypadku utraty pakietów nie ma potrzeby ponownego ich wysyłania. W momencie doręczenia takiego pakietu byłby on już od dłuższego czasu nieaktualny (nie dawałby użytecznych informacji klientowi a dodatkowo dezorientował gracza). Serwer tworzy pokój do gry na ustalonej mapie, a następnie oczekuje na połączenia klientów. Gdy do pokoju połączy się wystarczająco dużo klientów (co najmniej dwóch) rozpoczynana jest runda o ustalonym wcześniej typie potyczki. Symulacja odbywa się po stronie serwera, który wysyła obecny stan symulacji wszystkim klientom protokołem UDP. Aplikacja klienta łączy się z serwerem po podaniu adresu IP serwera. Po ustaleniu połączenia klient pobiera bazowe informacje na temat rozgrywki (tryb gry, mapa) i na ich podstawie ładuje odpowiednie zasoby. Po załadowaniu wymaganych zasobów renderowany jest ekran gry i klient zaczyna renderować kolejne klatki na podstawie pakietów danych regularnie wysyłanych przez serwer. Aplikacja klienta przyjmuje również input użytkownika (tj. w którą stronę chce się poruszyć gracz, w którą stronę chce celować oraz podejmowane akcje, np. strzał, przeładowanie czy dash) i wysyła te dane do serwera używając protokołu UDP. Łączenie z serwerem oraz rozłączanie będzie korzystało z dodatkowych mechanizmów nałożonych na protokół UDP, aby serwer miał pewne dane dotyczące połączonych klientów i wysyłał dane tylko do obecnie aktywnych klientów.
- Podstawowa mechanika gry
 - a) Poruszanie postacią - Klawisze W,A,S,D
 - b) HP (hit points) - Gracz rozpoczyna grę posiadając 100 HP. Postać gracza umiera, gdy jego HP osiągnie 0
 - c) Celowanie - Punkt wskazywany przez myszkę odpowiada za kierunek lotu pocisku
 - d) Przeładowanie broni - Klawisz R odpowiada za akcję przeładowania, która trwa średnio 1 sekundę
 - e) Zadawane obrażenia - Obliczane są na podstawie broni trzymanej przez gracza
- Podstawowe UI - Menu posiada 4 podstawowe przyciski
 - a) Set User Name - pozwala na edycję nazwy użytkownika wyświetlanej podczas rozgrywki
 - b) Join room - Inicjuje połączenie z serwerem o podanym adresie IP, jeśli nie podano żadnego adresu IP wykorzystywany jest adres z ustawień
 - c) Settings - Pozwala na ustawienie adresu IP domyślnego serwera z którym będzie łączyć się klient i konfiguracji gry
 - d) Exit - Wyjście z aplikacji

¹Podane niżej wartości numeryczne mogą ulec zmianie w trakcie developmentu.

- 3 rodzaje broni
 - a) Uzi - niskie obrażenia, 20 sztuk amunicji na magazynek, szybkie przeładowanie
 - b) Pistolet - średnie obrażenia, 7 sztuk amunicji na magazynek, szybkie przeładowanie
 - c) Shotgun - duże obrażenia, 2 sztuki amunicji na magazynek, powolne przeładowanie
- 1 tryb gry: deathmatch

Gracze po śmierci pojawiają się w losowym miejscu na mapie po upływie 5 sekund. Warunkiem zakończenia rozgrywki jest upłynięcie czasu lub osiągnięcie limitu zabójstw przez jednego z graczy.
- Globalne powiadomienie o zabójstwach
 - a) Serwer wyświetla powiadomienia o każdym zabójstwie w postaci komunikatu widocznego dla każdego gracza, w formacie *X got killed by Y*
 - b) Wyświetlane są także powiadomienia o dołączeniu do/opuszczeniu serwera przez graczy.
- Scoreboard
 - a) W trakcie rozgrywki można wyświetlić tablicę z wynikami, przytrzymując klawisz Tab
 - b) Scoreboard wyświetla statystyki graczy zależne od typu rozgrywki, takie jak liczba zabójstw i śmierci dla trybu deathmatch, czy liczba zdobytych przez drużynę flag w trybie CTF
 - c) Scoreboard wyświetla także warunki potrzebne do wygranej - liczba zabójstw/zdobytych flag

2. SHOULD have

- Dodatkowe tryby gry
 - a) Capture the flag (CTF) - Drużynowy tryb gry. Gracze muszą przejąć flagę wroga i dostarczyć ją do wyznaczonego punktu.
 - b) Team deathmatch - Drużynowa wersja trybu deathmatch.
- Chat dla graczy - po wybraniu klawisza T otwiera się okienko konwersacji, w którym gracz przechodzi do trybu pisania z innymi graczami. Wiadomości są widoczne dla wszystkich graczy.
- Dodatkowe VFX
 - a) Odgłosy broni, odgłosy trafienia
 - b) Wygląd pocisków
 - c) Światło spowodowane wystrzałem
 - d) Odgłosy kroków
- Dynamika - Kamera uwzględnia kierunek celowania przez gracza, przesuając się w kierunku w którym zwrócona jest postać gracza (stopień odsunięcia kamery zależy też od odległości kursora od postaci gracza).
- Dodatkowe mechaniki poruszania się postaci
 - a) Szybkie poruszanie - Trzymanie shifta pozwala na poruszanie się z prędkością równą 1.5 prędkości bazowej

- b) Dash - Po naciśnięciu spacji postać gracza bardzo szybko przemierza ustalony dystans w obecnym kierunku poruszania się
- c) Stamina - Energia wykorzystywana przy szybkim poruszaniu oraz dashach. Stamina regeneruje się o 10 punktów procentowych na sekundę, jeśli gracz nie jest w trybie szybkiego poruszania
- Synchronizacja symulacji klienta z serwerem
Jest to mechanizm który zmniejsza średnie opóźnienie symulacji po stronie klienta, tym samym zwiększając poczucie kontroli gracza nad awatarem. W tym rozwiązaniu aplikacja kliencka nie renderuje obecnej klatki tylko na podstawie danych odebranych od serwera, ale równolegle z serwerem wykonuje własną symulację na podstawie obecnie dostępnych danych. Dzięki temu gracz po dostarczeniu inputu będzie mógł od razu zobaczyć efekty swoich akcji. Ze względu na to, że klient nie dostaje na bieżąco wszystkich danych, jego symulacja może odbiegać od tej wykonywanej po stronie serwera, dlatego należy zaimplementować też mechanizm synchronizacji klienta z serwerem. W większości przypadków efekty tej synchronizacji są niezauważalne, ale gdy nastąpiła np. utrata pakietów może się pojawić nagle zmiana w obrazie po stronie klienta. W takim przypadku można zastosować interpolację (od symulacji klienta do symulacji serwera) i zamiast nagłego przeskoku, który może być bardzo dezorientujący, otrzymuje się tak zwany rubber-banding.

3. COULD have

- Dodatkowe mapy - Dodatkowy zestaw map, z z którego można korzystać, tworząc nową grę na serwerze
- System przedmiotów
 - a) Gracz posiada ekwipunek, dostęp do poszczególnych przedmiotów odbywa się wybierając klawisze 1, 2, ..., 0
 - b) Wybierając przedmiot i klikając X gracz wyrzuca go z ekwipunku, przedmiot pojawia się przed postacią na mapie
 - c) Liczba magazynków jest ograniczona, zebranie dodatkowego magazynku z planszy zwiększa liczbę dostępnych przeładowań
 - d) Po śmierci z gracza pojawiają się przedmioty, które posiadał w ekwipunku oraz magazynki
- Fog of war - Efekt rozmycia i przyciemnienia elementów planszy, które nie znajdują się w kierunku, w którym odwrócony jest gracz
- Minimapa - Miniaturowa mapa znajdująca się w rogu ekranu, na której widać poszczególne jej elementy. W przypadku, gdy wróg wejdzie w zasięg wzroku gracza lub gdy strzeli, pojawi się stosowna informacja na minimapie.

4. WON'T have

- Konta użytkowników
- Tworzenie roomów po stronie klienta