

PROGRAMOWANIE ZAAWANSOWANE – PROJEKT

DOKUMENTACJA GRY WARCABY

DIAGRAM KLAS

Okno extends JFrame implements ActionListener, MouseMotionListener
+ JPanel panel + JMenuBar menuBar + JMenu menuGra, menuInformacje + JMenuItem mNowa, mCofnij, mWyjscie + JMenuItem mOProgramie, mOAutorze
+ <u>void main(String args[])</u> + void actionPerformed((ActionEvent) e) + void mouseDragged(MouseEvent e) + void mouseMoved(MouseEvent e)

Panel extends JPanel implements ActionListener MouseListener
+ Graphics2D g2d + JButton nowaGra, cofnij + Color Black, White, Tlo + Plansza plansza + <u>boolean wyswietlanyStanGry = false</u>
void void paintComponent(Graphics g) + void actionPerformed((ActionEvent) e) + void mouseClicked(MouseEvent e) + void mouseEntered(MouseEvent e) + void mouseExited(MouseEvent e) + void mousePressed(MouseEvent e) + void mouseReleased(MouseEvent mysz) + void sprawdzWarunki() + void rysujMenu() + void rysujPlansze()

Gracz
+ <u>int ruch, IPionkow1, IPionkow2</u>
+ <u>void zmienGracza()</u> + <u>void ustawStanPocatkowy()</u>

Pole
<u>+ int szer = 48</u> - boolean dozwolone - boolean zaznaczonePole - boolean zaznaczonyPionek - boolean zaznaczonaDamka - boolean proponowane - boolean proponowaneBicie - char stan
+ void wyczysc()

Plansza
<u>+ int szer = 8</u> <u>+ Pole pole[][]</u> <u>+ int obecneX</u> <u>+ int obecneY</u> <u>+ int poprzednieX</u> <u>+ int poprzednieY</u> <u>+ int przeciwnikX</u> <u>+ int przeciwnikY</u> <u>+ boolean ustawionePionki</u> <u>+ boolean byloBicie;</u> <u>+ boolean poprzednioZaznaczonyPionek = false</u> <u>+ boolean mozliweCofniecie</u> <u>+ boolean cofniecie</u>
+ void ustawPionki + void wyswietl() + boolean sprawdzZaznaczeniePionka() + boolean sprawdzCzyDamka() + boolean zaznaczPionek(int myszX, int myszY) + void zaproponujRuch() + void zaproponujRuchDamka() + boolean wykonajRuch(int myszX, int myszY) + void cofnijRuch() + void wyczyscProponowaneRuchy() + void odznaczPionek()

OPIS KLAS

KLASA OKNO

- + JPanel panel // = new Panel() - obiekt klasy JPanel, z tym, że wskazuje referencyjnie na klasę potomną Panel
- + JMenuBar menuBar - obiekt klasy JMenuBar, czyli pasek menu
- + JMenu menuGra - opcja z menuBar, jest rozwijana
- + JMenu menuInformacje - opcja z menuBar, jest rozwijana
- + JMenuItem mNowa - konkretny przycisk w menu
- + JMenuItem mCofnij - j.w.
- + JMenuItem mWyjście - j.w.
- + JMenuItem mOProgramie - j.w.
- + JMenuItem mOAutorze - j.w.
- + void main(String args[]) - główna funkcja aplikacji, od niej zaczyna się działanie
- + void actionPerformed((ActionEvent e) - gdy zostanie wybrana opcja z menu, to ta funkcja wykonuje instrukcje dla danego przycisku
- + void mouseDragged(MouseEvent e)
- + void mouseMoved(MouseEvent e) - sprawdza czy myszka została poruszona; funkcja użyta do odświeżania stanu przycisku menu Cofnij

KLASA PANEL

- + Graphics2D g2d
- + JButton nowaGra - przycisk Nowa Gra
- + JButton cofnij - przycisk Cofnij
- + Color Black - kolor czarny (lekko rozjaśniony)
- + Color White - kolor biały (lekko przyciemniony)
- + Color Tlo - kolor tła
- + Plansza plansza - obiekt klasy Plansza
- + boolean wyswietlanyStanGry = false - flaga, przechowuje informacje czy ma być wyświetlane prawe menu ze stanem gry (liczba pionków, kolej gracza); nie jest wyświetlane przed rozpoczęciem gry i po zakończeniu
- # void void paintComponent(Graphics g)
- + void actionPerformed((ActionEvent e)

- + void mouseClicked(MouseEvent e)
- + void mouseEntered(MouseEvent e)
- + void mouseExited(MouseEvent e)
- + void mousePressed(MouseEvent e)
- + void mouseReleased(MouseEvent mysz)
- + void sprawdzWarunki()
- + void rysujMenu()
- + void rysujPlansze()

KLASA POLE

Dostęp do wszystkich zmiennych klasy Pole oprócz zmiennej statycznej int szer odbywa się za pomocą getterów i setterów.

+ int szer = 48 - zmienna statyczna definiująca szerokość boku pola w pikselach

- boolean dozwolone – pionki mogą znajdować i poruszać się jedynie po polach ciemnych; zmienna służy do pomalowania planszy

- boolean zaznaczonePole – zmienna przechowująca informację czy pole zostało zaznaczone przez użytkownika za pomocą myszki

- boolean zaznaczonyPionek – gdy na polu znajduje się pionek gracza, który posiada obecnie ruch i ten gracz zaznaczy pole, to jest ustawiana ta flaga na true

- boolean zaznaczonaDamka – podobnie jak w

- boolean proponowane – gdy gracz już zaznaczył swój pionek, to w tej zmiennej przechowywane są informacje, na które pole gracz może przenieść swój pionek

- boolean proponowaneBicie

- char stan – może mieć następujące wartości:

‘#’ – pole niedozwolone,

‘.’ – pole puste,

‘x’ – na polu znajduje się czarny pionek

‘X’ – na polu znajduje się czarna damka

‘x’ – na polu znajduje się biały pionek

‘X’ – na polu znajduje się biała damka

+ Pole() – wszystkie flagi oprócz boolean dozwolone ustawia na false, a stan pola na puste – ‘.’

+ void wyczyszc() – działa tak samo jak konstruktor

KLASA PLANSZA

- + int szer = 8 - zmienna statyczna definiująca szerokość boku plansza w polach
- + Pole pole[][] - dwuwymiarowa tablica 8x8 obiektów klasy Pole
- + int WspX - nieużywane
- + int WspY - nieużywane
- + int obecneX - istnieje specjalnie dla funkcji cofania ruchu, przetrzymuje współrzędną X aktualnie zaznaczonego pola
- + int obecneY - zawiera współrzędną Y aktualnie zaznaczonego pola
- + int poprzednieX
- + int poprzednieY
- + int przeciwnikX - przetrzymuje współrzędną X pionka przeciwnika, który może zostać pobity
- + int przeciwnikY - przetrzymuje współrzędną Y pionka przeciwnika, który może zostać pobity
- + boolean ustawionePionki - przechowuje informację czy pionki są ustawione tak jak na początku gry
- + boolean byloBicie - jeśli ostatnio było bicie, to zawiera true, jeśli nie false, jest używane
- + boolean poprzednioZaznaczonyPionek -
- + boolean mozliweCofniecie
- + boolean cofniecie
- + void ustawPionki - gdy gracz wybierze Nową grę, to ta funkcja zajmuje się czyszczeniem planszy i "ustawieniem pionków" - zmienia stan pól
- + void wyswietl() - wyświetla plansze w trybie tekstowym
- + boolean sprawdzZaznaczeniePionka() - jedna petla for zagnieżdżona w drugiej i wewnątrz if sprawdzający czy na jakimś polu został zaznaczony pionek
- + boolean sprawdzCzyDamka() - sprawdza, czy pionek doszedł do końca planszy, jeśli tak to staje się damką i zwraca true, jeśli nie zwraca false
- + boolean zaznaczPionek(int myszX, int myszY) - pobiera aktualne współrzędne, w których został wciśnięty przycisk myszy; jeśli współrzędne zawierają się w którymś polu, to zostaje zaznaczone pole; jeśli dodatkowo na tym polu znajdują się jakieś pionki to zwraca wartość true, jeśli jest to pole puste to zwraca wartość false
- + void zaproponujRuch() - gdy już jest zaznaczony jakiś pionek, to wtedy sprawdzane są możliwości ruchu, w tym bicia
- + void zaproponujRuchDamka() - jak wyżej, tyle, że dla damki
- + boolean wykonajRuch(int myszX, int myszY) - jeśli gracz miał zaznaczony wcześniej swój pionek i zaznaczy jakieś puste pole, to wtedy wykonuje się ta funkcja

+ void cofnijRuch() – sprawdza, kto miał ostatnio ruch, korzystając ze zmiennych statycznych poprzednieX, poprzednieY przywraca poprzednią pozycję pionka, sprawdza flagę byloBicie, jeśli tak to przywraca pionek przeciwnika z powrotem za pomocą zmiennych statycznych przeciwnikX, przeciwnikY, czyście obecną pozycję pionka za pomocą obecneX, obecneY i na koniec ustawia flagę mozliweCofniecie = false

+ void wyczyscProponowaneRuchy() – dla każdego pola ustawia flagi proponowane i proponowaneBicie na false

+ void odznaczPionek() – dla każdego pola ustawia flagi zaznaczonePole, zaznaczonyPionek i zaznaczonaDamka na false

KLASA GRACZ

Ogólnie można powiedzieć, że jest to “klasa statyczna”, gdyż wszystkie zmienne i funkcje są statyczne.

+ int ruch – przechowuje informacje, którego gracza jest teraz kolej, jeśli gracza 1. (pionki czarne) to przyjmuje wartość 1, jeśli gracza 2. (pionki białe) to 2

+ int IPionkow1 – obecna liczba pionków gracza 1., początkowo równa się 12

+ int IPionkow2 – obecna liczba pionków gracza 2., początkowo równa się 12

+ void zmienGracza() – gdy była kolej gracza 1. to dodaje 1, gdy była kolej gracza 2. to odejmuje 1

+ void ustawStanPoczątkowy() – ustawia liczbę pionków obu graczy na 12; używane przy rozpoczęciu gry od nowa

Karol Kiersnowski, indeks 59200