

Kamień Milowy I

Lista czynności zaplanowanych na pierwszy kamień milowy:

- Opracowanie tematyki gry.
- Zaprojektowanie sposobu przechowywania mapy w plikach.
- Zarządzanie plikami w aplikacji.
- Zaimplementować prototyp silnika gry z podstawowymi zmiennymi i funkcjami.
- Zaimplementowanie gracza z podstawowymi funkcjami np. poruszanie się.
- Krótkie sprawozdanie z postępu pracy.

Zrealizowaliśmy wszystkie zaplanowane przez nas punkty oraz poszerzyliśmy nasze cele o inne punkty z drugiego kamienia milowego.

Gra osadzona w czasach średniowiecznych z nutką fantasy.

Naszym założeniem było podzielić mapę na sektory, oraz przystosować kod do prostego dołączania obiektów do dowolnego sektora. Zrealizowany przez nas sposób na przechowywanie mapy w pliku został zaprojektowany następująco: Odczytujemy litery z pliku, gdzie każda litera ma swój odpowiednik dla danej tekstury podłoża sektora. Na podstawie litery w kontenerze (map) jest wyszukiwana nazwa odpowiedniej tekstury. Na podstawie nazwy w drugim kontenerze map jest wyszukiwany wskaźnik tej tekstury. Wskaźnik jest wysyłany do obiektu klasy MapSector, którego obiekty są przechowywane w wektorze wewnątrz obiektu GameMap, która jest częścią obiektu Engine.

Zoptymalizowaliśmy program poprzez zmianę ilości renderowanych sektorów. Aktualnie renderujemy tylko 15 sektorów, 5 w poziomie i 3 w pionie.

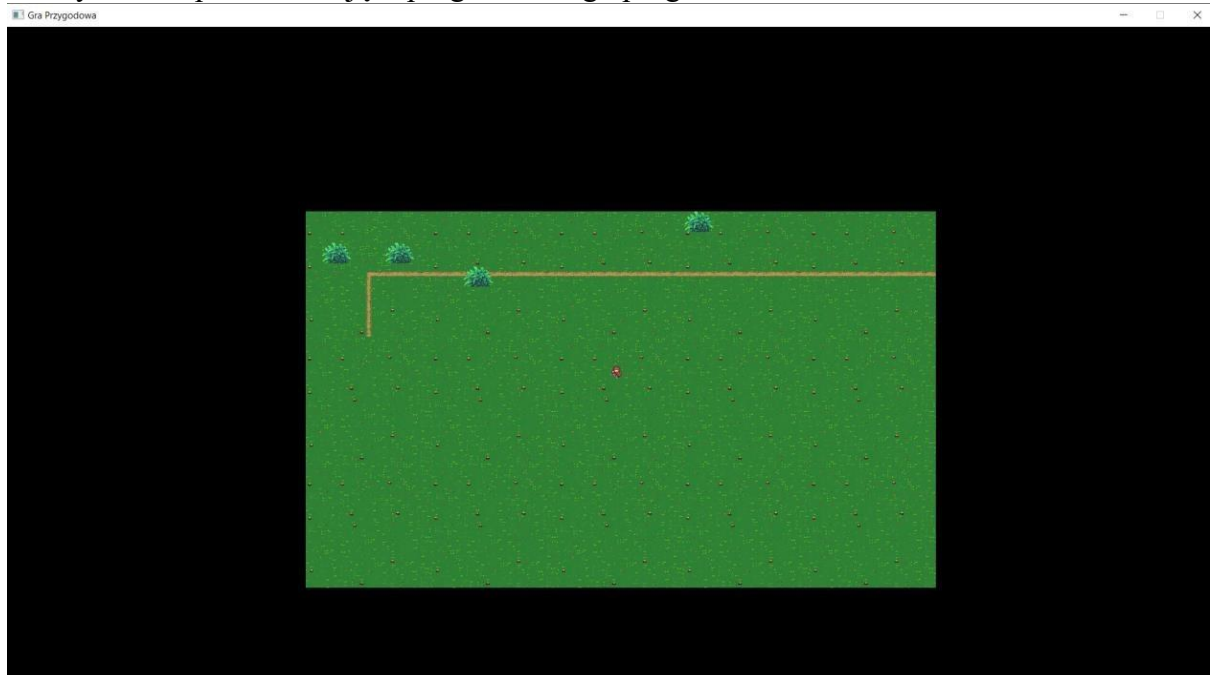
Silnik zbiera główne tworzone obiekty np. obiekt mapy, gracza. Zawierają się w niej funkcje aktualizujące oraz renderujące. Engine odpowiada za wykrywanie i obsługę kolizji oraz wydarzeń z biblioteki SFML. Mierzony jest czas jednego obrotu pętli, wykorzystywany przy obliczaniu dystansu jaki gracz pokona w ciągu jednego obrotu pętli. Stworzyliśmy kamerę podążającą za graczem, który może się poruszać po mapie. Kamera działa na zasadzie, że obiekt klasy View otrzymuje co każdą iterację pętli położenie gracza i patrzy na te współrzędne. A okno gry patrzy na kamerę.

Gracz jest obiektem, który tworzymy w obiekcie Engine, ma zdolność poruszania się w górę, dół, lewo, prawo. Posiada on ograniczenia w postaci braku możliwości wyjścia poza mapę. Gracz ma swoją pięcioklatkową animację, która jest wyświetlana w zależności od kierunku ruchu, animacje te są wyświetlane kolejno z określoną częstotliwością.

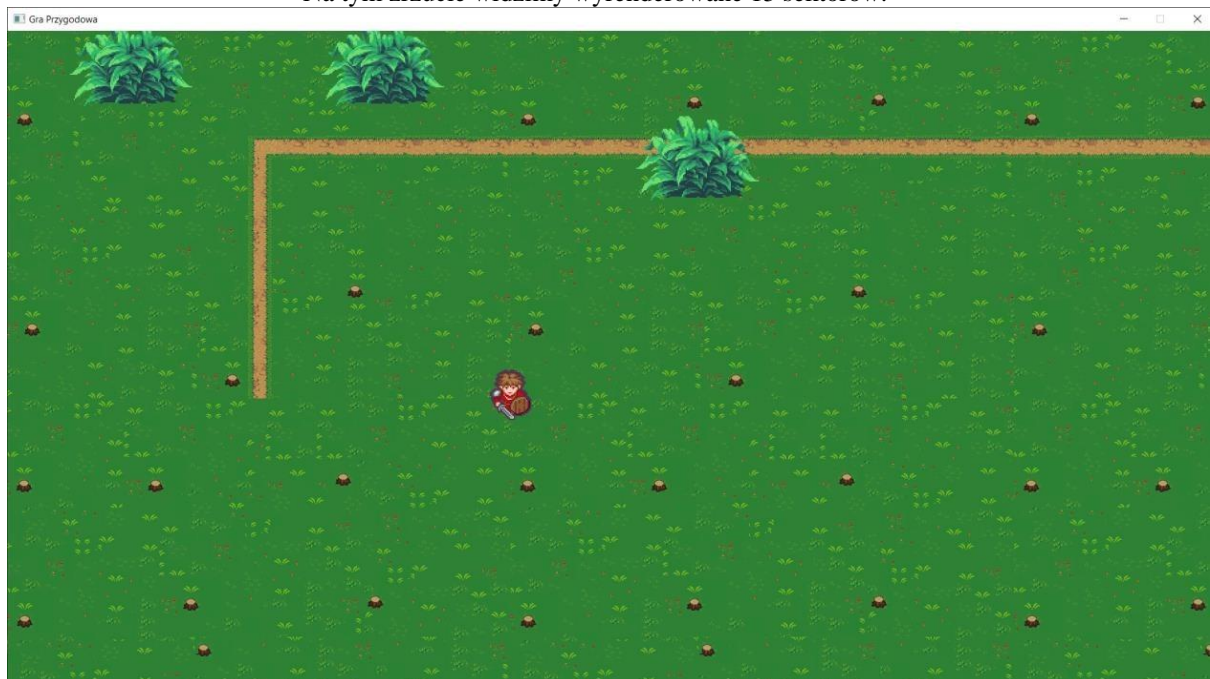
Czynności z drugiego kamienia milowego:

Zaprojektowaliśmy system umożliwiający wstawianie obiektów na mapie. Funkcja sprawdzania kolizji jest usprawniona o to, że sprawdzane są jedynie obiekty które znajdują się na tym samym sektorze co gracz.

Zrzuty ekranu przedstawiające progres naszego programu.



Na tym zrzucie widzimy wyrenderowane 15 sektorów.



Na tym zrzucie widzimy, że perspektywa blokuje nam widok czarnych krawędzi okna aplikacji.

Podsumowanie

Zrealizowane zostały wszystkie założenia na pierwszy kamień milowy wypisane w harmonogramie. Program poprawnie działa i się kompiluje

Kamień Milowy II

Lista czynności zaplanowanych na drugi kamień milowy:

- Rozbudowanie klasy gracza o dodatkowe elementy np. życie, inwentarz.
- Implementacja obiektów i ustawienie ich na mapie, dodanie stworzeń oraz NPC.
- Dodanie prostych zadań.
- Opracowanie menu głównego.
- Poprawa ewentualnych błędów i testowanie gry.
- Zoptymalizowanie działania programu.
- Sprawozdanie z postępu pracy.
- Opis działania kodu.
- Generacja dokumentacji w programie doxygen.

Klasa gracza została rozbudowana o elementy takie jak:

- możliwość zadawania obrażeń przeciwnikom.
- życie, gdy spadnie do 0, powracamy do menu głównego gry.
- ekwipunek, w którym posiadamy sloty na przedmioty, które nasza postać może założyć.
- inwentarz, który pozwala nam przetrzymywać przedmioty podniesione z ziemi, które zdobyliśmy zabijając potwory.
- statystyki, które zwiększają się wraz z założonymi przedmiotami. Każdy przedmiot ma swoje parametry. Każda statystyka wpływa na rozgrywkę, np. w zależności od tego ile mamy pancerza otrzymujemy mniej obrażeń.

Poprawiona została kolizja gracza z obiektami. Naszym wcześniejszym podejściem było wywołanie kolizji w momencie jej wykrycia, natomiast po przemyśleniach doszliśmy do wniosku, aby program w każdej iteracji pętli zapamiętywał pozycje gracza, jeżeli nie dochodzi do kolizji, w przypadku, gdy do kolizji dojdzie, gracz jest cofany na zapamiętaną bezpieczną pozycję.

Obiekty są dodawane z pliku tekstowego, za pomocą interpretera. Ustawiamy je na konkretnym sektorze na odpowiedniej pozycji, skalujemy je oraz przemieszczamy ich offset. Mamy możliwość dodania tablicy obiektów, bądź pojedynczego obiektu.

Nasza gra posiada aktualnie jeden typ NPC, oraz wroga szkieleta. Możemy ustawiać na mapie „dowolną” ilość obiektów entity, do których należą.

Czynności wykonywane przez wroga stworzenia:

Ich celem jest, gdy gracz będzie w pobliżu podążają w jego kierunku. Natomiast, gdy oddalimy się na pewną odległość, cofają się do swojej pozycji. Jeżeli nie ma gracza w pobliżu, szkielety patrolują okolicę w losowych kierunkach na określoną odległość. Po zabiciu szkieleta możliwe jest, że wypadnie przedmiot, który gracz może podnieść.

Czynności wykonywane przez NPC:

Sojusznice stworzenia również mogą(nie muszą) patrolować okolicę, nie wchodzi w interakcje z wrogimi stworzeniami, nie możemy ich zranić, natomiast możemy wziąć od nich misję. Misji możemy wziąć wiele, lecz tylko jedną od jednego NPC, wykonują się po kolei.

Menu główne to trzy przyciski ustawione na tle, jeden z nich EXIT wychodzi z gry, przycisku CONTINUE nie da się wcisnąć przed rozpoczęciem gry, a po rozpoczęciu gry, gdy wejdziemy do menu głównego możemy wrócić do naszego stanu gry. Przycisk NEW GAME pozwala na rozpoczęcie gry od nowa.

OPIS STEROWANIA W GRZE:

W-ruch gracza do góry

S- ruch gracza w dół

A- ruch gracza w lewo

D- ruch gracza w prawo

E- wyświetlenie okna ekwipunku

I-wyświetlenie okna inwentarza

ESC-wyjście do menu głównego

SPACE-atak

Q-wyświetlenie okna z zadaniami

R-pobranie zadania

Ruch w inwentarzu:

↑-ruch do góry

↓- ruch w dół

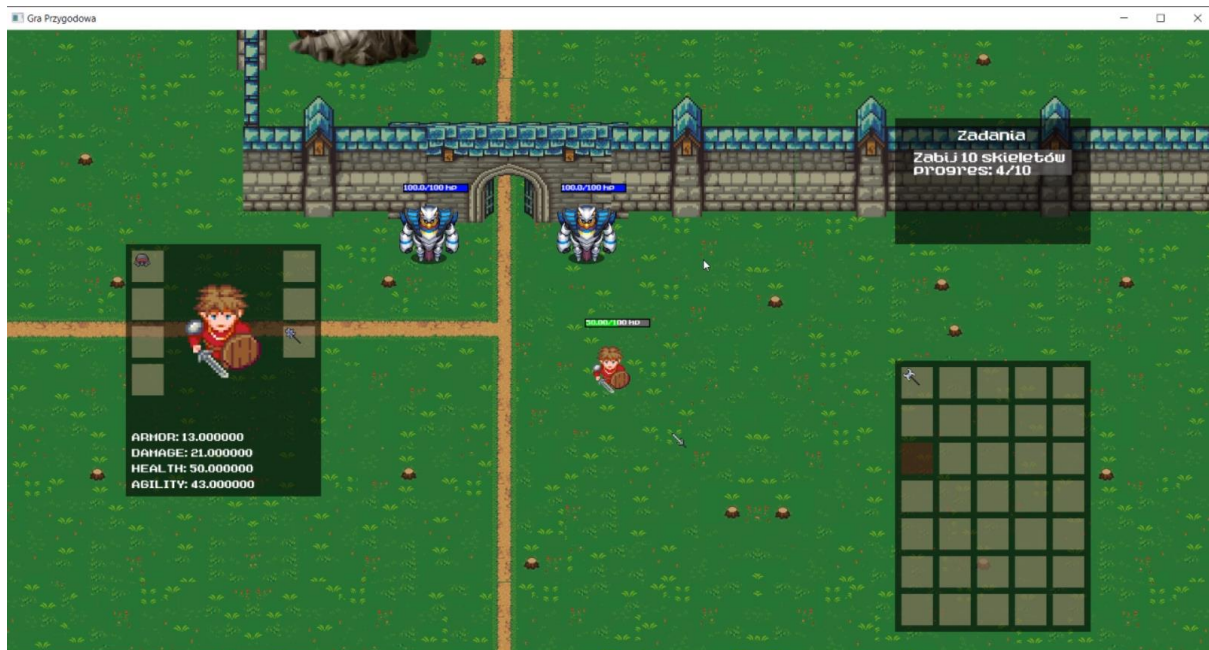
← - ruch w lewo

→- ruch w prawo

Backspace-usuwanie przedmiotu z inwentarza

ENTER-zakładanie przedmiotu

Zrzut ekranu z otwartym oknem zadań i progresem zadań. Widoczny ekwipunek oraz inwentarz, z kilkoma przedmiotami.



Zrzut ekranu z widocznym NPC mającym dla nas misje.



Zrzut ekranu z widokiem na menu główne



Skład zespołu:

1. Artur Graba
2. Karol Gardian
3. Damian Gajda

Podsumowanie

Zrealizowane zostały wszystkie założenia na drugi kamień milowy wypisane w harmonogramie. Program poprawnie działa i się kompiluje.

Obrona I termin

Na obronę projektu nasz zespół zgodnie z zaleceniami dodał funkcjonalność w postaci eliksirów otrzymywanych po ukończeniu zadania. Ten przedmiot tak jak pozostałe pojawia się w ekwipunku. Po wybraniu go i naciśnięciu klawisza ENTER przedmiot znika jednocześnie przywracając graczowi 30 punktów życia.



Rozwiązany został również problem z migającym ekranem podczas naciskania przycisku CONTINUE. Od tej pory po wciśnięciu tego przycisku po śmierci program nie podejmuje próby wznowienia rozgrywki. Jednak można ją zacząć od nowa przyciskiem NEW GAME.

Projekt uznajemy za zakończony, wszyscy członkowie zespołu mają taki sam wkład pracy. Kod był pisany w środowisku Visual Studio przy pomocy rozszerzenia Live Share.