PRM2T PROJEKT
Nurikabe

Kozłowski Jakub / 314656/ grupa T1
Olech Małgorzata / 311101 / grupa T1
Sosulski Jan / 311445 / grupa T1

Spis treści

1. ETAP 1

W tym etapie zajęliśmy się organizacją pracy przed rozpoczęciem realizacji konkretnych zadań projektowych i zbieraniem danych odnośnie logiki gry i jej możliwych opcji implementacji w Javie.

1.1. Zasady gry Nurikabe.

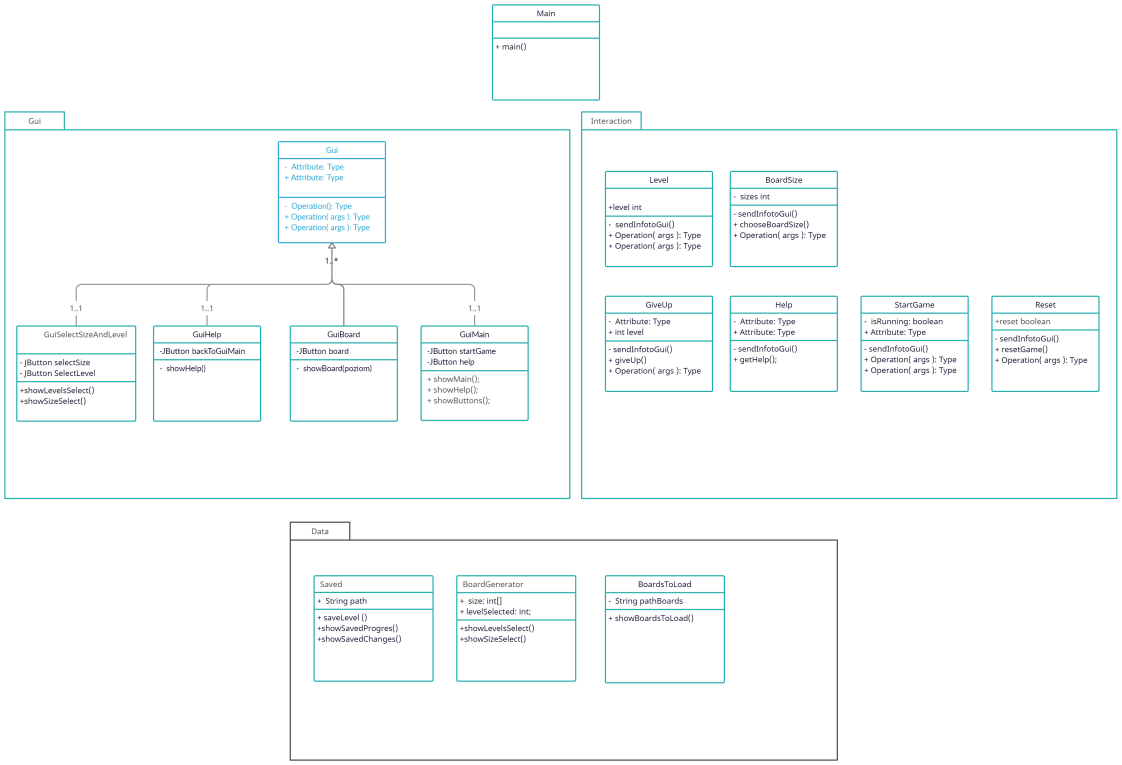
Nurikabe jest rozgrywana na prostokątnej siatce komórek, z których niektóre zawierają cyfry. Komórki mają początkowo nieznany kolor, ale mogą być tylko czarne lub białe. Dwie komórki tego samego koloru są uważane za „połączone", jeśli sąsiadują ze sobą pionowo lub poziomo, ale nie po przekątnej. Połączone białe komórki tworzą „wyspy", a połączone czarne komórki tworzą „morze". Ogólnie zasady można podsumować w punktach:
1. Każda ponumerowana komórka jest komórką wyspową, a liczba w niej to liczba komórek na tej wyspie.
2. Każda wyspa musi zawierać dokładnie jedną ponumerowaną komórkę.
3. Musi istnieć tylko jedno morze, które nie może zawierać „basenów", tj. 2×2 obszarów czarnych komórek.

1.2. Słowny opis przypadków użycia.

Użytkownik po uruchomieniu gry widzi interfejs startowy. Może w nim rozpocząć grę, uprzednio podając interesujący go poziom łamigłówki i rozmiar planszy. Ma również możliwość zobaczenia informacji o autorach i procesie tworzenia gry. Jeśli już wcześniej rozpoczął jakiś poziom może do niego powrócić, ew. zobaczyć plansze już rozwiązanych poziomów. Gracz może również wejść w help w którym będą wyjaśnione zasady gry.

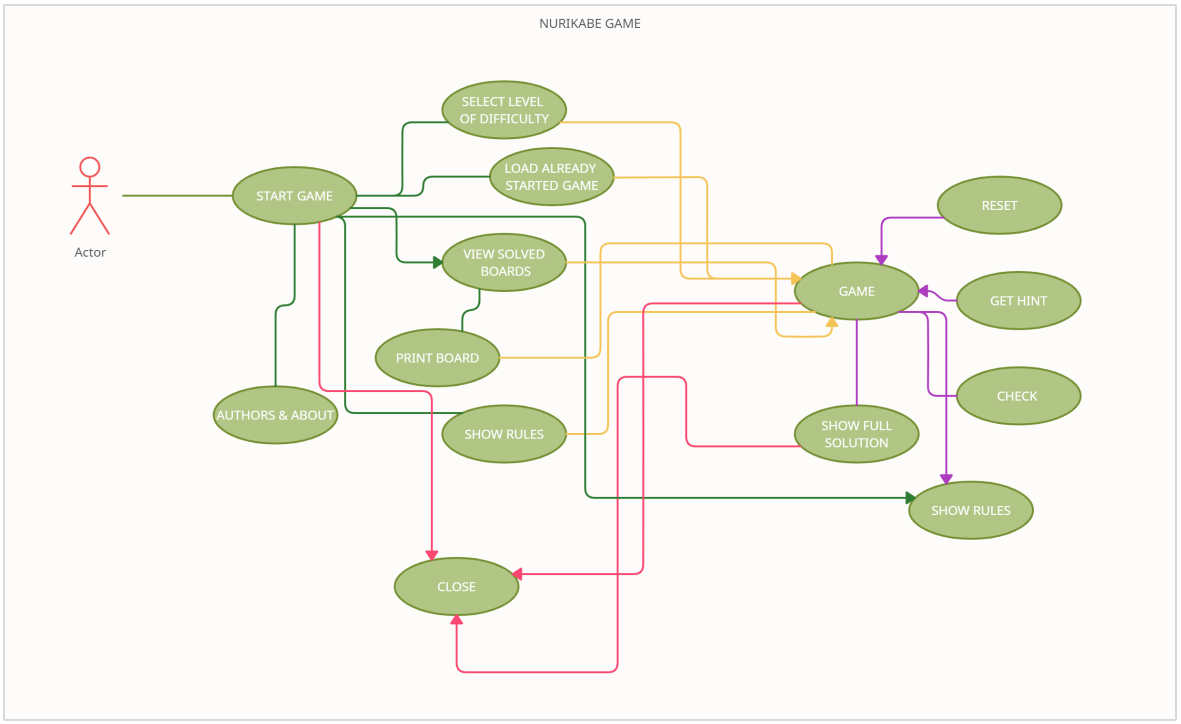
Po wybraniu poziomu gry i rodzaju planszy zostanie ona wyświetlona i gotowa do rozwiązywania. W przypadku kontynuacji zostanie załadowana plansza z momentu w którym została zapisana poprzednio i będzie możliwość jej kontynuacji. Od rozpoczęcia gry będzie odmierzany czas i po rozwiązaniu planszy zostanie ukazana możliwość zapisania czasu ukończenia. Możliwe jest uzyskanie podpowiedzi kolejnego kroku lub uzupełnienie całej planszy(jednoznaczne z rozwiązaniem). Także w trakcie gry będzie możliwość wydruku zapisanej planszy i rozwiązania dalszego w wersji np papierowej. W każdym momencie gry można wrócić do menu głównego w celu zapoznania się jeszcze raz z zasadami gry lub zmiany poziomu trudności/wielkości planszy.

1.3. Diagram UML naszego projektu.



Rysunek 1. Diagram UML (wersja do powiększonego odczytu w oddzielnym pliku)

1.4. Graficzne przedstawienie przypadków użycia.



Rysunek 2. Diagram przypadków użycia.

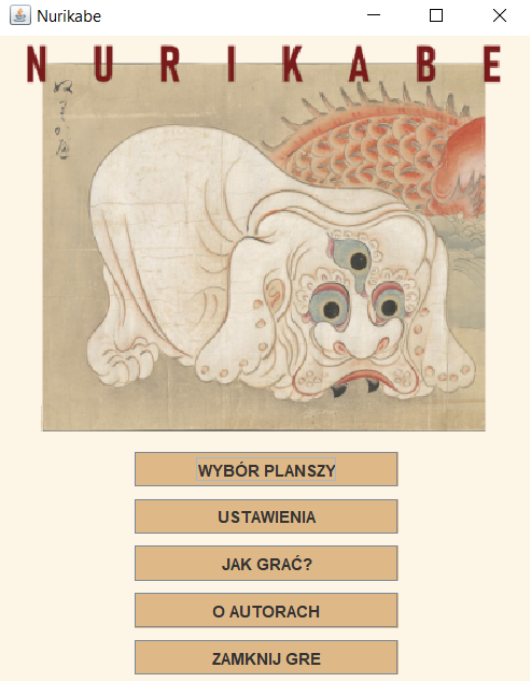
2. ETAP 2

W tym etapie zajęliśmy się realizacją naszej gry oraz weryfikacją naszych pomysłów na ich implementacje w Javie.

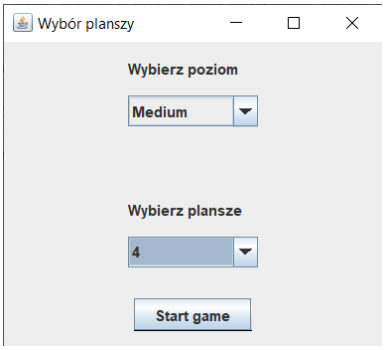
2.1. Interfejs graficzny użytkownika.

2.1.1. Menu główne

Po uruchomieniu aplikacji pojawia się okienko, będące panelem startowym, w którym za pomocą przycisków użytkownik może przejść do innych sekcji.

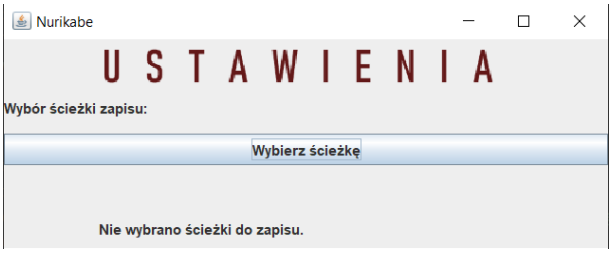


WYBÓR PLANSZ - po kliknięciu guzika przez użytkownika, wyskakuje okienko, w którym ma on możliwość wyboru trzech poziomów trudności (które odpowiadają trzem rozmiarom plansz: 4x4 - Easy, 5x5 - Medium, 7x7 - Hard).

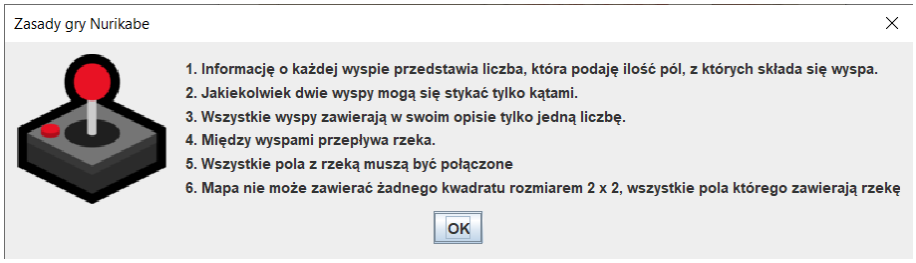


Przy pierwszym uruchomieniu gry na komputerze, program przed przejściem do panelu wyboru planszy poprosi użytkownika o podanie ścieżki do zapisu i późniejszego zapisu stanu gry (pliku **.csv** z wszystkimi planszami i wprowadzonymi przez użytkownika zmianami) oraz wydrukowanych plansz.

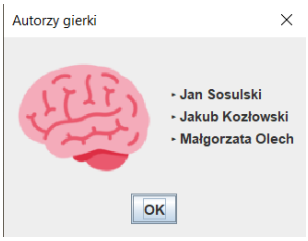
USTAWIENIA - po kliknięciu guzika przez użytkownika, otwiera się okno umożliwiające wybranie ściezki do zapisu i późniejszego odczytu stanu gry z pliku.



JAK GRAĆ ? - okienko JOptionPane umożliwia użytkownikowi na zapoznanie się z ogólnymi zasadami gry.



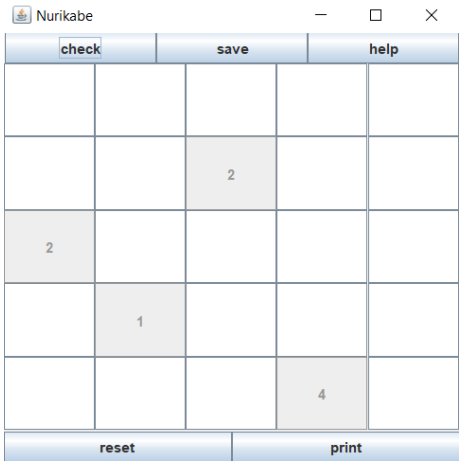
O AUTORACH - okienko JOptionPane informujące użytkownika o autorach gry.



ZAMKNIJ GRĘ - guzik zamykający program i kończący jego działanie.

2.1.2. Plansza

Po wybraniu ścieżki do zapisu/odczytu gry, wybraniu interesującego go poziomu trudności i numeru planszy użytkownikowi otwiera się okienko z właściwą grą.



Widoczna plansza składa się z pól określających liczbę pół białej wysepki oraz pól możliwych do kliknięcia przez użytkownika - czarne pole oznacza wyspę, a szare oznacza pewność użytkownika, że w tym miejscy spodziewa się "morza".

check - Przycisk, który umożliwia przekazanie użytkownikowi informacji zwrotnej, czy jego rozwiązanie jest poprawne. Jeśli zgadza się ono z rozwiązaniem z klucza rozwiązań użytkownik dostaje komunikat z gratulacjami.

save - W przypadku poprawnego wyboru ścieżki do zapisu/odczytu kliknięcie guzika umożliwia zapis stanu gry. W innym przypadku użytkownik zostanie poproszony o przejście do ustawień w Menu Głównym i wybranie ścieżki.

help - Przycisk odpowiadający za wyświetlenie podpowiedzi użytkownikowi.

reset - Przycisk umożliwiający wczytanie pustej planszy - czyli jej zresetowanie w celu rozpoczęcia rozwiązywania jej na nowo.

print - guzik odpowiadający za przygotowanie planszy do ew. wydruku jej przez użytkownika - zapisuje ona plansze w aktualnym stanie do pliku **.png** w lokalizacji wskazanej przez użytkownika w ustawieniach.

2.2. Omówienie klas z package'u csv.

Package zawiera jedną klasę: **OpenCsvData**. Jej działanie skupia się wokół obsługi pliku boards.csv, w którym to przechowywane są dane odnoście plansz wszystkich poziomów oraz wybrana przez użytkownika ścieżka do jego zapisu. Klasa korzysta z dodatkowego modułu OpenCSV. Konstruktor tej klasy, przyjmujący daną typu String z nazwą pliku, otwiera go, a odczytane dane zapisuje w specjalnej tablicy. Klasa posiada kilka metod get, umożliwiających dostęp do konkretnych danych z pliku - puste plansze, prawidłowe rozwiązanie, zapisany stan gry oraz metodę umożliwiającą zapis zmian użytkownika do pliku.

2.3. Omówienie klas z package'u Gui.

Package zawiera 6 klas, które odpowiedzialne są za graficzny interfejs użytkownika.
Gui jest naszym szkieletem który zawiera informacje o startowych parametrach okna Menu.
Poszczególne plansze do gry powstają w klasach kolejno **Board4_4**, **Board5_5** oraz **Board7_7**. Odpowiadają one za generowanie danej planszy którą użytkownik wybierze, wizualne jej przedstawienie oraz przyjmują informacje o interakcjach użytkownika.
Wyżej wymienione 3 klasy dziedziczą z klasy **Board** która przechowuje metody związane z sprawdzeniem planszy, uzyskiwaniem podpowiedzi, resetowaniem planszy jak i jej zapisem do pliku.

2.4. Omówienie klas z package'u interactions.

Package zawiera 7 klas, które odpowiedzialne są za prowadzenie interakcji z użytkownikiem - wyświetlanie komunikatów, wprowadzanie jego zmian i zapis wprowadzonych przez niego ustawień.

AboutAuthors - Klasa służąca prostemu wyświetleniu okienka z autorami gry, po wciśnięciu w Menu Głównym odpowiedniego przycisku.

GameRules - Klasa służąca prostemu wyświetleniu okienka z podstawowymi zasadami gry, po wciśnięciu odpowiedniego przycisku w Menu Głównym.

Print - Klasa odpowiedzialna za przygotowanie pliku, który użytkownik będzie w stanie wydrukować. Jej konstruktor przyjmuje JPanel, który to później będzie zapisywany do pliku typu png. Zabezpieczona jest na wypadek wcześniejszego niepodania przez użytkownika ścieżki do zapisu plików programu - wyświetlany jest wtedy użytkownikowi stosowny komunikat instruujący jakie kroki musi wykonać zanim zapisze plansze do wydruku oraz na wypadek błędów przy konwersowaniu w obiekt typu ImageIO - wtedy również wyświetlany jest stosowny komunikat. Użytkownik ma też możliwość nadania nazwy plikowi przed jego zapisem.

SaveFile - klasa umożliwiająca zapis po raz pierwszy pliku w podanej przez użytkownika lokalizacji. Zapisywany plik jest kopią obecnego w programie na stałe pliku boards.csv, pod taką nazwą będzie on też funkcjonował na dysku użytkownika.

SelectBoardSize - Klasa odpowiedzialna za umożliwienie użytkownikowi wyboru poziomu trudności (Easy / Medium / Hard) oraz numeru interesującej go planszy. Jeśli plik z zapisanymi stanami gry na komputerze użytkownika uległ uszkodzeniu/usunięciu lub nie został jeszcze utworzony to klasa korzysta z pliku boards.csv obecnego w kodzie programu. Posiada metodę, która umożliwia późniejszy dostęp do danych mówiących o poziomie i numerze wybranej planszy.

Settings - Klasa umożliwiająca użytkownikowi wybór ścieżki do zapisu i odczytu danych poprzez korzystanie z JFileChooser. Wybrana ścieżka jest zmienną statyczną, z który korzystamy potem w innych miejscach programu. W przypadku zamknięcia okienka umożliwiającego wybór, przed jego dokonaniem użytkownikowi wyświetla się stosowny komunikat ostrzegający przed tym.

StartGame - Klasa generująca główny interfejs graficzny użytkownika (opisana w punkcie 2.2.1 sprawozdania). Umożliwia wyświetlenie grafiki przedstawiającej chińskiego mitycznego potwora, od którego gra wzięła swoją nazwę oraz szeregu przycisków umożliwiających dalszą grę.

3. Podsumowanie projektu.

3.1. Co udało nam się zrealizować.

Zgodnie z wytycznymi przekazanymi na początku semestru udało nam się zrealizować następujące elementy tam wyszczególnione:

1. program umożliwiający grę w Nurikabe
2. plik i metody jego odczytu/zapisu umożliwiający generowanie plansz i istnienie do nich solwera
3. generowane przez nas plansze są dostępne w różnych poziomach trudności
4. użytkownik ma możliwość zapisu aktualnego stanu gry i późniejsze go wczytanie
5. użytkownik może uzyskać wskazówkę w trakcie rozwiązywania łamigłówki
6. możliwość sprawdzenia, czy wprowadzone rozwiązanie jest poprawne
7. reset planszy do stanu początkowego
8. zapis planszy umożliwiający czytelny wydruk łamigłówki

3.2. Czego nie udało nam się zrealizować.

Nie udało nam się stworzyć niezależnego generatora plansz oraz inteligentnego solwera. Wynika to z tego, że generowanie plansz wymagało by bardzo złożonego algorytmu, tak aby tworzone plansze były rozwiązywalne i równie złożonego algorytmu, który generowałby rozwiązanie. Poza tym jednym aspektem udało nam się zrealizować wszystkie inne założenia naszego projektu.