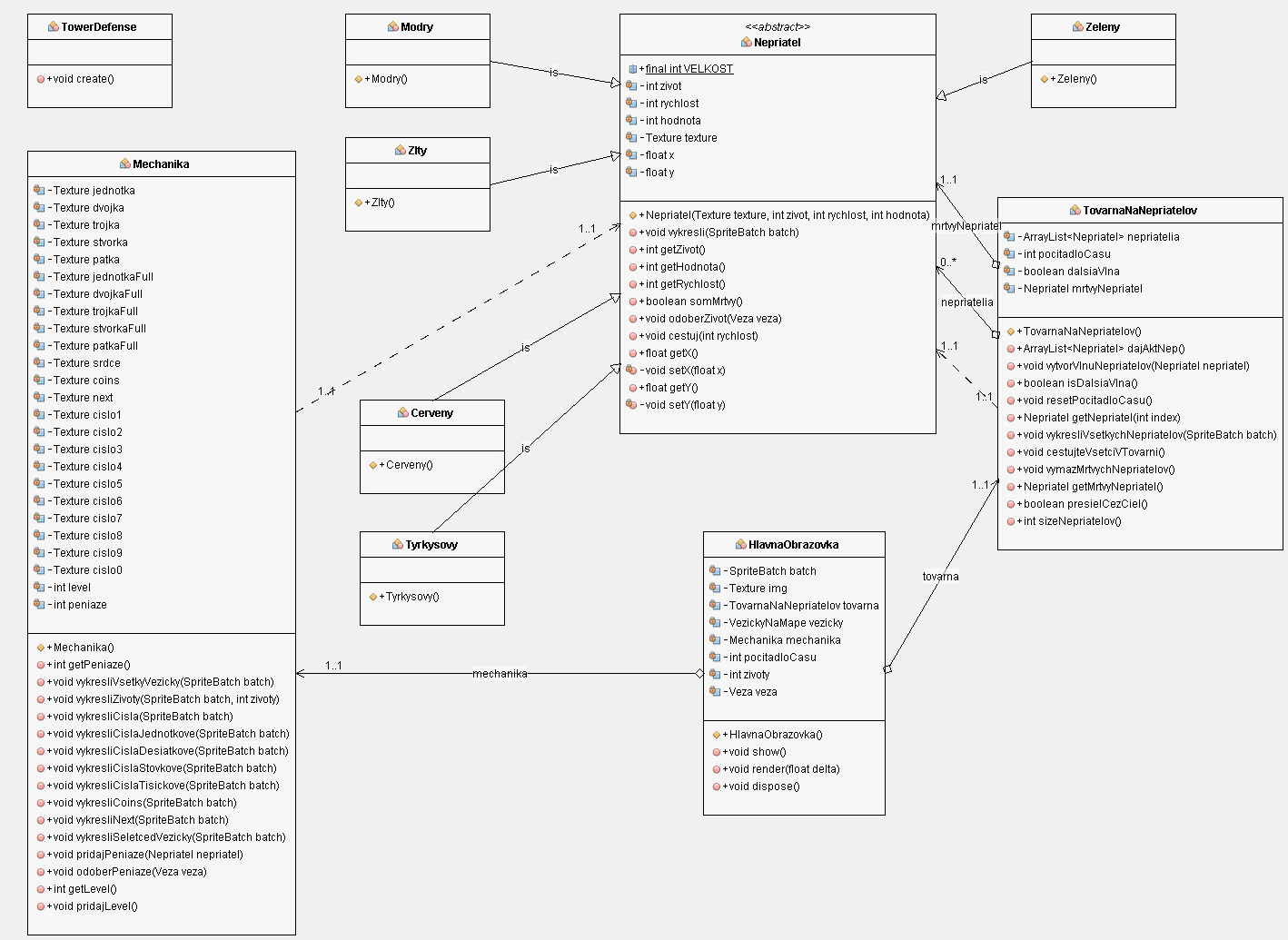
# VÃ½sledok vyhÄ¾adÃ¡vania obrÃ¡zkov pre dopyt fakulta riadenia a informatikyVÃ½sledok vyhÄ¾adÃ¡vania obrÃ¡zkov pre dopyt fakulta riadenia a informatikyŽilinská univerzita v Žiline

# Fakulta riadenia a informatiky Žilinskej univerzity v Žiline

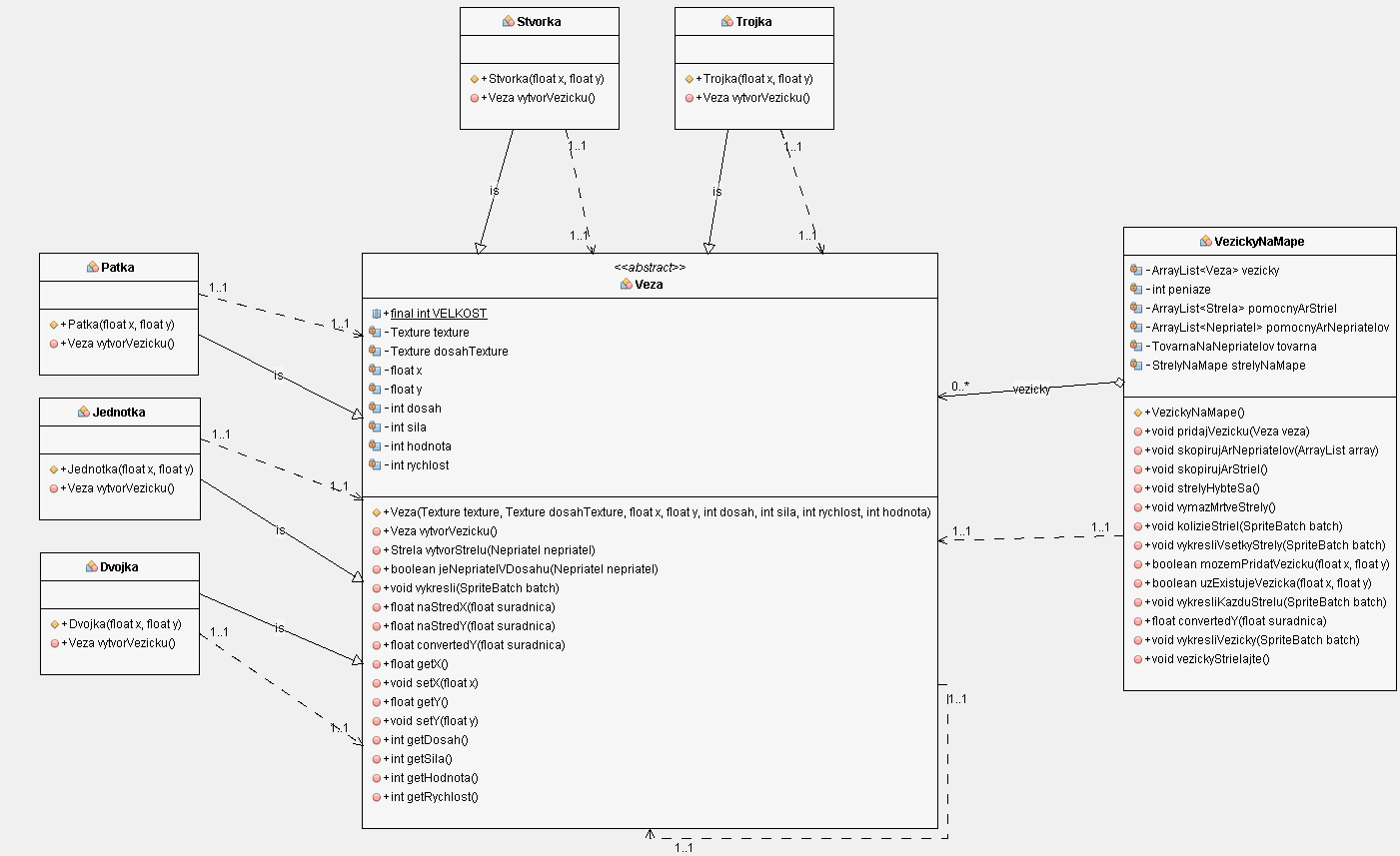
## **Informatika 2**

Semestrálna práca – Tower Defense

## **UML DIAGRAMY**

PACKAGE SEMESTRALKA

### PACKAGE VEZE



### PACKAGE NEPRIATEL

### PACKAGE STRELA

## **Úvod**

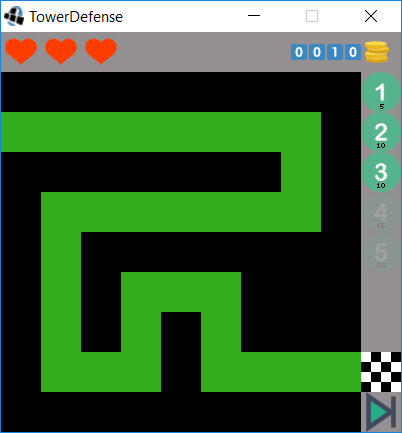
Za moju semestrálnu prácu som si si vybral hru Tower Defense, a vylepšil tak semestrálnu prácu z predmetu informatika 1.

V hre Tower Defense ide o to zničiť všetkých nepriateľov pomocou vašich vežičiek, ktoré postupne pridávate na mapu.

V momente keď vám určitý počet nepriateľov prejde do cieľa, hra skončila.

### **Návod na hranie**

Po spustení triedy DesktopLauncher cez NetBeans sa spustí mnou vytvorená hra.



V ľavom hornom rohu vidíte vaše životy, 

V pravom hornom rohu vidíte počet peňazí ktorý máte (momentálne 10) 

Pod peňiazmi sa už nachádzajú vežičky ktoré si môžete vybrať

Ak vežička nie je priehľadná, môžete na nu kliknuť a následne kliknuť na čiernu plochu, tak pridáte vežičku a bude vám odpočítané množstvo peňazí, ktoré má dána vežička napísané pod bielym číslo v tomto prípade to je veža č. 1 ktorá stojí 5 peňazí 

Vkladať vežičky môžete len na čiernu plochu, po zelenéj ploche sa budú pohybovať nepriatelia z ľavého horného rohu do pravého dolného, do cieľa.

Akonáhle chcete spustiť ďalšiu vlnu nepriateľov, musíte kliknúť na zelenú šípku pod cieľom 

Túto šípku však môžete stlačiť len v tom prípade, že na mape nie sú žiadni nepriatelia

### **Vlastnosti veží**

 - veža s číslom 1

- dosah : 1  
 - sila : 1  
 - rýchlosť : 1  
 - hodnota vežičky : 5

- veža s číslom 2

- dosah : 2  
 - sila : 1  
 - rýchlosť : 2  
 - hodnota vežičky : 10

 - veža s číslom 3

- dosah : 1  
 - sila : 2  
 - rýchlosť : 2  
 - hodnota vežičky : 10

 - veža s číslom 4

- dosah : 1  
 - sila : 1  
 - rýchlosť : 5  
 - hodnota vežičky : 15

 - veža s číslom 5

- dosah : 2  
 - sila : 2  
 - rýchlosť : 5  
 - hodnota vežičky : 20

### **Vlastnosti nepriateľov**

- Červený nepriateľ

- život : 1  
 - rýchlosť : 1  
 - hodnota nepriateľa : 1

- Modrý nepriateľ

- život : 2  
 - rýchlosť : 1  
 - hodnota nepriateľa : 1

- Tyrkysový nepriateľ

- život : 2  
 - rýchlosť : 2  
 - hodnota nepriateľa : 2

- Zelený nepriateľ

- život : 1  
 - rýchlosť : 4  
 - hodnota nepriateľa : 3

- Žltý nepriateľ

- život : 8  
 - rýchlosť : 1  
 - hodnota nepriateľa : 4

### **Počet životov**

Počet životov je nastavený na 3



Keď na obrazovke nebude ani jeden život, hra skončí

### **Popis tried a metód**

#### PACKAGE SEMESTRALKA

##### Trieda – HlavnaObrazovka

###### Metódy

**public HlavnaObrazovka()**

Konštruktor triedy HlavnaObrazovka, vytvorí pozadie,   
triedy : Mechanika, TovarenNaNepriatelov, VezickyNaMape   
Pocitadlo času nastaví na 1 a životy na 3

**public void render(float delta)**

Render skopíruje sa ArrayList Nepriatelov z TovarnaNaNepriatelov  
Skopíruje sa ArrayList Striel zo StrelyNaMape  
Podľa aktuálneho levelu sa pustí nová vlna nepriateľov  
Počítadlo času sa inkrementuje  
Pohne s vežičkami Pohne so strelami  
Kontroluje kolíziu striel s nepriateľmi Kontroluje akú vežičku má na mapu pridať   
Kontroluje, či môžem pridať vežičku na dané miesto, ak áno vežičku  
 pridá odobere peniaze za danú vežičku Kontroluje, či nepriateľ prešiel cez ciel, ak prešlo viac nepriateľov ako je možné tak hru ukončí  
Vymaže mŕtvych nepriateľov Vymaže strely ktoré mali kolíziu s nepriateľom  
Pridá peniaze za zabitého nepriateľa  
Každých 60 krát renderu vežičky vystrelia  
Ak je počítadlo vysoké tak sa zresetuje  
Čaká na pustenie nového levelu  
Vykreslí Nepriateľov, Vežičky, Strely, Mincí, Životov, Vežičky na výber, Šípku, Čísla mincí

##### Trieda - Mechanika

###### Metódy

**public Mechanika()**

Konštruktor  
Vytvorí všetky obrázky  
Nastaví level na -1  
Nastaví peniaze na 10

**public int getPeniaze()**

Vráti hodnotu peňazí

**public void vykresliVsetkyVezicky(SpriteBatch batch)**

Vykreslí všetky vežičky ktoré sa dajú vyberať

**public void vykresliZivoty(SpriteBatch batch, int zivoty)**

Vykreslí srdiečka, podľa počtu životov

**public void vykresliCisla(SpriteBatch batch)**

Vykreslí čísla

**public void vykresliCislaJednotkove(SpriteBatch batch)**

Vykreslí jednotkové číslo

**public void vykresliCislaDesiatkove(SpriteBatch batch)**

Vykreslí desiatkové číslo

**public void vykresliCislaStovkove(SpriteBatch batch)**

Vykreslí stovkové číslo

**public void vykresliCislaTisickove(SpriteBatch batch)**

Vykreslí tisíckové číslo

**public void vykresliCoins(SpriteBatch batch)**

Vykreslí obrázok mincí

**public void vykresliCoins(SpriteBatch batch)**

Vykreslí obrázok mincí

**public void vykresliNext(SpriteBatch batch)**

Vykreslí obrázok na štartovanie ďalšieho levelu

**public void vykresliSeletcedVezicky(SpriteBatch batch)**

Vykreslí vežičky ktoré sú na výber  
Ak sa dajú kúpiť nie sú priehľadné  
Ak sa nedajú kúpiť sú 60% nepriehľadné

**public void pridajPeniaze(Nepriatel nepriatel)**

Pridá do peňazí hodnotu daného nepriateľa

**public void odoberPeniaze(Veza veza)**

Odobere z peňazí hodnotu danej veže

**public int getLevel()**

Vráti level mechaniky

**public void pridajLevel()**

Inkrementuje level

##### Trieda – TowerDefense

###### Metódy

**public void create()**

Vytvorí Hlavnú obrazovku

#### PACKAGE STRELA

##### Trieda – Strela

###### Metódy

**public Strela(float x, float y, int rychlost, Nepriatel nepriatel, Veza veza)**

Konštruktor pre strelu  
Nastaví sa textúra  
pozícia X,Y  
rýchlosť strely  
nepriateľ strely  
veža z ktorej strela ide  
boolean žije, nastaví na pravdu

**public void strielajNaNepriatela()**

Strela sa začne pohybovať smerom ku nepriateľovi

**public Nepriatel getNepriatel()**

Vráti nepriateľa danej strely

**public boolean koliziaSNepriatelom(SpriteBatch batch)**

Ak má strela kolíziu s nepriateľom  
nastaví boolean zije na nepravdu  
Vráti pravdu

**public boolean isZije()**

Vráti pravdivostnú hodnotu booleanu "Zije"

**public boolean mrtvyNepriatel()**

ak je nepriateľ strely mŕtvy, vráti pravdu

**public void vykresli(SpriteBatch batch)**

ak strela žije, vykreslí sa

##### Trieda – StrelyNaMape

###### Metódy

**public StrelyNaMape()**

Konštruktor  
Vytvorí ArrayList striel

**public ArrayList<Strela> dajAktArrayStriel()**

Vráti ArrayList striel

**public void vymazMrtveStrely()**

Ak nepriateľ strely už zomrel, tak sa odstráni aj daná strela

**public void pridajStrelu(Strela strela)**

Do ArrayListu sa pridá daná strela

**public void odoberStrelu(Strela strela)**

Z ArrayListu vymaže danú strelu

**public void vytvorStreluPreVezu(Veza veza, Nepriatel nepriatel)**

do ArrayListu pridá novú strelu

**public void vsetkyVezickyStrielajteNaNepriatela()**

Každá strela v ArrayListe vystrelí na nepriateľa

#### PACKAGE VEZA

##### Trieda – Veza

###### Metódy

**public Veza(Texture texture, Texture dosahTexture, float x, float y, int dosah, int sila, int rychlost, int hodnota)**

Konštruktor  
priradí textúru vežičky, textúru dosahu, pozíciu X a Y, dosah, silu, hodnotu a rýchlosť

**public abstract Veza vytvorVezicku()**

vráti novú vežu

**public Strela vytvorStrelu(Nepriatel nepriatel)**

Vráti novú strelu

**public boolean jeNepriatelVDosahu(Nepriatel nepriatel)**

Ak sa nepriateľ nachádza v danom dosahu, vráti pravdu

**public void vykresli(SpriteBatch batch)**

Vykreslí vežičku v strede štvorca

**public float naStredX(float suradnica)**

vráti upravenú súradnicu obrázku na stred

**public float naStredX(float suradnica)**

vráti upravenú súradnicu obrázku na stred

**public float naStredY(float suradnica)**

vráti upravenú súradnicu obrázku na stred

**public float convertedY(float suradnica)**

vráti upravenú Y súradnicu myšky

**public float getX()**

vráti X súradnicu

**public void setX(float x)**

nastaví X súradnicu

**public float getY()**

vráti Y súradnicu

**public void setY(float y)**

nastaví Y súradnicu

**public int getDosah()**

vráti dosah veže

**public int getSila()**

vráti silu veže

**public int getHodnota()**

vráti hodnotu veže

**public int getRychlost()**

vráti rýchlosť

##### Trieda – VezickyNaMape

###### Metódy

**public VezickyNaMape()**

Konštruktor  
vytvorí ArrayList vežičiek, pomocných striel  
Továrnu  
Strely na mape  
nastaví peniaze na 0

**public void pridajVezicku(Veza veza)**

Do ArrayListu pridá danú vežu

**public void skopirujArNepriatelov(ArrayList array)**

Skopíruje ArrayList do pomocného ArrayListu

**public void skopirujArStriel()**

Skopíruje ArrayList striel do pomocného ArrayListu

**public void strelyHybteSa()**

Každá strela sa pohne smerom ku nepriateľovi

**public void vymazMrtveStrely()**

Vymaže mŕtve strely na mape

**public void kolizieStriel(SpriteBatch batch)**

Každá strela, ktorá má kolíziu s nepriateľom sa vymaže  
ak nastane kolízia, pridajú sa peniaze

**public void vykresliVsetkyStrely(SpriteBatch batch)**

vykreslí každú strelu

**public boolean mozemPridatVezicku(float x, float y)**

Ak sa bude vežička nachádzať na vhodnom mieste, vráti pravdu

**public boolean uzExistujeVezicka(float x, float y)**

Ak už na danej pozícii nachádza vežička, vráti pravdu

**public void vykresliKazduStrelu(SpriteBatch batch)**

Vykreslí každú strelu z ArrayListu

**public float convertedY(float suradnica)**

Konvertuje Y súradnicu

**public void vykresliVezicky(SpriteBatch batch)**

Vykreslí každú vežičku z ArrayListu

**public void vezickyStrielajte()**

Každá veža skontroluje každého nepriateľa  
ak má veža nepriateľa v dosahu, vytvorí strelu, pridá ju do ArrayListu  
Každá veža strieľa na nepriateľa v dosahu

##### Trieda – Jednotka

###### Metódy

**public Jednotka(float x, float y)**

Vytvorí novú vežu na danej pozícii  
1 dosah  
1 sila  
1 rýchlosť  
5 hodnota

**public Veza vytvorVezicku()**

Vytvorí novú vežu na danej pozícii  
vráti Jednotkovú vežičku, na danej pozícii

##### Trieda – Dvojka

###### Metódy

**public Dvojka(float x, float y)**

Vytvorí novú vežu na danej pozícii  
2 dosah  
1 sila  
2 rýchlosť  
10 hodnota

**public Veza vytvorVezicku()**

Vytvorí novú vežu na danej pozícii  
vráti Dvojkovú vežičku, na danej pozícii

##### Trieda – Trojka

###### Metódy

**public Trojka(float x, float y)**

Vytvorí novú vežu na danej pozícii  
1 dosah  
2 sila  
1 rýchlosť  
10 hodnota

**public Veza vytvorVezicku()**

Vytvorí novú vežu na danej pozícii  
vráti Trojkovú vežičku, na danej pozícii

##### Trieda – Stvorka

###### Metódy

**public Stvorka(float x, float y)**

Vytvorí novú vežu na danej pozícii  
1 dosah  
1 sila  
5 rýchlosť  
15 hodnota

**public Veza vytvorVezicku()**

Vytvorí novú vežu na danej pozícii  
vráti Štvorkovú vežičku, na danej pozícii

##### Trieda – Patka

###### Metódy

**public Patka(float x, float y)**

Vytvorí novú vežu na danej pozícii  
2 dosah  
2 sila  
5 rýchlosť  
20 hodnota

**public Veza vytvorVezicku()**

Vytvorí novú vežu na danej pozícii  
vráti Päťkovú vežičku, na danej pozícii

#### PACKAGE NEPRIATEL

##### Trieda – Nepriatel

###### Metódy

**public Nepriatel(Texture texture, int zivot, int rychlost, int hodnota)**

Konštruktor  
priradí textúru, život, rýchlosť, hodnotu a pozíciu X,Y

**public void vykresli(SpriteBatch batch)**

Nepriateľ sa vykreslí

**public int getZivot()**

Vráti život nepriateľa

**public int getHodnota()**

Vráti hodnotu

**public int getRychlost()**

Vráti rýchlosť

**public boolean somMrtvy()**

Ak má nepriateľ 0 alebo menej životov, vráti pravdu

**public void odoberZivot(Veza veza)**

Odobere nepriateľovi život o hodnote sily veže

**public void cestuj(int rychlost)**

Nepriateľ sa pohne o danú rýchlosť po danej ceste

**public float getX()**

vráti X súradnicu

**private void setX(float x)**

Nastaví pozíciu X

**public float getY()**

vráti Y súradnicu

**private void setY(float y)**

nastaví Y súradnicu

##### Trieda – TovarnaNaNepriatelov

###### Metódy

**public TovarnaNaNepriatelov()**

Konštruktor  
vytvorí nový ArrayList nepriateľov  
nastaví počítadlo času na 0, ďalšiu vlnu na pravdu  
Do mŕtveho nepriateľa nepridá nič

**public ArrayList<Nepriatel> dajAktNep()**

Vráti kópiu ArrayListu nepriateľov

**public void vytvorVlnuNepriatelov(Nepriatel nepriatel)**

Vytvorí nepriateľa vždy keď : počítadlo času modulo 40 sa rovná 0  
počítadlo času sa inkrementuje

**public boolean isDalsiaVlna()**

vráti pravdivostnú hodnotu ďalšej vlne

**public void resetPocitadloCasu()**

Resetne počítadlo času

**public Nepriatel getNepriatel(int index)**

Vráti nepriateľa daného indexu

**public void vykresliVsetkychNepriatelov(SpriteBatch batch)**

Vykreslí každého nepriateľa

**public void cestujteVsetciVTovarni()**

Každý nepriateľ sa posunie o svoju rýchlosť

**public void vymazMrtvychNepriatelov()**

vymaže "mŕtveho" nepriateľa  
ak je nepriateľ mŕtvy, tak ho vymaže z ArrayListu

**public Nepriatel getMrtvyNepriatel()**

Vráti mŕtveho nepriateľa

**public boolean presielCezCiel()**

Ak nepriateľ prešiel cez cieľ, vráti pravdu a odstráni ho z ArrayListu

**public int sizeNepriatelov()**

Vráti veľkosť ArrayListu nepriateľov

##### Trieda – Cerveny

###### Metódy

**public Cerveny()**

Konštruktor, vytvorí nového nepriateľa   
1 život  
1 rýchlosť  
1 hodnota

##### Trieda – Modry

###### Metódy

**public Modry()**

Konštruktor, vytvorí nového nepriateľa   
2 životy  
1 rýchlosť  
1 hodnota

##### Trieda – Tyrkysovy

###### Metódy

**public Tyrkysovy()**

Konštruktor, vytvorí nového nepriateľa   
2 životy  
2 rýchlosť  
2 hodnota

##### Trieda – Zeleny

###### Metódy

**public Zeleny()**

Konštruktor, vytvorí nového nepriateľa   
1 život  
4 rýchlosť  
3 hodnota

##### Trieda – Zlty

###### Metódy

**public Cerveny()**

Konštruktor, vytvorí nového nepriateľa   
8 životov  
1 rýchlosť  
4hodnota