

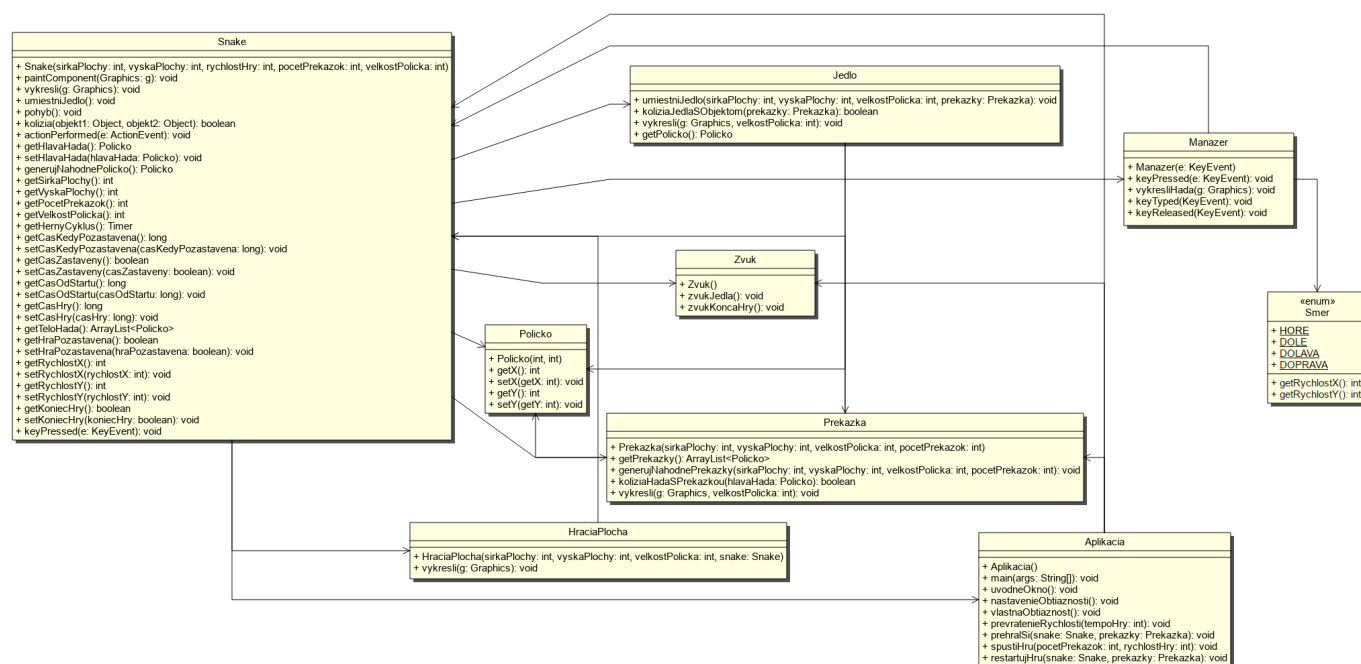
Hra Snake - Filip Klein

Vo videohre "Snake" preberá hráč kontrolu nad hadom, ktorý sa pohybuje po obdĺžnikovej hernej ploche, zbierajúc jedlo a získavajúc body. Ovládanie je implementované pomocou klávesnice, kde hráč môže používať šípky alebo klávesy WASD na nasmerovanie hada. S každým zjedeným jedlom sa had predlžuje, čo pridáva na obtiažnosti hry. Hráč sa musí vyhnúť kolíziám s okrajmi plochy, prekážkami a vlastným telom.

V hre môžu byť pridané aj prekážky, ktoré zvyšujú náročnosť a strategický prístup hráča. Obtiažnosť hry je nastavovateľná, ovplyvňujúc rýchlosť hada a počet prekážok. Po skončení hry hráč dostane možnosť zmeniť obtiažnosť, znovu spustiť hru alebo ju ukončiť. Celkový vizuálny dojem hry je zdokonaľovaný farebnou grafikou a obsahuje zvukové efekty, čo prispieva k celkovej zábavnosti herného zážitku. Ponúka jednoduchú, no návykovú hru s možnosťou prispôsobenia obtiažnosti a vizuálnych prvkov.

Hra pozostáva z deviatich tried: **Snake**, **Aplikacia**, **Manazer**, **HraciaPlocha**, **Jedlo**, **Policko**, **Prekazka**, **Smer**, **Zvuk**. Všetky tieto triedy sú stručne popísané v dokumentácii.

Vonkajší pohľad UML diagramu obsahuje triedy a ich vzájomné prepojenia. Tento diagram ilustruje, ako tieto triedy spolupracujú a ako sú medzi sebou prepojené v rámci hry Snake. Diagram poskytuje stručný prehľad o štruktúre hry, čo uľahčuje rýchle pochopenie hlavných komponentov a ich vzájomných vzťahov.



Snake.java

Trieda **Snake** implementuje hru Snake v jazyku Java s použitím grafickej knižnice Swing. Zabezpečuje herný cyklus, pohyb hada, kolízie, generovanie jedla a prekážok, a ovládanie klávesnicou.

Premenné triedy **Snake**

Nastavenie hry

- sirkaPlochy**: Šírka hracej plochy.

- **vyskaPlochy**: Výška hracej plochy.
- **velkostPolicka**: Veľkosť políčka hracej plochy.
- **pocetPrekazok**: Počet prekážok na hracej ploche.
- **rychlostX**: Rýchlosť pohybu hada v smere osi X.
- **rychlostY**: Rýchlosť pohybu hada v smere osi Y.

Stav hry

- **koniecHry**: Indikuje, či sa hra skončila.
- **hraPozastavena**: Indikuje, či je hra pozastavená.
- **casOdStartu**: Čas od štartu hry.
- **casKedyPozastavena**: Čas, kedy bola hra pozastavená.
- **casZastaveny**: Indikuje, či je čas pozastavený.
- **casHry**: Čas trvania hry.

Herné objekty a nástroje

- **zvuk**: Inštancia triedy **Zvuk** pre spustenie zvukov v hre.
- **hlavaHada**: Inštancia triedy **Policko** reprezentujúca hlavu hada.
- **hraciaPlocha**: Inštancia triedy **HraciaPlocha**, reprezentujúca hraciu plochu hry.
- **teloHada**: Zoznam políčok tvoriacich telo hada.
- **jedlo**: Inštancia triedy **Jedlo**, reprezentujúca jedlo v hre.
- **nahodne**: Využíva sa na generovanie náhodných hodnôt, napríklad pri umiestňovaní prekážok alebo jedla.
- **hernyCyklus**: Časovač pre herný cyklus.
- **prekazky**: Inštancia triedy **Prekazka** sa používa na manipuláciu s prekážkami na hracej ploche.
- **manazer**: Inštancia triedy **Manazer** pre ovládanie klávesnice.

Metódy triedy Snake

```
public Snake(int sirkaPlochy, int vyskaPlochy, int rychlostHry, int pocetPrekazok, int velkostPolicka)
```

Konštruktor triedy Snake. Inicializuje premenné a objekty potrebné pre spustenie hry.

```
public void paintComponent(Graphics g)
```

Metóda pre vykreslenie komponentu. Vykresľuje aktuálny stav hry.

```
public void vykresli(Graphics g)
```

Metóda volá vykreslenie v triede HraciaPlocha.

```
public void umiestniJedlo()
```

Umiestňuje nové jedlo na hraciu plochu.

```
public void pohyb()
```

Ošetruje pohyb hada, kolízie s jedlom, prekážkami a hranami hracej plochy.

```
public static boolean kolizia(Object objekt1, Object objekt2)
```

Metóda na detekciu kolízií medzi objektmi na hracej ploche.

```
public void actionPerformed(ActionEvent e)
```

Implementuje akcie, ktoré sa majú vykonať v každom hernom cykle.

```
public void keyPressed(KeyEvent e)
```

Volá metódu z triedy Manazer, ktorá riadi pohyb hada.

Aplikacia.java

Trieda **Aplikacia** riadi priebeh hry a zabezpečuje užívateľské rozhranie.

Premenné triedy **Aplikacia**

- **zvuk**: Inštancia triedy **Zvuk** slúži na spracovanie zvukových efektov.

Metódy triedy **Aplikacia**

```
public Aplikacia()
```

Konštruktor triedy **Aplikacia**, inicializuje inštanciu triedy **Zvuk**.

```
public static void main(String[] args)
```

Hlavná metóda pre spustenie aplikácie. Vytvorí inštanciu **Aplikacia** a zavolá jej metódu **uvodneOkno()**.

```
public void uvodneOkno()
```

Zobrazuje úvodné okno s inštrukciami pre hru a ponúka možnosti pre pokračovanie alebo ukončenie hry.

```
public void nastavenieObtiaznosti()
```

Zobrazuje dialógové okno na nastavenie obtiažnosti hry podľa výberu užívateľa.

```
public void vlastnaObtiaznost()
```

Zobrazuje dialógové okno pre nastavenie vlastnej obtiažnosti hry.

```
public int prevratenieRychlosti(int tempoHry)
```

Prevráti rýchlosť hry podľa zadaného tempa.

```
public void prehralSi(Snake snake, Prekazka prekazky)
```

Zobrazuje dialógové okno pri prehre a ponúka možnosti pre opakovanie, ukončenie alebo zmenu obtiažnosti hry.

```
public void spustiHru(int pocetPrekazok, int rychlostHry)
```

Spúšťa hru s definovaným počtom prekážok a rýchlosťou.

```
public void restartujHru(Snake snake, Prekazka prekazky)
```

Reštartuje hru po prehre.

Manazer.java

Trieda **Manazer** spracováva užívateľské vstupy z klávesnice a ovláda pohyb hada.

Premenné triedy **Manazer**

- **snake**: Inštancia triedy **Snake**, ktorú ovláda tento manažér.
- **hadikHore**: Obrázok hada pohybujúceho sa hore.
- **hadikDole**: Obrázok hada pohybujúceho sa dole.
- **hadikVpravo**: Obrázok hada pohybujúceho sa vpravo.
- **hadikVlavo**: Obrázok hada pohybujúceho sa vľavo.

Metódy triedy **Manazer**

```
public Manazer(Snake snake)
```

Konštruktor triedy **Manazer**, inicializuje inštanciu triedy **Snake**, ktorú bude ovládať.

```
public void keyPressed(KeyEvent e)
```

Riadi pohyb hada podľa stláčania kláves. Ak je hra pozastavená, je umožnená iba klávesa ESC na vrátenie sa do hry. Ostatné ovládanie hada pomocou kláves WASD alebo šípok na klávesnici je vypnuté.

```
public void vykresliHada(Graphics g)
```

Vykresľuje obrázky hada podľa smeru jeho pohybu.

HraciaPlocha.java

Trieda **HraciaPlocha** má za úlohu vizualizáciu herného prostredia a poskytuje informácie o aktuálnom stave hry.

Premenné triedy **HraciaPlocha**

- **sirkaPlochy**: Šírka hernej plochy.
- **vyskaPlochy**: Výška hernej plochy.
- **velkostPolicka**: Veľkosť políčka na hernej ploche.

- **snake**: Inštancia triedy **Snake** reprezentujúca hada.

Metódy triedy **HraciaPlocha**

```
public HraciaPlocha(int sirkaPlochy, int vyskaPlochy, int velkostPolicka, Snake snake)
```

Konštruktor triedy **HraciaPlocha**. Inicializuje premenné potrebné pre hernú plochu.

```
public void vykresli(Graphics g)
```

Vykresľuje hernú plochu, mriežku a informácie o stave hry.

- Vykresľuje mriežku herného poľa.
- Nastavuje šachovnicový štýl herného poľa.
- Zobrazuje informácie o dĺžke hada a čase trvania hry.
- Spravuje pozastavenie a koniec hry, vrátane aktualizácie času v hre - ak hráč hru pozastaví, čas bude pozastavený, dokým sa znova do hry nevráti.

Jedlo.java

Trieda **Jedlo** predstavuje objekt jedla v hre. Zabezpečuje jeho umiestňovanie, kolízie s prekážkami a vizualizáciu na hernú plochu.

Premenné triedy

- **jedlo**: Inštancia triedy **Policko** reprezentujúca polohu jedla na hernom poli.
- **jablkoImage**: Obrázok jedla (jablko).

Metódy triedy **Jedlo**

```
public void umiestniJedlo(int sirkaPlochy, int vyskaPlochy, int velkostPolicka, Prekazka prekazky)
```

Náhodne vygeneruje pozíciu a umiestňuje jedlo na hernú plochu.

```
public boolean koliziaJedlaSObjektom(Prekazka prekazky)
```

Kontroluje, či nedochádza ku kolízii medzi umiestnením jedla a prekážkami na hernom poli a v prípade kolízie vygeneruje nové umiestnenie jedla.

```
public void vykresli(Graphics g, int velkostPolicka)
```

Vykresľuje jedlo na hernú plochu, ktoré sa zobrazuje ako obrázok jablka.

Policko.java

Trieda **Policko** predstavuje jednoduchý objekt reprezentujúci polohu na hernom poli. Obsahuje informácie o x-ovej a y-ovej súradnici.

Premenné triedy **Policko**

- **x**: Celé číslo predstavujúce x-ovú súradnicu polohy políčka.
- **y**: Celé číslo predstavujúce y-ovú súradnicu polohy políčka.

Metódy triedy **Policko**

```
public Policko(int x, int y)
```

Konštruktor triedy **Policko**. Inicializuje objekt polohy s danými x-ovou a y-ovou súradnicou.

Prekazka.java

Trieda **Prekazka** reprezentuje prekážky na hracej ploche. Prekážky sú generované náhodne a môžu ovplyvňovať pohyb hada.

Premenné triedy **Prekazka**

- **prekazky**: Zoznam políčok predstavujúcich prekážky.
- **hlavaHada**: Inštancia triedy **Policko** reprezentujúca hlavu hada.
- **snake**: Inštancia triedy **Snake** reprezentujúca hada.
- **sirkaPlochy**: Šírka hracej plochy.
- **vyskaPlochy**: Výška hracej plochy.
- **velkostPolicka**: Veľkosť políčka hracej plochy.
- **pocetPrekazok**: Počet prekážok na hracej ploche.

Metódy triedy **Prekazka**

```
public Prekazka()
```

Konštruktor triedy **Prekazka**. Inicializuje potrebné premenné, zoznam prekážok a volá metódu na vykreslenie náhodných vygenerovaných prekážok na hraciu plochu.

```
public void generujNahodnePrekazky(int sirkaPlochy, int vyskaPlochy, int  
velkostPolicka, int pocetPrekazok)
```

Generuje náhodné prekážky na hracej ploche.

```
public boolean koliziaHadaSPrekazkou(Policko hlavaHada)
```

Kontroluje, či nedochádza ku kolízii medzi umiestnením hada a prekážkami na hernom poli a v prípade kolízie vygeneruje nové umiestnenie hada.

```
public void vykresli(Graphics g, int velkostPolicka)
```

Vykreslí prekážky na hracej ploche.

Smer.java

Enumerácia **Smer** definuje rôzne smery pohybu hada na hracej ploche. Každý smer má priradené hodnoty rýchlosti na osiach X a Y.

Enumy triedy **Smer**

- **HORE**: Smer s rýchlosťou -1 na osi Y (hore).
- **DOLE**: Smer s rýchlosťou 1 na osi Y (dole).
- **DOLAVA**: Smer s rýchlosťou -1 na osi X (vľavo).
- **DOPRAVA**: Smer s rýchlosťou 1 na osi X (vpravo).

Premenné triedy **Smer**

- **rychlostX**: Udávaná rýchlosť hada na osi X.
- **rychlostY**: Udávaná rýchlosť hada na osi Y.

Metódy triedy **Smer**

Smer(int rychlostX, int rychlostY)

Inicializuje nový smer so zadanými hodnotami rýchlosti na osiach X a Y.

public int getRychlostX()

Vráti rýchlosť na osi X pre daný smer.

public int getRychlostY()

Vráti rýchlosť na osi Y pre daný smer.

Zvuk.java

Trieda **Zvuk** zodpovedá za spúšťanie zvukov v hre.

Premenné triedy **Zvuk**

- **zvukJedla**: Inštancia **Clip** pre zvuk pri jedení jedla.
- **zvukKoncaHry**: Inštancia **Clip** pre zvuk pri prehre.

Metódy triedy **Zvuk**

public Zvuk()

Konštruktor triedy **Zvuk**. Inicializuje premenné pre zvuky jedla a konca hry.

public void zvukJedla()

Prehrá sa zvukový efekt zakaždým, ak had zje jedlo.

```
public void zvukKoncaHry()
```

Ak hráč prehrá, tak sa spustí tento zvukový efekt.