



ŽILINSKÁ UNIVERZITA V ŽILINE

Fakulta riadenia  
a informatiky

Semestrálna práca z predmetu informatika 1

**CENTIPEDE**

vypracoval:

Adrián Ličko

študijná skupina:

5ZYS12

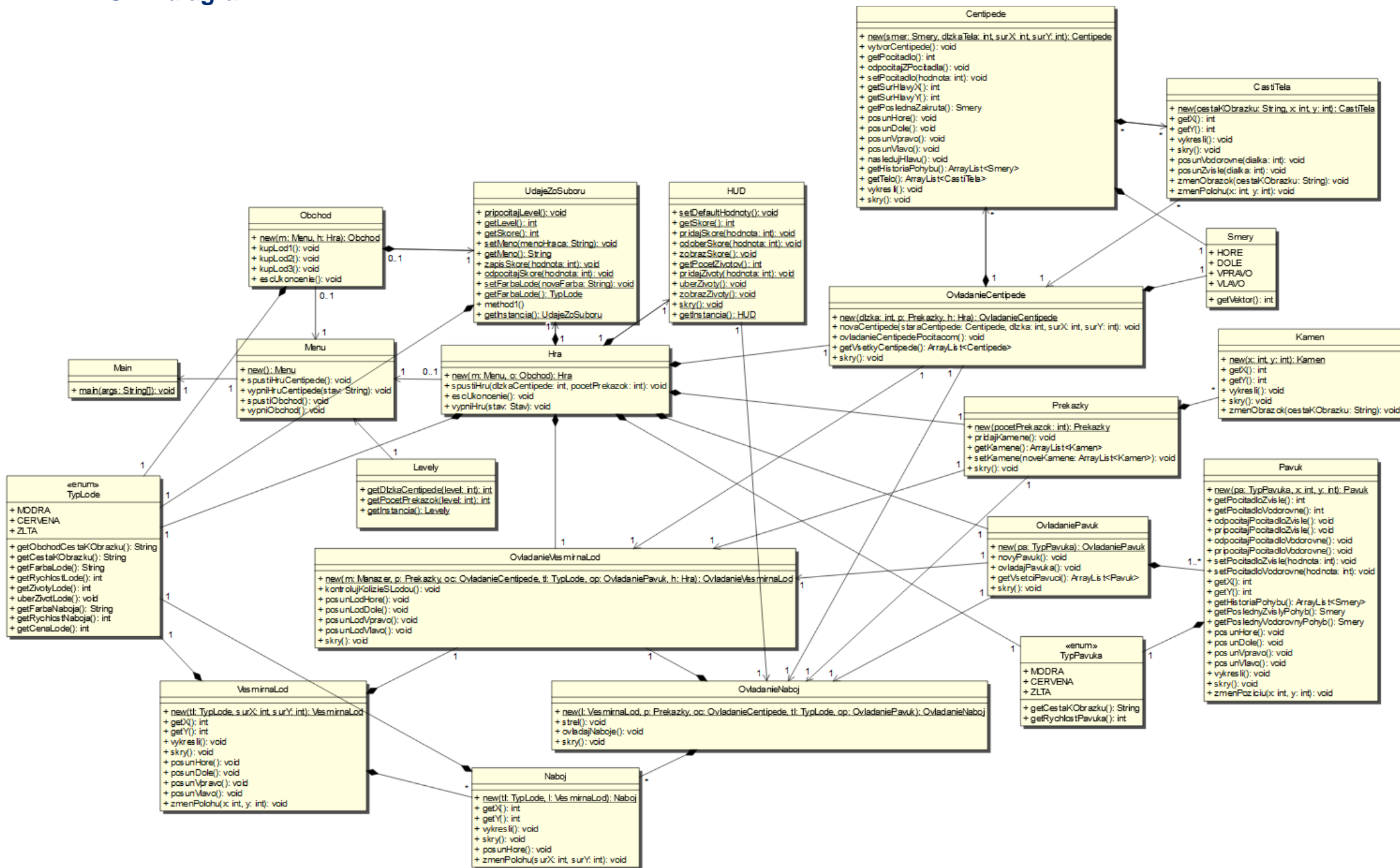


## Obsah

UML diagram .....	3
Používateľská príručka.....	4
Programátorská príručka.....	5
Cudzí kód .....	7
Použité obrázky .....	7
Zdroje .....	8

Obrázok 1 Žltá vesmíra loď.....	7
Obrázok 2 Červená vesmírna loď .....	7
Obrázok 3 Modrá vesmírna loď.....	7
Obrázok 4 Možnosť Obchod.....	7
Obrázok 5 Možnosť Hra.....	7
Obrázok 6 Nápis Centipede .....	7
Obrázok 7 Nápis Obchod.....	7
Obrázok 8 Srdiečko.....	7
Obrázok 9 Skóre .....	7
Obrázok 10 Kameň .....	7
Obrázok 11 Telo Centipede .....	7
Obrázok 12 Hlava Centipede .....	7

## UML diagram



## Používateľská príručka

Hra sa spúšťa cez (batch) súbor **PlayCentipede.bat**.

Po spustení sa od hráča vyžaduje zadanie mena. Toto meno je dobré si zapamätať, pretože sa naň bude ukladať počet skóre ktoré hráč nazbieral, aktuálna zakúpená loď a posledný level ktorý prešiel.

Po zadaní mena sa hráčovi zobrazí okno s menu, kde si môže zvoliť možnosť či chce ísť do obchodu alebo či chce ísť hrať. Tieto možnosti si hráč môže vybrať pomocou stlačenia daného čísla na klávesnici, ktoré je pri danej možnosti napísané. Ďalej sa nad nimi zobrazí aj level, do ktorého hráč vstúpi pri zvolení hrania hry.

V obchode si hráč môže kúpiť novú vesmírnu loď. Lode si kúpi stlačením klávesy, podobne ako v menu. Na loď ktorú si chce kúpiť musí mať ale dostatok skóre, inak si ju kúpiť môcť nebude. Pri každej z lodí sú aj jej vlastnosti, napr. aká je rýchla, koľko ma životov a podobne. Podľa zvolenej lodi sa zmení aj pavúk, ktorý môže byť rýchlejší alebo pomalší. Ďalej, na spodku obrazovky je možno vidieť aktuálne dosiahnuté skóre hráča a aktuálne kúpenú loď. Pri kúpe novej lodi, hráč o predošlú príde. Pre odchod z obchodu hráč stlačí tlačidlo escape.

Po zvolení možnosti pre hru, sa hráčovi zobrazí hra. Hráčovi sa zvolí vesmírna loď podľa toho ktorú vlastní. Taktiež sa mu zobrazí aj level, kde má stonožka rôznu dĺžku a počet prekážok na hracej ploche je tiež iný.

V hre môže hráč vyhrať alebo prehrať. Vyhrá tým, že stonožku do poslednej časti zničí. Iba vtedy sa mu započíta dosiahnuté skóre. Prehrať môže tak, že stratí životy. O životy príde, keď narazí do steny, stretne sa s pavúkom alebo sa stretne so stonožkou. Svoje životy danej lode a dosiahnuté skóre v danom leveli môže vidieť na HUD na spodku obrazovky.

Hráč a teda vesmírna loď sa pohybuje pomocou šípok. Strieľať môže cez medzerník.

Po vyhratí daného levelu, sa hráčovi na dané meno aktualizuje nazbierané skóre a level. Levelov je 10.



## Programátorská príručka

Pojekt je postavený na knižnici ShapesGE.

**Centipede**, stonožka sa skladá z častí. Na to je vytvorená trieda menom **CastiTela**. Táto trieda obsahuje metódy na získanie súradníc pozície, jej posuny a tak ďalej.

Objekty týchto častí sú uložené v ArrayListe v atribúte u triedy **Centipede**. Stonožka sa vytvorí automaticky hneď, ako sa vytvorí na ňu nový objekt. Pre jej vytvorenie treba zadať jej súradnice kde sa zobrazí, akú bude mať dĺžku a akým smerom bude smerovať. Napríklad ak smeruje doprava, tak jej telo sa zobrazí naľavo od hlavy. Trieda obsahuje ArrayList takýchto smerov kde sa ukladá každý pohyb stonožky, respektíve, hýbe sa iba hlava a telo ju nasleduje. Tento ArrayList maže svoje prvky, ktoré už pre pohyb tela nepotrebuje. Ďalej obsahuje počítadlo, ktoré nastavuje trieda **OvladanieCentipede**. Toto počítadlo slúži na nastavenie hodnoty toho, koľko krát sa má niečo udiať. Viac je vysvetlené pod triedou **OvladanieCentipede**.

Viacero stonožiek vytvára trieda **OvladanieCentipede**. Pre vytvorenie inštancie na túto triedu, sa zadávajú hodnoty pre prvú stonožku ktorá sa vytvorí. Každá stonožka je jeden prvok v ArrayListe centipede. Metódu **novaCentipede** volá metóda v triede **OvladanieNaboj**, zavolá sa vtedy, kedy náboj trafí stonožku, jej časti od zasiahnutého miesta sa vymažú a z vymazaných nezasiahnutých častí sa vytvorí nová stonožka. Tu sa nastavuje aj počítadlo, ktoré znamená, koľko krát má stonožka v nasledujúcich krokoch vykonať posledný pohyb ktorý spravila. Ďalej je tu metóda na ovládanie každej zo stonožiek. Je zodpovedná za jej pohyb a správanie. Ovláda ju manžér.

Trieda **Prekazky** obsahuje metódy na tvorbu **Kamen** objektov.

**Pavuk** predstavuje prekážku a stratu života pre hráča. Pri jeho tvorbe treba zadať jeho typ a súradnice. Typ pavúka závisí od typu rakety ktorú hráč vlastní. Iný typ pavúka znamená jeho iná farba a jeho iná rýchlosť. Jeho metódy sú podobné ako pri stonožke.

Týchto pavúkov ovláda **OvladaniePavuk**. Na začiatku keď objekt na túto triedu vytvorí, tak sa náhodne zvolí počet pavúkov. Ďalej pri zavolaní metódy **novyPavuk** sa náhodne zvolí smer do ktorého sa pavúk bude uberať. Metódu **ovladajPavuka** spravuje manažér a zabezpečuje jeho pohyb, ktorý je náhodný.

**VesmírnaLod** tvorí objekt s ktorým hráč pohybuje nepriamo. To ako loď bude vyzeráť a ako rýchlo sa s ňou bude dať pohybovať, závisí od nastavených hodnôt v enume **TypLode**.

Vesmírnu loď pohybuje trieda **OvladanieVesmírnaLod**, ktorá obsahuje metódy na posun, ktoré spravuje manžér, pri stlačení klávesy šípok. Ďalej táto trieda obsahuje metódy na kontrolu kolízie lodi s prekážkami, stonožkou alebo pavúkmi. Tieto metódy s návratovou hodnotou boolean vyvoláva metóda, zodpovedná za kontrolu všetkých už spomenutých kolízií. Túto metódu spravuje manažér a v prípade aspoň jednej z týchto kolízií, vesmírnu loď vráti na štartovnú pozíciu a hráčovi uberie život. Toto sa stane aj keď hráč výjde z povolenej hracej plochy.

**OvladanieNaboj** vytvára, vykresluje a hýbe s nábojmi, strelami. Objekty **Naboj** vytvára cez metódu **strel**, ktorá sa vykoná v prípade ak hráč stlačí medzerník. V tejto metóde sa nachádza časť kódu kde sa používa timer a timertask. Pri tvorbe tejto časti som si dopomohol



návodmi od autora *Bro Code*. Rovnako aj v metóde *ovladajNaboje*. V tejto časti som narazil na chybu, ktorú som vyriešil pomocou iterátora. Chybu som vyriešil vďaka autorovi menom *Lakshman Reddy* a jeho návodmi na YouTube. Táto metóda ovláda vytvorené náboje a v prípade kolízie zničí náboj aj danú vec do ktorého náboj narazil. V prípade, ak náboj narazil do stonožky, čiže do jej časti, tak sa ešte preveruje do ktorej. Na základe toho potom stonožka príde o svoju časť.

Všetky tieto objekty spúšťa trieda **Hra**. Cez metódu *spustiHru*, ktorú spúšťa menu, sa vytvárajú a zobrazujú objekty. Tu sa nastavuje aj typ lode. Typ lode sa číta so súboru "udaje.txt". V tomto súbore sa na základe mena, ktoré hráč na začiatku zadával, číta aký typ lode má. Táto informácia, spolu so skóre a levelom ktorý akurát prešiel sa ale zadáva až vtedy, keď hru hráč vyhrá, takže ak sa tam ešte táto informácia neprečítala, tak sa zvolia základné hodnoty. Pre skóre a levej je to 0, pre typ lode a teda farbu je to modrá. Na základe typu vesmírnej lode sa zvolí aj typ pavúka, ktorý je farebne rovnaký. Trieda so zapisovaním a čítaním zo súboru je cudzí ako sa píše nižšie.

Ďalej táto metóda *spustiHru* zobrazuje HUD. **HUD** je zodpovedné zobrazenie životov lode z enumu *TypyLode*, a zobrazenie skóre ktoré hráč v danom leveli nazbieral.

Ak sa hra skončí výhrou, tak ako bolo spomínane v používateľskej príručke, sa pripočíta skóre do súboru s údajmi.

**UdajeZoSuboru**, ako je napísané aj v dokumentačných komentároch, túto triedu som poskladal z veľa rôznych diskusií na GeeksForGeeks a Stack Overflow. Na prispôsobenie mojich požiadaviek mi značne dopomohol ChatGPT. Táto trieda obsahuje metódy na zápis a čítanie zo súboru "udaje.txt". Tieto údaje sú uložené vo formáte "meno = skóre, farba, level". Meno je také, aké hráč na začiatku zadal, skóre je v základe 0, farba je v základe modrá a level je tiež v základe 0. Tieto hodnoty sú nastavené vtedy, keď užívateľ zadá meno, ktoré sa v súbore ešte nenachádza. Meno, nové skóre a prejdený level sa zapíše iba ak hra skončila výhrou.

**Levely**, v tejto triede sú metódy, kde keď sa zadá číslo levelu, tak sa vrátia rôzne hodnoty pre počet prekážok a pre dĺžku stonožky. Tieto informácie sa zadávajú pri tvorbe hry a teda ich vyvoláva trieda **Menu**.

**Menu** je zodpovedná za úvodné menu do ktorého sa hráč dostane. Vypisuje sa tam stav predchozej hry, či vyhral alebo prehral a taktiež dve možnosti. Na základe stlačenej klávesy, manažér spustí metódu pre vytvorenie inštancie a objektov v hre alebo obchode. Metódy sú patrične obstarané aby manažér nespúšťal napríklad obchod, keď hráč hrá. To môže iba vtedy ak je v menu.

Keď sa hráč presúva medzi obchodom, menu a hrou, vždy sa vytvorí inštancia na dané objekty nanovo, aby ich garbage collector mohol odstrániť a nemuseli byť niekde zbytočne uložené.

**Obchod** zobrazí všetky lode a ich vlastnosti ktoré sa nachádzajú v enumu *TypyLode*. Ukladajú sa do poľa *ArrayListu* *udajeVsetkychLodi*, kde jeden prvok je jeden z enumov, ktorý obsahuje v *ArrayListe* svoje vlastnosti a jednu vlastnosť pavúka. Viac menej celá funkcionálna bola vysvetlená v používateľskej príručke.



Trieda **Main** je posledná trieda a je to trieda cez ktorú sa hra spúšťa. Vďaka tomu, že metóda sa viaže na triedu a nie na inštanciu som vytvoril súbor "PlayCentipede.bat", ktorý spúšťa "PlayCentipe.jar", ktorý práve vyvoláva túto metódu. A tak užívateľ, hráč nemusí mať nainštalovaný BlueJ.

## Cudzí kód

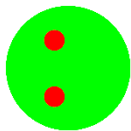
Tak ako sa píše v programátorskej príručke, cudzí kód sa náchádza v triedach **UdajeZoSuboru** a **OvladanieNaboj**.

V **UdajeZoSuboru** som si domohol takmer so všetkým. Metódy ktoré je tam vidieť sú poskladané z rôznych diskusií a doladené chatbotom ChatGPT na prispôbenie hry.

V **OvladanieNaboj** je časť kde je použitý Timer a TimerTask. Táto časť je inšpirovaná návodmi od autora *Bro Code*. Ďalej metóda ovladajNaboje, kde som narazil na chybu „ConcurrentModificationException“ s ktorou mi pomohol návod od autora *Lakshman Reddy*.

Zvyšok kódu je môj a nikto a nič mi nedopomáhalo.

## Použité obrázky



Obrázok 11 Hlava Centipede, vlastná tvorba



Obrázok 12 Telo Centipede, vlastná tvorba



Obrázok 3 Modrá vesmírna loď, vlastná tvorba



Obrázok 2 Červená vesmírna loď, vlastná tvorba



Obrázok 1 Žltá vesmírna loď, vlastná tvorba



Obrázok 10 Kameň, autor Faviola, free license



Obrázok 9 Skóre, autor jcomp, free license



Obrázok 8 Srdiečko, autor pokpak, free license

# OBCHOD

Obrázok 7 Nápis Obchod, vlastná tvorba

# CENTIPEDE

Obrázok 6 Nápis Centipede, vlastná tvorba



Obrázok 5 Možnosť Hra, vlastná tvorba



Obrázok 4 Možnosť Obchod, vlastná tvorba



## Zdroje

Code, B. (dátum neznámy). *YouTube*. Dostupné na Internete:

<https://www.youtube.com/watch?v=QEF62Fm81h4>

doc. Ing. Ján Janech, P. (dátum neznámy). *GitHub*. Dostupné na Internete:

<https://github.com/infjava/shapesge>

*Freepik*. (dátum neznámy). Dostupné na Internete: <https://www.freepik.com/>

GeeksforGeeks. (dátum neznámy). Dostupné na Internete: <https://www.geeksforgeeks.org/>

*Open Game Art*. (dátum neznámy). Dostupné na Internete: <https://opengameart.org/>

Reddy, L. (dátum neznámy). *YouTube*. Dostupné na Internete:

[https://www.youtube.com/watch?v=83Omy\\_C8t24](https://www.youtube.com/watch?v=83Omy_C8t24)

*Stack Overflow*. (dátum neznámy). Dostupné na Internete: <https://stackoverflow.com/>

*Vecteezy*. (dátum neznámy). Dostupné na Internete: <https://www.vecteezy.com/>