

Semestrálna práca Entity Devastator

Samuel Fáber 5ZYS21



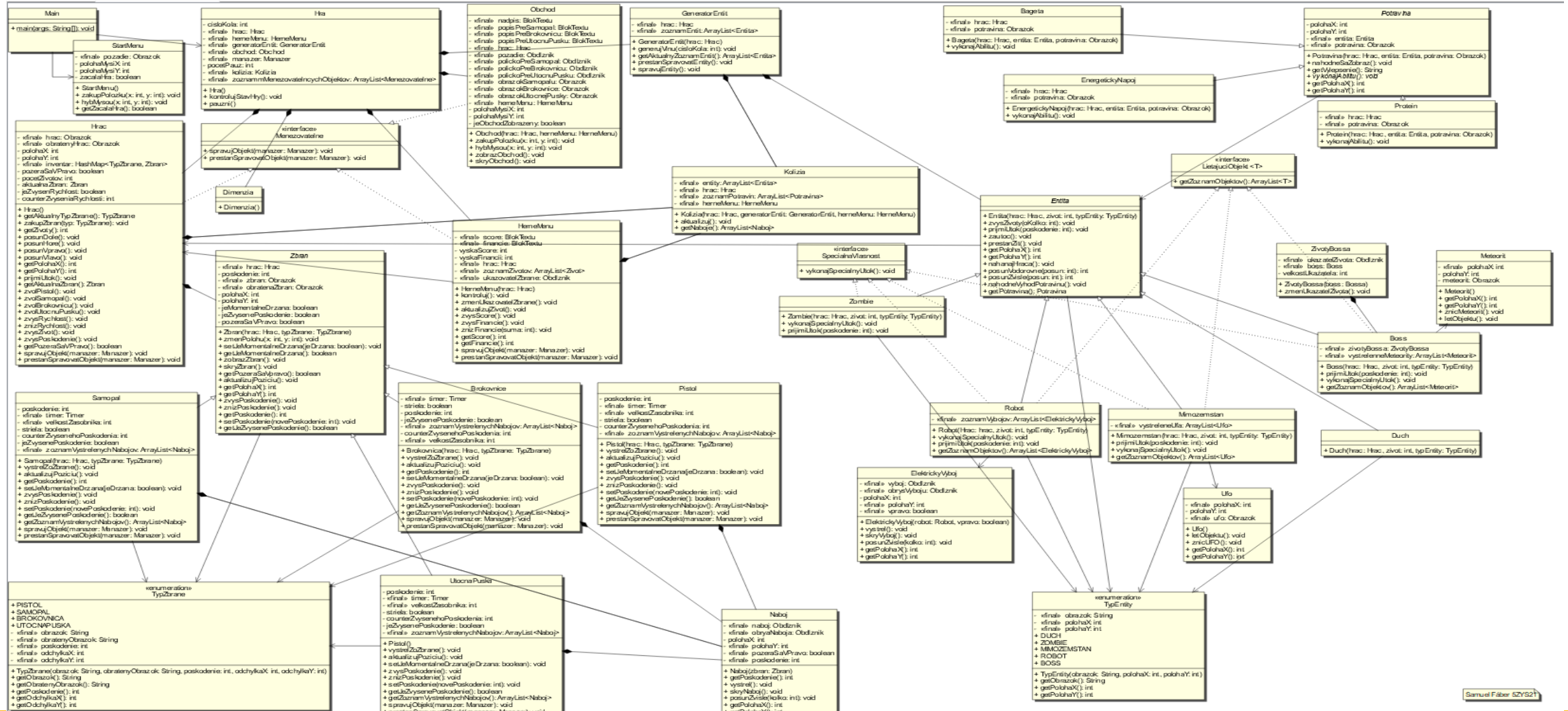
ŽILINSKÁ UNIVERZITA V ŽILINE
Fakulta riadenia
a informatiky

Myšlienka semestrálnej práce

Hráč sa ocitá na bojisku, kde čelí nepriateľským entitám prichádzajúcim vo vlnách. Jeho úlohou je, zničiť všetkých nepriateľov a zbierať z nich **hernú menu** (peniaze). Tieto peniaze môže následne investovať do **vylepšovania zbraní**, čo mu umožní efektívnejšie eliminovať ďalšie vlny nepriateľov. Celkovo musí hráč prežiť **4 vlny**, pričom každá je náročnejšia ako predchádzajúca. Na konci ho čaká **boss fight**, ktorý otestuje jeho pripravenosť a schopnosti. Výhra nad bossom znamená úspešné dokončenie hry.



UML diagram projektu



Ovládanie

Klávesa – **hore** (hráč sa posúva v danom smere)

Klávesa – **dole** (hráč sa posúva v danom smere)

Klávesa – **vpravo** (hráč sa posúva v danom smere)

Klávesa – **vľavo** (hráč sa posúva v danom smere)

Klávesa **1** – zvolenie pištole

Klávesa **2** – zvolenie samopalu

Klávesa **3** – zvolenie brokovnice

Klávesa **4** – zvolenie útočnej pušky

Klávesa **medzerník** – vystrelenie zo zbrane

Klávesa **p** – pauznutie hry s možnosťou zakúpenia zbrane

Ľavé tlačidlo **myši** – zakúpenie zbrane



Inšpirácia z internetu

```
this.striela = true;
this.timer.schedule(new TimerTask() {
    @Override
    public void run() {
        Pistol.this.striela = false;
    }
}, delay: 380);
Naboj nabojePistole = new Naboj(zbran: this);
this.zoznamVystrelenychNabojov.add(nabojePistole);
```

```
int dx = this.hrac.getPolohaX() - this.polohaX - 100;
int dy = this.hrac.getPolohaY() - this.polohaY - 100;
double vzdialenost = Math.sqrt(dx * dx + dy * dy);
int moveX = (int) (6 * dx / vzdialenost);
int moveY = (int) (6 * dy / vzdialenost);
this.posunVodorovne(moveX);
```

1. Java.util.TimerTask

Predkovia abstraktnej triedy Zbraň a Entity majú každý oneskorenie pri streľbe alebo útočení. napríklad pištoľ má pri streľbe oneskorenie 380ms. Funguje na princípe, že bool premenná strieľa sa na 380 ms zamkne na false takže hráč nemôže „spamovať“ streľbu.

2. Prenasledovanie hráča entitami

Metoda **aktualizujPoziciu()** v triede Entita ktorá riadi Pohyb Entít smerom ku hráčovi s cieľom mu ublížiť. Táto metóda ma tiež oneskorenie podobne ako Zbrane ale aby neútočili na hráča príliš rýchlo.



Inšpirácia z internetu pokračovanie

```
//Táto časť je z internetu maže dané objekty ak ich daný zoznam obsahuje
this.entity.removeIf(entitNa0dstranenie::contains);
zoznamNabojov.removeIf(nabojeNa0dstranenie::contains);
ArrayList<Potravina> potravinyNa0dstranenie = new ArrayList<>();
//Kolízia hráča s potravinou
for (Potravina potravina : this.zoznamPotravín) {
    if (potravina != null){
        int pX = potravina.getPolohaX();
        int pY = potravina.getPolohaY();
        if (pX >= this.hrac.getPolohaX() && pX <= this.hrac.getPolohaX() + 100 &&
            pY >= this.hrac.getPolohaY() && pY <= this.hrac.getPolohaY() + 100) {
            potravina.vykonajAbilitu();
            potravinyNa0dstranenie.add(potravina);
        }
    }
}
this.zoznamPotravín.removeIf(potravinyNa0dstranenie::contains);
```

3. Mazanie objektov pokiaľ sa nachádzajú v zozname

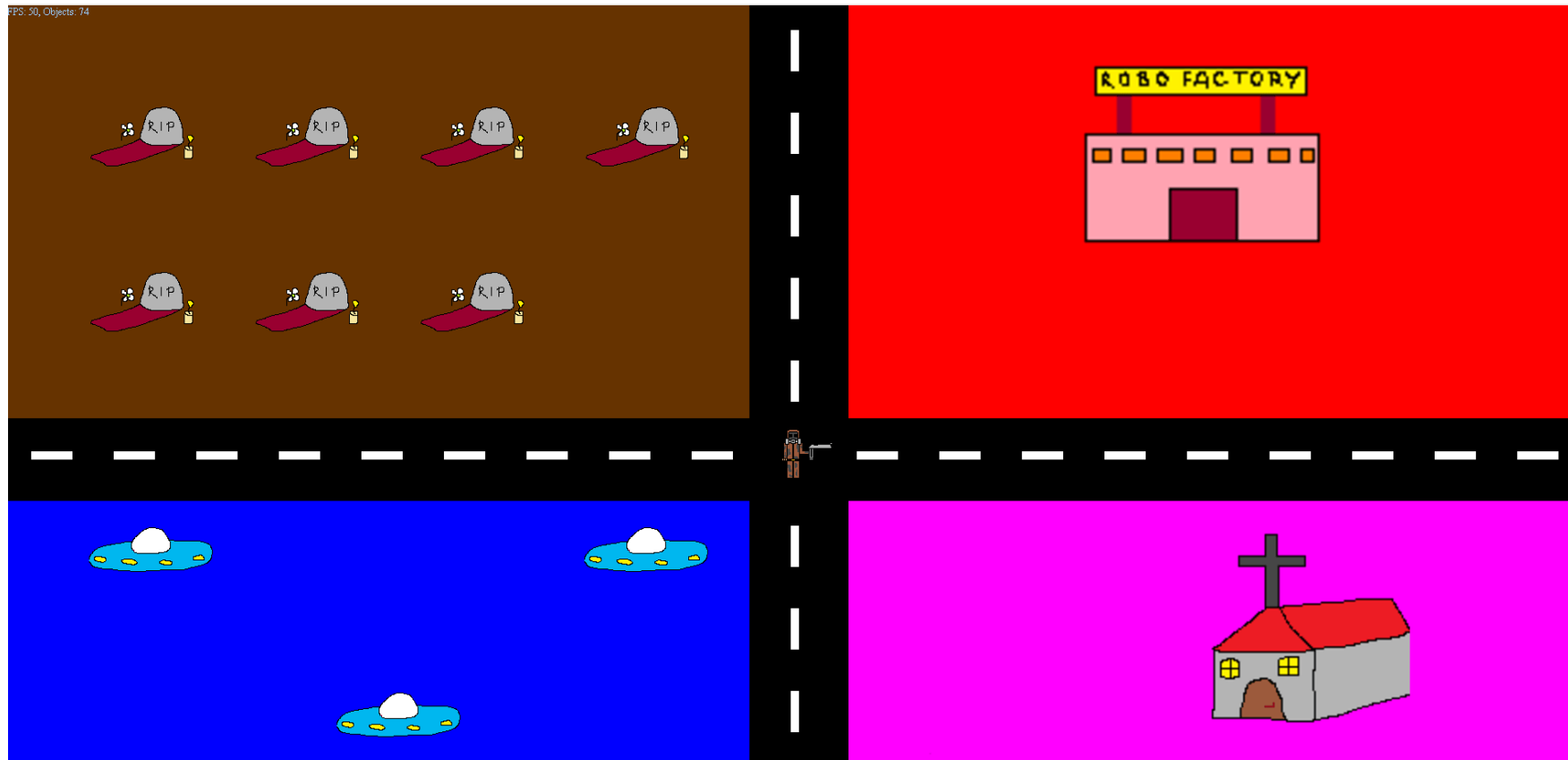
Metoda aktualizuj v triede Kolizia ktorá riadi strety objektov medzi sebou a maže objekty ktoré už prestali existovať.

<https://www.geeksforgeeks.org/arraylist-removeif-method-in-java/>



Trieda Dimenzia

Trieda **Dimenzia** reprezentuje herné prostredie kde hráč prekonáva vlny nepriateľov. Dimenzia je vytvorená pomocou Tvarov z ShapesGE.



Trieda Hra

Hlavná trieda projektu ktorá má za úlohu spravovať logiku hry ako napríklad: vytvorenie objektov, generovanie entít podľa dosiahnutého skóre alebo spustenie obchodu podľa stlačenia klávesy p.



Entity

Entity predstavujú nepriateľov ktorých má hráč za úlohu poraziť. Máme 5 rôznych entít a to: Duch, Zombie, Mimoszemšťan, Robot a Boss. Logika je riešená pomocou dedičnosti kde máme hlavného predka Entita ktorý má na starosti pohyb, útok, prijímanie poškodenia. Taktiež pri entitách môžeme vidieť polymorfizmus pomocou Interface **SpecialnaVlastnost** ktorý núti implementujúce triedy mať metódu **vykonajSpecialnyUtok()** ktorý vytvorí špeciálny útok podľa danej entity. Taktiež entity sa líšia veľkosťou svojho života

Entita **Boss** má špeciálnu vlastnosť, že zakaždým čo prijme útok tak náhodne na osi x vygeneruje meteorit ktorý keď sa len dotkne hráča tak ho okamžite usmrtí.

Entita **Zombie** má špeciálnu vlastnosť že podľa akej si náhody si vie doplniť život.

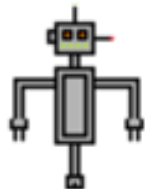
Entita **Mimoszemšťan** má špeciálnu vlastnosť, že podľa akej si náhody vie vygenerovať ufo ktoré sa náhodne na osi x vygeneruje meteorit ktorý keď sa len dotkne hráča tak ho okamžite usmrtí.



Entity pokračovanie

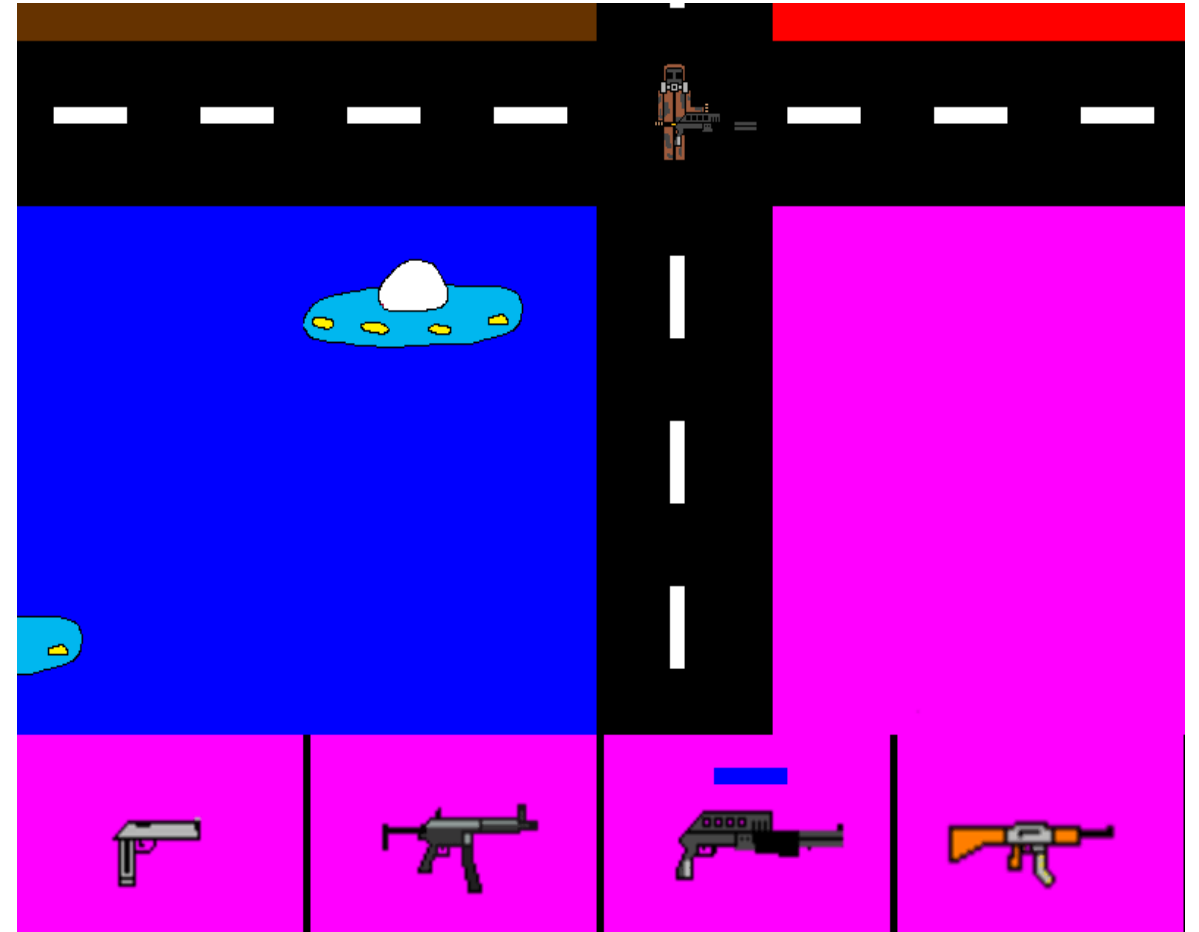
Entita **Robot** má špeciálnu vlastnosť, že vie podľa akej si náhody vystreliť elektrický výboj ktorý keď sa len dotkne hráča tak ho okamžite usmrtí.

Entita **Duch** nemá žiadnu špeciálnu vlastnosť.



Zbrane

Zbrane majú za úlohu pomôcť hráčovi prejsť cez nepriateľské vlny. Zbrane sú podobne ako Entity riešené pomocou dedičnosti kde rodičovská trieda **Zbran** má 4 predkov a to: **Pistol**, **Samopal**, **Brokovnica** a **UtocnaPuska**. Každá zbraň vie strieľať a zväčšiť si poškodenie po konzumácii špeciálnej potraviny. Rozdiel medzi zbraňami je hlavne, že každá strieľa inou rýchlosťou a taktiež že hráč má od začiatku pištoľ ale ostatné zbrane si musí zakúpiť.



Potraviny

Potraviny majú podobne ako zbrane hráčovi uľahčiť priebeh hry a to tak že mu dávajú akési vylepšenia. Štruktúra je taktiež riešená pomocou dedičnosti a to nasledovne. Rodičovská trieda **Potravina** má 3 predkov a to: **Bageta**, **Proteín** a **EnergetickyNapoj**. Pridávanie vylepšenia hráčovi je riešené pomocou polymorfizmu v rámci dedičnosti a to tak že každý potomok zdedí od triedy Potravina abstraktné metódu **vykonajAbilitu()** ktorá sa vykoná pri kolízii hráča s danou potravinou.

Potravina **Bageta** hráčovi vylieči jeden život 

Potravina **Proteín** zvýši poškodenie hráčovej momentálne držanej zbrane 

Potravina **Energetický nápoj** zvýši hráčovi rýchlosť 

Trieda GeneratorEntit

Trieda **GeneratorEntit** nám generuje entity podľa čísla kola ktorý je parametrom s logikou **1 inštancia každej entity * číslo kola**. Špeciálny príklad je parameter **5** ktorý vygeneruje inštanciu **Bossa**. Táto trieda taktiež uchováva Entity v **ArrayListe zoznamEntit** ktorý je potrebný pre triedu **Kolizia**.



Trieda Kolizia

Táto trieda sa má starať o kolízie medzi objektami v hre a to konkrétne: Kolízia hráča s meteoritom, kolízia náboja s entitou, kolízia hráča s potravinou. Najkomplexnejšou metódou triedy (možno aj projektu) je **aktualizuj()** ktorá manažuje kolízie entít, hráča, nábojov a lietajúcich objektov. V metóde ide iba o getovanie polohy objektov a porovnávanie s druhým objektom a následné vykonanie danej akcie.



Trieda HerneMenu

Herné menu kontroluje životy hráča, financie, score a aktuálnu zbraň. Je vytvorené za pomoci shapesGE.



Trieda Hrac

Trieda Hrac ma funkcionality: pohyb hráča, kúpa zbraní, používanie zbraní, využívanie špeciálnych vlastností. Ale jeho hlavnou úlohou je poraziť všetkých nepriateľov.



Záver

Projekt som s cieľom precvičiť si objektovo-orientované programovanie, hlavne dedičnosť a polymorfizmus. Snažil som sa aplikovať poznatky z cvičení a pochopiť ich praktické využitie. Hra mi pomohla lepšie porozumieť návrhu a štruktúre programovania v jave.



Ďakujem za pozornosť

Samuel Fáber 5ZYS21



ŽILINSKÁ UNIVERZITA V ŽILINE
Fakulta riadenia
a informatiky