Connect 4 GUI

Należy napisać klasy pozwalające na przeprowadzenie pojedynczej rozgrywki gry Connect 4 dla wybranej przez użytkownika: ilości kolumn i wierszy, ilości graczy (jeśli wybrany jest jeden gracz, to ruchy drugiego gracza realizuje komputer).

W pisanej wcześniej klasie Connect4 należy dodać:

* metodę setGameStateListener(), która zapamięta odbiornik GameStateListener
* wywołanie metody gameFinished() odbiornika z argumentem: 0 - jeśli jest remis, 1 lub 2 jeśli wygrał jeden z graczy
* wywołanie metody waitingForPlayer() odbiornika z argumentem: 1 lub 2 w momencie, gdy zmienia się wartość zwracana przez czyjaKolej()
* metodę: public int dajWartosc(int wiersz, int kolumna) - dającą zawartość odpowiedniego pola planszy
* metody pozwalające sprawdzić rozmiar planszy, np. getLiczbaKolumn() i getLiczbaWierszy()
* statyczne pole (stałą) TRYB\_GRY\_Z\_KOMPUTEREM\_DOSTEPNY o wartości true lub false w zależności od zaimplementowanej funkcjonalności
* domyślny rozmiar planszy jako dwie stałe

Wybór parametrów gry realizowany jest w oknie dialogowym - klasa Connect4Config. Przykładowy wygląd interfejsu:

Obraz zawierający tekst

Opis wygenerowany automatycznie

Przykładowy szkielet klasy Connect4Config:

**public** **class** Connect4Config **extends** JDialog {

        // pola

**public** Connect4Config() {

**super**((JDialog)**null**, **true**);

                createGUI();

        }

**private** **void** createGUI() {

                // utworzone i ułożone komponenty

                getRootPane().setDefaultButton(okButton);

                setDefaultCloseOperation( JDialog.***DO\_NOTHING\_ON\_CLOSE*** );

setLocationRelativeTo(**null**);

// obsługa zdarzeń dla okButton i cancelButton

        }

**public** **int** getRows() { …. }

**public** **int** getCols() { ….

**public** **int** getNumberOfPlayers() { …. }

}

W metodzie createGui() należy utworzyć wszystkie niezbędne komponenty i zadbać o ich ułożenie w oknie (można to uzyskać np. korzystając z SpringLayout). Listy rozwijane (ilości wierszy i kolumn) to JComboBox<Integer>, dla których należy ustawić początkowe wartości (standardowe 6 wierszy i 7 kolumn) przy pomocy setSelectedItem(). Jeśli tryb gry z komputerem nie jest wspierany, to pole powinno być wyłączone (setEnabled(false)).

Odbiorniki zdarzeń dla przycisków Zakończ (cancelButton) i Graj (okButton) wywołują odpowiednio:

                        System.*exit*(0);

i

                        setVisible(**false**);

                        dispose();

Metody getRows(), getCols(), getNumberOfPlayers() zwracają wartości wybrane w odpowiednich polach.

W metodzie Run.main() - pamiętając o np. SwingUtilities.invokeLater() - należy pokazać okno Connect4Config. Okno to jest modalne, zatem kolejne polecenia wykonane zostaną po jego zamknięciu.

Jeśli wciśnięty był przycisk Graj, to tworzony i wyświetlany jest obiekt Connect4GUI, któremu jako parametry konstruktora przekazywane są ilości wierszy, kolumn i graczy.

Szkielet klasy:

**public** **class** Connect4GUI **extends** JFrame **implements** ColumnClickListener, GameStateListener {

**// pola**

**public** Connect4GUI(**int** rows, **int** cols, **int** numberOfPlayers) { … }

**private** **void** createGUI() { … }

        @Override

**public** **void** columnClicked(**int** column) { … }

        @Override

**public** **void** gameFinished(**int** winner) { … }

        @Override

**public** **void** waitingForPlayer(**int** player) { … }

}

W konstruktorze:

* tworzony jest obiekt klasy Connect4 z planszą o podanych wymiarach
* dla obiektu Connect4 ustawiany jest GameStateListener jako Connect4Gui.this
* wywołana metoda createGui()
* ustalany rozmiar okna i jego położenie

W metodzie createGUI():

* tworzony interfejs składający się z etykiety (pokazującej którego gracza kolej) - wyświetlanej tylko, jeśli gra dwóch graczy i obiektu Connect4Plansza
* dla obiektu Connect4Plansza ustawiany jest ColumnClickListener jako Connect4Gui.this

Obraz zawierający stół

Opis wygenerowany automatycznie

Metoda columnClicked() z interfejsu ColumnClickListener:

* nie robi nic jeśli gra jest zakończona
* wywołuje metodę wrzucKrazek( kolumna ) i ew. wykonajRuch() w przypadku gry z komputerem
* aktualizuje planszę
* jeśli kolumna była pełna wyświetla stosowny komunikat

Metoda gameFinished() z interfejsu GameStateListener:

* zapamiętuje, że gra jest zakończona
* pokazuje informację o zwycięzcy lub remisie

Metoda waitingForPlayer() z interfejsu GameStateListener:

* aktualizuje etykietę pokazującą czyja kolej

Klasa Connect4Plansza dziedziczy z JPanel i wyświetla siatke etykiet JLabel:

​​**public** **class** Connect4Plansza **extends** JPanel {

// pola

**public** Connect4Plansza(**final** Connect4 gra) {

                // zapamiętać obiekt gra

                // ustawić layout

// utworzyć i zapamiętać w tablicy obiekty JLabel

                // dla każdego zarejestrować odbiornik MouseInputAdapter

                // w mouseClicked() wywołać makeMove()

// w mouseEntered() i mouseExited() wywołać odpowiednio selectColumn(..., true) lub selectColumn( …, false)

        }

**protected** **void** selectColumn(**int** column, **boolean** select) {

// zmienić tło - zaznaczyć lub odznaczyć całą kolumnę pól

        }

**protected** **void** makeMove(**int** columnNumber) {

                // wywołać columnClicked() odpowiedniego odbiornika

        }

**public** **void** aktualizuj() {

                // korzystąc z metody dajWartosc(wiersz, kolumna ) zaktualizować zawartość etykiet

        }

**public** **void** setColumnClickListener(ColumnClickListener listener) {

**this**.columnClickListener = listener;

        }

}

Podpowiedzi:

* dla etykiet JLabel należy ustawić nie-przezroczystość:

etykieta.setOpaque(**true**);

aby móc zmieniać kolor tła, np.

etykieta.setBackground( select?Color.***white***:UIManager.*getColor* ( "Panel.background" ));

Na wikampie należy przesłać sprawozdanie w postaci pliku PDF, w którym:

* pokazany zostanie kod wszystkich napisanych klas (wraz z opisem, jeśli klasy są inne niż proponowane)
* zrzuty ekranu ilustrujące działanie poszczególnych funkcji programu
* w komentarzu należy podać procent zdobytych punktów (razem 15p):
* nowe funkcje dodane do klasy Connect4 (1p)
* parametry gry (ilość wierszy i kolumn, ilość graczy) ustalane w oknie dialogowym zbliżonym do pokazanego w instrukcji (4p)
* ustawione początkowe wartości list rozwijanych zgodnie ze stałymi z klasy Connect4 (1p)
* pole wyboru gry z komputerem zablokowane jeśli taka gra nie jest wspierana (1p)
* rozmiar okna z planszą dostosowany do ilości wierszy i kolumn (1p)
* etykieta z informacją o kolejce gracza widoczna tylko jeśli jest dwóch graczy i aktualizowana po ruchu gracza (1p)
* zawartość planszy aktualizowana po wykonaniu ruchu (1p)
* kolumna pod kursorem myszy podświetlana (1p)
* pokazywany komunikat o zakończeniu gry (wygraną lub remisem) (1p)
* brak możliwości wykonywania kolejnych ruchów po wygranej (1p)
* wrzucany krążek po kliknięciu kolumny i wykonywany ruch przez komputer (jeśli tryb gry z komputerem) (1p)
* próba wrzucenia krążka do pełnej kolumny daje odpowiedni komunikat (1p)

Literatura:

SpringLayout - [https://docs.oracle.com/javase/tutorial/uiswing/layout/spring.html](https://www.google.com/url?q=https://docs.oracle.com/javase/tutorial/uiswing/layout/spring.html&sa=D&source=editors&ust=1642956859629416&usg=AOvVaw2JKels8Y9Kj6AUI7_L2KMh)