**Dokumentace – Dáma na neobvyklé šachovnici**

**Stručné zadání**

Hra Dáma na netradiční válcovité šachovnici v prostředí Pygame. Uživatel má možnost si vybrat ze tří možností, buď hru dvou hráčů, uživatel proti počítači, nebo počítač proti počítači. Počítač využívá jednoduchý rozhodovací způsob pro výběr tahů. Herní logika zajišťuje platnost tahů, vybírání figurek a vyhodnocení vítěze.

**Přesné zadání**

Úlohou programu je vytvořit funkční aplikaci dámy, kde:

* Hrací plocha má kruhové propojení ve vodorovném směru (sloupce se opakují).
  + Lze si představit, jako bychom klasickou šachovnici ohnuli a spojili do válce tak, že podél horního kruhu jsou rozestaveny jedny figurky a podél dolního druhé. Tyto dvě kružnice, horní a dolní pak určují kde se protivníkovy figurky mění na dámu.
  + V programu zobrazujeme jako osm soustředných mezikruží.
* Každá figura se pohybuje podle standardních pravidel dámy.
  + Každá barva má 8 figurek rozestavěných po tmavých polích ve dvou řadách. Šachovnice má 8x8 políček.
  + Pinšlíci se pohybují pouze dopředu, a to diagonálně o 1 místo. Figurku lze přeskočit pouze pokud je za ní volno.
  + Dáma se získá tak, že pinšlík dojde na protější kraj šachovnice. Dáma se může pohybovat i zpět po šachovnici.
  + Pokud nějaká figurka může skákat, musí. Každá figurka zahraje právě jeden tah a potom hraje protihráč.
  + Začíná hráč s bílými figurky.
* Hra končí, pokud jeden hráč ztratí všechny figurky, nebo nemá možný tah.
* Ovládání je myší.
  + Nejdříve hráč kliknutím vybere figurku a potom místo kam se má přesunout.
  + Program vždy čte dvojici kliknutí, tzn. Že pokud hráč poprvé klikne špatně, druhé kliknutí musí vypotřebovat a až třetí se bere jako nové vybrání figurky.
  + Program vypíše chybu vždy když hráč udělá nějaký neplatný tah, tah se nepočítá a neprovede a hráč hraje znovu.
* AI hráč vybírá tah na základě jednoduchého hodnocení.

**Zvolený algoritmus**

* **Lidský hráč**: vybírá tah kliknutím – první klik vybere figuru, druhý klik určí cílové pole.
* **AI hráč**: prochází všechny možné tahy, preferuje:
  1. Tahy, které vytvoří dámu a nejsou ohrožené.
  2. Bezpečné tahy.
  3. Libovolné dostupné tahy, pokud žádné bezpečné nejsou.

Bezpečnost tahu se určuje simulací pozice soupeřových figurek a kontrolou, zda by mohly okamžitě sebrat novou pozici.

**Algoritmus hry (pravidla a tahy)**

Hra dáma probíhá na **válcové šachovnici**, což znamená, že levý a pravý okraj desky jsou propojené (hráč může projít „dokola“).  
Algoritmus hry se stará o správný průběh:

1. **Inicializace hry**
   * Načte se hrací deska, rozloží se figurky podle pravidel. Začíná bílá.
2. **Tah hráče**
   * Program čeká na vstup: buď kliknutí myší (člověk), nebo volbu algoritmu (AI).
   * Vybere se figurka a cílové pole.
   * Program ověří, zda je tah podle pravidel platný:
     + pohyb jen diagonálně, nelze hrát soupeřovou figurkou,
     + může-li brát, musí, pohybuje se jen po tmavých políčkách,

**Provedení tahu**

* + Figurka se přesune, případně se odstraní soupeřova figurka.
  + Pokud figurka dojde na poslední řadu, stane se dámou.

1. **Kontrola stavu hry**
   * Zkontroluje se, zda některý hráč neztratil všechny figurky.
   * Pokud ano, nastaví se vítěz a hra končí.
   * Jinak se střídají hráči a hra pokračuje.

**Diskuse výběru algoritmu**

Původně jsme zvažovali implementaci **minimax algoritmu s prohledáváním do hloubky** pro silnější AI, ale pro účely zápočtu jsme zvolili jednodušší heuristiku založenou na prioritách tahů, aby byl kód přehledný a hra svižná. Minimax by sice poskytl lepší strategii, ale vyžadoval by více optimalizace kvůli velkému množství herních stavů.

**Program**

Struktura programu je rozdělena do několika tříd:

* **Game** – hlavní herní logika, střídání hráčů, kontrola vítěze.
* **Board** – reprezentace šachovnice, uložení figurek, kontrola platnosti tahů.
* **Piece** – jednotlivá figura, zná svůj typ (pěšec/dáma) a možné směry pohybu.
* **Player** – abstraktní hráč.
* **HumanPlayer** – implementace hráče ovládaného myší.
* **AIPlayer** – jednoduchá umělá inteligence.
* **Hlavní smyčka** – vykreslení herní plochy, zpracování událostí, volání tahů hráčů.

Projekt je kvůli přehlednosti rozdělen do více souborů, které na sebe odkazují:

* **piece.py**  
  Obsahuje třídu Piece, která reprezentuje jednu figurku.
  + Udržuje barvu, pozici a informaci, zda je figurka dáma.
  + Definuje metody pro možné tahy (krok(), skok()).
* **board.py**  
  Definuje třídu Board, která reprezentuje herní plochu.
  + Obsahuje matici polí a aktuální rozložení figurek.
  + Umí kontrolovat platnost tahu (check\_valid\_move).
  + Metoda je\_v\_poli() zajišťuje kontrolu, zda souřadnice spadají do hrací plochy (včetně „obalení“ při válcové šachovnici).
* **game.py**  
  Obsahuje logiku celé hry.
  + Uchovává informace o hráčích, stavu hry a tahu na řadě.
  + Implementuje play\_turn(), která provádí tah a vyhodnocuje jeho důsledky.
  + Metoda winner() určuje, zda už někdo vyhrál.
* **player.py**  
  Definuje obecnou třídu Player a její potomky:
  + HumanPlayer – hráč ovládaný člověkem (klikání myší).
  + AIPlayer – počítačový hráč, který používá algoritmus minimax.
* **visual.py**  
  Obsahuje funkce pro grafické vykreslování přes knihovnu pygame.
  + Funkce run\_gui\_game(p1, p2) spustí hru mezi dvěma hráči (člověk vs. člověk, člověk vs. AI, AI vs. AI).
  + Obsahuje kreslení šachovnice, figurek a obsluhu událostí myši.
* **main.py**  
  Hlavní soubor programu, který propojuje vše dohromady.
  + Načte hráče (typicky HumanPlayer a AIPlayer).
  + Inicializuje Game a volá run\_gui\_game()

**Alternativní programová řešení**

* **Plná AI s minimaxem** – zavrženo kvůli složitosti a době výpočtu.
* **Oddělené GUI a logika** – pro přehlednost by bylo vhodné, ale pro zápočtovou verzi jsme použili jednodušší strukturu s herní logikou přímo propojenou s vykreslováním.

**Reprezentace vstupních dat a jejich příprava**

Vstupy od hráče jsou kliknutí myší:

1. **První klik** – výběr figury (musí patřit hráči na tahu).**Druhý klik** – určení cílového pole. Pokud je tah neplatný, figuru je třeba vybrat znovu.  
   AI vstupy jsou generovány interně.
2. **Druhý klik** – určení cílového pole. Pokud je tah neplatný, figuru je třeba vybrat znovu.  
   AI vstupy jsou generovány interně.

**Reprezentace výstupních dat a jejich interpretace**

* **Herní plocha zobrazuje aktuální stav figurek.**
* **Zprávy o chybách (např. neplatný tah) se vypisují do textového pole ve spodní části.**
* **Po vítězství jednoho hráče se zobrazí obrazovka „Game Over“ s barvou vítěze.**

**Průběh prací**

* **Nejprve byla vytvořena základní verze šachovnice a figura s pohybem.**
* **Poté byly doplněny kontroly tahů a logika střídání hráčů.**
* **Následně přidána AI podle priorit bezpečných tahů.**
* **Poslední fází bylo vykreslení kruhového propojení šachovnice.**

**Co nebylo doděláno**

* **Minimax AI pro hlubší strategii.**
* **Možnost více herních módů (např. AI vs AI).**
* **Ukládání a načítání rozehrané hry.**

**Závěrečný povzdech**

**Bylo zajímavé řešit netradiční kruhovou šachovnici, ale ladění AI zabralo více času, než jsme čekali. Kdybychom začali s minimaxem od začátku, možná by hra byla chytřejší, ale asi bychom nestihli finální ladění.**