

Kostecki Kacper
300236

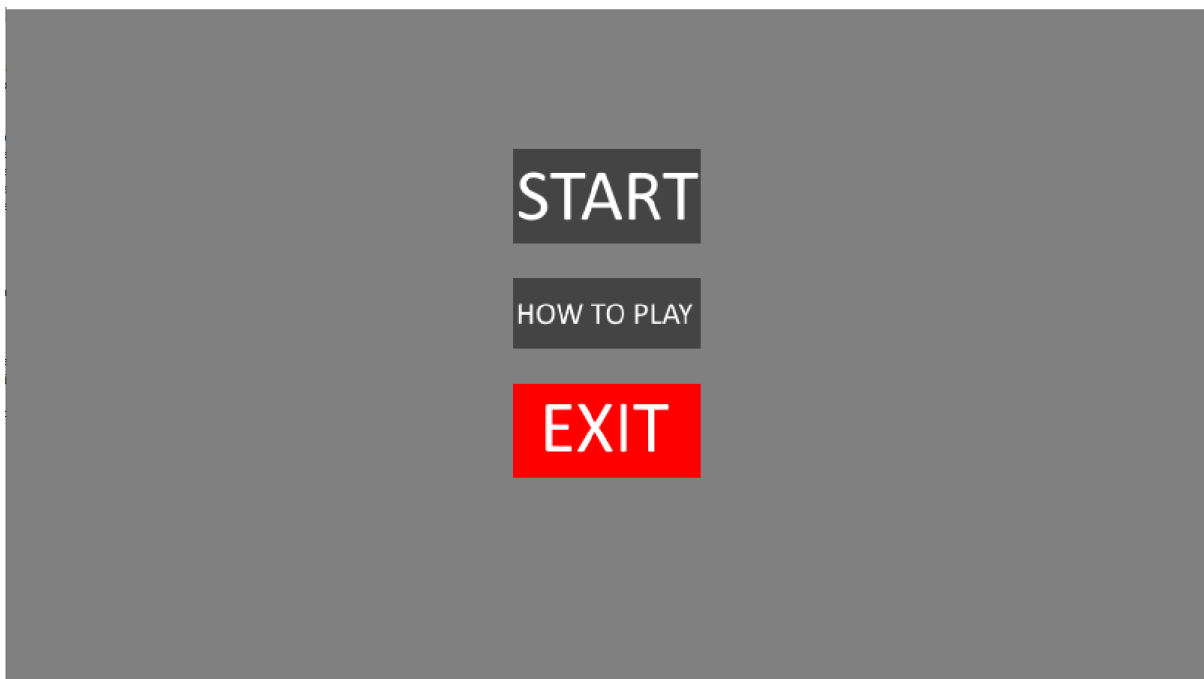
Kulpa Katarzyna
300248

Projekt PROZ sprawozdanie

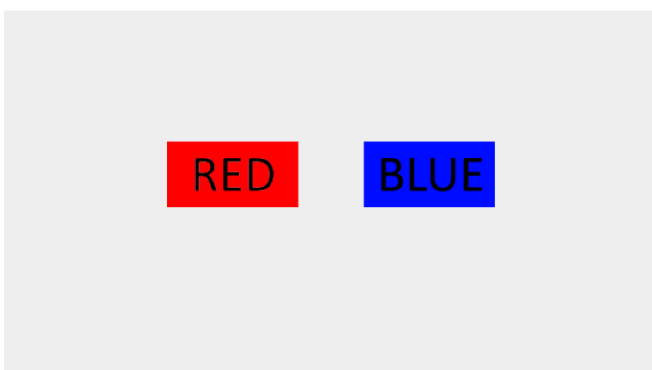
Tower defense multiplayer – gra dla dwóch lub więcej graczy.

Ostateczna wersja projektu różni się lekko od planowanej wersji. Zasady ostatecznej wersji:

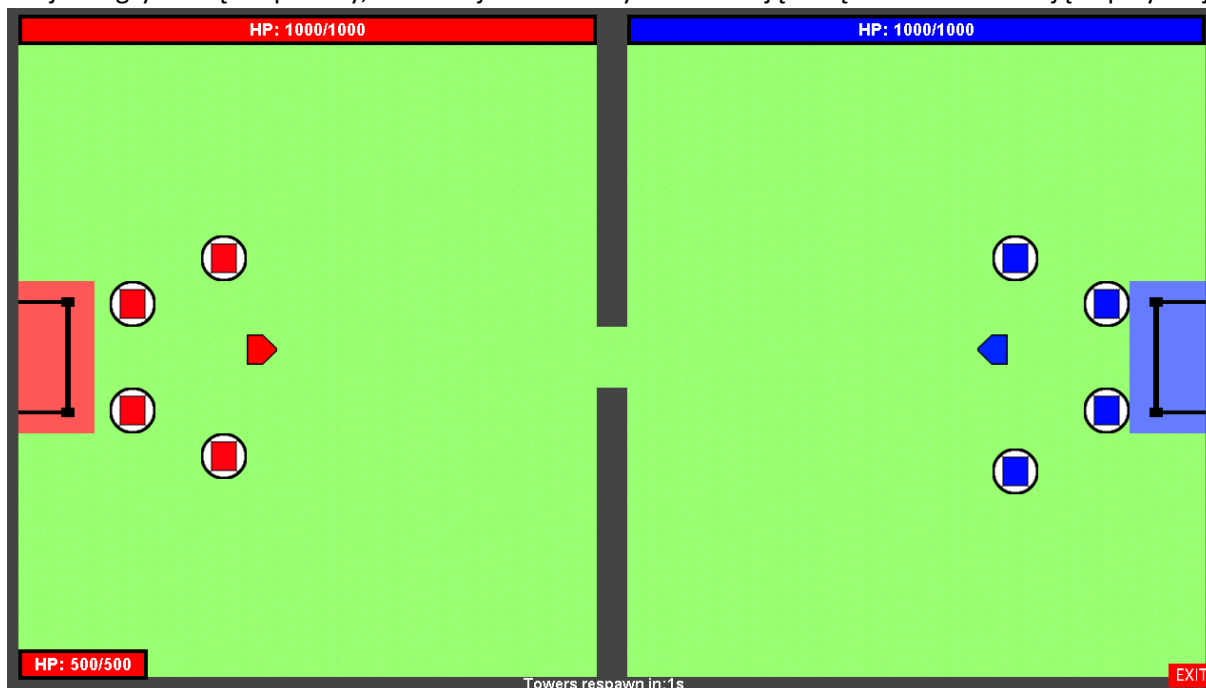
Zaczynając grę wyświetlone zostaje GUI.



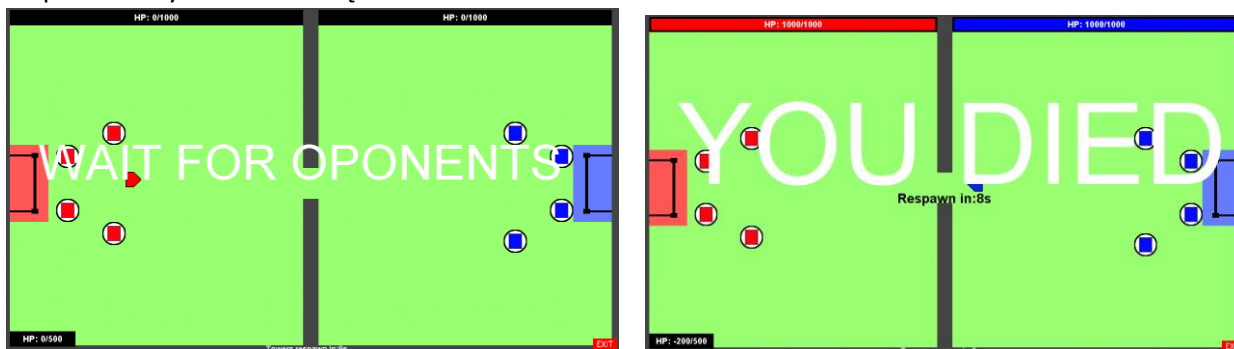
Aby zobaczyć zasady gry należy nacisnąć HOW TO PLAY, aby zacząć grę START, a by wyjść EXIT. Kiedy wybierzemy opcję START musimy dokonać wyboru w której drużynie chcemy grać (RED lub BLUE). Następnie wybieramy czy chcemy utworzyć nowy serwer lub czy chcemy dołączyć do już istniejącego serwera. W następnym kroku jeżeli zdecydowaliśmy o dołączeniu do serwera musimy podać jego ip. Następnie podajemy naszą nazwę użytkownika. Po jej podaniu przechodzimy do właściwej rozgrywki. Jeżeli jesteśmy hostem i żaden z przeciwników jeszcze nie dołączył do rozgrywki, wyświetla się napis, iż musimy poczekać na przeciwnika. Dopóki nikt nie dołączy do drużyny przeciwnej gracze nie mogą się ruszać, atakować itd.



Akcja rozgrywa się na planszy, na której każda drużyna ma swoją bazę oraz 4 wieże stojące przy niej.



Gracze poruszają się przyciskami WSAD. Ponad to mogą strzelać do siebie poprzez kliknięcie spacji, każdy trafiony w przeciwnika pocisk odbiera mu 100 jednostek życia. Użytkownik ma łącznie 500 jednostek życia (aktualny stan swojego zdrowia może zobaczyć w rogu), kiedy je straci pojawia mu się komunikat, że umarł i musi poczekać 10 sekund, aby móc grać dalej. Użytkownicy mogą się leczyć poprzez powrót do bazy – pola odpowiedniego koloru drużyny leczą każdego gracza stojącego na nich 10 punktów życia na sekundę.



Gra kończy się w momencie, kiedy któraś z drużyn zniszczy bazę przeciwnika. Można ją zacząć niszczyć tylko wtedy, kiedy wszystkie wieże są zniszczone. Każdy trafiony pocisk zabiera 100 jednostek życia bazy z 1000 możliwych. Utrudnieniem jest to, że wszystkie wieże odbudowują się po upływie konkretnego czasu, który wyświetla się na dole planszy. Kiedy któraś z drużyn zniszczy bazę wyskakuje komunikat o wygranej, natomiast drużynie przegranej wyskakuje komunikat o przegranej. Paski zdrowia obydwu baz są pokazane na górze planszy.

Kiedy gracz chce opuścić grę może nacisnąć przycisk EXIT lub po prostu zamknąć okno gry.

Przydział obowiązków:

- Kacper Kostecki:
 - Podstawy mechaniki gry: system kolizji, generowanie mapy, obsługa jednostek (entity),
 - Stworzenie gui oraz wszelkich aspektów wizualnych gry,
 - Stworzenie silnika gry – obsługa „renderowania” oraz „ticków”,
 - Testy jednostkowe.
- Katarzyna Kulpa:
 - Obsługa sieciowa gry: stworzenie modułu klienta oraz serwera,
 - Obsługa przesyłania pakietów między serwerem oraz klientami,
 - Implementacja mechaniki gry na platformę sieciową – odpowiednie dopasowanie mechaniki gry, aby było możliwe granie sieciowe w czasie rzeczywistym,
 - Usuwanie wszelkich błędów, które wystąpiły w trakcie implementacji nowych mechanik.

Opis wykorzystanych rozwiązań w kodzie źródłowym:

Najważniejszym elementem kodu źródłowego gry jest klasa Game. To w niej inicjalizowany jest główny wątek odpowiedzialny za działanie gry oraz tzw. „game loop”, dzięki któremu nie wykonujemy określonych działań „renderowania” oraz „tickowania” cały czas, a jedynie przez pewną określoną liczbę razy na sekundę, co znacząco poprawia wydajność. Wszelkie ważne zmienne lub metody określonych klas, do których chcielibyśmy się dostać z dowolnego miejsca kodu są przechowywane w klasie Handler.

Plansza gry jest tworzona na bazie „tile”, czyli pojedynczych, powtarzalnych kwadratów, które razem tworzą całą mapę. Dzięki takiemu rozwiązaniu możemy z łatwością modyfikować planszę dodając kolejne przeszkody lub wieżyczki.

Wszelkie jednostki są tworzone według tej samej koncepcji, dlatego została stworzona abstrakcyjna klasa Entity. W przypadku przemieszczających się jednostek została również stworzona klasa Creature, która dziedziczy po klasie Entity oraz obsługuje poruszanie się, czego główna klasa jednostek nie obsługuje, ponieważ nie każda jednostka musi się przemieszczać po planszy.

Wszystkie jednostki są przechowywane w menedżerze, który jest przypisywany do mapy. Każdy gracz posiada swój własny menedżer. Wszelkie wejścia od gracza (klawiatura, myszka, okno) również są obsługiwane w odpowiednich menedżerach.

Wszelkie sieciowe klasy są przechowywane w paczce net. Klasa GameServer obsługuje serwer stworzony przez gracza. Jest ona wywoływany tylko i wyłącznie przez gracza, który chce uruchomić serwer, czyli zostać hostem gry. Wywołuje on również, tak jak wszyscy inni gracze, którzy chcą dołączyć do rozgrywki, klasę GameClient, w której obsługiwane są wszystkie aspekty rozgrywki po stronie klienta np. tworzenie pocisków wystrzelonych przez przeciwników lub dodawanie samych przeciwników do gry. GameClient oraz GameServer komunikują się między sobą za pomocą protokołu UDP wysyłając informacje w pakietach. Każdy pakiet to string zawierając na samym początku swoje ID oraz wszelkie potrzebne informacje, które muszą dotrzeć od jednego gracza do wszystkich pozostałych.

Testy jednostkowe głównie skupiają się na kolizji jednostek.

Dokładny opis kodu źródłowego to komentarze w nim zawarte:

Kod źródłowy: <https://github.com/kkosteck/PROZ>