The Forest – dokumentacja

# Ogólny opis projektu

Gra The Forest to prosta gra symulująca życie rozbitka, który w celu przetrwania musi wykonywać różne czynności. Celem gry jest pokonanie finałowego potwora. Aby być do tego zdolnym, należy zdobyć odpowiedni ekwipunek - broń i zbroję oraz schronienie.

# Interfejs użytkownika



## Silnik graficzny

Interfejs graficzny został zaprojektowany w oparciu o silnik graficzny WPF. Okno gry posiada możliwość skalowania (minimalne wymiary okna to 600 x 1067 px).

## Komunikaty

W centralnej części ekranu znajduje się pole z komunikatami tekstowymi. Znajdują się w nim opisy wszystkich wykonywanych na bieżąco akcji.

## Ekwipunek

W lewej części ekranu znajduje się ekwipunek gracza. Składają się na niego jedzenie oraz 4 podstawowe surowce: drewno, kamień, skóra i metal. Są one potrzebne do konstruowania przedmiotów, które można zakupić w sklepie. Poniżej znajduję się pole, w którym wyświetlona jest aktualnie posiadana broń i zbroja.

## Statystyki

Powyżej ekwipunku na żółtym polu znajdują się statystyki gracza. Pokazana jest ilość posiadanych pieniędzy, aktualna godzina, dzień gry oraz punkty doświadczenia. Czas gry jest czasem wirtualnym i jest on przyspieszony 200 razy.

## Zapis gry i pauzowanie

Poniżej ekwipunku znajduje pole, w którym możemy zapisać lub wczytać zapis gry. Jest też przycisk pauza, który umożliwia zapauzowanie oraz wznowienie gry. Wówczas zostanie wstrzymany czas i wszystkie aktywności zostaną zablokowane.

## Sklep

W górnej części ekranu znajduje się sklep. Wybierając odpowiednią rzecz z listy, możemy ją kupić lub sprzedać. Należy jednak zwrócić uwagę, żeby mieć odpowiednią ilość pieniędzy i surowców. Informacja o tym znajduje się tuż obok nazwy produktu. Oprócz tego przy broniach zbrojach i schronieniach znajdują się ich odpowiednio atak, obrona i wytrzymałość.

## Życie i energia

Na górze pośrodku znajduje się pole z informacją o punktach energii i życia. Należy zwracać szczególną uwagę na punkty życia, ponieważ gdy osiągniemy 0 punktów, gra się zakończy i trzeba będzie zacząć grać od nowa.

## Obrona i atak

Na prawo od życia i energii znajduje się wskaźnik punktów ataku i obrony. Mają one wpływ na łupy z polowania lub zbierania surowców.

## Las

W prawej części ekranu znajdują się przyciski, które umożliwiają interakcje z lasem. Możemy udać się na polowanie, gdzie zdobędziemy jedzenie. Wówczas jednak stracimy kilka punktów energii i życia - zależnie od tego, z jaką bronią i zbroją się wybierzemy. Oprócz tego możemy zbierać surowce. Wtedy tracimy tylko energię. Możemy również spróbować zawalczyć z finałowym potworem, aby ukończyć grę.

## Jezioro

Jezioro to miejsce rekreacji, możemy tutaj odpocząć i naładować punkty energii albo udać się na wędkowanie i zyskać jedzenie.

## Schronienie

W schronieniu można zjeść jedzenie albo udać się spać. Warto dbać o rozbudowanie schronienia, aby być przygotowanym na niespodziewany atak nocnych potworów.

## Czy zapisać stan gry?

Przy zamykaniu okna wyświetla się okno z zapytaniem o zapisanie stanu gry.

# Mechanika gry

## Początek gry

Na start gracz posiada 3 sztuki jedzenia i 10 monet. Jest w pełni wypoczęty i ma 100 punktów życia.

## Doświadczenie

W czasie gry warto dbać o zróżnicowane aktywności takie jak zbieranie surowców, polowania lub łowienie. Dodają one punkty doświadczenia, co wpływa na efektywność naszych akcji.

## Czas gry

Warto zwrócić uwagę na biegnący czas. Każda aktywność w grze zajmuje pewien czas. Warto zerkać na zegar w porach wieczornych i wykorzystać noc do spania.

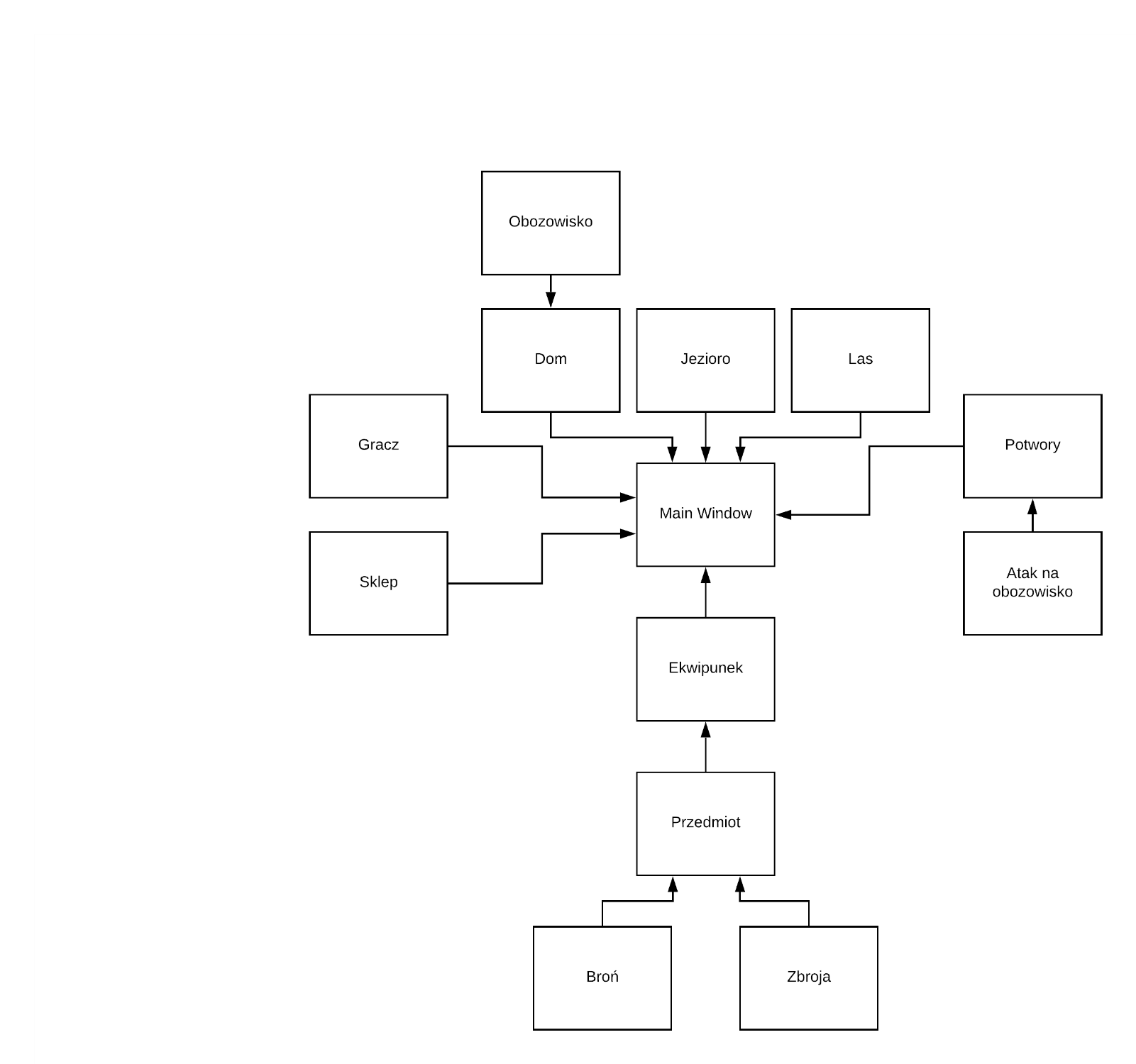
## Atak na obozowisko

W czasie nieprzespanej nocy, hałas zdradza potworom naszą lokalizację. Mogą się one pokusić na zaatakowanie naszego obozowiska. Dlatego warto ulepszać nasze schronienie, co zmniejszy szansę na zaatakowanie naszego domostwa.

## Finał gry

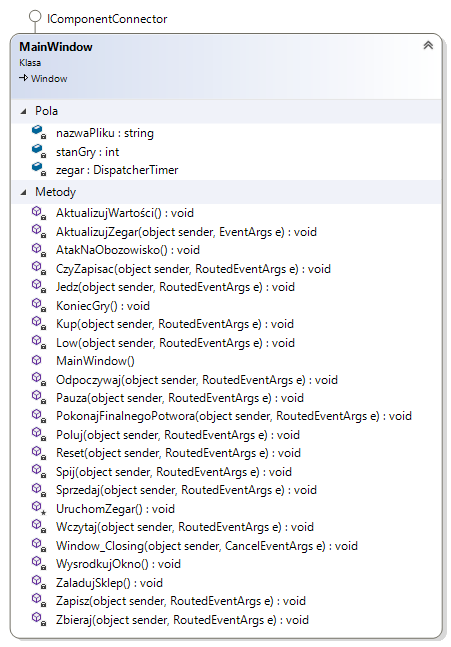
W celu ukończenia gry należy rozprawić się z finałowym potworem. Aby tego dokonać, wymagane są najlepsza broń i zbroja oraz najlepsze zbudowane schronienie. Ponadto trzeba mieć maksymalną ilość zdrowia i energii.

# Uproszczony diagram klas



# Opis klas

## MainWindow



Klasa jest klasą główną programu. Jest odpowiedzialna za inicjalizacje komponentów gry oraz reagowanie na wszystkie interakcje z użytkownikiem. Obsługuje również liczenie czasu w grze oraz zdarzenia losowe związane z porą dnia.

Pola klasy:

* nazwaPliku : private readolny string – przechowuje nazwę pliku, w którym jest zapisywany stan gry
* stanGry : private int – przechowuje informacje o tym czy gra trwa, jest w stanie pauzy lub się zakończyła
* zegar : private readolny DispatcherTime – zmienna obsługująca bieg czasu w grze

Metody klasy:

* MainWindow() : void – konstruktor klasy, inicjalizuje interfejs i pozostałe komponenty programu
* Kup(object sender, RoutedEventArgs e) : void – na podstawie wybranego elementu na w ListBoxie *sklep* wywołuje funkcję Sklep.Kup() i wyświetla komunikat tekstowy o zakupie. Następnie wywołuje funkcję AktualizujWartości(), która aktualizuje wszystkie pola tekstowe.
* Sprzedaj(object sender, RoutedEventArgs e) : void - na podstawie wybranego elementu na w ListBoxie *sklep* wywołuje funkcję Sklep.Sprzedaj() i wyświetla komunikat tekstowy o sprzedaży. Następnie wywołuje funkcję AktualizujWartości(), która aktualizuje wszystkie pola tekstowe.
* Poluj(object sender, RoutedEventArgs e) : void – wywołuje funkcję Las.Poluj() i wyświetla komunikat o polowaniu. Następnie wywołuje funkcję AktualizujWartości(), która aktualizuje wszystkie pola tekstowe.
* Zbieraj(object sender, RoutedEventArgs e) : void – wywołuje funkcję Las.Zbieraj() i wyświetla komunikat o zebranych surowcach. Następnie wywołuje funkcję AktualizujWartości(), która aktualizuje wszystkie pola tekstowe.
* Odpoczywaj(object sender, RoutedEventArgs e) : void – wywołuje funkcję Jezioro.Odpoczywaj() i wyświetla komunikat o zyskanej energii. Następnie wywołuje funkcję AktualizujWartości(), która aktualizuje wszystkie pola tekstowe.
* Low(object sender, RoutedEventArgs e) : void – wywołuje funkcję Jezioro.Low() i wyświetla komunikat o złowionych rybach. Następnie wywołuje funkcję AktualizujWartości(), która aktualizuje wszystkie pola tekstowe.
* Jedz(object sender, RoutedEventArgs e) : void – wywołuje funkcję Obozowisko.Jedz() i wyświetla komunikat o zyskanym zdrowiu i energii. Następnie wywołuje funkcję AktualizujWartości(), która aktualizuje wszystkie pola tekstowe.
* Spij(object sender, RoutedEventArgs e) : void – wywołuje funkcję Jezioro.Low() i wyświetla komunikat o złowionych rybach. Następnie wywołuje funkcję AktualizujWartości(), która aktualizuje wszystkie pola tekstowe.
* PokonajFinalnegoPotwora(object sender, RoutedEventArgs e) : void – symuluje pokonywanie potwora. Sprawdza czy gracz ma wystarczający ekwipunek do pokonania potwora i wyświetla komunikat o pokonaniu lub porażce. Następnie wywołuje funkcję AktualizujWartości(), która aktualizuje wszystkie pola tekstowe.
* AktualizujWartości() : void – aktualizuje wartości we wszystkich polach tekstowych oraz sprawdzająca czy gracz żyje
* KoniecGry() : void – wywoływana gry gracz umiera – wyświetla komunikat o końcu gry, blokuje interakcję z przyciskami i wyświetla pole końca gry
* ZaladujSklep() : void – dodaje wszystkie elementu podlegające kupnie lub sprzedaży do ListBoxi *sklep*
* WysrodkujOkno() : void – wywoływana podczas inicjalizacji gry – ustawia okno na środku ekranu w przypadku, gdy gra nie jest uruchamiana w trybie pełnoekranowym
* UruchomZegar() : void – inicjalizuje działanie zegara liczącego czas w grze
* AktualizujZegar() : void – wywoływana z każdym cyklem zegara, co 300 ms – w odpowiednich godzinach wywołuje funkcję AtakNaObozowisko(). Aktualizuje pola tekstowe godziny. Jeżeli gracz żyje – dodaje minutę do czasu gry.
* Zapisz(object sender, RoutedEventArgs e) : void – tworzy plik w którym zapisuje wszystkie zmienne związane z aktualnym stanem gry i wyświetla komunikat o pomyślnym zapianiu.