**ŽILINSKÁ UNIVERZITA V ŽILINE**

**FAKULTA RIADENIA A INFORMATIKY**

**INFORMATIKA 1**

Semestrálna práca

Pavol Brišák 5ZYI15

**Školský rok 2021/2022**

**ZADANIE SEMESTRÁLNEJ PRÁCE**

Ako tému semestrálnej práce som si zvolil hru hadíka. Musel som vyriešiť výber počtu hráčov, typu mapy, typu hada a samotnú hru hadíka. Práca obsahuje viacero verzií ako hru hrať a po skončení hry sa získané skóre uloží do súboru.

**Typy máp:**

* **Klasická**: Had sa postupne predlžuje jedením jabĺčka, ktoré je na mape iba jedno. Narazením do seba, do nejakej steny mapy alebo do druhého hada, hra končí.
* **Ľahká**: Had sa predlžuje jedením jabĺk, ktorých je na mape viacero. Je dovolené narazenie do steny, pričom vtedy had cez stenu prejde a zobrazí sa na opačnej strane mapy. Hra končí, ak had narazí do seba alebo do druhého hada. Mapa je podstatne väčšia.
* **Ťažká**: Had sa postupne predlžuje jedením jabĺčka, ktoré je na mape iba jedno. Na mape sú vytvorené prekážky na stranách popri stenách. Hra končí, ak had narazí do seba, do steny, do druhého hada alebo do prekážky. Mapa je o niečo väčšia než klasická mapa.

**Triedy**

**HraHadíka**: Trieda spúšťa samotnú hru. Obsahuje metódu main, ktorá zapína triedu VyberHracovAMapy.

**VyberHracovAMapy**: Trieda využíva metódy tried JFrame a MouseListener, pomocou ktorých hráč môže klikať na vytvorené okno, na ktorom sú nastavené tlačidlá. Hráč si môže vybrať, či chce hrať sám alebo s druhým hráčom. Taktiež má na výber, ktorú mapu si želá hrať a s ktorým typom hada chce hrať. Keď hráč klikne na niektoré tlačidlo, tlačidlo sa rozsvieti a daný údaj sa zapamätá do atribútu. Tlačidlá sa rozsvietia keď hráč vojde myškou do tlačidlo a zhasnú keď z neho odíde. Táto funkcia funguje iba, ak na dané tlačidlo hráč ešte neklikol. Oba hráči majú na výber zo 4 typov hada: Hráč 1: zelený, modrý, ružový a farebný. Hráč 2: fialový, oranžový, čierny a farebný. Farebný a čierny had sa kreslí pomocou obrázkov zo súboru. Čierny had má vytvorenú špeciálnu mapu, kde ho je lepšie vidno. Pri tomto výbere využívam zoznam stringov, pomocou ktorých čítam zvolený obrázok hada zo súboru, ktorý sa následne zobrazí na tlačidle. Oba tlačidla sú nastavené tak, aby ich bolo vidno iba ak hráč klikne na príslušné tlačidlo počtu hráčov. Hráč 1 hrá pomocou šipiek a hráč 2 pomocou kláves WASD. Na tlačidlo Pokračovať môže hráč kliknúť iba ak si zvolil počet hráčov a typ mapy. Potom sa hráčovi zobrazí okno Input, kde napíše meno, pod ktorým si želá hrať (toto meno slúži iba pre zápis do súboru), následne sa spustí trieda HraciaPlocha.

**HraciaPlocha**: Táto trieda využíva metódy triedy JFrame, ktorá vytvára okno, na ktoré sa vloží hra spustením triedy Hadik1Hrac alebo Hadik2Hraci. V konštruktore sa inicializuje parameter okna a zavolá sa metóda spustiHru, v ktorej sa podľa počtu hráčov spustí príslušná trieda hry hadíky. Ďalej sa zavolá metóda nastavOkno, kde sa nastavuje názov, rozmery a lokácia okna.

**Hadik1Hrac**: Trieda využíva metódy tried JPanel, KeyListener a ActionListener, pomocou ktorých sa hra vloží na už vytvorené okno a hráč môže pomocou klávesnice ovládať hada. Nechal som sa inšpirovať kódom zo stránky <https://www.youtube.com/watch?v=bI6e6qjJ8JQ&ab_channel=BroCode> Hra funguje na princípe opakovaného kreslenia plochy každých niekoľko sekúnd. Spustí sa časovač, ktorého vytvorenie mám požičané zo stránky <https://stackoverflow.com/questions/4044726/how-to-set-a-timer-in-java> a pomocou metódy ActionPerformed sa vykonajú dané metódy. V konštruktore sa inicializujú atribúty, podľa typu mapy sa nastavia rozmery, uloží sa meno hráča, typ hada, nastaví sa mu náhodný smer, následne sa nastaví plocha a zavolá sa metóda spustiHru. V tejto metóde sa podľa typu mapy vytvoria jablká, prekážky, had a zapne sa časovač. Had sa vytvára pomocou náhodne vytvorených pozícií jeho hlavy, tie môžu byť iba v strednej časti mapy. Zvyšné telo sa pridáva k hlave podľa smeru hada. Pozície sa vytvárajú nasledovne: zistí sa koľko je možných miest pre vytvorenie pozície hada podľa šírky a výšky mapy, ktorá je vydelená veľkosťou tela hada. Z týchto možných miest sa jedno náhodne vyberie a vynásobí sa veľkosťou tela hada. Daná pozícia sa skontroluje, či sa vytvorila v správnej časti mapy, pokiaľ nie, metóda sa zavolá znova. Pozície prekážok a jabĺk sa vytvárajú rovnako, až na to, že prekážky sú práve v tej časti mimo oblasti, v ktorej sa vytvára had. Všetky tieto vytvorené pozície útvarov sa kontrolujú, či sa vytvorili na správnych pozíciách, ak nie, metódy sa spúšťajú znova. Na všetkých týchto vytvorených pozíciách sa pomocou grafiky kreslí obdĺžnik alebo sa kreslí obrázok zo súboru. V metódach, v ktorých sa kreslia dané útvary využívam správy grafiky setColor, drawRect a fillRect, ktoré mi nakreslia útvar nastavenou farbou a vyplnia ho. Pokiaľ sa kreslí útvar pomocou obrázkov zo súboru, najprv sa vytvorí nový ImageIcon s cestou daného obrázka, ďalej sa vytvorí lokálna premenná Image pomocou get Image a getScaledInstance, ktorá obrázku nastaví veľkosť na veľkosť tela hada. Potom sa vytvorí nový ImageIcon z tohto obrazu a metódou paintIcon sa kreslí na daných pozíciách. Dôležitá je metóda paintComponent z triedy JPanel, ktorá dokáže kresliť na danú plochu. V nej som musel vyriešiť problém, aby sa mi celá plocha prekresľovala. To zabezpečuje správa super.paintComponent, ktorú mám požičanú zo stránky <https://docs.oracle.com/javase/tutorial/uiswing/painting/problems.html> Had sa posúva pomocou kontroly jeho smeru. K príslušnému smeru sa zmenia pozície hlavy hada a zvyšné pozície tela sa len zmenia na pozície tela pred nimi. Najprv sa had posunie a potom sa kontroluje, či nenarazil a či nezjedol jablko. Po narazení sa had prestane pohybovať, časovač sa zastaví. Pokiaľ had zjedol jablko, jeho telo a skóre sa zvýšia o jeden. V celej hre je 10% šanca pre vytvorenie zlatého jablka, ktoré hráčovi po zjedení pridá 5 bodov. Ak sa had prestal pohybovať, plocha sa ďalej nekreslí ale zobrazí sa na plochu text, ktorý hráča informuje o výhre, prehre, jeho skóre a mene. Údaje sa vtedy automaticky zapíšu do súboru a ak si hráč želá hrať tak isto nastavenú mapu, musí stlačiť medzerník. V tom momente sa hadovi nastavia atribúty na pôvodné hodnoty a zavolá sa metóda spustiHru. Smer hada sa mení pomocou stlačení klávesy, ktorá sa číta v metóde KeyPressed. V nej bolo dôležité kontrolovať, či nový smer, ktorý chceme nastaviť je vôbec možný (had napr. nemôže zmeniť smer na pravý ak ide doľava). Zápis do súboru je nastavený tak, že sa údaje zapíšu do súboru bez jeho prepísania. Zapíše sa dátum hrania danej hry, meno hráča a jeho skóre. Dátum sa zapisuje pomocou správy java.time.LocalDate.now(), ktorú mám z tejto webovej stránky <https://www.javatpoint.com/java-get-current-date> Hra končí, ak telo hada dosiahlo najväčšiu možnú dĺžku.

**Hadik2Hraci**: V tejto triede bolo nutné dodať vytvorenie druhého hada. Pozície hlavy druhého hada sú vytvorené v tej istej oblasti v strede mapy, ale nesmie byť k prvému hadovi bližšie než je dvojnásobok veľkosti tela hada. Musel som doplniť vo všetkých kontrolách pozícií útvarov kontrolu, či sa daný útvar nenachádza na nejakej pozícii druhého hada. Taktiež som musel pridať klávesy WASD do metódy KeyPressed, aby sa mohol druhý had pohybovať. Do súboru sa zapisujú mená a skóre oboch hráčov a taktiež sa zistí a zapíše, ktorý hráč mal vyššie skóre.

**Smer**: Ide o triedu typu enum, ktorá predstavuje smery hada (vpravo, vľavo, hore, dole).