

PAP 2024Z ZESPÓŁ 16 ETAP I

Dokument, z wymaganiami i funkcjonalnościami, dotyczący projektu zgodny z dostarczonymi instrukcjami.

W przypadku jakichkolwiek niejasności prosimy o kontakt.

Spis Treści

- Skład Osobowy Zespołu
- Tytuł i Opis Projektu
- Funkcjonalności Podstawowe
- Funkcjonalności Dodatkowe
- Wykorzystane Technologie

Skład Osobowy Zespołu

- Maciej Jan Bogusławski
- Bartosz Żelazko
- Hubert Kaczyński
- Amadeusz Lewandowski

Tytuł i Opis Projektu

Gra “Habemus Rex!”

Projekt polega na stworzeniu turowej gry strategicznej osadzonej w czasach XVIII wieku na kontynencie europejskim. Gracz kontroluje imperium wielkiej Polski w celu odwrócenia biegu historii i zapobiegnięciu rozbiorom. Tylko silne i stabilne państwo będzie w stanie przeciwstawić się potęgze trzech wrogich mocarstw. Gra umożliwi graczowi wcielenie się w króla elekcyjnego. Świeżo po jego wyborze na przywódcę narodu musi stawić czoła wielu przeciwnościom losu - nierozwinięte ziemie, pusty skarbiec, zuchwali sąsiedzi i co najgorsze mankamenty demokracji szlacheckiej. Celem gracza jest optymalne zarządzanie zasobami poprzez wznoszenie budynków, rekrutowanie jednostek wojskowych, podbijanie nowych ziem, aby pokonać przeciwników i doprowadzić Rzeczpospolitą Obojga Narodów do prawdziwego rozkwitu.

Funkcjonalności Podstawowe

1. Implementacja funkcjonalności interaktywnej mapy

Etap ten polega na stworzeniu graficznej reprezentacji mapy świata w grze, na której toczy się rozgrywka, a także na zaimplementowaniu funkcjonalności obiektów na mapie pozwalających graczowi na interakcje.

2. Implementacja funkcjonalności związanych z turowością gry

Etap ten odpowiada za zaimplementowanie głównej pętli odpowiedzialnej za wywoływanie wydarzeń, kontrolę zasobów, jednostek, przychodów, kosztów i budynków gracza. Etap ten odpowiada za płynny przebieg rozgrywki, a także utrzymanie poprawnej komunikacji pomiędzy poszczególnymi komponentami gry.

3. Implementacja ekranu startowego gry

Etap ten odpowiada za stworzenie graficznych elementów menu startowego, pozwalających graczowi na rozpoczęcie nowej rozgrywki bądź wczytanie zapisanej wcześniej gry. Etap ten równocześnie polega zatem na pobraniu informacji o dostępnych zapisach gry i graficznej ich prezentacji w celu umożliwienia ich uruchomienia.

4. Implementacja mechaniki pól

Etap ten odpowiada za elementy bazodanowe związane z przypisaniem budynków do konkretnych pól na mapie, a także wyliczanie statystyk oraz informacji o danych polach.

5. Implementacja funkcjonalności wizualnego menu budowy budynków

Etap ten odpowiada za implementację graficznych elementów pozwalających graczowi na budowę dostępnych budynków oraz sprawdzenia informacji o nich, ich kosztu oraz płynących z ich budowy korzyści.

6. Implementacja logiki budynków

Etap ten odpowiada za zaimplementowanie mechaniki przychodów i kosztów wynikających z posiadanych budynków, które bezpośrednio wpływają na posiadane przez gracza surowce.

7. Implementacja logiki zasobów

Etap ten odpowiada za zaimplementowanie bazodanowych elementów związanych z modyfikacją stanu posiadanych przez gracza surowców, informacjami o dostępnych w grze surowcach oraz aktualnych przychodach i wydatkach surowców.

8. Implementacja wizualnych elementów odpowiedzialnych za prezentację statystyk

Etap ten polega na stworzenie graficznych elementów umożliwiających zapoznanie się z graczem z obecnym stanem zasobów, jednostek, przychodów i kosztów. Równocześnie etap ten odpowiada za odświeżanie się wartości stanu poszczególnych parametrów rozgrywki.

9. Implementacja logiki działania przeciwników

Etap ten polega na implementacji algorytmów odpowiedzialnych za zachowanie się pozostałych, wirtualnych graczy w grze. Etap ten ma na celu kontrolę i modyfikację zasobów, przychodów, kosztów i jednostek przeciwników gracza.

10. Implementacja funkcjonalności odpowiedzialnej za logikę walki

Etap ten odpowiada za obliczanie strat gracza oraz wirtualnego przeciwnika w trakcie walki oraz implementacji algorytmu decydującego o zwycięstwie bądź przegranej gracza na podstawie informacji takich jak siła armii, jednostki itp.

11. Implementacja funkcjonalności wizualnego przedstawienia walki

Etap ten odpowiada za graficzną reprezentację informacji związanych z walkami toczonymi przez gracza oraz umożliwia graczowi rozpoczęcie walki oraz przedstawienie informacji dotyczących toczonej walki oraz jej rezultatu.

12. Implementacja funkcjonalności wizualnego menu rekrutowania jednostek

Etap ten odpowiada za implementację graficznych elementów pozwalających graczowi poznać koszty i statystyki dostępnych jednostek oraz umożliwia ich rekrutację.

13. Implementacja logiki jednostek bojowych

Etap ten odpowiada za implementację bazodanowego interfejsu jednostek oraz wyposażenie bazy danych w dostępne w grze jednostki oraz informacje o ich statystykach. Jednocześnie etap ten odpowiada za wyliczanie siły bojowej wszystkich jednostek gracza oraz przeciwnika.

14. Implementacja reagowania na zdarzenia awaryjne

Etap ten obejmuje utworzenie mechanik kontrolujących i reagujących na zdarzenia awaryjne, polegające na wykrywaniu i reagowaniu na niedobory surowców wykorzystywanych przez obiekty lub jednostki. Etap ten zakłada wobec tego np. dodanie mechaniki zatrzymania pracy budynków w przypadku wyczerpania się zasobów, gdy koszt danego budynku nie zostanie zaspokojony bądź eliminacji jednostek z armii gracza, gdy ich koszt utrzymania przekracza stan zasobów gracza.

15. Implementacja mechaniki powiadomień

Etap ten polega na implementacji algorytmów oraz elementów graficznych odpowiedzialnych za prezentowanie oraz informowanie gracza o istotnych wydarzeniach w trakcie rozgrywki. Część powiadomień zależna jest od obecnego stanu gry, inna zaś stanowi losowe uatrakcyjnienie rozgrywki, wzbogacając ją o elementy fabularne.

16. Stworzenie mechaniki punktacji oraz zwycięstwa

Etap ten obejmuje implementację algorytmów odpowiedzialnych za obliczanie wartości tzw. punktów zwycięstwa gracza oraz jego wirtualnych przeciwników.

Punkty te zależne są od czynników takich jak posiadane pola, zbudowane budynki, liczebność i siła armii, posiadane zasoby itp.

17. Kreacja oprawy graficznej rozgrywki

Etap ten polega na tworzeniu elementów graficznych mających na celu uatrakcyjnienie rozgrywki, ułatwienie zrozumienia poszczególnych części gry oraz informowaniu gracza o aktualnym stanie i przebiegu rozgrywki. Etap ten obejmuje tworzenie elementów takich jak ikony, miniatury oraz tła jednostek, budynków, zasobów, poszczególnych rodzajów pól itp.

18. Implementacja funkcjonalności zapisu i odczytu stanu gry

Etap ten odpowiada za implementację funkcjonalności odpowiedzialnych za zachowanie aktualnych parametrów rozgrywki, takich jak obecny moment gry, wybudowane budynki, stan zasobów gracza i przeciwników, rodzaj i ilość wybudowanych jednostek, przynależność pól do gracza lub przeciwników itd. Etap ten równocześnie pozwala na odtworzenie zapisanego stanu rozgrywki.

19. Implementacja mechaniki upływu czasu

Etap ten obejmuje uwzględnienie mechanik oraz elementów graficznych informujących gracza o czasie oraz turze w danej chwili rozgrywki oraz implementacja mechanizmu zatrzymania gry po określonej liczbie tur. Mechanizm ten odpowiedzialny jest również za informowanie gracza o nadchodzącym końcu gry.

20. Stworzenie ekranu pomocy

Etap ten polega na utworzeniu graficznego interfejsu prezentującego graczowi podstawowe informacje dotyczące gry, zawierającego wskazówki, instrukcje oraz objaśnienie fundamentalnych części rozgrywki, takich jak walka, zarządzanie jednostkami oraz zasobami, budowanie budynków, interakcja z elementami mapy itp.

Szczegóły dotyczące funkcjonalności mogą się trochę zmienić podczas tworzenia projektu ze względu na zmiany koncepcji, utrudnienia implementacyjne bądź małą ilość czasu (np. zrobienie czegoś w danej formie zajmie zbyt dużo czasu, który można byłoby poświęcić na stworzenie innych bardziej istotnych funkcjonalności).

Funkcjonalności Dodatkowe

1. Serwer z REST API do pobierania i aktualizacji danych połączony z bazą danych: - system logowania i rejestracji do gry z poziomu klienta gry - system odzyskiwania hasła przez maila - system zapisu i odczytu gry z chmury (zapisy przechowywane w bazie danych np. w postaci JSONów, aby w łatwy sposób umożliwić odtworzenie stanu gry) - webowy mini panel administracyjny do zarządzania zarejestrowanymi użytkownikami (możliwość zmiany np. nazwy, maila, usunięcia konta) - weryfikacja dwu-etapowa w panelu administracyjnym

W przypadku braku implementacji podpunktu bezpośrednio połączenie klienta gry z bazą danych bez serwera (gra offline).

Implementacja powyższej funkcjonalności zależy od możliwości czasowych i przerobowych.

Chcemy na początku stworzyć małe demo na użytek prywatny i przetestować czy w ogóle miałyby to sens i jak dobrze by to działało. Po tym podejmiemy właściwą decyzję czy chyba, że zostanie nam to odradzone.

2. Drzewko technologii w grze

Możliwość wykupienia różnego rodzaju bonusów, za zasoby, ułatwiających rozgrywkę.

3. Wydarzenia losowe

Wydarzenia losowe, które mogą się pojawić w danej turze i zmienić przebieg rozgrywki np. plaga, napad bandytów.

4. Dyplomacja

Mechaniki dyplomacji z przeciwnikami.

5. Handel

Mechaniki handlu z przeciwnikami.

6. Wybieranie poziomu trudności

Możliwość wybrania poziomu trudności gry przed stworzeniem nowego save'a.

Funkcjonalności mogą, ale nie muszą zostać zaimplementowane. Mogłyby się pojawić w dalszej perspektywie rozwijania projektu. W trakcie implementacji projektu mogą pojawić się nowe, które zostaną do niego wdrożone.

Wykorzystane Technologie

Wszystkie wykorzystane technologie są technologiami Java'owymi zgodnie z wymaganiami projektowymi.

1. JavaFX

2. Hibernate (modele bazodanowe)

3. Spring/Spring Boot (do serwera z REST API, w przypadku implementacji funkcjonalności dodatkowej)

4. Thymeleaf (do mini panelu webowego, w przypadku implementacji funkcjonalności dodatkowej)

Lista wykorzystanych technologii może się powiększyć w czasie implementacji danych funkcjonalności, gdyż nie jesteśmy w stanie przewidzieć czy coś dodatkowego może się nie przydać.