



**Politechnika
Śląska**

Grafika Komputerowa - Projekt

Temat: Gra FPS Hunter

Autorzy:

Michał Urbanek

Przemysław Ziobrowski

1. Treść zadania

Tematem projektu będzie gra typu FPS (First Person Shooter). Gra będzie polegała na sterowaniu myśliwym, polującym na różne moby. Dostępne będą dwa tryby gry. W pierwszym trybie, gracz będzie musiał w określonym czasie (wybrany przed rozgrywką) lub do pierwszej śmierci, zdobyć jak najwięcej punktów. W drugim trybie na gracza nie zostanie nałożony limit czasowy i będzie mógł polować aż do pierwszego zgonu. Gracz będzie posiadał pulę 250 punktów zdrowia.

W grze będą występowały różne rodzaje mobów (m.in. posiadające indywidualny sposób poruszania się oraz zróżnicowaną ilość punktów życia), za których zabicie można dostać różną ilość punktów. Część gatunków będzie agresywna, tzn. w momencie zauważenia gracza, zaatakują go, natomiast pozostałe gatunki będą uciekały. Na mapie będzie stała liczba mobów, tzn. po zabiciu moba, na mapie w losowym miejscu pojawi się dokładnie taki sam. Pozwoli to na uniknięcie opustoszenia mapy. Zwierzęta będą się w inteligentny sposób poruszały po mapie. W ramach sterowania dostępna będzie opcja skradania (np. kucnięcie). Pozwoli to na zmniejszenie dystansu z jakiego moby będą w stanie wykrywać gracza. Dostępna będzie również opcja biegu, jednak gracz będzie posiadał określoną liczbę punktów wytrzymałości (50). Jeżeli gracz będzie biegł zbyt długo punkty mogą się wyczerpać. Będzie można je zregenerować powstrzymując się od biegu.

Gracz będzie mógł wybierać pomiędzy dwiema mapami. Model map będzie bardzo zróżnicowany, zawierający różne ukształtowania terenu, rodzaje drzew, roślinności oraz będzie wyposażony w realistyczne tekstury. Wprowadzimy również dodatkowe elementy fauny jak różne rodzaje ptaków na niebie oraz ryby w wodzie. Dodatkowo będą zmienne warunki atmosferyczne oraz cykl dobowy (dzień i noc). Zaimplementujemy również realistyczny model strzelania tj. model odrzutu, kiwania celownika, przeładowania itd. Do wyboru będzie wiele modeli broni np. strzelba, pistolet maszynowy, karabin snajperski czy nawet łuk. Każdą z broni będzie posiadała określone parametry. Bronie będzie można modyfikować i personalizować poprzez np. wymianę celownika, dodanie tłumika, zmianę rodzaju amunicji czy zmianę kamuflażu broni. Każda modyfikacja będzie miała wpływ na parametry broni.

Interfejs gracza (HUD) będzie obejmował pasek życia, pasek wytrzymałości, miniaturę trzymanej broni wraz z ilością amunicji oraz minimapę, która będzie pozwalała na wyświetlenie obecnej pozycji gracza na mapie oraz pozycję pobliskich mobów.

Gra została uzupełniona o dodatkowe, w pełni animowane menu. Znajdują się w nim ustawienia pozwalająca maniulować ilością mobów w grze, apteczek, amunicji oraz czasu gry.

Nie zabrakło również globalnych ustawień głośności muzyki oraz efektów dźwiękowych. Ze względu na zbyt wysoką złożoność projektu usunięto możliwość wyboru broni oraz dodatkowe mniej znaczące elementy jak: elementy fauny czy system dobowy.

2. Analiza zadania

2.1. Podstawy teoretyczne problemu

W związku z realistycznym zachowywaniem się broni, konieczna będzie implementacja algorytmu realizującego realistyczne zachowywanie się pocisków. Uwzględnimy działanie siły grawitacji na pocisk, tzn. wraz z czasem lotu, pocisk będzie zbliżał się do powierzchni ziemi z przyspieszeniem $9,81 \text{ m/s}^2$. Jest to istotny element zważając na to, iż gracz często będzie strzelał z dużych odległości. Zaimplementowany zostanie realistyczny model odrzutu broni, co będzie miało przełożenie na jej celność. Celność będzie również zależała od sposobu w jaki porusza się gracz, podczas skradania celność będzie większa, a odrzut mniejszy.

Został zaimplementowany realistyczny model strzału, natomiast nie uwzględnia on trajektorii lotu pocisku. Celność strzału zależy od długości serii - dłuższa seria jest obarczona mniejszą celnością każdego pocisku. Kucanie oraz skradanie powoduje wzrost celności.

2.2 Wykorzystywane zagadnienia grafiki komputerowe

- Wstęp do programowania w Unity
- Wykrywanie kolizji
- Animacja komputerowa
- Techniki animacji szkieletowej

Wyszczególnione powyżej zagadnienia grafiki komputerowej zostały wykorzystane zgodnie z założeniami. Dodatkowo skorzystaliśmy z zagadnienia - Modele oświetlenia, w celu urzeczywistniania terenu gry.

2.3 Wykorzystywane biblioteki i narzędzia programistyczne

Program zostanie stworzony w zintegrowanym środowisku do tworzenia grafiki trójwymiarowej Unity. Całość zostanie napisana w języku C#. Wykorzystamy również zintegrowane środowisko programistyczne Microsoft Visual Studio.

Wymienione powyżej narzędzia programistyczne zostały wykorzystane. Dodatkowo pracowaliśmy z wykorzystaniem oprogramowania Visual Studio Code, rozszerzeń VS Tools for Unity oraz dedykowanego debuggera. Zastosowaliśmy system kontroli wersji GitHub w celu wspólnej pracy nad projektem oraz tworzeniu kopii zapasowych.

3. Plan pracy (wyszczególnienie kamieni milowych)

- 1) Stworzenie mapy
- 2) Stworzenie postaci gracza oraz mechanizmu poruszania się
- 3) Dodanie nieinteligentnych zwierząt
- 4) Dodanie prostego modelu strzelania
- 5) Dodanie menu głównego i systemu zliczania punktów
- 6) Dodanie zaawansowanego modelu strzelania
- 7) Dodanie inteligencji zwierząt

Plan pracy znacząco się nie zmienił. Po zakończeniu pracy i oddaniu finalnego komponentu, łączyliśmy efekty pracy z wykorzystaniem GitHub oraz lokalnego przenoszenia gotowych elementów (tzw. prefabrykatów)

4. Wstępny podział pracy między osoby w zespole

Michał Urbanek	Przemysław Ziobrowski
modele zwierząt	model gracza
stworzenie mapy (teren)	stworzenie mapy (elementy naturalne)
skrypt AI mobów	skrypt przygotowania gry
menu główne	system zliczania punktów
interfejs gracza (HUD)	model broni
wprowadzenie animacji mobów	system modyfikacji broni

Podział pracy również znacząco się nie zmienił. Dodatkowo często wymienialiśmy się własnymi spostrzeżeniami oraz wspólnie rozwiązywaliśmy większe problemy programistyczne. Ze względu na wycofany już na etapie założeń punkt “system modyfikacji broni” - nie został on zrealizowany.

5. Specyfikacja zewnętrzna

5.1 Wymagania sprzętowe

System operacyjny: Windows 7, 8, 8.1 i 10 (64-bit);

Procesor: Intel Core i3/AMD FX (lub lepszy);

Karta graficzna: zgodna z DirectX 10 i obsługą shader model w wersji 4.0;

Miejsce na dysku: 10 GB wolnego.

5.2 Instalacja

Gra nie wymaga instalacji. Jest dostarczona w postaci pliku wykonywalnego, który można uruchomić bez dodatkowego rozpakowywania gry. Do prawidłowego uruchomienia gry, niezbędna jest jedynie maszyna spełniająca minimalne wymagania sprzętowe.

5.3 Interfejs użytkownika

Główne menu gry składa się z trzech paneli. Są to kolejno:

- Start the game- rozpoczęcie gry
- Options- ustawienia
- Exit- wyjście z gry.

Panel ustawień zawiera zestaw suwaków pozwalających na bardzo dokładne spersonalizowanie rozgrywki. Znajdują się tam suwaki:

MUSIC VOLUME - poziom głośności odtwarzanej muzyki w grze

FX VOLUME - poziom głośności efektów w grze(m.in odgłosy tła postaci oraz mobów)

AMOUNT OF ZOMBIES - liczba aktywnych "Zombie" w rozgrywce

AMOUNT OF WORMS - liczba aktywnych "Robaków" w rozgrywce

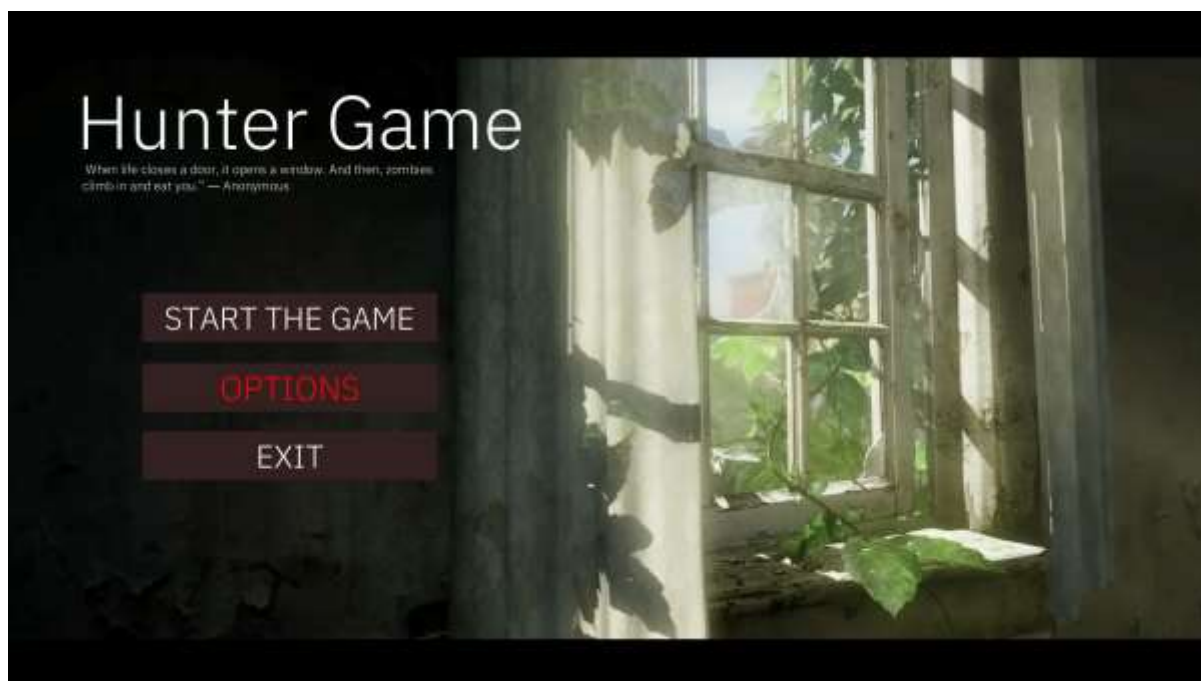
AMOUNT OF FIRST AID KIT - liczba dostępnych na mapie apteczek

AMOUNT OF AMMO - liczba dostępnych na mapie zestawów amunicji

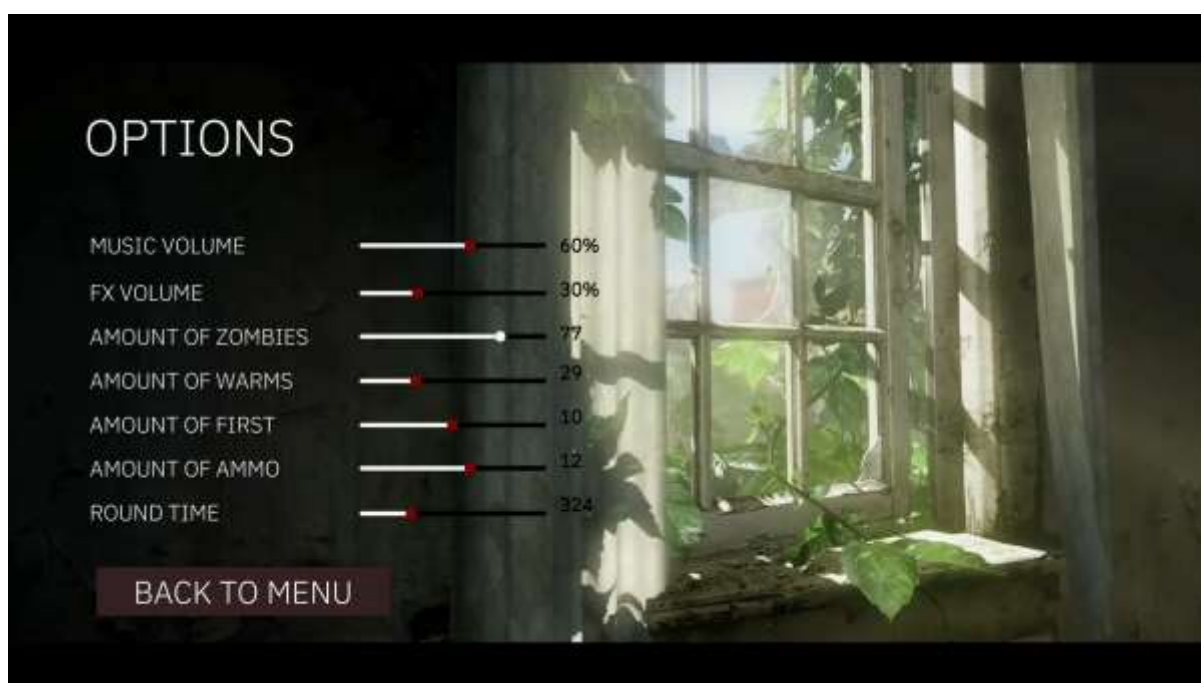
ROUND TIME - czas trwania jednej rundy (pojedynczej rozgrywki)

Całe menu główne jest w pełni animowane oraz oprawione w efekty dźwiękowe. Panele menu oraz suwaki są duże i wygodne dla poprawienia komfortu użytkowania. Całość została skomponowana w stonowanej, nie narzucającej się stylistyce.

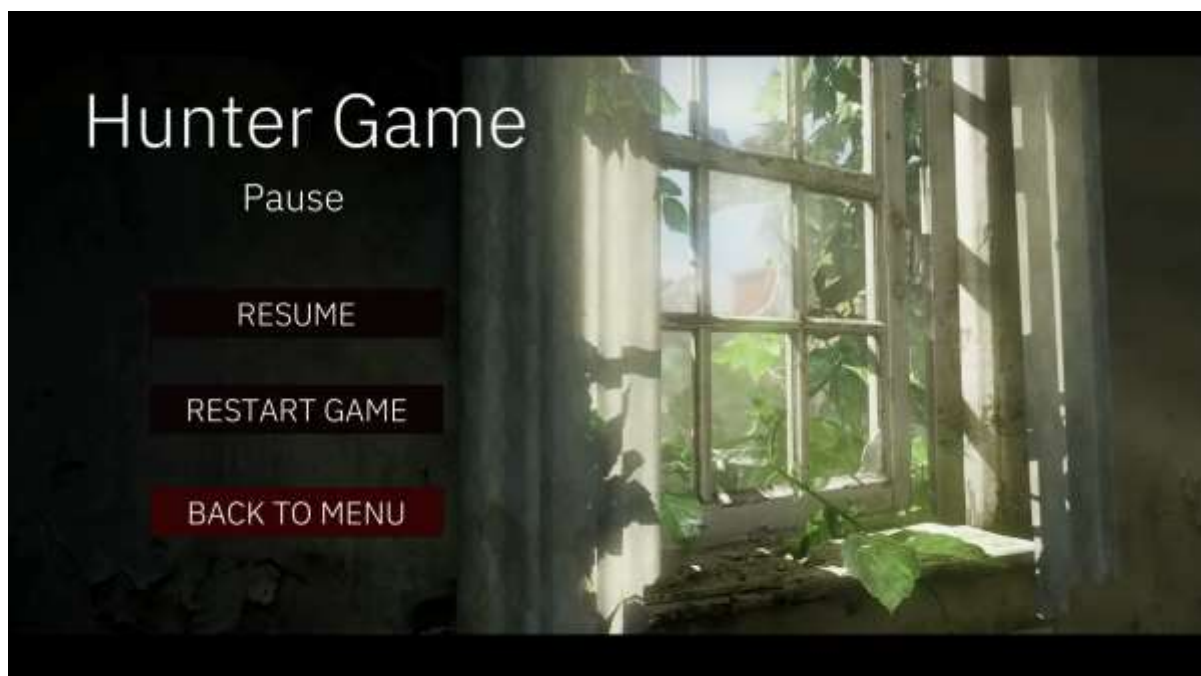
Poniżej zostały zamieszczone zrzuty ekranu, obrazujące rzeczywisty wygląd menu gry



Zrzut ekranu 1. Menu główne gry z dostępnymi wariantami wyboru



Zrzut ekranu 2. Panel ustawień menu głównego z suwakami personalizującymi rozgrywkę



Zrzut ekranu 3. Panel zatrzymanej rozgrywki z dostępnymi przyciskami

5.4 Sposób uruchomienia i obsługi programu.

Gra jest uruchamiana za pomocą pliku wykonywalnego "Unity_HunterGame.exe" znajdującego się w paczce gry "Unity Hunter Game". Użytkownik po uruchomieniu aplikacji powinien przejść do ustawień, wciskając przycisk "OPTIONS", aby skonfigurować rozgrywkę oraz dostosować jej poziom trudność. Po wybraniu interesujących parametrów, należy powrócić do menu głównego naciskając przycisk "BACK TO MENU" oraz rozpocząć nową grę naciskając przycisk "START THE GAME".

Aby zatrzymać rozgrywkę, należy wcisnąć przycisk ESC na klawiaturze. Gracz zostaje wtedy przeniesiony do menu pauzy. Aby zakończyć rozgrywkę należy nacisnąć przycisk "BACK TO MENU". Aby zresetować grę należy wcisnąć przycisk "RESTART GAME" - gra zostanie zresetowana z parametrami zgodnymi z poprzednią turą. Aby kontynuować aktualną rozgrywkę należy wcisnąć przycisk "RESUME"

5.5 Sterowanie w grze

Gra posiada rozbudowany kontroler postaci oraz interakcji w grze. Poruszanie postacią odbywa się za pomocą klawiszy WSAD (odpowiednio W - ruch do przodu, S - ruch do tyłu, A - ruch w lewo, D - ruch w prawo). Przytrzymanie klawisza SHIFT powoduje przyspieszony ruch postaci (bieg). Wciśnięcie klawisza CTRL powoduje zwolnienie ruchu postaci (skradanie się).

Lewy przycisk myszy jest odpowiedzialny za pojedynczy strzał. Istnieje również możliwość wyprowadzenia serii z karabinu, po przytrzymaniu klawisza strzału. Wciśnięcie klawisza Q spowoduje samo-uleczenie postaci (w przypadku gdy gracz posiada dodatkowe apteczki)

5.6 Format danych wejściowych i wyjściowych.

Gra nie przyjmuje żadnych danych wejściowych i wyjściowych, stąd punkt ten nie zostanie szerzej omówiony. Stan rozgrywki również nie jest zapisywany, po opuszczeniu gry.

6. Przykład działania programu



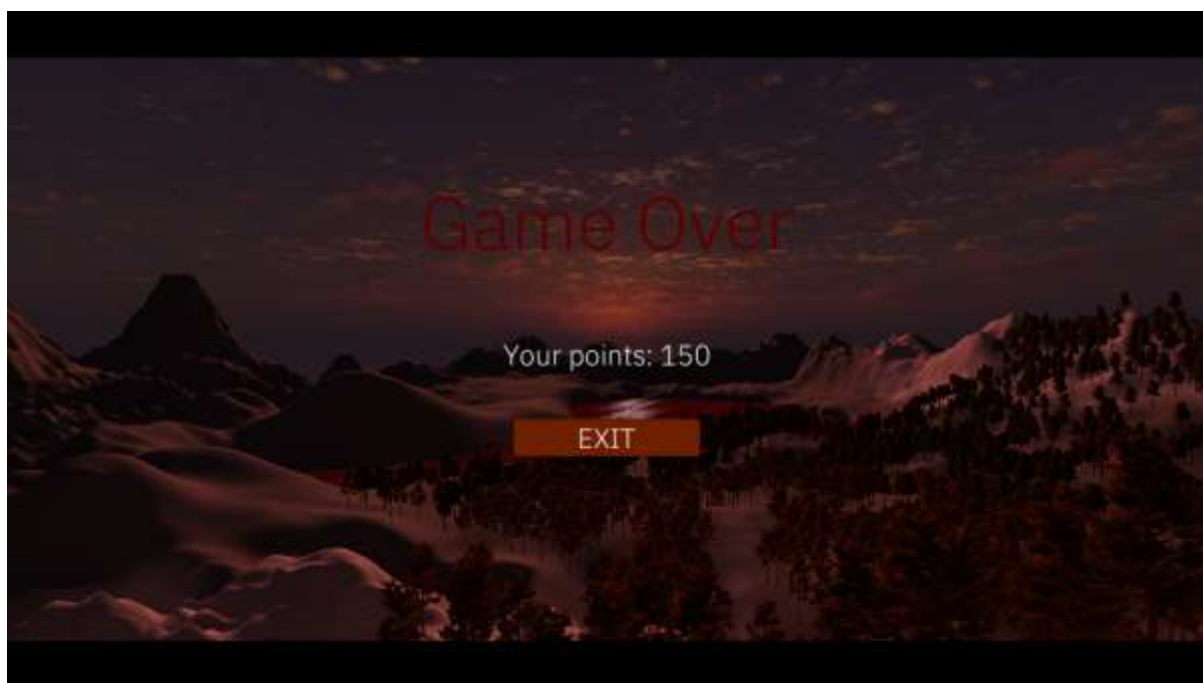
Zrzut ekranu 4. Atak wściekłych Zombie. Widoczny HUD gracza oraz mini-mapa. Strefa leśna.



Zrzut ekranu 5. Egzekucja robala. Widoczny radar oraz paski życia i wytrzymałości. Strefa mroku



Zrzut ekranu 6. Plaga Zombie oraz robaków. Chwila przed katastrofą. Ostatnie punkty wytrzymałości



Zrzut ekranu 7. Ekran końca gry z widoczną ilością zdobytych punktów



Zrzut ekranu 8. Egzekucja zombie. W tle "chmura" przeciwników



Zrzut ekranu 9. Jezioro Czerwone. Krajobraz gry rodem z światowych produkcji



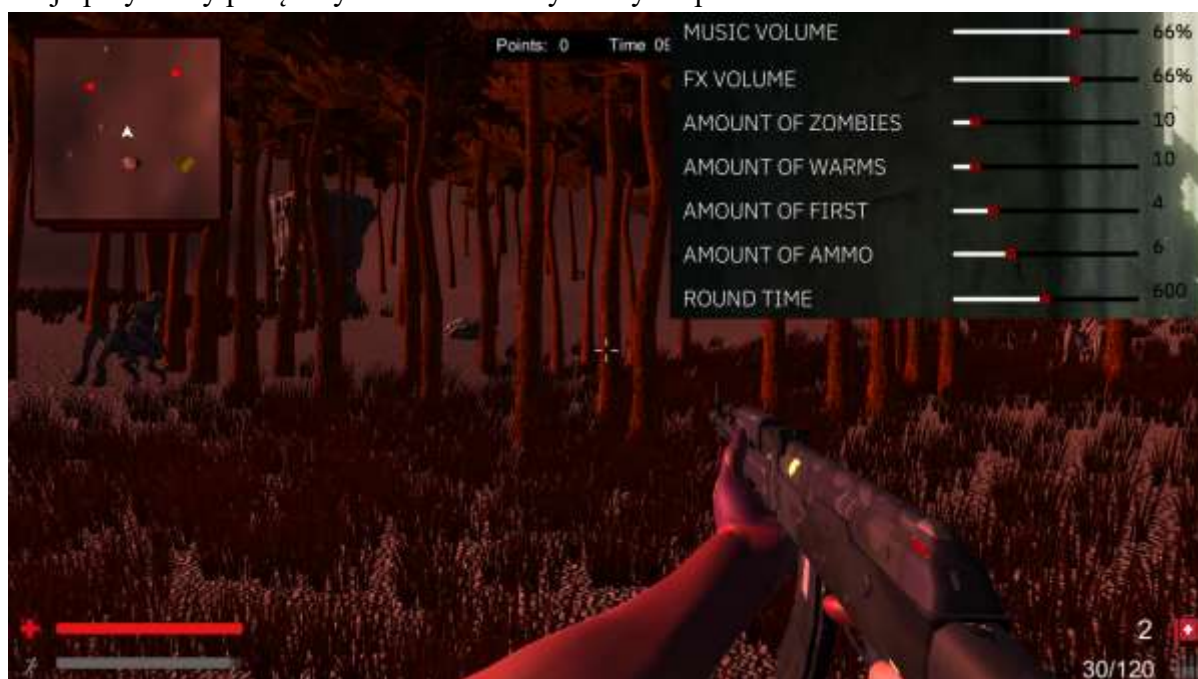
Zrzut ekranu 10. Zestaw apteczki do samodzielnego opatrzenia się

7. Specyfikacja wewnętrzna

Link do dokumentacji generowanej automatycznie (Doxygen), został dodany jako załącznik do raportu końcowego.

8. Testowanie i uruchamianie

Program został wielokrotnie przetestowany na etapie rozwoju gry. Został przetestowany na dwóch komputerach o dobrej specyfikacji sprzętowej. Nie wykazuje nieprawidłowego działania. Został przetestowany pod kątem różnych ustawień ilości zombie/robaków, apteczek/amunicji oraz czasu gry. Zmiany ustawień nie wpływały na stabilność programu, a jedynie obciążenie komponentów. Poniżej zostały zaprezentowane 2 konfiguracje ustawień, odpowiednio dla różnych poziomów trudności. Program został przetestowany również pod kątem zdarzeń nietypowych, jak próba ucieczki poza mapę, wielokrotnego zebrania tego samego ekwipunku czy zabicia, już umarłego moba. Wszystkie akcje przyniosły pożądany rezultat i nie wykazały nieprawidłowości.



Zrzut ekranu 11. Test gry z wykorzystaniem "niskiego" poziomu trudności gry.



Zrzut ekranu 12. Test gry z wykorzystaniem “średniego” poziomu trudności gry.



Zrzut ekranu 13. Test gry z wykorzystaniem “wysokiego” poziomu trudności gry. Totalna demolka.
Nawałnica Zombie uniemożliwiła zrobienie ładnego zrzutu ekranu

9. Wnioski

Stworzenie tak rozbudowanego programu pozwoliło nam odkryć prawdziwy programistyczny potencjał. W nawiązaniu do pierwotnej analizy zadania - była ona bardzo rozbudowanym opisem pięknej gry, która mogłaby powstać, ale... w kilka miesięcy, z wykorzystaniem zespołu programistów. Pracując w zespole 2 osobowym trzeba liczyć się z tym, że nie wszystko uda się zrobić. Skupiliśmy się na rzeczach najważniejszych. Chcieliśmy stworzyć ciekawą grę, z przyjemnym dla oka interfejsem. Uwzględniliśmy najważniejsze elementy mechaniki gry, aby dostarczała ona wiele radości użytkownikowi.

Przekonaliśmy się, że nawet niewielkim nakładem pracy można uzyskać wspaniałe efekty wykorzystując bardzo zaawansowany silnik gry Unity. Jest to narzędzie dość intuicyjne, ale wymaga wielu godzin spędzonych na projektowaniu, aby nabrać doświadczenia i obycia z programem. Nieodzownym elementem Unity jest również technologia C#, wykorzystywana do pisania skryptów gry. Sam kod jest bardzo "wysoko-poziomowy" i w relatywnie krótkim czasie i niewielkiej ilości kodu, można zaprogramować bardzo złożony algorytm (jak np. sztuczna inteligencja postaci w grze). Dzięki początkowej konfiguracji narzędzi programistycznych (debugger VSC), problemu nie sprawiło nam naprawianie błędów popełnionych na etapie rozwoju projektu.

Projekt rozwinął nasze umiejętności pracy w zespole oraz wspólnego rozwiązywania problemów. Regularny kontakt z opiekunem projektu przyczynił się do ukończenia projektu na czas, a cenne uwagi i spostrzeżenia, zostały głęboko przeanalizowane oraz wdrożone. Finalnie, jesteśmy bardzo zadowoleni z produktu końcowego i planujemy w przyszłości dalej rozwijać swoje umiejętności w tej dziedzinie.