

SPRAWOZDANIE

z realizacji projektu Programowanie obiektowe Java

Temat projektu: Transylwania - gra horror – survival – multiplayer.

Zespół projektowy: Maciej Bandura, Marcin Ślusarczyk, Szymon Śmigłaski; 2ID14B 2021/2022

Środowisko programistyczne (IDE): JetBrains IntelliJ

Język programowania: Java – wersja (17.02 LTS). Framework(okna): Swing. Połączenie serwerowe –Java socket's – TCP.

Koordynator projektu: mgr inż. Grzegorz Pawiński

1. Projekt

Gra została wykonana jako projekt z przedmiotu: Programowanie obiektowe (Java). Kod źródłowy napisany w całości w języku Java. Bazowany na wersji 17.02 LTS. Nie została zastosowana żadna zewnętrzna – dodatkowa biblioteka graficzna. Wszystko zostało oparte o funkcjonalności zawarte w podstawowej wersji języka Java.

Pomysłów na grę było kilka, podczas pierwszych spotkań projektowych w zespole toczyliśmy ożywione dyskusje, padło kilkanaście propozycji odnośnie samej tematyki gry, jak i jej typu oraz funkcjonalności. Padały propozycje gry w rzucie izometrycznym, jak i widoku z trzeciej osoby. W ostateczności urodził się pomysł, gry typu survival, oferującej dla gracza widok „z góry”. Nie chcieliśmy tworzyć gry opartej o dużą ilość graczy, gdyż wiedzieliśmy, że będzie to trudne do testowania.

2. O grze

Gra w systemie multiplayer (Jeden serwer – Wiele klientów). Jest to: asymetryczny survival horror. Minimalna ilość graczy biorących udział w rozgrywce to 2. Maksymalna ilość graczy wynosi 8. Mechanika gry składa się z czterech modułów/etapów: poczekalnia, wstęp do rozgrywki, rozgrywka i podsumowanie (restart rozgrywki).

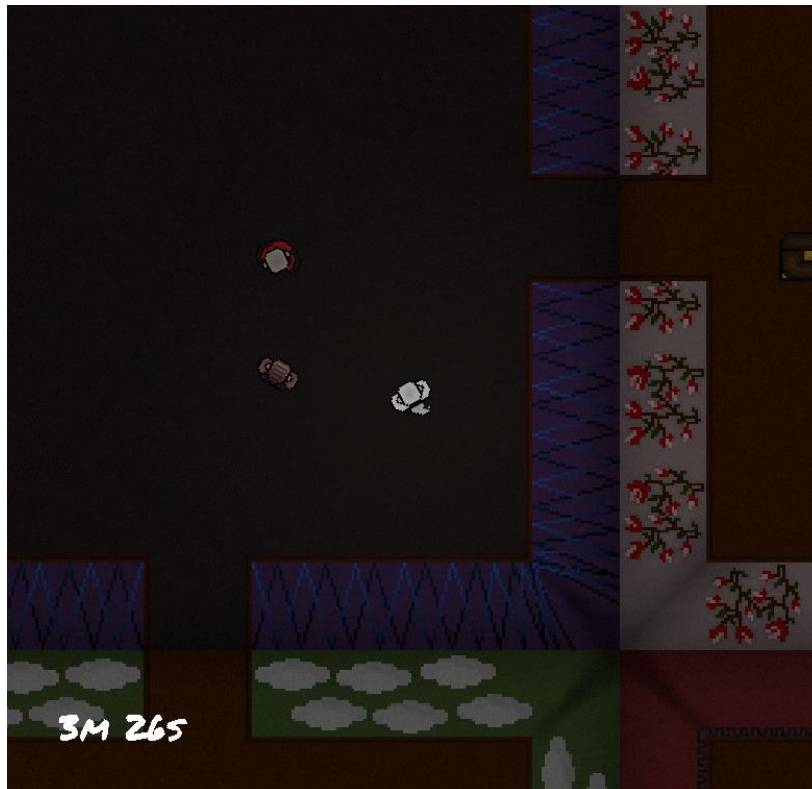
Po uruchomieniu gry ukazuje nam się **menu** w którym gracz wpisuje: nazwę użytkownika oraz adres IP i port serwera do którego chce dołączyć. Mamy dwie możliwości: możemy dołączyć do istniejącego już serwera lub możemy stworzyć nową instancję serwera i od razu do niej dołączyć. Po poprawnym dołączeniu do serwera i pojawieniu się jeszcze minimum jednego gracza, rozpoczyna się odliczanie do rozpoczęcia rozgrywki. Jest to czas pozwalający na dołączenie większej ilości graczy. Będzie on wyliczany na nowo za każdym razem gdy na serwer dołączy jakiś nowy gracz.



menu gry

3. Rozgrywka

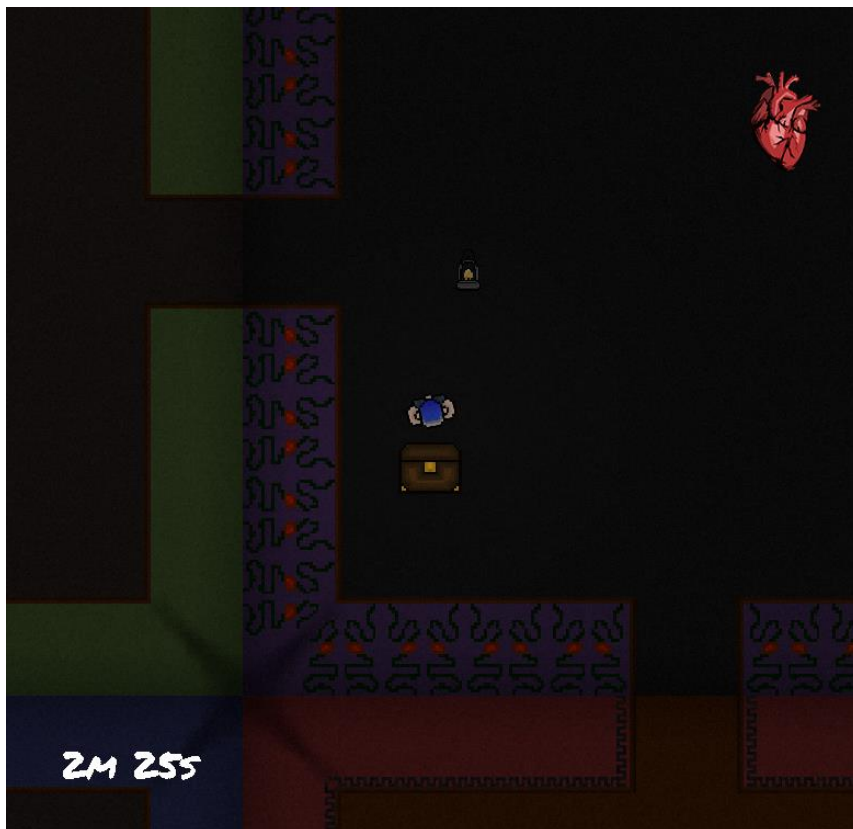
Rozgrywka opiera się na zasadzie: jeden gracz jest „wampirem” pozostali gracze „ludźmi”, tzw. Survivorzy. Gracz ma do wyboru kilka postaci, różniących się wyglądem. Każdy „człowiek” ma trzy życia. Poziom życia gracza jest widoczny w prawym górnym rogu ekranu, jest prezentowany za pomocą „animacji” serca. Zadaniem wampira jest „dopaść” wszystkich ludzi przed upłynięciem określonego czasu. Czas rozgrywki jest ograniczony. Jest on zależny od ilości graczy biorących udział w rozgrywce. Jeżeli wampir dopadnie wszystkich ludzi – wygrywa. Jeżeli jakiemuś człowiekowi uda się „ujść z życiem” wtedy wygrywają Survivorzy. Jeżeli czas rozgrywki minie i wampirowi nie uda się „zabić” wszystkich ludzi wtedy wampir przegrywa. W przypadku gdy jakiś człowiek umiera, zostaje on wyłączony z rozgrywki do postaci ducha – może on obserwować rozgrywkę lecz nie może w niej brać czynnego udziału. Nie jest on wtedy widoczny przez pozostałych – „żyjących” graczy.



postacie dostępne w grze: wampir, survivor i duch

Podczas trwania rozgrywki, dla gracza dostępne są lampki oraz kryjówki w formie skrzynek - są one przydatne w szczególności dla postaci Survivora, dzięki schowaniu się w skrzynce, przez pewien czas wampir go nie widzi oraz nie działa „instynkt” wampira który podpowiada mu pozycje „ludzi”. Również wampir ma możliwość chowania się w skrzynce dzięki czemu może zaskoczyć Survivora w najmniej oczekiwanym momencie. Lampki natomiast służą do manipulowania czasem rozgrywki. Zarówno „ludzie” jak i „wampir” mogą z nich korzystać. Postać Survivora ma możliwość zapalania lampek co umożliwia oślepianie wampira i dzięki temu czas rozgrywki zostaje skrócony. Wściekły wampir ma możliwość niszczenia zapalonych lampek co powoduje wydłużenie czasu rozgrywki.

Po zakończeniu rozgrywki każdemu graczowi ukazuje się komunikat informujący o tym kto wygrał. Po chwili gracze zostają przeniesieni do „lobby” gdzie będą oczekiwać na rozpoczęcie kolejnej rozgrywki. Jeżeli podczas trwania rozgrywki dołączą nowi gracze do serwera, to w tym momencie zostaną oni dołączeni do poczekalni i kolejna rozgrywka będzie mogła się rozpocząć z większą ilością graczy.



Wybrane przedmioty dostępne w grze: skrzynka i lampka

4. Zabezpieczenia powiązane z rozgrywką typu multiplayer

Nasza gra została zabezpieczona pod kątem wszelkich wypadków mogących wystąpić w trakcie rozgrywki typu multiplayer takich jak:

- losowe odłączenie się graczy od serwera,
- dołączenie graczy podczas trwania rozgrywki,
- problemy z łączem internetowym i synchronizacją.

Przyjęte założenia:

- jeżeli na serwerze pozostanie jeden gracz to wygrywa on rozgrywkę i zostaje przeniesiony do „lobby” gdzie będzie oczekiwał na dołączenie innych graczy,
- jeżeli podczas rozgrywki grę opuści wampir to automatycznie rozgrywka się kończy i wygrywają Survivorzy,
- jeżeli podczas trwania rozgrywki dołączy nowy gracz to staje się On postacią ducha.
- W przypadku gdy, gracz który utworzył instancję serwera zakończy rozgrywkę i wyłączy grę, pozostali gracze również zostaną odłączeni od serwera i zostaną poinformowani o tym fakcie odpowiednim komunikatem na ekranie.

5. Komunikacja Klient – Serwer

Jest to pierwszy projekt tego typu jakiemu musieliśmy stawić czoła. W początkowych fazach projektu pomiędzy klientem a serwerem przesyłaliśmy tylko ciągi znaków. Lecz po kilku godzinach pracy stwierdziliśmy, że serializacja danych będzie bardzo trudna w przypadku samych obiektów typu String. Ostatecznie przesyłamy pakiety zawierające złożone przez nas obiekty. Wyróżniamy kilka typów pakietów jakie przesyłamy:

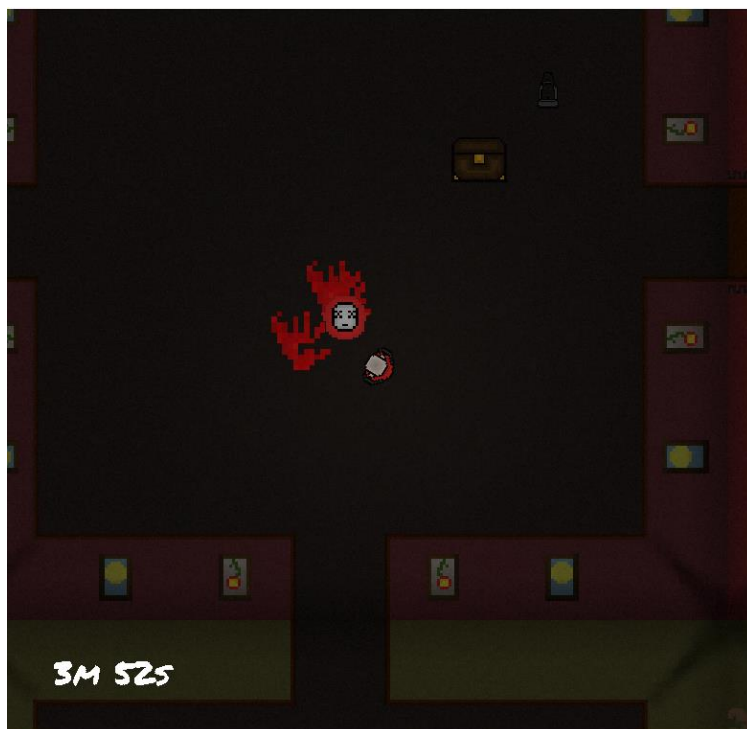
- PlayerListPacket – pakiet zawierający wszystkie informacje o graczach.
- PropsPacket – pakiet zawierający informacje o propach (plamy krwi, trupy)
- FlagPacket – pakiet potrzebny do synchronizacji najważniejszych flag na serwerze. Takich jak: status gry czy manipulacja czasem gry.

- LampsListPacket – pakiet odpowiadający za synchronizację statusu lamp na mapie. Synchronizuje informacje o zapaleniu lub zniszczeniu lampek.
- ChestsListPacket – pakiet odpowiedzialny za synchronizację statusu skrzynek na mapie. Pozwala on na uzyskanie informacji czy dana skrzynka jest pusta, czy może jest w niej schowany jakiś gracz.

Wszystkie pakiety są priorytetowane, tzn. pakiety mają różny poziom ważności. W celu optymalizacji wybrane pakiety synchronizujemy częściej od pozostałych. Jest to między innymi pakiet z graczami, chcemy aby on był najczęściej synchronizowany, tak aby ruch pozostałych graczy po mapie był jak najbardziej płynny.

6. Inne informacje o sposobie działania gry

Po ataku wampira odgrywany jest dźwięk, zmienia się on w zależności od tego, czy jest to atak udany czy nie. Również gracze będący w pobliżu wampira słyszą dźwięk ataku. Po udanej ofensywie wampira zraniony „survivor” zostawia plamę krwi w miejscu gdzie został zaatakowany przez wampira.



Sam atak wampira jest limitowany czasowo. Gdyż sama próba „odebrania człowiekowi życia” może wampira „zmęczyć”, limitujemy czasowo próby podjęcia ataku przez wampira. Aktualnie ten czas jest ustawiony na około 30 sekund. Taka funkcjonalność ma zarówno dać czas „survivorowi” uciec przez wampirem gdy ten próbuje go po raz kolejny zaatakować. Jak i utrudnić zadanie wampirowi. Warto zaznaczyć, że ten czas jest różny w przypadku gdy wampirowi uda się zadać atak raniący „człowieka” niż w przypadku gdy atak będzie chybiony. Opóźnienie istnieje również w przypadku próby zapalenia / zniszczenia lampki.

Warto wspomnieć też o generowaniu mapy i pozycji propów. Sama mapa jest generowana w momencie wejścia do lobby. Z każdą rozgrywką mapa się różni. W szczególności można to zauważyć po przejściach między pokojami. To czy dane przejście jest otwarte decyduje czynnik losowy. A więc z każdą kolejną rozgrywką gracz zostaje postawiony w sytuacji gdzie nie zna mapy i nie wie, czy gdy wejdzie do danego pokoju, to znajdzie z niego inne wyjście niż te którym do niego wszedł. Pozycje skrzynek i lampek też zmieniają się co kolejną rozgrywkę.

7. Działania podjęte przez członków zespołu

Praca nad projektem wymagała od nas poświęcenia wielu godzin spędzonych przed ekranami komputerów. W tym czasie prowadziliśmy ożywione dyskusje dotyczące poszczególnych zadań, którym musieliśmy sprostać i strategii działania. Ten etap pracy możemy określić jako efektywną pracę zespołową. Nie chcemy wyszczególniać poszczególnych obowiązków członków zespołu, gdyż granice były płynne i często kompetencje się zmieniały. Podczas prezentacji projektu opowiedzieliśmy trochę o podstawowym podziale prac.

Podczas prac nad projektem nie ucierpiał żaden student.