



Informatyka – zaawansowana inżynieria oprogramowania

Twórcy projektu:

Rafał Dobrowolski, Wiesław Bikowski, Kacper Buczkowski, Łukasz Darmochwał

rok 3, semestr VI

Temat projektu

Gra typu Space-shooter, utrzymana w klimacie retro

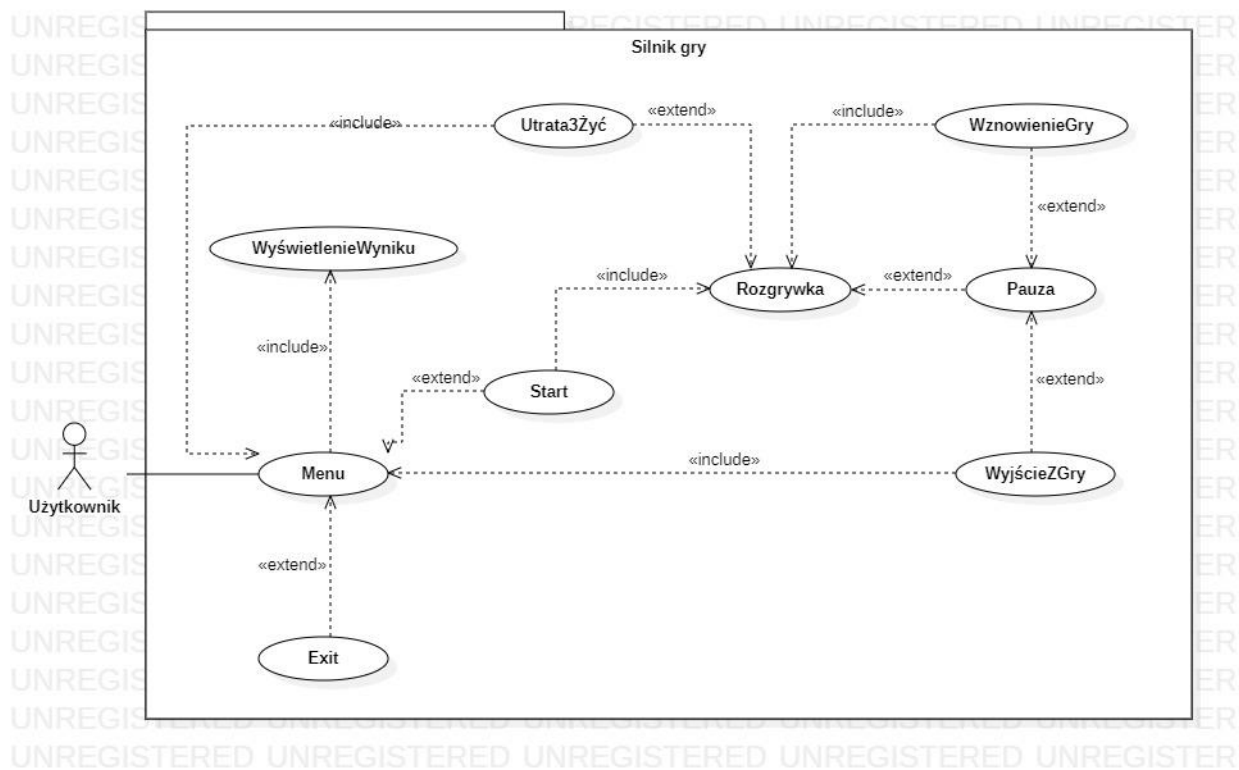
Spis treści:

1. Założenia projektu.....	4
2. Diagramy UML.....	
• Diagram przypadków użycia.....	5
• Diagramy aktywności.....	6-8
• Diagram klas.....	9
• Diagram wdrożenia.....	10
3. Instrukcja użytkowania.....	11-12
4. Podsumowanie.....	13

Założenia projektu:

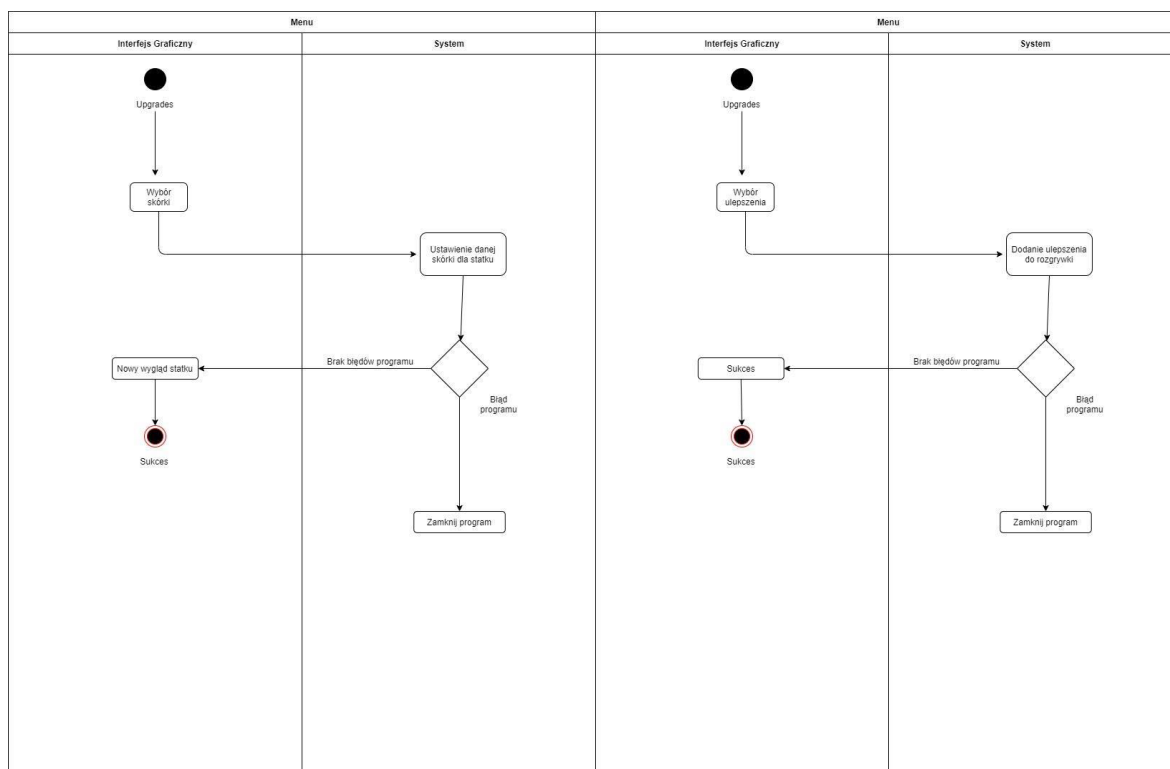
- stworzona zostanie gra typu space-shooter utrzymywana stylistyce pixelartowej,
- gra przeznaczona na komputery z systemem Windows oraz na urządzenia mobilne z systemem Android,
- rozgrywka pozwoli na sterowanie statkiem kosmicznym w celu zestrzelenia jak największej ilości wrogów oraz dostania się na coraz wyższe poziomy,
- stworzone zostanie rozbudowanie menu pozwalające na: start rozgrywki, wyjście z gry, wyświetlenie okna skórek, pauza, wznowienie, restart, wyświetlenie okna ulepszeń,
- okno skórek pozwoli na wybranie jednej z pięciu dostępnych wariantów statku – jest to element czysto preferencyjny i nie wpływa na właściwości statku,
- podczas rozgrywki zliczany będzie wynik gracza na podstawie liczby zestrzelonych wrogów, zostanie on również wyświetlony podczas zakończenia rozgrywki,
- gracz kończy rozgrywkę w momencie zakończenia jej z poziomu menu lub po utracie trzech żyć,
- grafiki użyte w projekcie tworzone są w stanowczej większości własnoręcznie,
- rozgrywkę umilać będą klimatyczne ścieżki dźwiękowe, jedna przeznaczona dla głównego menu oraz drugo towarzysząca graczowi podczas rozgrywki,
- wersja desktop:
 - a. sterowanie strzałkami,
 - b. możliwość ulepszania funkcjonalności statku wraz ze zdobywaną podczas rozgrywki walutą,
- wersja mobile:
 - a. sterowanie dotykiem,
 - b. stworzenie "skill check" - elementu sprawdzającego czas reakcji gracza
- dodatkowo zostanie stworzona strona internetowa promująca grę oraz przedstawiające podstawowe informacje dotyczące rozgrywki,
- strona internetowa umożliwi wybranie oraz pobranie odpowiedniej dla siebie wersji gry,

Diagramy UML



Rysunek 1: Diagram przypadków użycia

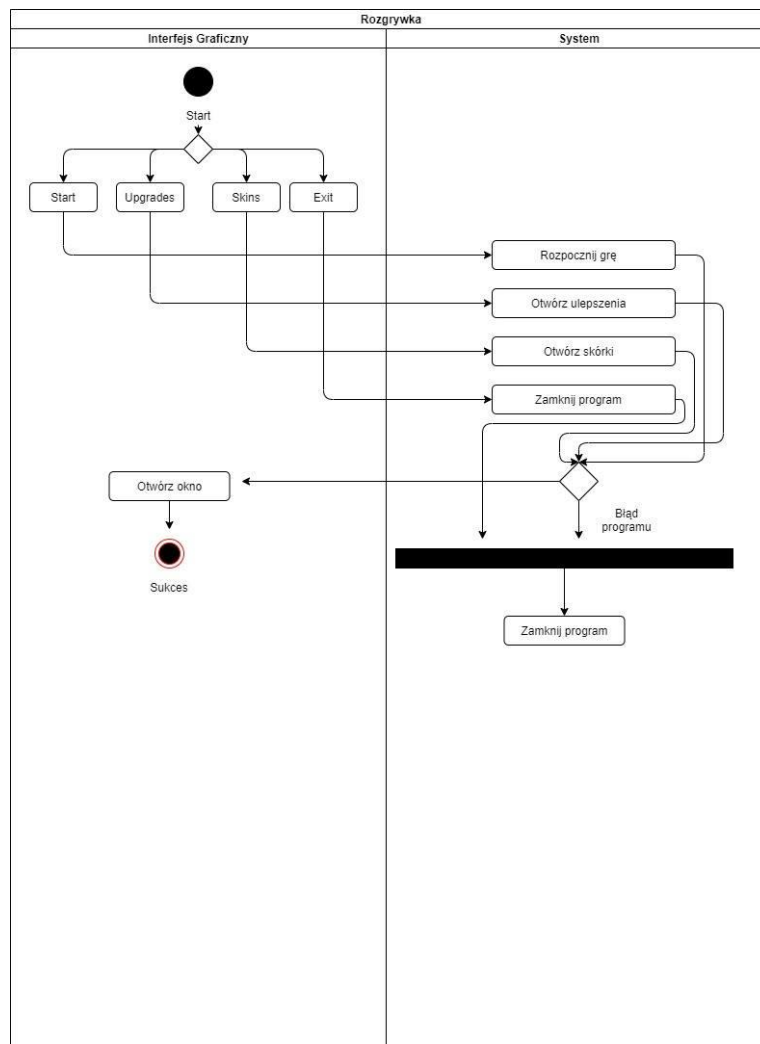
Rysunek 1 przedstawia ogólny diagram przypadków użycia tworzonej gry. Całość skupia się wokół pola "Rozgrywka" symbolizującego stan gry będący główną częścią grywalną. Na diagramie brakuje pola "Ulepszenia" oraz "Skill Check" z racji iż są to mechanizmy dedykowane dla konkretnej wersji gry (desktop lub mobile).



Rysunek 2: Diagram aktywności - skórki

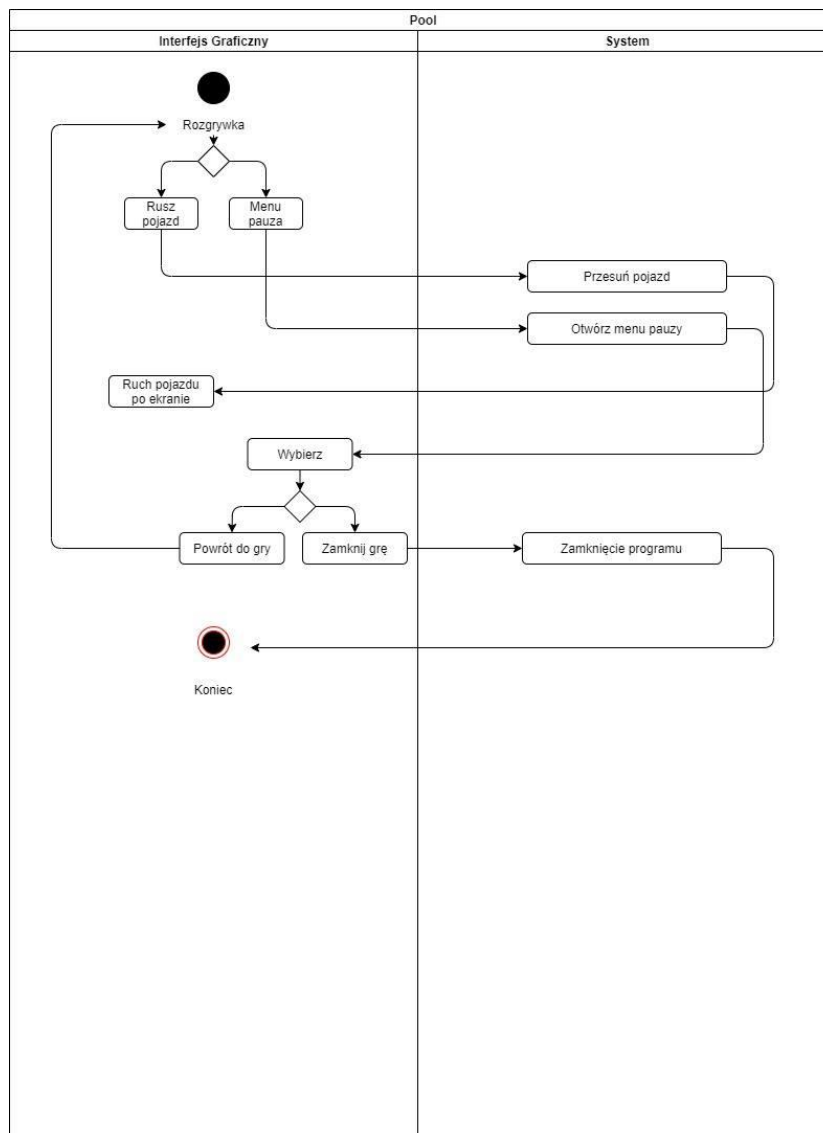
Rysunek 3: Diagram aktywności - ulepszenia

Rysunek 2 oraz Rysunek 3 przedstawiają diagramy aktywności dwóch pobocznych mechanizmów urozmaicających rozgrywkę. Nie są one skomplikowane, bo i część kodu stojąca za tymi mechanizmami nie jest złożona.



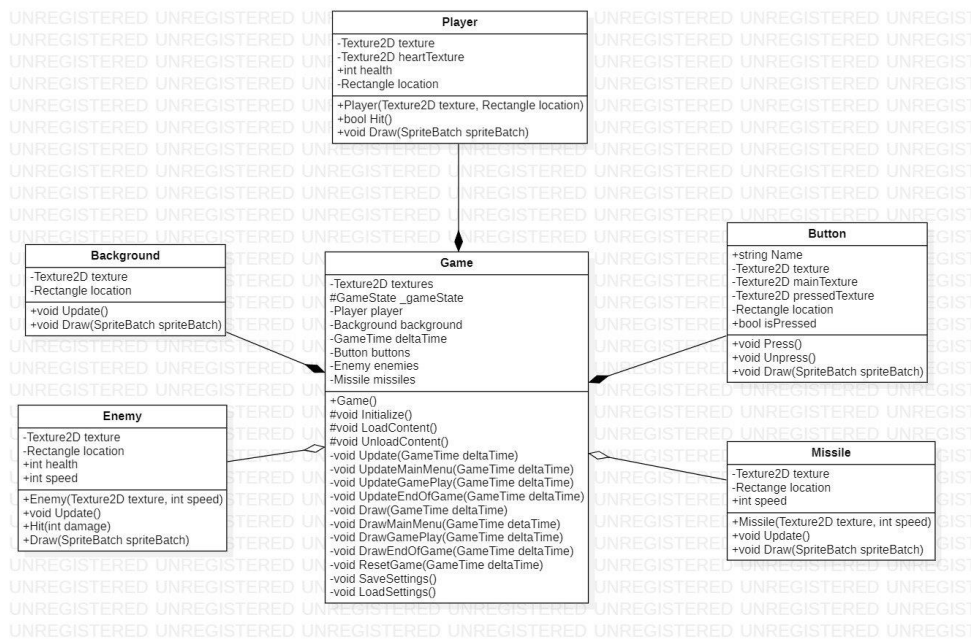
Rysunek 4: Diagram aktywności - menu

Rysunek 4 przedstawia diagram aktywności dla kilku opcji menu takich jak START, UPGRADES, SKINS czy EXIT. Wybranie którejkolwiek opcji przekierowuje program do odpowiedniej funkcji obsługującej zdarzenie kliknięcia w dany przycisk.



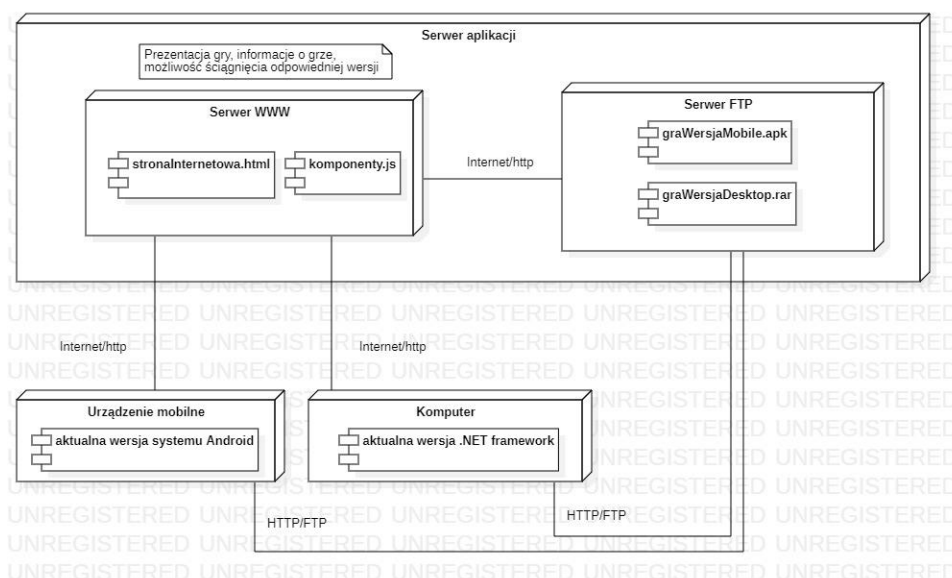
Rysunek 5: Diagram aktywności - rozgrywka

Rysunek 5 przedstawia ostatni diagram aktywności reprezentujący tym razem samą rozgrywkę. Klikając przycisk START możemy wprowadzić tak naprawdę tylko dwie dane wejściowe - poruszanie strzałkami powodujące ruch statku na ekranie lub wciśnięcie przycisku "ESC" powodujące wyświetlenie okna pauzy.



Rysunek 6: Diagram klas

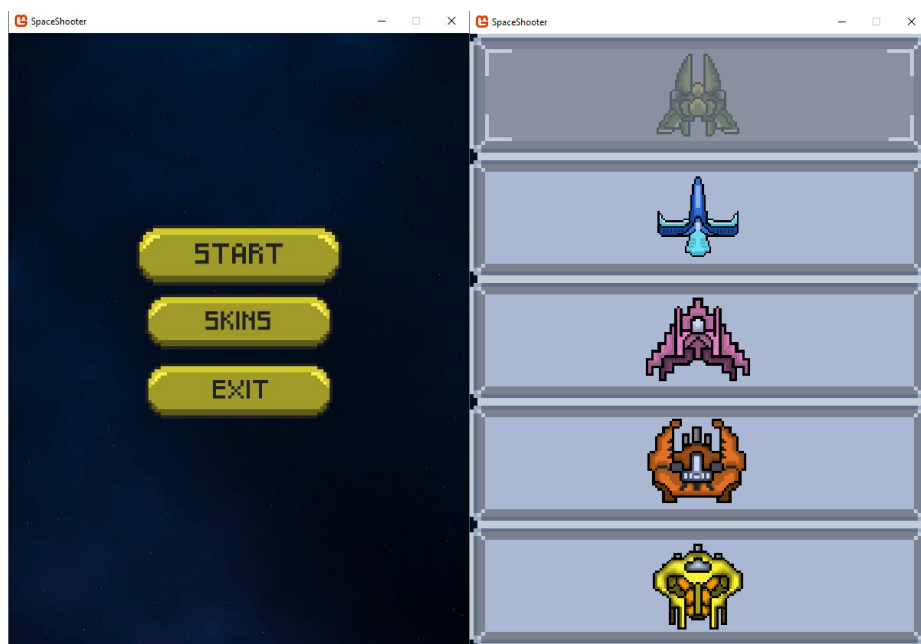
Rysunek 6 przedstawia diagram klas tworzonej gry. Wszystko skupia się wokół głównej klasy "Game" sterującej większością zasobów oraz zawierającej większość logiki gry. Klasy poboczne reprezentują przeciwników, tła, statek samego gracza, przyciski oraz pociski. Część klas jest dość bliźniacza, aczkolwiek powielenie ich i lekka zmiana znacznie ułatwiła zarządzanie obiektami tychże klas.



Rysunek 7: Diagram wdrożenia

Diagram 7 prezentuje system wdrożenia tworzonej aplikacji. Do jej prawidłowego działania wymagany jest komputer z systemem Windows 10 oraz aktualną wersją .NET Framework lub urządzenie mobilne z aktualną wersją systemu Android. Użytkownik w celu pobrania gry udaje się na stronę internetową i wybiera odpowiednią dla siebie wersję. Całość jest umieszczona na zewnętrznym serwerze plików.

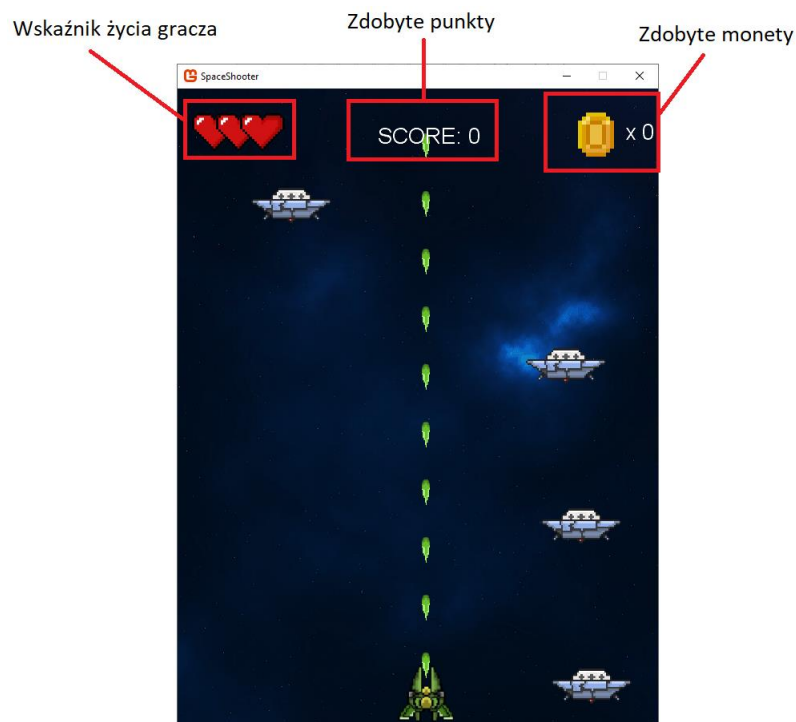
Instrukcja użytkownika



Rysunek 10: Główne menu

Rysunek 11: Okno skórek

Po włączeniu gry ukaże się ekran startowy wraz z trzema opcjami do wyboru. START umożliwia nam przejście bezpośrednio do rozgrywki, SKIN umożliwia wybór alternatywnego wyglądu statku, EXIT pozwala zamknąć aplikację. Z okna skórek można wyjść naciskając przycisk "ESC".



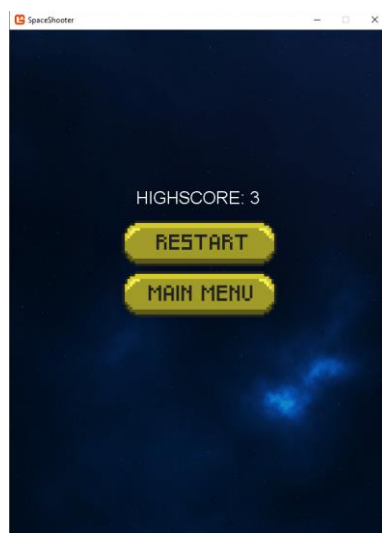
Rysunek 12: Okno rozgrywki



Rysunek 13: Okno pauzy

Rysunek 14: Okno ulepszeń

Okno pauzy uruchamiamy wciskając podczas rozgrywki przycisk "ESC". Z opcji możemy kolejno wznowić rozgrywkę, zacząć od nowa, przejść do okna ulepszeń lub wyjść do menu głównego. Uruchamiając okno ulepszeń możemy wykupić rozbudowę statku wydając zebrane wcześniej monety. Ulepszenie kupujemy klikając na nie.



Rysunek 15: Okno końca gry

Okno końca gry wyświetla się po utracie trzech żyć. Umożliwia ono szybkie wznowienie gry lub przejście do głównego menu. Dodatkowo pokazuje ono liczbę zdobytych punktów.

Użyte oprogramowanie

W celu wykonania założonej gry wykorzystujemy język C# wraz z frameworkiem Monogame oraz środowiska programistycznego Visual Studio. Strona internetowa wykonywana jest w środowisku Visual Studio Code w oparciu o HTML, CSS, JS, JQuery. Wykorzystywane grafiki tworzone są w programie Aseprite a diagramy użyte w dokumentacji pochodzą z StarUML

Podsumowanie

- Cele zrealizowane / niezrealizowane

Stanowczą większość zakładanych celów udało się zrealizować bez większych problemów a także udało się zaimplementować wiele mechanizmów które początkowo nie były brane pod uwagę (system ulepszeń oraz skill check). Udało się stworzyć w pełni grywalną i funkcjonalną wersję desktop dla systemów Windows 10 oraz mobile dla telefonów z systemem android. Jedynym mankamentem, który wymagałby dopracowania to sama rozgrywka. Miała ona mocno opierać się na zręczności gracza, lecz finalny produkt nie do końca spełnia to założenie. Gra wymagałaby gruntownej przebudowy by to zmienić. Nie udało nam się również popracować z animacjami (np. wybuchy, animowane pociski itp..) gdyż silnik, na którym pracowaliśmy miał co to tego spore ograniczenia. Przesiadka np. Na platformę UNITY znacznie by to ułatwiła.

- Możliwe kierunki rozwoju

Przesiadka na platformę UNITY była by dobrym pomysłem. W łatwy sposób można by było rozwinąć animacje oraz ulepszyć sprawdzanie kolizji (Monogame oferuje to w ograniczonym zakresie). Co do samej gry warto by było dodać różne trajektorie lotu przeciwników, większą ich różnorodność oraz o czym wspominałem wcześniej, wynajmować ich ruch i kolizję. Znacznie by to ulepszyło wizualny odbiór rozgrywki. Można by było się zastanowić nad modelem sieciowym – nad jakąś prostą rywalizacją w zdobywaniu punktów i ewentualnych nagrodach.

- Wnioski

Korzystając z frameworku Monogame nauczyliśmy się naprawdę sporo na temat tworzenia gier i ich mechanizmów. Tworząc projekt skupialiśmy się na obiektowym podejściu do rozwiązywania problemów co w znaczącym stopniu rozwinęło nasze umiejętności programistyczne.

Praca grupowa przebiegała bez przeszkód z racji jasnego i konkretnego podziału zadań do wykonania. Zauważyliśmy również pisząc dwie wersje - desktop i mobile - zdrową napędzającą pracę konkretnie.