UNIWERSYTET WARMIŃSKO-MAZURSKI

W OLSZTYNIE

WYDZIAŁ MATEMATYKI I INFORMATYKI

**Kamil Zieliński**

**Kierunek: Informatyka**

**Realizacja gry 2D z gatunku strategicznych gier czasu rzeczywistego w silniku Unity**

Praca inżynierska wykonana

w katedra Analizy Zespolonej

pod kierunkiem

*dr Jacek Marchwicki*

Olsztyn 2022

UNIVERSITY OF WARMIA AND MAZURY

IN OLSZTYN

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

**Kamil Zieliński**

**Field of study: Computer Science**

**Development of 2D real-time strategy game in Unity engine**

Engineering Thesis written in

Chair of Complex Analysis

under supervision of

*dr Jacek Marchwicki*

Olsztyn 2022

Streszczenie

Praca zawiera dokumentację na temat gry czasu rzeczywistego „Bulwarriors” w 2D zbudowanej na silniku Unity. Projekt został stworzony na potrzeby realizacji pracy inżynierskiej. Dokumentacja zawiera opis gatunku jakim są gry RTS (Real-Time Strategy), użytych narzędzi z którymi aplikacja została zaprojektowana oraz przede wszystkim szczegółowy opis samej gry, składający się z objaśnienia rozgrywki, czyli mechanik występujących w grze, wyjaśnienia sterowania oraz przedstawienie stworzonego świata gry.

Summary

This paper contains documentation on the real-time game "Bulwarriors" in 2D built on the Unity engine. The project was created for an engineering thesis. The documentation contains a description of the game genre RTS (Real-Time Strategy), used tools with which the application was designed and most of all a detailed description of the game itself, consisting of an explanation of the gameplay, or mechanics occurring in the game, explanation of the control and a presentation of the game world created.

Spis treści

[1. Wprowadzenie 4](#_Toc93508018)

[2. Opis gatunku 5](#_Toc93508019)

[2.1. RTS jako gatunek gier 5](#_Toc93508020)

[2.2. Przeznaczenie 5](#_Toc93508021)

[2.4. Znane tytuły 6](#_Toc93508022)

[2.4.1. StarCraft 6](#_Toc93508023)

[2.4.2. The Lord of the Rings: The Battle for Middle-Earth 8](#_Toc93508024)

[3. Założenia techniczne 9](#_Toc93508025)

[3.1. Platforma 9](#_Toc93508026)

[3.2. Unity 9](#_Toc93508027)

[3.3. C# 10](#_Toc93508028)

[3.4. GIMP 10](#_Toc93508029)

[3.5. GitHub 10](#_Toc93508030)

[4. Rozgrywka 11](#_Toc93508031)

[4.1. Interakcja z otoczeniem 11](#_Toc93508032)

[4.2. Przebieg rozgrywki 11](#_Toc93508033)

[4.4. Podsumowanie rozgrywki 12](#_Toc93508034)

[4.5. Opis UI 13](#_Toc93508035)

[4.6. Sterowanie 14](#_Toc93508036)

[4.6.1. Myszka 14](#_Toc93508037)

[4.6.2. Klawiatura 16](#_Toc93508038)

[4.7. Mechanika 17](#_Toc93508039)

[4.7.1. Jednostki 17](#_Toc93508040)

[4.7.2. Budynki 18](#_Toc93508041)

[Występujące w grze typy budynków podobnie jak jednostki posiadają przypisane obiekty oraz statystki. 18](#_Toc93508042)

[4.7.3. Walka 19](#_Toc93508043)

[4.7.4. Szukanie ścieżki 20](#_Toc93508044)

[4.7.5. Ekonomia 21](#_Toc93508045)

[5. Świat gry 22](#_Toc93508046)

[6. Zakończenie 28](#_Toc93508047)

[7. Bibliografia 29](#_Toc93508048)

1. Wprowadzenie

Celem pracy inżynierskiej jest przedstawienie wizji oraz implementacja gry strategicznej czasu rzeczywistego w widoku dwuwymiarowym. Polegającej na pokonaniu przeciwnika równocześnie obronie przed jego atakami. Gracz na początku rozgrywki posiada pewną ilość jednostek oraz budynków z pomocą, których musi zacząć rozwijać swoją siedzibę, w celu zgromadzenia większych sił w postaci jednostek bojowych oraz budowie kolejnych struktur umożliwiających zbieranie zasobów jak również tworzeniu lepszych oddziałów. Zwycięstwo gracz może osiągnąć poprzez zniszczenie wszystkich budynków przeciwnika. Przeciwnik również dysponuję jednostkami oraz budynkami tego samego typu co gracz. Przeciwnik, sterowany stworzonym algorytmem próbuje gracza wyeliminować niszcząc wszystkie postawione przez gracza budynki, powodując jego porażkę.

Aplikacja została zaimplementowana z użyciem silnika Unity 2D, oprogramowania GIMP 2.8, git oraz językiem programowania C#. Projekt gry był inspirowany znanymi tytułami, o których mowa w dalszej części pracy jak również własnymi pomysłami na świat gry oraz implementacje mechanik i logiki gry.

Dokumentacja zawiera opis gatunku jakim są gry RTS w dodatku tytuły, którymi był inspirowany projekt, użytych narzędzi z którymi aplikacja została zaprojektowana oraz przede wszystkim szczegółowy opis samej gry, składający się z objaśnienia rozgrywki, czyli mechanik występujących w grze, wyjaśnienia sterowania oraz przedstawienie stworzonego świata gry.

1. Opis gatunku
   1. RTS jako gatunek gier

Gra RTS oznacza gatunek „strategicznych gier czasu rzeczywistego” (ang. Real‑time strategy), w których gracz podejmuje decyzje w celu osiągnięcia warunków zwycięstwa lub zrealizowania jakiegoś założonego celu. Gry tego typu często stawiają na szybkość podejmowania strategicznych decyzji, pozwalających zyskać przewagę nad przeciwnikiem, zwiększają szanse przetrwania lub wpływają korzystnie na wynik końcowy. Niezbędna szybkość podejmowanych działań jest narzucona ze względu, że gra trwa w czasie rzeczywistym, nie jest ograniczona turami, gdzie gracze mają pewną ilość czasu na podjęcie kluczowych decyzji. W tym trybie, każdy gracz w tym przeciwnik wykonuje ruch lub podejmuje jakąś decyzję w tym samym czasie.

* 1. Przeznaczenie

W gatunku RTS, sporo osób może znaleźć element, którym się zainteresuje. Przez budowanie olbrzymich siedzib, zbieraniu surowców, stworzeniu ekonomicznej potęgi, czy niezwyciężonych fortec, poprzez tworzenie armii zdolnej niszczyć przeciwników i zdobywaniu terytorium do dalszej ekspansji.

Sama mechanika bywa dla graczy satysfakcjonująca, dodatkowo nietrudno dodać do gier RTS fabułę, czy swego rodzaju kampanię, która może dodatkowo urozmaicić grę, wprowadzając kolejny cel i pozwalając na satysfakcjonującą wielogodzinną rozgrywkę.

Istnieje również grupa graczy, która jest zainteresowana ściśle mechaniką oraz statystykami, wiele gier posiada podsumowania rozegranych potyczek, zawierającej spis zebranych zasobów czy jednostek w czasie, co stanowi satysfakcjonujące podsumowanie całej sesji.

* 1. Znane tytuły
     1. StarCraft

Cała seria gier elektronicznych stworzona przez studio Blizzard Entertainment, seria jest jedną z najpopularniejszych gier RTS. Wydana w 1998 roku, stała się jedną z najbardziej popularnych gier tego gatunku.

StarCraft posiadał zaimplementowane funkcje, które były standardem dla tego typu gier. Gra wyróżniła się ze względu na urozmaicenie gry wynikające z wprowadzenia różnych ras, różniących się sposobem gry oraz funkcjami. Wymagało to od gracza zmianę strategii w zależności do wybranej nacji.



Rysunek fragment rozgrywki podczas walki dwóch nacji z gry StarCraft

Wydany w 2010 StarCraft II był kolejną odsłoną gry, będącą ulepszeniem poprzedniej części. Produkcja cieszyła się większym zainteresowaniem od poprzedniego wydania, szczególnie w branży sportu internetowego.



Rysunek 2 fragment rozgrywki z gry StarCraft

Gracz rozpoczyna grę z czterema robotnikami umożliwiających zbieranie zasobów, czyli kryształów i gazu, jak również tworzenie nowych struktur, zapewniających produkcje różnego typu jednostek jak również ulepszanie ich zwiększając poszczególne statystki danego rodzaju jednostki. Zasoby pełnią rolę waluty, za pomocą której gracz może zlecić budowę, ulepszenie czy produkcję.

Celem gry jest wyeliminowanie przeciwnika, którego celem jest doprowadzenie do przegranej gracza.

Szata graficzna, mechaniki oraz kampania gry stanowiły odpowiednią symulacje świata gry stworzonego w scenerii wojen kosmicznych.

* + 1. The Lord of the Rings: The Battle for Middle-Earth

Gra stworzona przez EA Los Angeles w 2004 roku, jest oparta na filmowej trylogii Władca Pierścieni w reżyserii Petera Jacksona, która z kolei jest na podstawie powieści J.R.R. Tolkiena oryginalnej powieści.

Produkcja jest grą gatunku strategii czasu rzeczywistego, gracze jak również przeciwnicy gromadzą zasoby, wykorzystując je do budowy baz wojskowych i armii. Struktury są budowane na z góry ustalonych działkach na mapie gry. W grze jest jeden zasób, produkowany za pośrednictwem dedykowanych budynków.

Armia jaką gracz dysponuje składa się z grup jednostek, jak również pojedynczych, gdzie każda posiada swoje statystyki. Frakcje do dyspozycji posiadają różne typy jednostek, mniej lub bardziej różniące się założeniami i mechaniką.

Bohaterowie są specjalnym rodzajem pojedynczej jednostki, jest ona wyposażona w szereg najróżniejszych umiejętności, zwiększających tymczasowo siłę jednostki czy zadających duże obrażenia.

Rysunek 3 fragment kampani z gry The Lord of the Rings: Battle for Middle-Earth

1. Założenia techniczne
   1. Platforma

Gra została zaprojektowana na system Windows, który jest dominującym system operacyjnym. Globalnych udział w rynku systemów operacyjnych wśród komputerów stacjonarnych wynosi ponad 70%. Co więcej system Windows jest kompatybilny z większością współczesnych gier, dlatego posiada go większość użytkowników.

Problematycznymi wydają się platformy mobilne czy konsole, ze względu na charakterystykę sterowania, która w tych przypadkach jest utrudniona. Sterowanie w tego typu grze jest znacząco uproszczone w przypadku użycia myszy komputerowej.

Wybierając platformy powinno się uwzględnić przede wszystkim grupę użytkowników, która najchętniej zainteresuje się tytułem. Dlatego w przypadku tego projektu został wybrany system Windows.

* 1. Unity

Aplikacja została zaimplementowana z użyciem silnika Unity w wersji 2020.3.20f1 przy pomocy komponentów potrzebnych do stworzenia gry 2D.

Silnik Unity został opracowany przez Unity Technologies, produkt został wypuszczony w czerwcu 2005 roku i od tamtego czasu jest stale rozwijany i aktualizowany.

Silnik służy do wszechstronnych projektów zarówno 2D jak i 3D, poza obsługą elementów fizycznych, pozwalających na tworzenie mechanik gry, umożliwiona jest również opcja tworzenia animacji na samej platformie Unity jak i prostych modeli 3D.

* 1. C#

C# język programowania jest głównie używany w silniku unity do pisania skryptów do aplikacji. Istnieją również wtyczki oraz pakiety, które umożliwiają na pisanie skryptów w innych językach.

Do tworzenia skryptów w języku C# użyte zostało zintegrowane środowisko programistyczne Microsoft Visual Studio 2019 firmy Microsoft.

* 1. GIMP

Narzędziem użytym do tworzenia oraz obrabiania grafik użytych w grze posłużył program GIMP 2.8.18. Program jest darmowym edytorem grafiki rastrowej.

* 1. GitHub

GitHub – jest serwisem internetowym przeznaczonym do projektów programistycznych, wykorzystujący system kontroli wersji Git.

Git jest systemem kontroli wersji, niemal niezbędnym w realiach dzisiejszej pracy z kodem oraz aplikacjami. Najważniejszym zastosowaniem oprogramowania git jest możliwość efektywnej pracy zespołowej na jednym projektem programistycznym.

W tym projekcie mimo pracy w jednoosobowym zespole, był niezbędny ze względu na łatwą i bezpieczną ingerencje w stworzony projekt. Pozwalającą na pracę na tak zwanych gałęziach, które stanowią pewnego rodzaju kopię projektu oraz historię zmian na danej gałęzi co umożliwia łatwy powrót w przypadku popełnienia jakiegoś błędu i chęci powrotu do poprzedniej wersji.

1. Rozgrywka
   1. Interakcja z otoczeniem

Gracz w ramach sterowania obiektami będzie mógł sterować jednostkami oraz budynkami, które są do niego przydzielone. Przeciwnik dysponuje tymi samymi rodzajami obiektów. Gracz może zlecić swoim jednostkom różne polecenia, takie jak:

* Przemieszczenie jednostek
* Atak na jednostkę przeciwnika
* Atak na budynek przeciwnika
* Budowa budynku

Budynki gracza dysponują swoimi funkcjami aktywnymi oraz pasywnymi. Do aktywnych należy rekrutacja jednostek, natomiast do pasywnych dodawanie do waluty w grze.

* 1. Przebieg rozgrywki

Gracz zaczyna z jednym budynkiem głównym „Cytadelą” oraz grupą „podstawowych robotników” oraz z pewną ilością złota. Wraz z czasem budynek generuje przychód, który pozwala na budowę struktur przy pomocy robotników, które pozwalają na zwiększenie zarobków oraz rekrutacje jednostek.

Przeciwnik w tym czasie również wykonuje swoje ruchy w tym wykonując co pewną ilość czasu atak grupą jednostek na bazę gracza usiłując ją zniszczyć.

* 1. Cel gry

Wymiana ataków w pewnym momencie doprowadzi do zyskania przewagi jednej ze stron, strona, która doprowadzi do zniszczenia budynków strony przeciwnej wygrywa i gra się kończy, wyświetlając statystki gracza, informując czy wygrał lub przegrał.

* 1. Podsumowanie rozgrywki

Rozgrywka kończy się w momencie zniszczenia wszystkich struktur jednej ze stron. Skutkuje to pojawieniem się ekranu zwycięstwa lub przegranej. Gracz ma możliwość wyjścia do menu głównego oraz wyświetlenia szczegółowych statystyk z gry.

Dane są pokazane w postaci wykresu określającego, ile gracz posiadał jednostek, złota lub punktów w określonym czasie.



Rysunek przedstawia ekran po zakończonej rozgrywce z wyświetlonym wykresem zależności między złotem a czasem rozgrywki

Liczba punktów jest obliczana na podstawie ilości jednostek, zdobytych zasobów oraz czasu.



liczbaJednostek – liczba określająca, ile gracz zdobył jednostek

modyfikatorJednostek – modyfikator określający jak cenne są jednostki dla punktacji

liczbaZłota – liczba określająca, ile gracz zdobył zasobów

czas – czas trwania rozgrywki, wyrażony w sekundach

* 1. Opis UI

Rysunek przedstawia fragment rozgrywki z wyszczególnionymi elementami interfejsu gracza

1. Ramka zasobów fragment UI przedstawiający aktualne ilości jednostek, złota oraz czas trwania rozgrywki.
2. Ramka akcji zawiera przyciski akcji, które się wyświetlają, jeżeli zostanie zaznaczona jednostka lub budynek zawierający akcje związane z rekrutacją lub budowaniem.
3. Ramka listy przedstawione są tutaj wszystkie zaznaczone jednostki w postaci miniaturki.
4. Ramka statystyk przedstawia szczegółowe statystyki pojedynczego obiektu. W przypadku zaznaczenia wielu obiektów, nic nie będzie w niej wyświetlane.
   1. Sterowanie
      1. Mysz komputerowa

Sterowanie opiera się głównie na myszy komputerowej, z jej pomocą będzie można wykonywać niemal wszystkie operacje.

* + - 1. Zaznaczanie obiektów



Rysunek przedstawia w jaki sposób zwizualizowane jest zaznaczanie jednostek w grze

Zaznaczanie obiektów, polega na wciśnięciu lewego klawisza myszy, przytrzymanie go i przeciągnięcie, pojawi się zielone pole, które oznacza fragment, w którym obiekty zostaną zaznaczone, po puszczeniu lewego przycisku myszy. W sytuacji, kiedy w obrębie pola znajdą się jednostki oraz budynki to tylko jednostki zostaną zaznaczone. Jeżeli tylko pojedynczy budynek znajdzie się w polu to zostanie zaznaczony. Nie ma możliwości zaznaczenia kilku budynków.

* + - 1. Przemieszczanie jednostek



Rysunek zaznaczone jednostki gracza

Zaznaczonym jednostkom można wydać polecenie przemieszczenia się nakierowując kursor w wyznaczone miejsce a następnie kliknięcie prawego przycisku myszy. Jednostki przemieszczając się w wyznaczone miejsce ustawiając się obok siebie wypełniając koło wielkości zależnej od zaznaczonego oddziału.

* + - 1. Atakowanie oraz budowanie

Zaznaczone jednostki mogą atakować lub budować w przypadku pracowników. Komendę wydaje się poprzez najechanie kursorem na obiekt, następnie kliknięcie prawego przycisku myszy. Nakierowanie kursora na obiekty przeciwnika spowoduje wydanie polecenia ataku na budynek lub jednostkę.

Polecenie budowy nastąpi w sytuacji, kiedy wśród zaznaczonych jednostek znajdzie się co najmniej jedna jednostka budowniczych, którzy po naciśnięciu prawego klawisza myszy na budynku nie zbudowanym przemieszczą się do budynku i rozpoczną budowę.

* + - 1. Obsługa funkcji obiektów



Rysunek fragment interfejsu zawierający przyciski

Część obiektów posiada dodatkowe funkcje pokazane w UI w sekcji „Akcji” w postaci kwadratowych guzików, które po naciśnięciu lewym przyciskiem myszy zostaną użyte. Pozwalają one na tworzenie nowych obiektów.

* + 1. Klawiatura
       1. Sterowanie kamerą

Obraz zawierający tekst

Opis wygenerowany automatycznieDo przemieszczania kamery przypisane są klawisze „w”, „s”, „a” oraz „d”, kolejno do przemieszczania w górę, dół, lewo, prawo. Ruch kamery jest ograniczony krawędziami, za które, gracz nie może przemieścić kamery.

Rysunek 9 fragment skryptu odpowiedzialnego za ruch kamery

* 1. Mapa

Jest to obszar na którym toczy się rozgrywka. Gracz na mapie jest umiejscowiony w lewym dolnym rogu obszaru, przeciwnik z kolei na przeciwnej stronie, w prawym górnym rogu.

Mapa pełni funkcję wizualną, urozmaicając pole walki oraz orientację gracza na mapie.

Rysunek 10 mapa gry



* 1. Obiekty statyczne

Kamienie – obiekt stanowiący przeszkodę, przez którą jednostki nie mogą przejść, służy do stworzenia granicy mapy. Jednostki nie mogą w żaden sposób podjąć interakcji z obiektem.

Rysunek 11 fragment granicy mapy

Budynki – stanowią obiekt statyczny, który stanowi przeszkodę, przez którą jednostki nie mogą przechodzić. Interakcje dostępne dla jednostek to budowanie w przypadku budynków sojuszniczych oraz ataku dla budynków przeciwnika.

Obraz zawierający tekst, grafika wektorowa

Opis wygenerowany automatycznie

Rysunek 12 różne typy budynków

* 1. Obiekty dynamiczne

Jednostki – są obiektami, które mają możliwość przemieszczania się po mapie, zarówno samoistnie jak również za pośrednictwem komend gracza. Oddziały mają zaimplementowane animacje, aby najlepiej uwydatnić funkcje jakie wykonują oraz nadać dynamikę do gry.

Jednostki mogą podejmować interakcje z niektórymi obiektami na mapie.

Rysunek 13 różne typy jednostek

1. Implementacja
   1. Główny kontroler

Obiekt w aplikacji odpowiedzialny za prawidłowe uruchomienie i zainicjowanie rozgrywki. Komponenty przypisują odpowiednie obiekty do gracza lub przeciwnika oraz uzupełnia statystyki obiektów znajdujących się na mapie w momencie rozpoczęcia gry na głównej scenie.

Jeden z komponentów obiektu jest odpowiedzialny za kontrole sytuacji na mapie i kontroluje stan budynków obu stron. W przypadku, gdy jedna wyniesie zero, skrypt przerywa rozgrywkę i wyświetla ekran podsumowania.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek przedstawia komponenty w głównym kontrolerze gry

* 1. Sterowanie obiektami

Główny kontroler zawiera skrypt odpowiedzialny za odczytywanie i obsługę urządzeń wejścia dotyczących sterowania obiektami na mapie przy pomocy zaznaczania prostokątem, który jest wyrysowany na mapie od miejsca, w którym użytkownik wcisnął lewy przycisk myszy do aktualnej pozycji kursora, jeżeli klawisz jest przytrzymany, w momencie puszczenia obiekty w obszarze zostają zaznaczone. Komponent został tak skonstruowany, aby możliwe było zaznaczenie wielu jednostek lub jednego budynku.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek przedstawia fragment kodu odpowiedzialnego za zaznaczanie myszą komputerową

* 1. Budowanie

Gracz tworzy struktury jednostek robotników. Zaznaczenie robotnika skutkuje wyświetleniem w ramce akcji możliwych opcji budowania. Po naciśnięci przycisku, w miejscu kursora pojawia się schemat, czyli, przezroczysta grafika przedstawiająca wybraną strukturę, która do momentu podjęcia kolejnej czynności będzie śledziła pozycję kursora. Kliknięcie lewego przycisku myszy powoduje postawienie struktury, jeżeli nie występuje kolizja z już istniejącymi obiektami na mapie. Użycie prawego przycisku powoduje anulowanie budowania i usunięcie schematu.

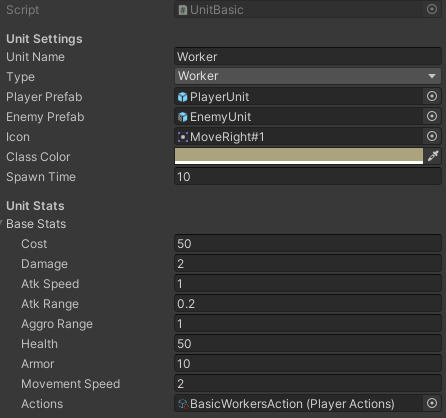
Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek fragment skryptu odpowiedzialnego za budowanie

* 1. Jednostki

Wszystkie występujące w grze jednostki mają zestaw obiektów, odpowiedzialnych za wizualną część obiektu oraz podstawowe statystyki odpowiedzialne za mechanikę, czyli koszt, punkty zadawanych obrażeń, prędkość ataku, zasięg ataku, zasięg agresji, punkt życia, pancerz, prędkość poruszania się oraz opcjonalny zestaw akcji do wykonania, zawierający dostępne struktury do zbudowania przez daną jednostkę.



Rysunek zestaw statystyk oraz obiektów przypisanych dla jednostki pracownika

Dane o każdej jednostce są przechowywane w kontenerze danych „ScriptableObject” co pozwala zmniejszyć zużycie pamięci, poprzez unikanie tworzenia dużej ilości prefabrykatów.

Jednostki dodane do sceny umieszczone w specjalnej strukturze, obiekcie przeznaczonym wyłącznie dla danego typu jednostki, który z kolei znajduje się w jednym z dwóch głównych obiektów nazwanych „PlayerUnits” oraz „EnemyUnits”, odpowiednio dla gracza oraz przeciwnika.

Obraz zawierający tekst

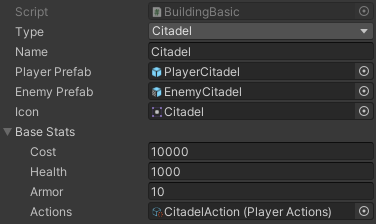
Opis wygenerowany automatycznie Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek przedstawia strukture przechowywania obiektów

* 1. Budynki

Występujące w grze typy budynków podobnie jak jednostki posiadają przypisane obiekty oraz statystki. W odróżnieniu od jednostek, struktury mają mniej wartości, pole akcji przechowuje informacje, które jednostki struktura może zrekrutować.



Rysunek zestaw statystyk oraz obiektów przypisanych do budynku cytadela

Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek przedstawia strukturę przechowywania budynków

Sposób implementacji budynków różni się od jednostek, tym, że prefabrykaty zawierające graficzne elementy dodatkowo zawierają skrypty odpowiedzialne za funkcjonalność danej struktury, zwiększanie ilość złota właściciela budynku oraz skrypt obsługujący rekrutację jednostek.

* 1. Walka

Walka następuje, kiedy jednostka atakuje obiekt nieprzyjaciela, nie musi w tym samym czasie, być celem ataku. Celem może zostać jednostka jak również budynek.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek 21 funkcja odpowiedzialna za poszukiwanie wrogiej jednostki

Jednostki posiadają funkcje za pomocą, której szukają przeciwnika w zasięgu ustalonym w specyfikacji jednostki, jeżeli przeciwnik znajdzie się w promieniu zasięgu agresji, jednostka rozpocznie pościg, który może doprowadzić do dwóch sytuacji:

* Porzucenie – następuje w momencie, kiedy dystans między agresorem a celem będzie większy od dwukrotności promienia zasięgu agresji.
* Atak – agresor zbliża się do celu, jeżeli znajdzie się w zasięgu ataku zadaje obrażenia celowi, następnie kontynuuje pościg.

Gracz posiada możliwość nadania komendy ataku jednostce, w takiej sytuacji jednostka kontynuuje atak do momentu otrzymania innej komendy np. ruchu. W przypadku nadania komendy, jednostka nie może porzucić ścigania przeciwnika.

Doprowadzenie do utraty wszystkich punktów życia przez cel powoduje zniszczenie jednostki, a agresor ponownie przechodzi do szukania nowych przeciwników w obrębie promienia agresji.

Wzór na podstawie, którego są wyliczane obrażenia:

* 1. Szukanie ścieżki

W grze występują obiekty, które będą stanowiły przeszkodę dla przemieszczających się jednostek, przykład stanowią struktury. W celu uniknięcia niepożądanego zatrzymywania się jednostek w miejscu, użyty został system „A\* Pathfinding Project”. System został użyty ze względu na obsługę wyszukiwania ścieżek w projektach 2D.

System bazuje na algorytmie wyszukiwania A\* działającego na grafach, system obsługując aplikację 2D używa kafelkowej mapy, w której kafelki pełnią rolę wierzchołków grafu, z kolei połączenia między kafelkami są krawędziami.

Algorytm A\* zaczyna od pierwszego węzła i uwzględnia wszystkie sąsiednie wierzchołki, które są niedostępne i stanowią przeszkodę, są odfiltrowane, następnie algorytm podejmuje decyzje o wybraniu wierzchołka o najniższej wartości ruchu od punktu startowego. Wartość ruchu składa się z sumy kosztu przejścia z komórki początkowej do komórki bieżącej oraz wartości heurystycznej, czyli oszacowanemu kosztowi przejścia z bieżącej komórki do ostatniej. Algorytm powtarzany jest rekurencyjnie do uzyskania najkrótszej ścieżki do punktu docelowego.

Do oszacowania wartości heurystycznej używa się heurystyki, odpowiedniej to sposobu poruszania się obiektu.

The Manhattan Distance (metryka miasto) – używana jest, gdy obiekt ma się poruszać wyłącznie w czterech kierunkach.

The Euclidean Distance (metryka euklidesowa) – stosowana jest w sytuacji, kiedy kierunek w którym obiekt się porusza nie jest ograniczony.

* 1. Ekonomia

Walutą w grze jest złoto, za które gracz może rekrutować i budować nowe struktury. W grze znajdują się budynki odpowiedzialne między innymi za zwiększanie zarobków, do tych budynków należą:

* Cytadela
* Bank
* Dom
* Farma

Zarobek jest obsługiwany za pomocą skryptu, który w określonym czasie dodaje wyznaczoną ilość złota dodanego do głównej puli.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek funkcja używana do okresowego dodawania złota

* 1. Strategia przeciwnika

Przeciwnik zawiera określone funkcje, odpowiadające za ruch jednostek oraz rozbudowę bazy. Program w sposób losowy wybiera jaką czynność ma podjąć.

* + 1. Jednostki

Jednostki przeciwnika są sterowane za pomocą skryptu, który wydaje komendy, mające na celu atak na bazę gracza. Skrypt wybiera losową ilość jednostek przeciwnika dostępnych na mapie, których liczba stanowi od 10% do 50% wszystkich.

Wybrany oddział jest przemieszczany do wyznaczonego punktu zbiórki skąd po określonym czasie grupa rusza w kierunku struktur gracza. Priorytetem ataku jest główny budynek gracza, jeżeli został zniszczony wybierana jest inna struktura gracza na mapie, a następnie tam wysyłane są kolejne ataki. Zapobiega to sytuacji, w której gracz buduje struktury daleko do swojego głównego budynku.

Jednostki po otrzymaniu komendy zostają usunięte z listy dostępnych jednostek, jednocześnie będąc oznaczone jako wykonujące komendę, dzięki czemu nie zostaną one wybrane ponownie.

W sytuacji, kiedy jednostka zostanie bezczynna po wykonaniu komendy, zostaje ona zawrócona do punktu zbiórki oraz przydzielona do listy dostępnych jednostek.Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek 23 funkcja odpowiedzialna za powrót jednostki

* + 1. Budynki

Skrypt odpowiedzialny za budynki zawiera funkcje obsługujące budowę struktur jak również rekrutację jednostek. Wymienione funkcję są wykonywane w sytuacji, kiedy przypisane przeciwnikowi złoto osiągnie odpowiednią wartość.

Złoto przeciwnika jest przydzielane do dwóch zmiennych, jedna połowa wszystkich środków jest przydzielona na cel rekrutacji jednostek, druga do budowy struktur.

Rekrutacja jednostek jest wykonywana w losowym budynku przeciwnika z dostępnych na mapie. Z listy dostępnych dla danego typu budynku losowana jest jednostka, która zostaje wyznaczona do rekrutacji. Skrypt będzie czekał do momentu, aż przydzielone złoto osiągnie wymaganą wartość po czym zleci rekrutację.

Budowa struktur jest realizowana, jeżeli odpowiednia ilość złota jest przydzielona oraz istnieje wolne miejsce na dostępnej liście pozycji. Algorytm budowy struktur w pierwszej kolejności wybierze budynek o nazwie Cytadela, jeżeli został zniszczony oraz pozostałe warunki budowy są spełnione. W przeciwnym wypadku, jeżeli Cytadela nie została zniszczona, skrypt wybierze losową strukturę z listy dostępnych budynków, jeżeli odpowiednie warunki są spełnione, istnieją robotnicy oraz przeciwnik ma uzbieraną odpowiednią ilość złota.

* 1. Statystki

W grze stale są zapisywane dane wykorzystywane w podsumowaniu rozgrywki.[End this, fajny graf i wogule]

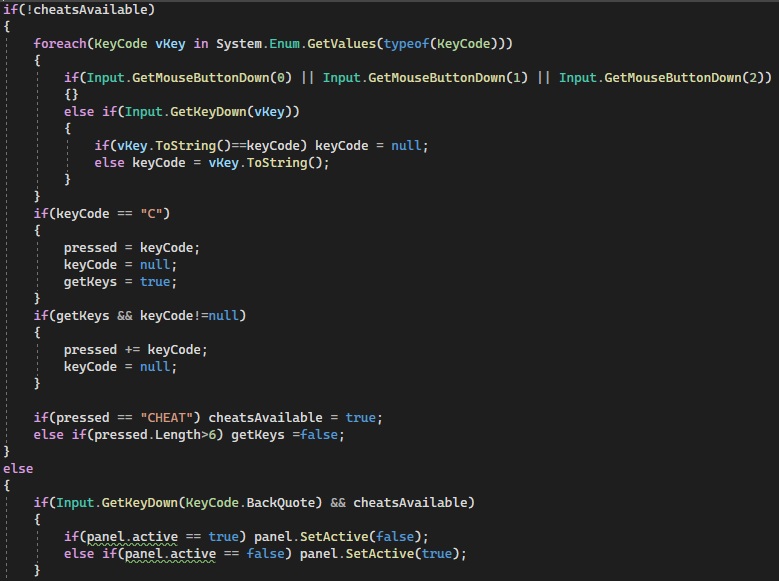
* 1. Komendy

Do gry w celu ułatwienia testowania funkcji użytych w grze została zaimplementowana konsolka oraz komendy wywołujące pewne zachowania, które w normalnej rozgrywce nie są pożądane.

W celu osiągnięcia zezwolenia na używanie komend, należy wpisać następujący ciąg znaków na scenie głównej: „cheat”. Kolejnym krokiem jest kliknięcie klawisza „ ` ” (grawis), spowoduje to wyświetlenie konsolki, w której wpisanie komendy spowoduje odpowiedni efekt.

Zaimplementowane komendy:

* Help – wyświetla dostępne komendy
* Win – powoduje usunięcie budynków przeciwnika, którego efektem jest wygrana.
* Lose - powoduje usunięcie budynków gracza, którego efektem jest przegrana.
* Gold – dodaje do konta gracza 10 000 złota



Rysunek przedstawia fragment skryptu odpowiedzialnego za aktywację konsoli

* 1. Grafika

Grafiki wszystkich obiektów zostały stworzone za pośrednictwem programu GIMP. Jednostkami są uzbrojone „ziemniaki”, animacja przedstawiająca ich ruch oraz nadająca życia obiektom, została stworzona na podstawie klatek będących pojedynczymi grafikami.

Struktury przedstawiają budynki inspirowane przykładowymi tytułami oraz średniowiecznymi motywami architektury.

* 1. Czcionka

Czcionka „Bubble font” w wersji darmowej została zaczerpnięta z Unity Asset Store, jej twórcą jest Jazz Create Games. Użyta, czcionka nadaje grze bajkowego uroku. Czcionka nadaje dodatkowy efekt wizualny, jednocześnie pozostając przejrzysta.

1. Zakończenie

Wynikiem przeprowadzonej pracy jest gra strategiczna czasu rzeczywistego, gra zawiera niezbędne funkcjonalności do prawidłowego działania. Gra zawiera znaczącą większość założonych funkcjonalności. Architektura gry pozwala na dalsze ulepszenia czy zmiany w elementach gry, które ją urozmaicą.

W przyszłości planowane jest wprowadzanie sugestii jakie powstaną w wyniku testowania przez grupę odbiorców, naprawianie ewentualnych błędów oraz ulepszenie skryptów odpowiedzialnych za decyzje przeciwnika.

1. Bibliografia

* <https://www.britannica.com/topic/StarCraft>
* <https://influencermarketinghub.com/biggest-esports-games/>
* <https://www.ign.com/articles/2016/03/22/the-rise-and-fall-of-starcraft-ii-as-an-esport>
* <https://en.wikipedia.org/wiki/The_Lord_of_the_Rings:_The_Battle_for_Middle-earth>
* <https://docs.unity3d.com/Manual/index.html>
* <https://en.wikipedia.org/wiki/Unity_(game_engine)>
* <https://docs.microsoft.com/pl-pl/dotnet/csharp/>
* <https://www.youtube.com/c/Brackeys>
* <https://www.youtube.com/c/CodeMonkeyUnity>
* <https://arongranberg.com/astar/>
* <https://arongranberg.com/astar/docs/>
* <https://www.educative.io/edpresso/what-is-the-a-star-algorithm>
* <https://brilliant.org/wiki/a-star-search/>
* <http://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html>
* <https://assetstore.unity.com/packages/2d/fonts/bubble-font-free-version-24987>
* <https://www.chosic.com/download-audio/28027/>