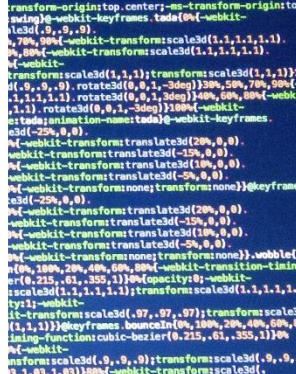


PROJEKT TANKY

OBJEKTOVÝ PRÍSTUP K RIEŠENIU PROBLÉMOV [MATERIÁL PRE ŠTUDENTOV]

MICHAL PETRÍK
NATÁLIA RÉVÉSZOVÁ
MICHAL VARGA
NORBERT ADAMKO



EURÓPSKA ÚNIA
Európsky sociálny fond
Európsky fond regionálneho rozvoja



OPERAČNÝ PROGRAM
ĽUDSKÉ ZDROJE



MINISTERSTVO
ŠKOLSTVA, VEDY,
VÝSKUMU A ŠPORTU
SLOVENSKEJ REPUBLIKY



Tento projekt sa realizuje vďaka podpore z Európskeho sociálneho fondu
a Európskeho fondu regionálneho rozvoja v rámci Operačného programu Ľudské zdroje

www.minedu.sk www.employment.gov.sk/sk/esf/ www.itakademia.sk

Projekt Tanky

Objektový prístup k riešeniu problémov

[Materiál pre študentov]

Spracované v rámci národného projektu IT Akadémia – vzdelávanie pre 21. storočie

Žilina, 2021

Projekt Tanky - Objektový prístup k riešeniu problémov [Materiál pre študentov]

Spracované s finančnou podporou národného projektu [IT Akadémia – vzdelávanie pre 21. storočie](#)

Autori: Michal Petrík, Natália Révészová, Michal Varga, Norbert Adamko

Neprešlo jazykovou úpravou.

Vydavateľ: Žilinská univerzita v Žiline

Rok vydania: 2021

Vydanie: 1.

Obsah podlieha licencií Creative Commons CC BY 4.0.

Tento projekt sa realizuje vďaka podpore z Európskeho sociálneho fondu v rámci Operačného programu Ľudské zdroje.

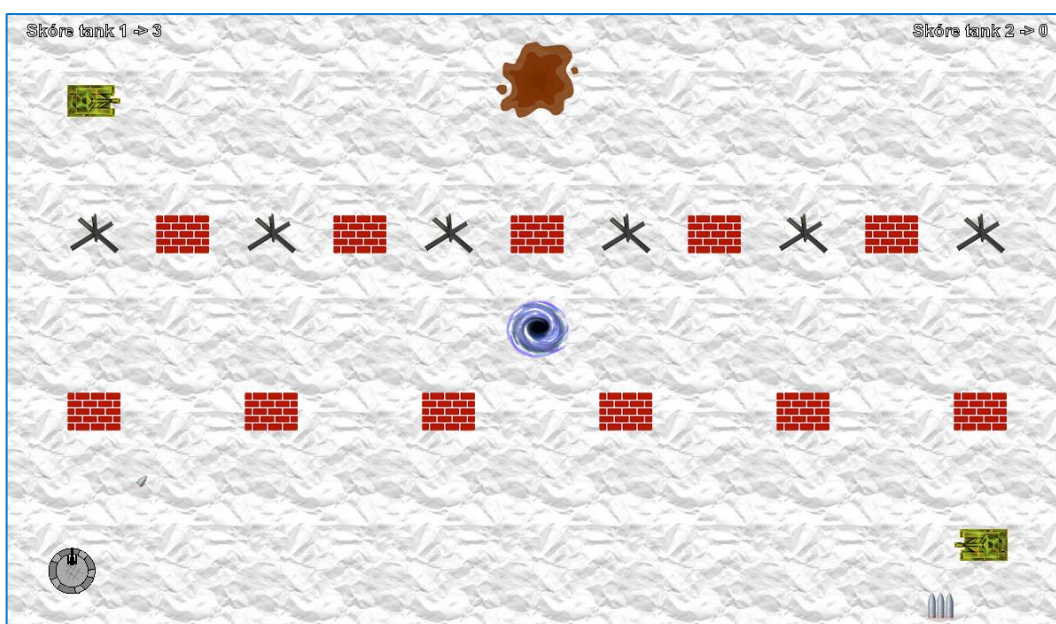
OBSAH

Úvod	5
1 Tanky	6
2 Metódy a trieda Strela.....	10
3 Pohyb tanku a bariéra	12
4 Atribúty a lokálne premenné	14
5 Strieľanie tanku	18
6 Prvok bojiska	22
7 Úrovne	24
8 Bahno a skóre hráčov	31
9 Bonusy	35
10 Generovanie bonusov	39
11 Menu	41
12 Ovládanie.....	45
13 Nepriateľská vežička.....	47
14 Čierna diera	48
Index obrázkov	50
Bibliografia.....	52

ÚVOD

POPIS A PRAVIDLÁ HRY

Tanky je hra pre dvoch hráčov. Každý hráč ovláda pohyb a strieľanie svojho tanku na bojisku. Tank má obmedzený počet nábojov a danú rýchlosť. Na bojisku sa nachádzajú zábrany a rôzne bonusy (doplnenie nábojov, zvýšenie rýchlosti pohybu tanku), ktoré sa generujú automaticky. V hre sú tri úrovne odstupňované od najľahšej po najťažšiu. Na bojisku sú ďalšie prekážky ako bahno, nepriateľská vežička a čierna diera, ktoré sťažujú hru. Pri zapnutí hry sa hráčom zobrazí menu a po skončení poslednej úrovne sa zobrazí ukončovacie menu, v ktorom je zobrazené skóre obidvoch hráčov s možnosťou pokračovania v hre alebo jej skončenia. Cieľom hry je získať vyššie skóre ako má nepriateľ, pričom skóre sa zvyšuje zneškodnením protivníka.



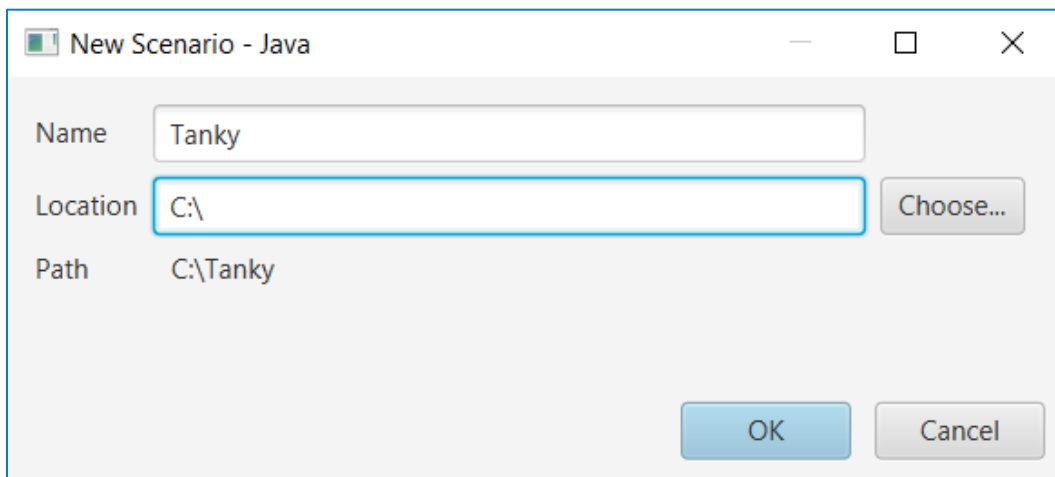
Obrázok 1: Hra Tanky

1 TANKY

V prvej kapitole založíte nový projekt pre hru *Tanky*. Svetu nastavíte pozadie a veľkosť. Vytvoríte hráča – tank, ktorý bude v hre súperiť s iným tankom o víťazstvo. Vyskúšate si príkazy na vytvorenie tanku, jeho otočenie a posúvanie v rámci sveta.

ÚLOHA 1.1

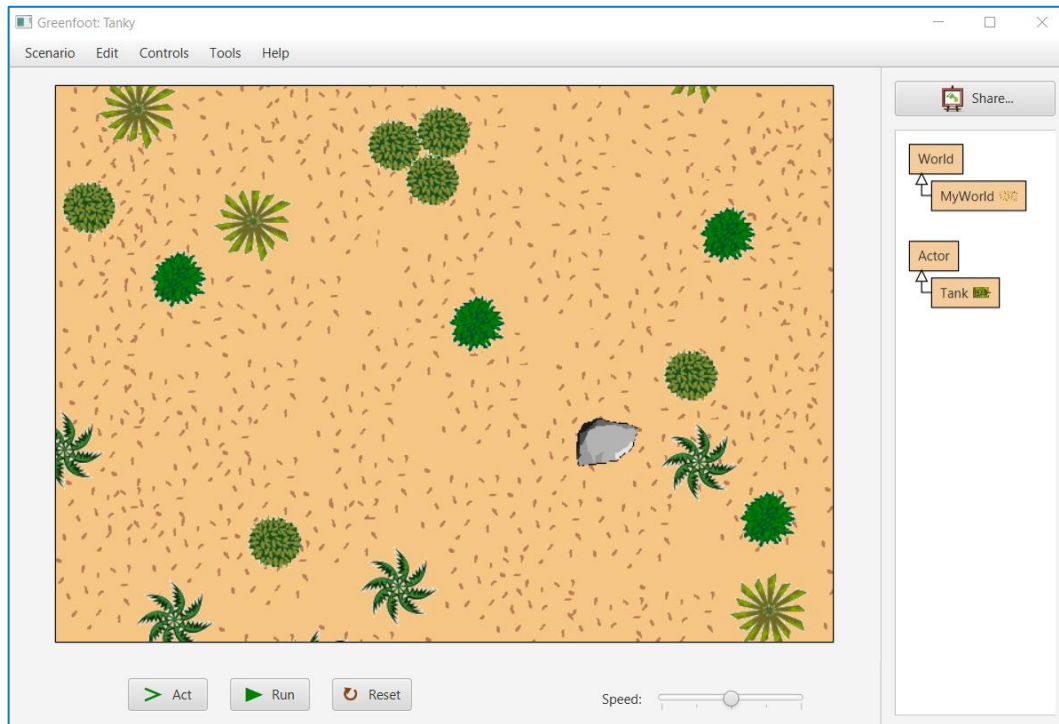
Na úvod si vytvorte nový projekt. Ako názov projektu nastavte pomenovanie *Tanky*. Ďalej si zvolte vhodnú zložku, do ktorej projekt uložíte.



Obrázok 2: Vytvorenie nového projektu

ÚLOHA 1.2

V ďalšom kroku nájdite výstižný obrázok bojiska, na ktorom sa bude odohrávať hra *Tanky*. Pridajte tento obrázok svetu – **MyWorld**. Inšpiráciu môžete nájsť na nasledujúcom obrázku.



Obrázok 3: Pozadie v hre Tanky

ÚLOHA 1.3

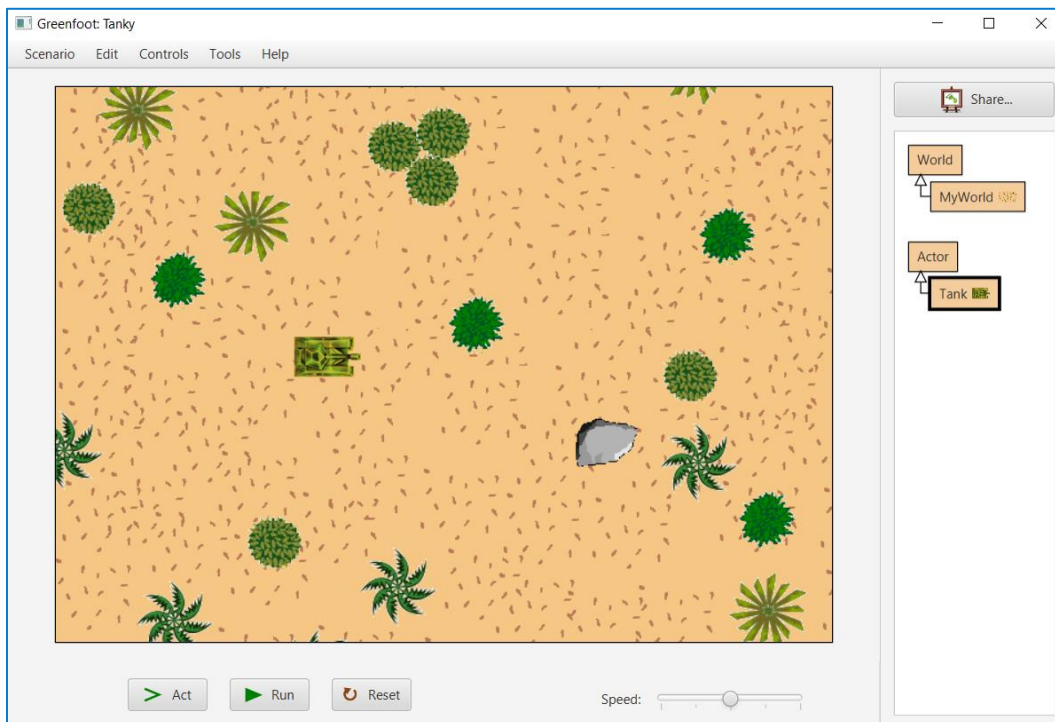
V editore kódu triedy **MyWorld** upravte konštruktor tak, aby sa vytvoril svet o veľkosti 700x500 buniek, pričom každá bunka bude veľká jeden bod.

ÚLOHA 1.4

Vytvorte hráča. Hráč v tejto hre bude reprezentovaný tankom, ktorý bude súperiť s iným tankom o víťazstvo. Ďalej vyhľadajte obrázok tanku a priradte ho novému objektu.

ÚLOHA 1.5

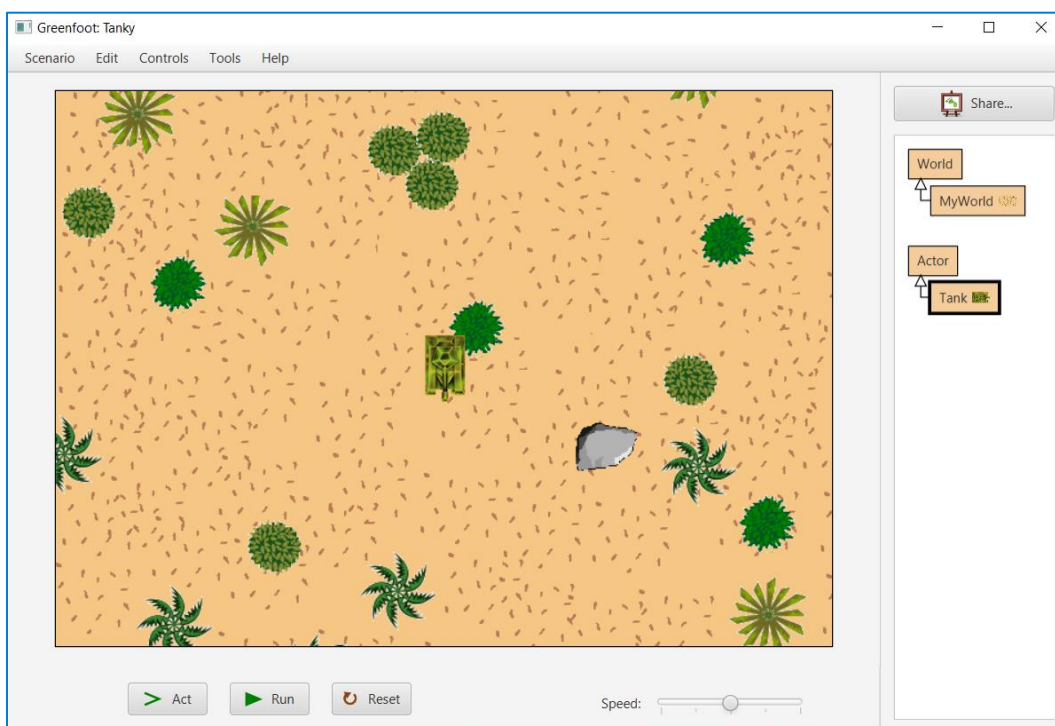
Vytvorte inštanciu tanku a umiestnite ju do sveta.



Obrázok 4: Vloženie tanku do sveta




ÚLOHA 1.6

Preskúmajte nad inštanciou triedy **Tank** metódy, ktoré ponúka nástroj Greenfoot. Sledujte pritom zmeny vo vnútornom stave inštancie. Umiestnite tank do ľavého horného rohu. Pomocou metód **move** a **turn** prikážte tanku, aby sa presunul do stredu sveta, ako je zobrazené na nasledujúcom obrázku.



Obrázok 5: Zmena pozície tanku

Tank : Tank

int x	350	Inspect
int y	250	
private int mySequenceNumber	0	
private int lastPaintSequenceNumber	0	
int rotation	90	
World world		
private GreenfootImage image		
private Object data	null	
private Rect boundingRect		
private int[] boundingXs		
private int[] boundingYs		
private int imageWidth	60	
private int imageHeight	36	

Show static fields

Close

Obrázok 6: Vnútroň stav tanku

2 METÓDY A TRIEDA STRELA

V tejto kapitole vytvoríte metódy na pohyb, otáčanie tanku a opis trojuholníka pohybom tanku. Vyskúšate si tiež písanie dokumentačných komentárov. Vytvoríte nového aktora hry – strelu, ktorá po spustení hry prejde celým svetom v smere, ktorým je otočená.

ÚLOHA 2.1

Vytvorte metódu **vpred**, ktorá posunie inštanciu tanku o 5 buniek. Zavolajte vytvorenú metódu.

ÚLOHA 2.2

Ďalej vytvorte metódy na otočenie tanku o 3 stupne vpravo s názvom **otocSaDoprava** a otočenie tanku o 3 stupne vľavo s názvom **otocSaDolava**. Opäť overte svoje riešenie zavolaním týchto metód.

ÚLOHA 2.3

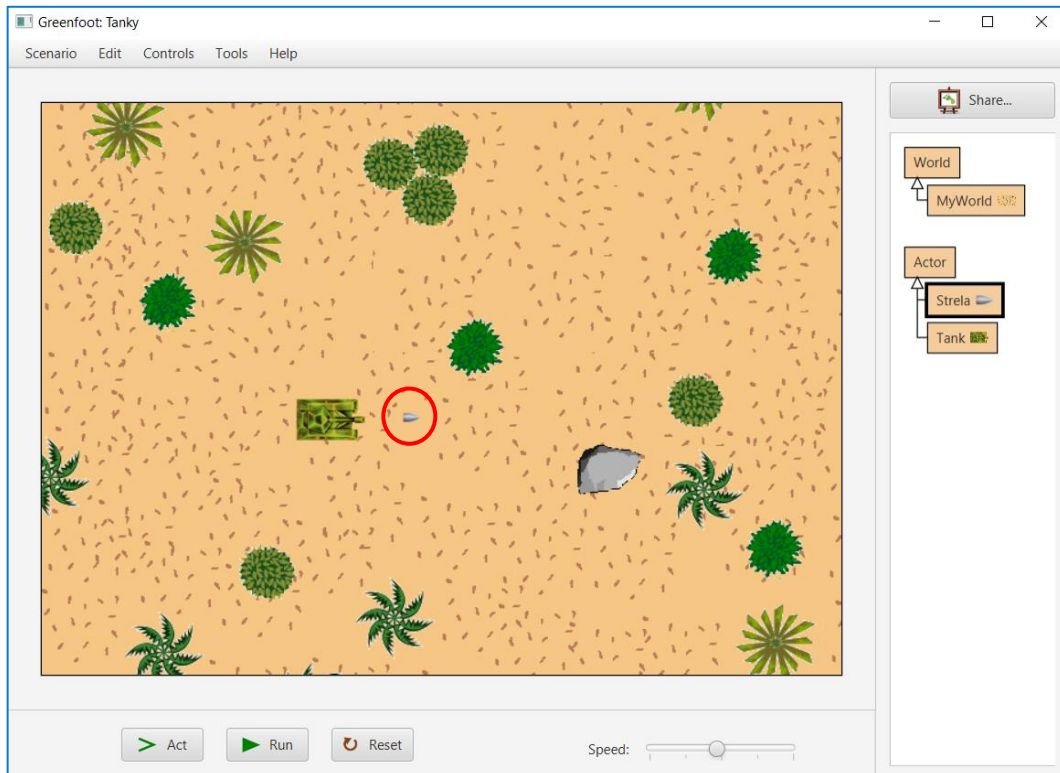
Doplňte tanku metódu **opisTrojuholnik**, pomocou ktorej tank opíše svojím pohybom trojuholník.

ÚLOHA 2.4

Doplňte dokumentačné komentáre pre metódy **vpred**, **otocSaDoprava** a **otocSaDolava**. Triede **Tank** upravte dokumentačný komentár, pridajte do neho verziu a autora. Pomocou dokumentačného okna si pozrite zmenené a pridané dokumentačné komentáre.

ÚLOHA 2.5

Vytvorte nového aktora hry *Tanky*, strelu. Priradte strele obrázok podobne, ako ste ho pridali tanku.



Obrázok 7: Strely umiestnené vo svete

ÚLOHA 2.6

Naprogramujte strelu tak, aby po svojom vytvorení a spustení hry, bez potreby vyvolania akejkoľvek metódy, prešla celým svetom v smere, ktorým je otočená.

3 POHYB TANKU A BARIÉRA

V tretej kapitole zmeníte strele smer jej pohybu, keď sa dotkne okraja sveta. Pomocou klávesnice budete ovládať pohyb tanku. Vytvoríte bariéru, cez ktorú tank nedokáže prejsť, bariéra teda v hre *Tanky* predstavuje prekážku.

ÚLOHA 3.1

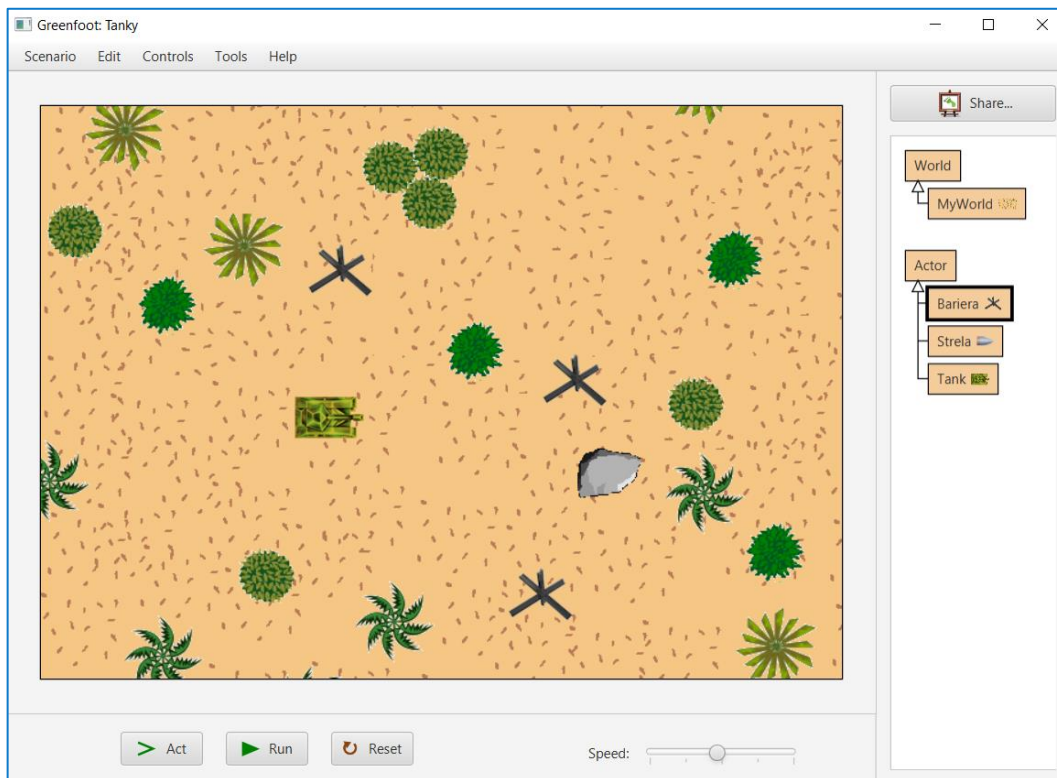
V tejto úlohe naprogramujte strelu tak, aby sa odrazila od okraja sveta.

ÚLOHA 3.2

Nasledujúcou úlohou je naučiť tank reagovať na stlačený kláves na klávesnici, a tak vykonávať pohyb tanku. Pridajte kód, ktorý po stlačení šípok na klávesnici umožní tanku pohybovať sa dopredu a otáčať sa do strán. Zisťovanie stlačenia klávesu je možné pomocou metódy **Greenfoot.isKeyDown**, ktorej vstupným parametrom je anglický názov klávesu.

ÚLOHA 3.3

Vytvorte bariéru ako nového aktora hry. Bariéra predstavuje objekt, cez ktorý sa vo finálnej podobe hry tank nedokáže dostať, ale dokáže cez neho prestreliť. Okrem toho nájdite a prideľte bariére prislúchajúci obrázok.



Obrázok 8: Pridanie bariér do bojiska

ÚLOHA 3.4

Vytvorte aspoň jednu inštanciu bariéry na takých pozíciách v bojisku, ktoré nezasahujú do už vytvoreného tanku. V triede **Tank** upravte metódu **vpred** tak, aby sa tank posunul iba v prípade, že sa nedotýka bariéry.

Negáciu alebo zápor môžete získať pomocou logického operátora „!“, ktorý pridáte pred logický výraz. Príkladom je nasledujúci kód, ktorý posunie tank vpred v prípade, ak nie je stlačená šípka hore.

```
if (!Greenfoot.isKeyDown("up")) {
    this.vpred();
}
```

Predchádzajúcu ukážku kódu si môžete skúsiť vo svojom projekte. V triede **Tank** v metóde **act** pridajte pred podmienku stlačenia šípky hore logický operátor negácie. Po spustení hry sa bude tank neustále pohybovať, až pokiaľ nestlačíte šípku hore alebo pokiaľ nenarazí na okraj sveta. Nezapomnite nakoniec logický operátor odstrániť, pretože v hre *Tanky* na tomto mieste nemá správny význam.

4 ATRIBÚTY A LOKÁLNE PREMENNÉ

V tejto kapitole pripravíte hru pre dvoch hráčov. Každý tank bude reagovať na iné klávesy a môže mať inú rýchlosť ako jeho súper. Tanku pridáte nový atribút reprezentujúci počet životov. Vytvoríte novú prekážku múr, ktorá zatiaľ bude fungovať rovnako ako bariéra, ale v ďalších kapitolách ju bude možné zničiť po viacerých zásahoch strelou.

ÚLOHA 4.1

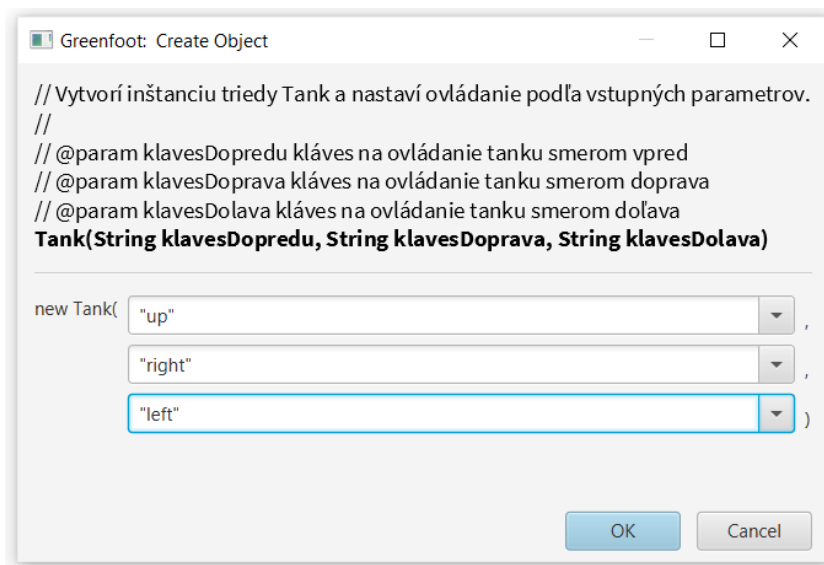
Vytvorte si dve inštancie tankov a vyskúšajte ich pohyb prostredníctvom kláves. Sledujte pohyb tankov.

ÚLOHA 4.2

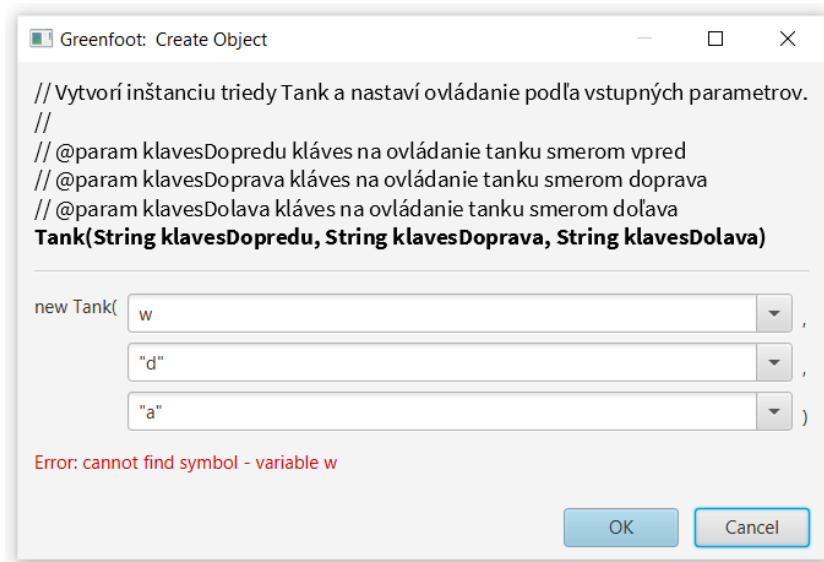
Ako ste mohli vidieť v predošlej úlohe tanky reagujú na rovnaké klávesy. Vyriešte tento problém pomocou parametrického konštruktora tanku a atribútov **klavesDopredu**, **klavesDoprava** a **klavesDolava**, čím získate nezávislé ovládanie každej inštancie tanku. Dopíšte vytvorenému konštrukturu dokumentačný komentár s popisom vstupných parametrov.

ÚLOHA 4.3

Upravte kód tanku, ktorý reaguje na stlačenie klávesov tak, aby ste použili atribúty reprezentujúce klávesy. Vo svete vytvorte dve inštancie triedy **Tank**. Pre každú inštanciu nastavte iné klávesy na vykonávanie pohybu. Vyskúšajte riadenie pohybu tankov prostredníctvom klávesov. Ich pohyb je po vykonaných zmenách nezávislý.



Obrázok 9: Vytvorenie inštancie tanku



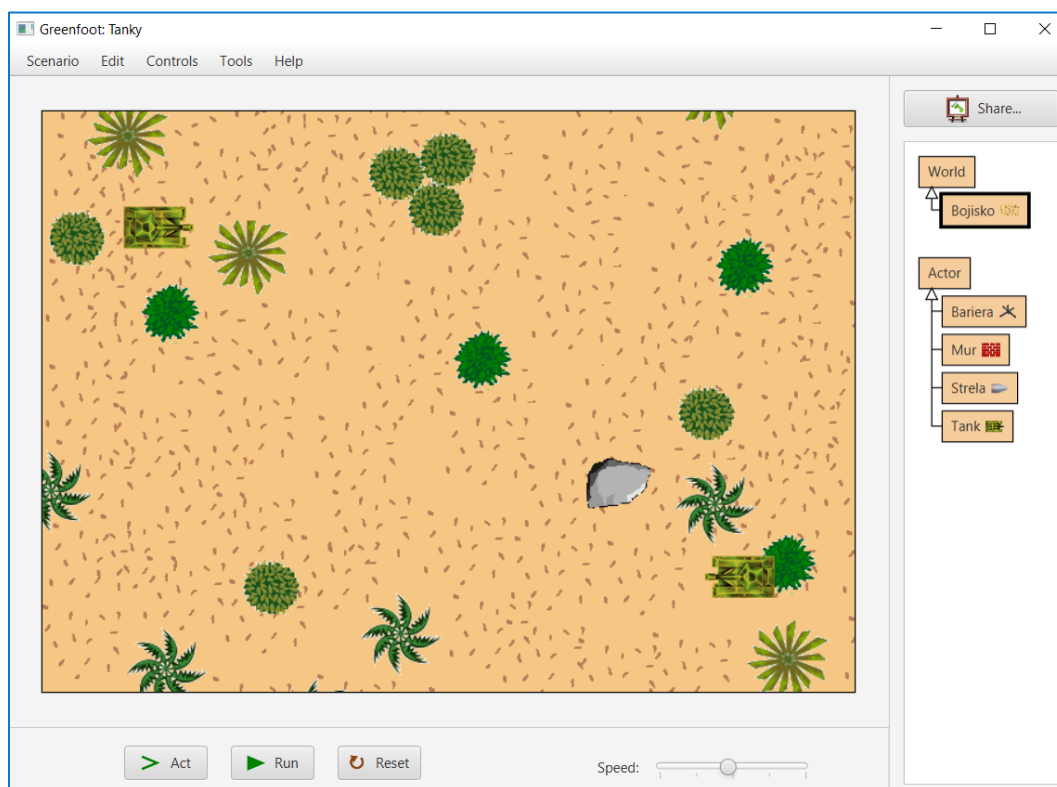
Obrázok 10: Nesprávne nastavenie kláves

ÚLOHA 4.4

Pridajte tanku atribút reprezentujúci jeho rýchlosť. Inicializujte ho na hodnotu 5. Ďalej preťaďte konštruktor triedy **Tank** pridaním nového vstupného parametra rýchlosť. Umožníte tak tento atribút meniť pre jednotlivé inštancie tanku. Atribút rýchlosť použijete pri pohybe tanku dopredu.

ÚLOHA 4.5

Premenujte objekt **MyWorld** na výstižnejšie pomenovanie sveta, napríklad na **Bojisko**. V bojisku vytvorte a uložte dva nové referenčné atribúty tankov, ktoré budú vykonávať pohyb prostredníctvom rôznych kláves. Pridajte vytvorené tanky do sveta tak, aby boli umiestnené podobne ako na nasledujúcom obrázku.



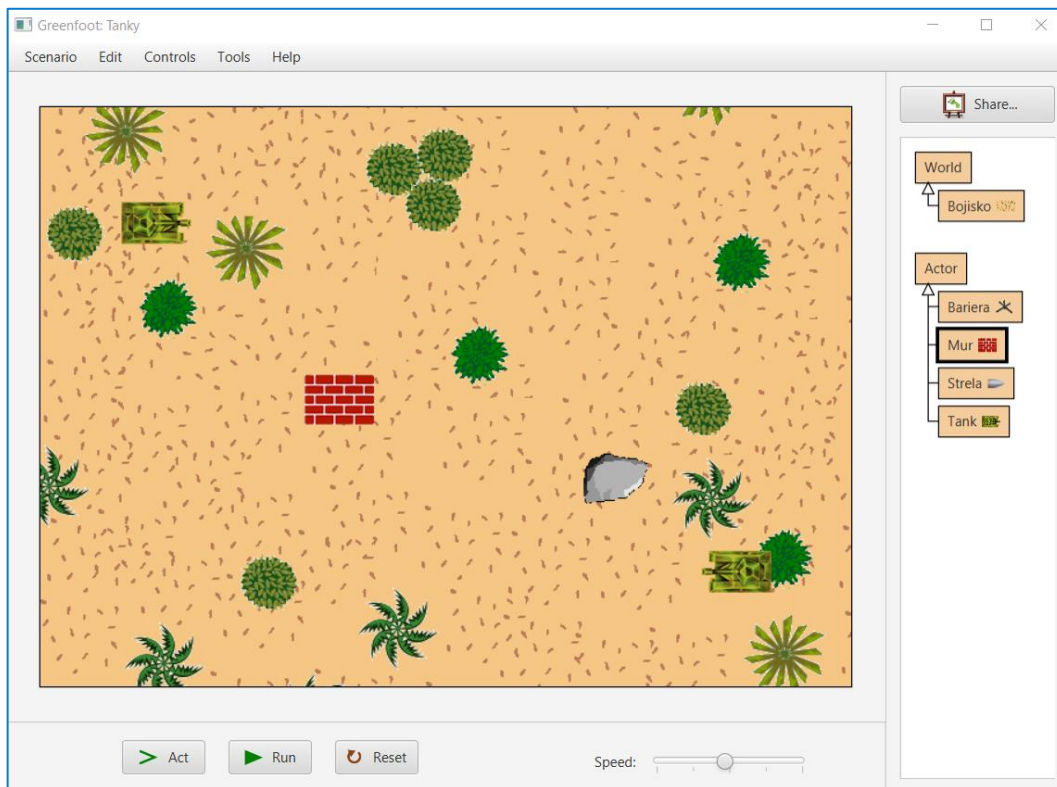
Obrázok 11: Pridanie tankov do bojiska

ÚLOHA 4.6

Pridajte tanku atribút s názvom **pocetZivotov**, predstavujúci počet jeho životov a inicializujte ho na hodnotu 3. Inicializáciu urobte tak, aby boli životy nastavené pri zavolaní ktoréhokoľvek konštruktora.

ÚLOHA 4.7

Vytvorte novú prekážku múr, za ktorú sa bude môcť tank skryť a nebude ju možné zničiť iba jednou strelou. Pridajte múru vhodný obrázok a vložte ho do bojiska, zatiaľ iba cez kontextové menu. Pridajte múru atribút **pocetZivotov**, ktorý nastavte na hodnotu 2.



Obrázok 12: Vloženie prekážky múr do bojiska

ÚLOHA 4.8

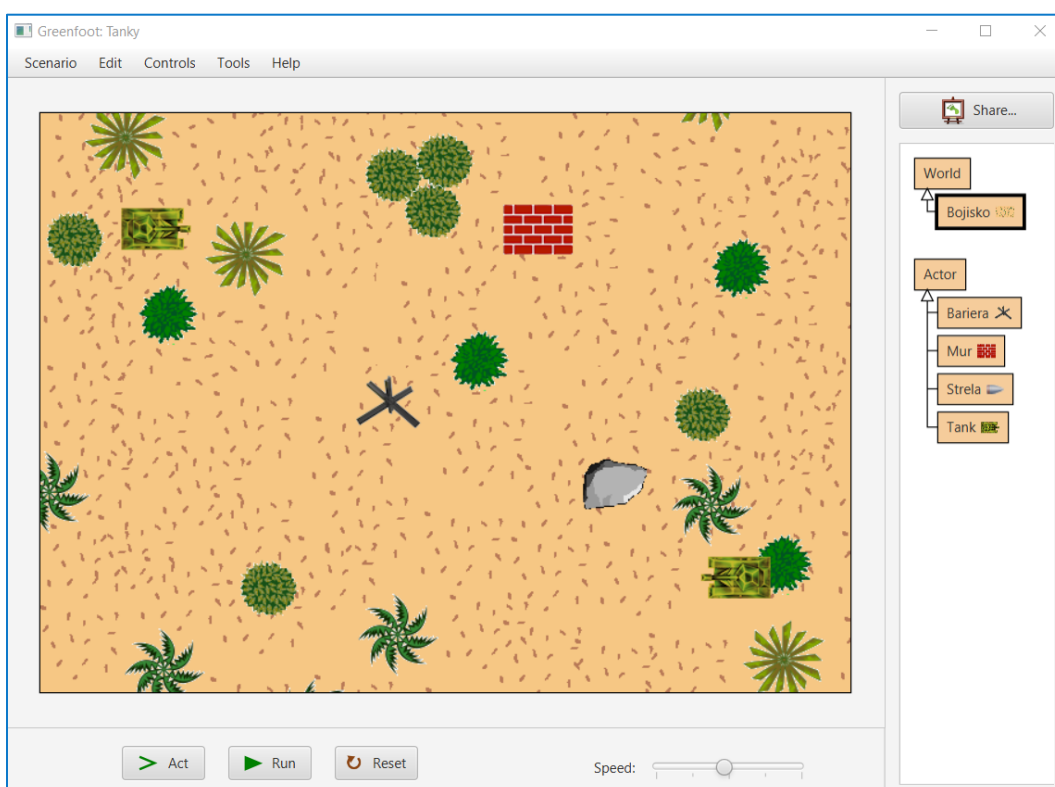
Upravte tank tak, aby nedokázal cez prekážku múr prejsť.

5 STRIEĽANIE TANKU

V kapitole Strieľanie tanku pridáte do bojiska prekážky ako stacionárne objekty hry. Vyskúšate si tvorbu metódy s návratovou hodnotou. Zvyšok kapitoly sa venuje implementácii strieľania tanku, pričom tank bude mať obmedzenú muníciu a obmedzený čas medzi jednotlivými strelami.

ÚLOHA 5.1

Vytvorte na bojisku bariéru a múr. Tieto prekážky nie je potrebné ukladať ako atribúty v triede **Bojisko**.



Obrázok 13: Pridanie prekážok do bojiska

ÚLOHA 5.2

Vytvorte tanku metódu **mozelstVpred** s návratovou hodnotou typu **boolean**, ktorá bude vracieť hodnotu **true** alebo **false** v závislosti od toho, či sa tank dotýka niektorej z prekážok alebo nie. Doplňte aj podmienku, aby tank nemohol prejsť cez iný tank. Túto metódu vhodne použite.

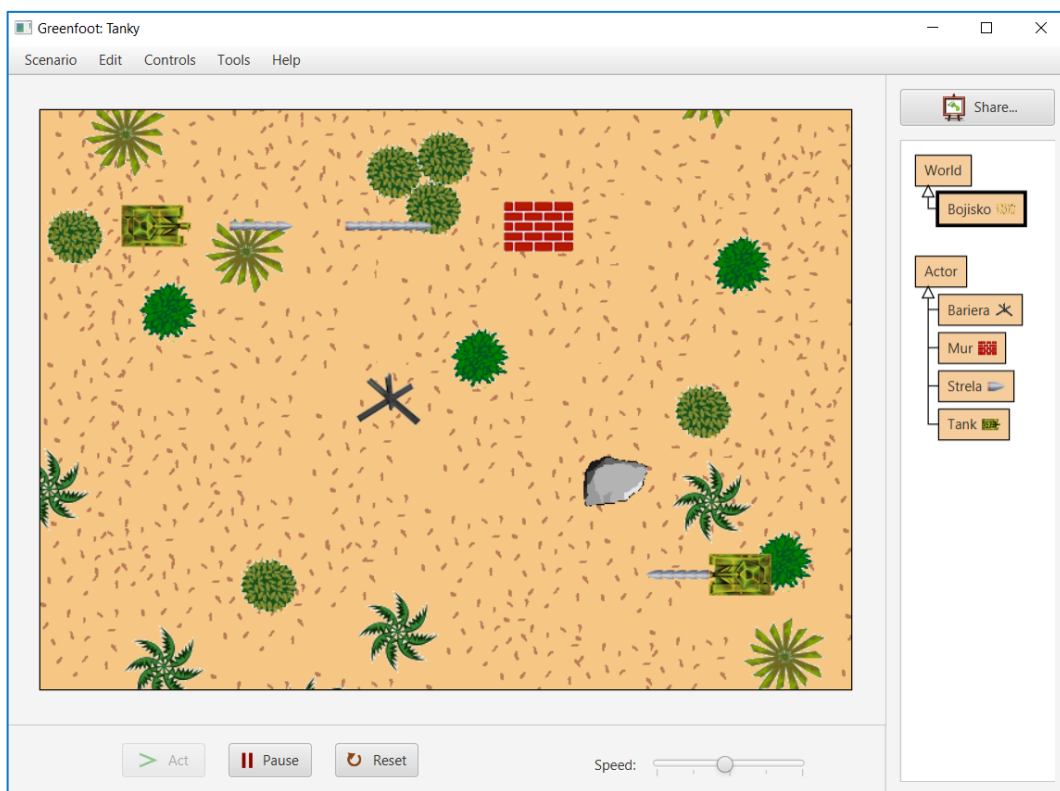
ÚLOHA 5.3

V nasledujúcich častiach pridáte tanku schopnosť strieľať. Dopíšte do tanku metódu **vystrel** na vytvorenie strely, ktorá bude letieť v smere, ktorým je otočený tank.

Keďže sa už nenachádzate v triede **Bojisko**, ktorá je potomkom triedy **World**, ale v triede **Strela**, ktorá je potomkom triedy **Actor**, budete potrebovať najprv získať samotný svet, aby ste do neho mohli strelu vložiť. Svet získate pomocou metódy **getWorld**. Do takto získaného sveta môžete následne pridať vytvorenú strelu. Funkčnosť tejto metódy môžete overiť pri jej zavolaní nad inštanciou tanku z kontextového menu.

ÚLOHA 5.4

Pridajte klávesu, pomocou ktorej tank vystrelí strelu. Naučte tank reagovať na túto klávesu vystrelením. Doplnite referenčným atribútom tankov umiestnených v triede **Bojisko** klávesy vystrelenia.



Obrázok 14: Strieľanie tanku pomocou určeného klávesu

ÚLOHA 5.5

Upravte kód strely tak, aby po dotyku s okrajom sveta nezmenila svoj smer, ale aby zanikla.

ÚLOHA 5.6

Zničenie strely nenastane len v prípade, keď sa dostane na okraj sveta, ale aj v prípade, keď zasiahne tank. Získanie tanku, ktorý strela zasiahla, je možné zavolaním metódy

this.getOneIntersectingObject(Tank.class), ktorá vráti referenciu na aktora, ktorý následne musíte ešte pretypovať na **Tank**. Uložte tento tank do lokálnej premennej, pretože budete s ním v ďalšej úlohe pracovať.

Porozmýšľajte, ako sa dá pomocou získaného zasiahnutého tanku napísať podmienka, v ktorej ak strela zasiahne tank, tak táto strela bude vymazaná zo sveta. Po implementácii kódu si vyskúšajte svoje riešenie.

Strela bude ničiť aj prekážku múr, avšak až po viacerých zasiahnutiach. V aktuálnom stave hry však strela cez túto prekážku preletí. Nepriestrelnosť múru sa bude riešiť v neskoršej kapitole.

ÚLOHA 5.7

V aktuálnej verzii hry sa vystrelená strela okamžite zničí, pretože sa dotkne tanku, ktorý ju vystrelil. Takto by ale hra *Tanky* nefungovala. Pouvažujte, ako by sa to dalo napraviť. Riešenie implementujte a overte.

ÚLOHA 5.8

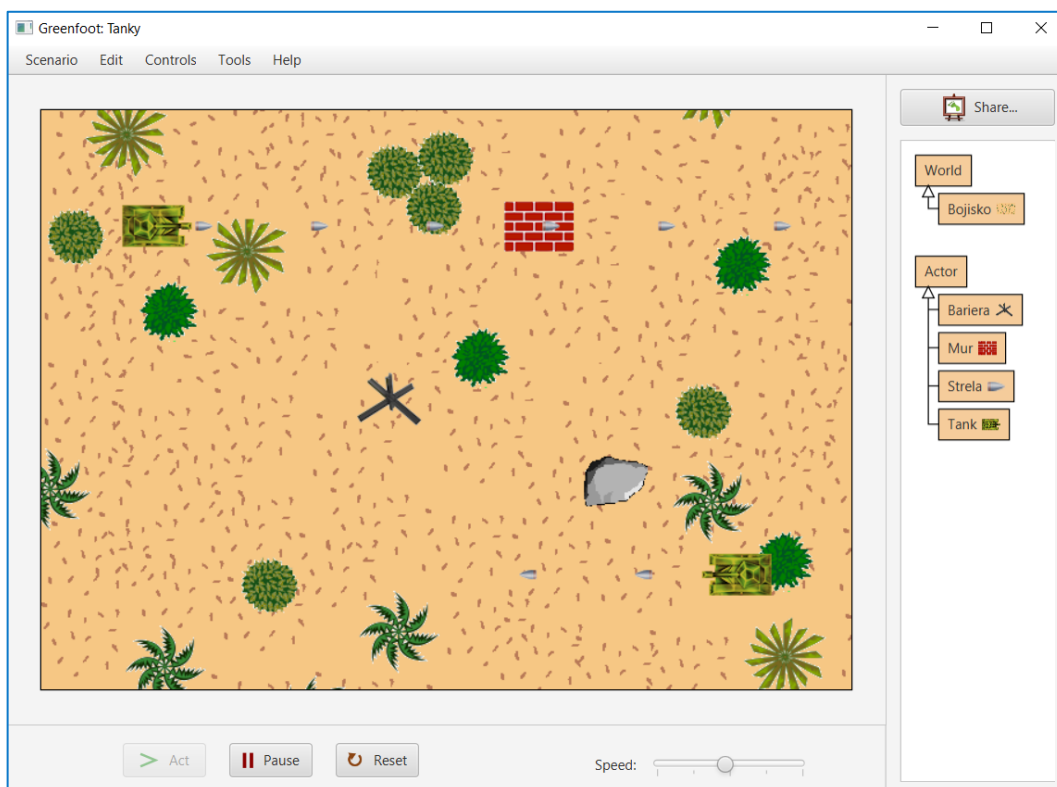
Vašou ďalšou úlohou je znížiť tanku počet životov, keď ho zasiahne strela. Dopíšte tanku bezparametrickú metódu **zasah** bez návratovej hodnoty, ktorá zníži počet životov tanku. Ak počet životov tanku klesne na hodnotu nula, tak tank vymažte z bojiska. Metódu **zasah** zavolajte na správnom mieste v triede **Strela**.

ÚLOHA 5.9

Obmedzte rýchlosť strieľania tanku dobíjaním munície. Pridajte tanku nový atribút **casDoDobitia**, ktorý bude reprezentovať čas do dobitia munície. Inicializujte ho a v metóde **act** ho znižujte. Keď dosiahne nulu, tak zase môže tank reagovať na výstrel. Vytvorte metódu s návratovou hodnotou **mozeVystrelit**, ktorá bude kontrolovať, či tank stihol dobiť muníciu. Túto metódu zavolajte na správnom mieste. Po vystrelení aktualizujte atribút času dobíjania munície na hodnotu 10.

ÚLOHA 5.10

Obmedzte muníciu tanku prostredníctvom nového atribútu **municia**, predstavujúceho počet jeho nábojov. Inicializujte muníciu tanku na hodnotu 10. Pred výstrelom kontrolujte, či má tank dostatok munície. Po každom výstrele počet munície znížte.



Obrázok 15: Strieľanie s dobíjaním munície

6 PRVOK BOJISKA

Jednotlivé prvky hry majú spoločné vlastnosti a delia sa na zasiahnutelné a nezasiahnutelné strelou. Rovnaké vlastnosti je možné zastrešiť v spoločnom predkovi a rozdiely špecifikovať v potomkoch. V nasledujúcej kapitole vytvoríte prvok bojiska, ktorý bude predkom všetkých aktorov hry.

ÚLOHA 6.1

Vytvorte spoločného predka pre prvky bojiska, všetkých aktuálnych aktorov. Spoločnému predkovi nastavte atribút **pocetZivotov** predstavujúci počet životov, ktorý bude dostupný jeho potomkom a inicializujte ho pomocou vstupného parametra konštruktora. Všetkým prvkom bojiska tento atribút nastavte pomocou zavolania konštruktora predka, čiže príkazu **super**.

Vymažte tanku a múru atribút reprezentujúci počet životov. Tento atribút už budú dediť od svojho predka, triedy **PrvokBojiska**.

ÚLOHA 6.2

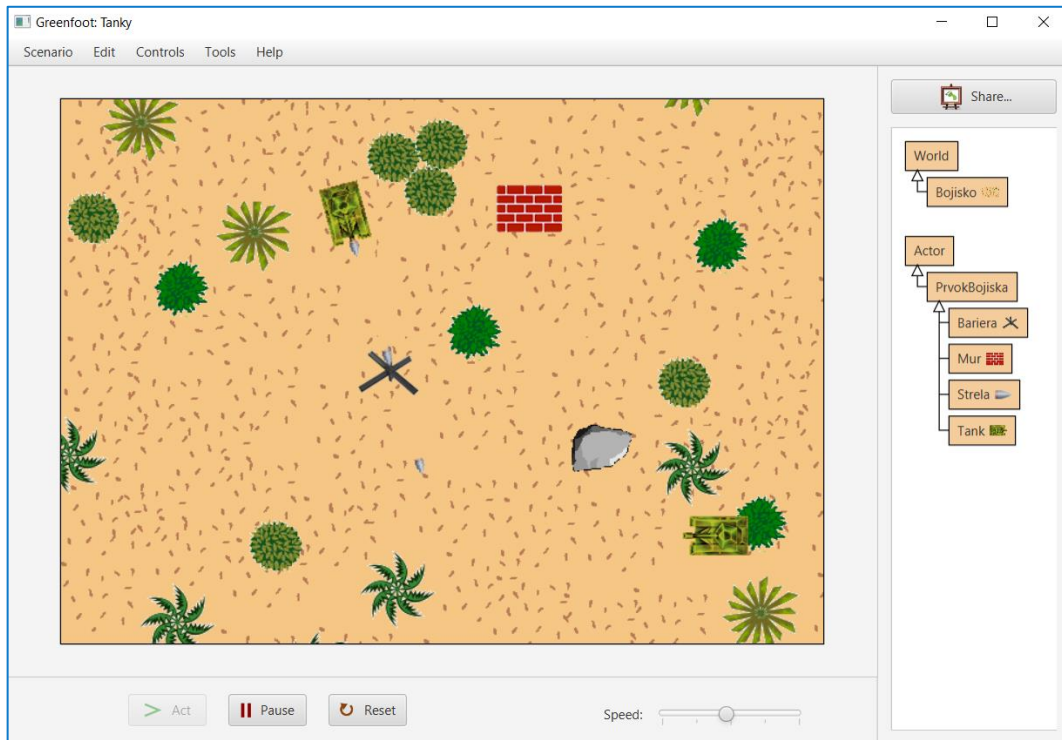
V triede **PrvokBojiska** vytvorte metódu s návratovou hodnotou, ktorá bude predstavovať to, či je daný prvok bojiska zasiahnutelný strelou. Táto metóda bude vracať implicitne hodnotu **true**. V prvkoch bojiska, ktoré nebudú zasiahnutelné strelou (zatiaľ len bariéra) prekryte túto metódu a zmeňte návratovú hodnotu na **false**.

ÚLOHA 6.3

Vašou ďalšou úlohou je vytvoriť metódu, v ktorej zadefinujete čo sa stane v prípade, že prvok bojiska bude zasiahnutý strelou. Všimnite si, že také správanie ste už naprogramovali v triede **Tank** v metóde **zasah**. Keďže takéto správanie by mal mať každý prvok bojiska presuňte metódu **zasah** do predka, čím dosiahnete to, že každý prvok bojiska sa pri zásahu strelou odstráni z bojiska.

ÚLOHA 6.4

V hre budete mať rôzne prekážky a bonusy, ktoré budú zasiahnutelné strelou. Preto upravte strelu tak, aby mohla zasiahnúť ktorýkoľvek prvok bojiska, ktorý je zasiahnutelný. Po tejto úprave je tank schopný odstreliť prvok bojiska múr.



Obrázok 16: Nezasiahnuteľnosť bariéry

7 ÚROVNE

Doteraz hra skončila zničením jedného z tankov. V tejto kapitole si vytvoríte rôzne úrovne, medzi ktorými sa bude hra striedať.

Pre pokračovanie si potrebujeme zdefinovať nový pojem, abstraktné metódy. Ich využitie si vysvetlíme na príklade, ktorým môžu byť opäť hudobníci v orchestri. Hudobníci hrajú na rôzne hudobné nástroje. Ich spoločnou vlastnosťou však je, že vedia hrať. Preto predok všetkých hudobníkov - hudobník, by obsahoval abstraktnú metódu hraj, ktorá nemá telo metódy, pretože hudobník nevie špecifikovať, na aký nástroj bude hrať. Jeho potomkovia, napr. trubkári, huslisti a pod., by túto metódu prekryli a hrali v nej na konkrétne hudobné nástroje.

Pre abstraktné metódy platí niekoľko pravidiel. Sú to metódy, ktoré sa môžu vyskytovať iba v abstraktnej triede. V neabstraktných potomkoch, ktoré dedia od takejto abstraktnej triedy, musia byť tieto metódy prekryté. Predok neobsahuje telo metódy, iba hlavičku. Kód abstraktnej metódy je implementovaný v potomkoch.

ZAPAMÄTAJTE SI!

```
[modif. prístupu] abstract class NazovTriedy {  
    [modif. prístupu] abstract [návratový typ] nazovMetody(typ parameter);  
}
```

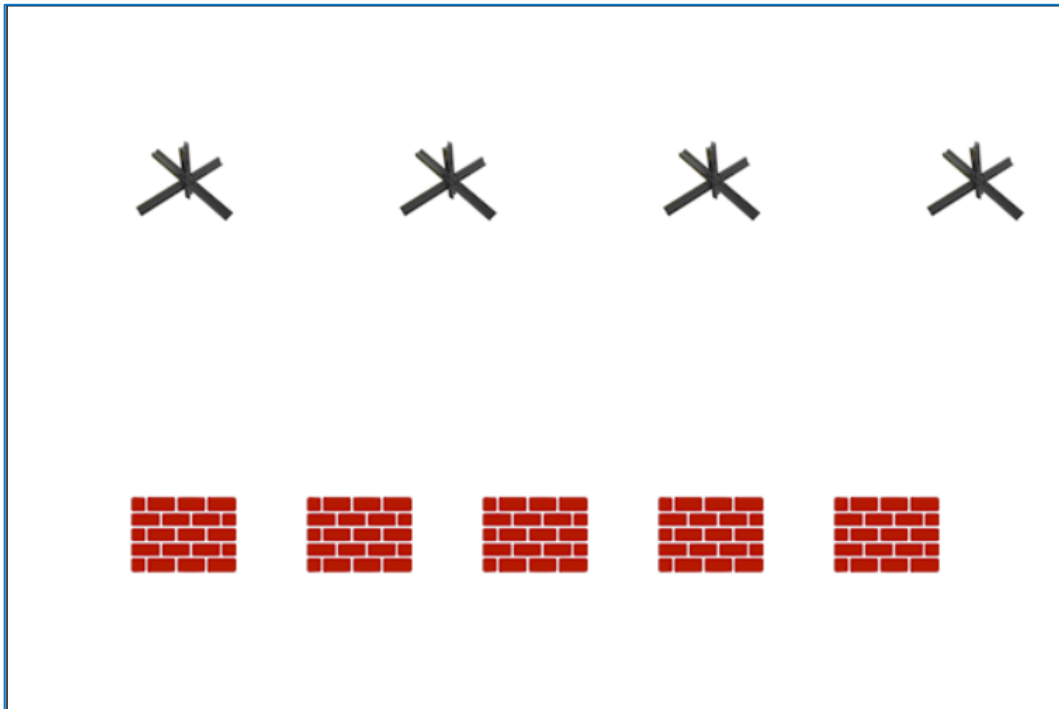
ÚLOHA 7.1

Prvým krokom tejto kapitoly je nastavenie šírky a výšky bojiska prostredníctvom vstupných parametrov konštruktora triedy **Bojisko**. Na základe šírky a výšky nastavte pozíciu druhého tanku na opačnú stranu bojiska vzhľadom na prvý tank. Kód na vytváranie prekážok v bojisku vymažte. Nahraďte ho chránenými (**protected**) metódami, ktoré budú využívať potomkovia na pridanie bariéry s názvom **pridajBarieru** a múru s názvom **pridajMur** do bojiska na pozície zadané cez vstupné parametre.

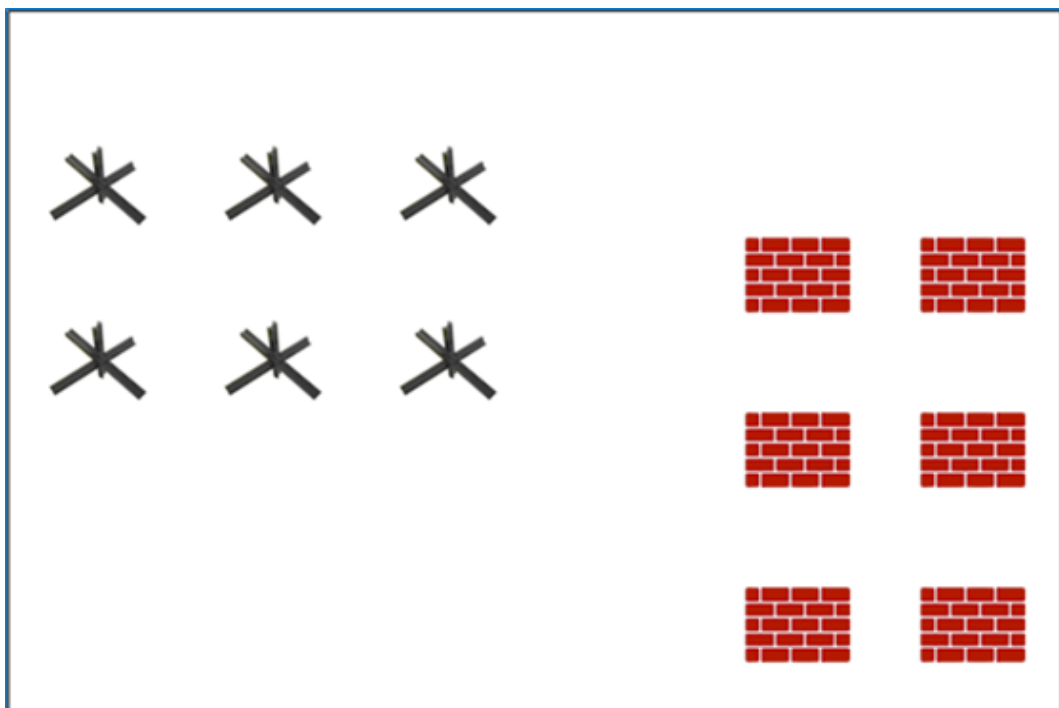
Je to potrebné pretože v ďalších krokoch budete vytvárať jednotlivé úrovne, v ktorých budú prekážky rozmiestnené na rôznych pozíciách.

ÚLOHA 7.2

Vytvorte chránené metódy **vytvorRiadokBarier**, **vytvorRiadokMurov**, **vytvorObdlnikBarier** a **vytvorObdlnikMurov**, v ktorých použijete cyklus s pevným počtom opakovaní, ktorý rozmiestni zadaný počet bariér a múrov. Prvý typ metódy dostane pozíciu prvej prekážky, veľkosť medzery medzi prekážkami a počet prekážok ako vstupné parametre a rozmiestni ich do riadku. Druhý typ metódy bude fungovať rovnako ale rozmiestni prekážky do obdĺžnika, pričom použije metódu na rozmiestnenie prekážok do riadku.



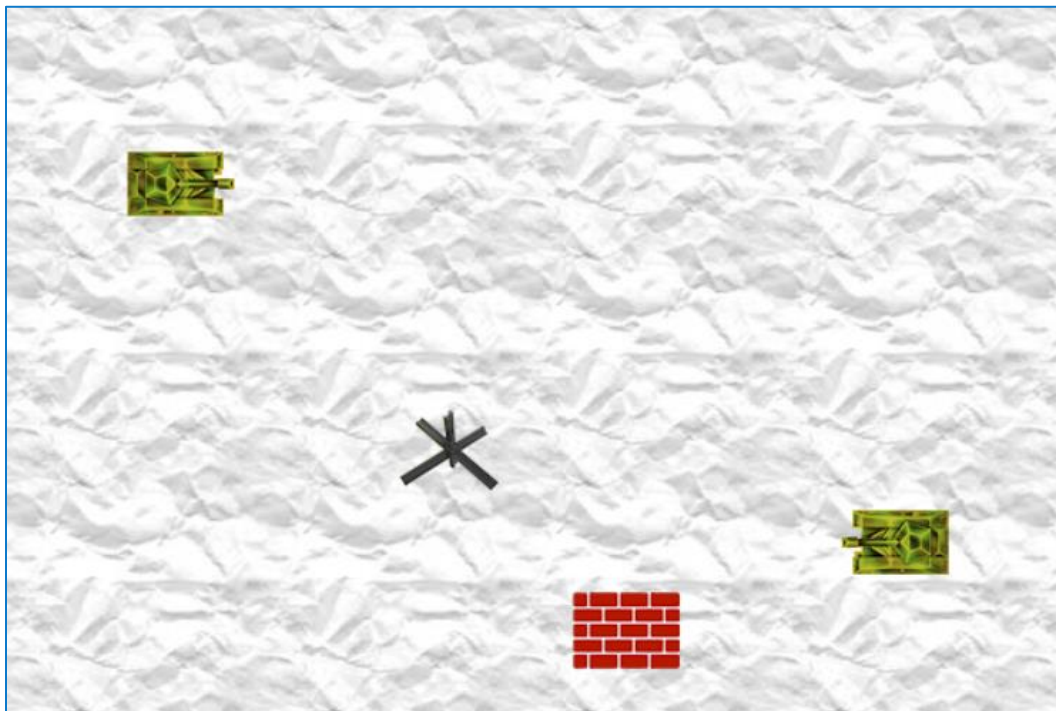
Obrázok 17: Umiestnenie prekážok do riadku



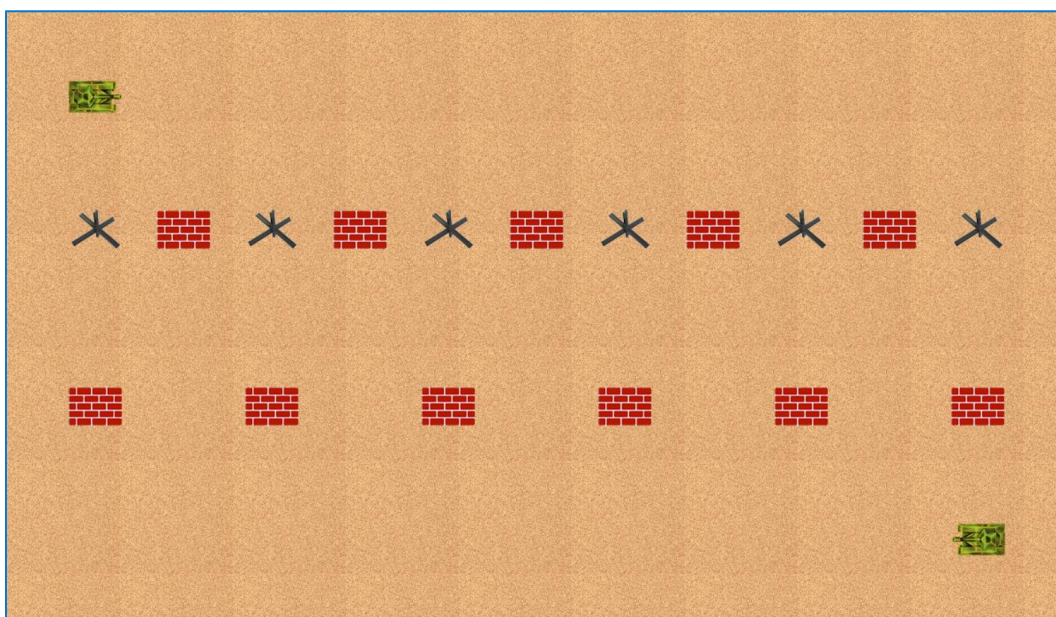
Obrázok 18: Umiestnenie prekážok do obdĺžnika

ÚLOHA 7.3

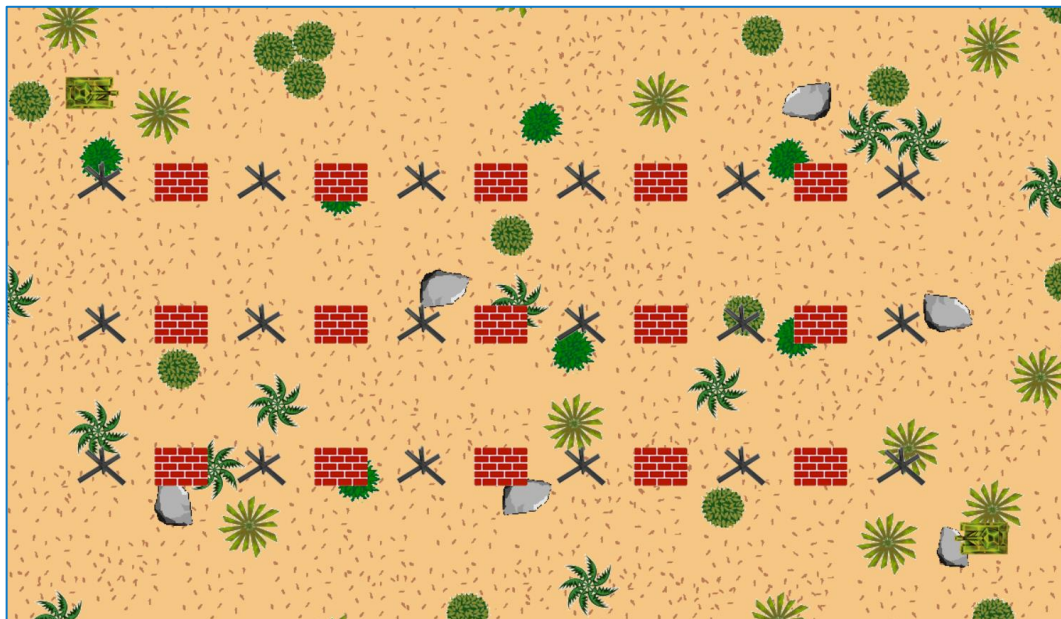
Vytvorte tri typy úrovní, ktoré budú potomkami už vytvoreného bojiska. Použite rôzne kombinácie metód na rozmiestnenie prekážok. Môžete použiť rôzne pozadia bojísk pre jednotlivé úrovne. Po naprogramovaní rôznych typov bojísk si môžete vytvoriť a zobrazíť bojiská pomocou príkazu z kontextového menu **new NazovBojiska()**. Inšpiráciu nájdete na nasledujúcich obrázkoch.



Obrázok 19: Ukážka najľahšej úrovne



Obrázok 20: Ukážka stredne ťažkej úrovne



Obrázok 21: Ukážka najťažšej úrovne

V ďalšej časti budete využívať na ukladanie prvkov pole, ktoré si najprv vysvetlíme.

Údaje v počítači sa dajú ukladať rôznymi spôsobmi. Jedným z nich je ukladanie prostredníctvom poľa. Pole si môžete predstaviť napríklad ako skriňu so zásuvkami, kde v každej zásuvke môže byť vec takého typu, akého typu je pole – napríklad číslo, tričko, tank. Pole umožňuje odložiť si niečo do zásuvky (do jednej zásuvky/poličky jednu vec) a prípadne sa pozrieť, čo je tam uložené. Veľkosť poľa je obmedzená a nedá sa meniť počas behu programu. Definuje sa pri vytvorení vymenovaním prvkov poľa alebo zadaním veľkosti. Do poľa sa vždy ukladajú prvky rovnakého typu. Stav poľa je možné v nástroji Greenfoot zobrazíť pomocou voľby **Inspect** nad objektom, ktorý pole obsahuje, a následným dvojklikom na vybrané pole. Takto zobrazený stav poľa obsahuje názov, typ a veľkosť poľa a prvky na jednotlivých pozíciách, ako môžete vidieť na obrázkoch 22 a 23.

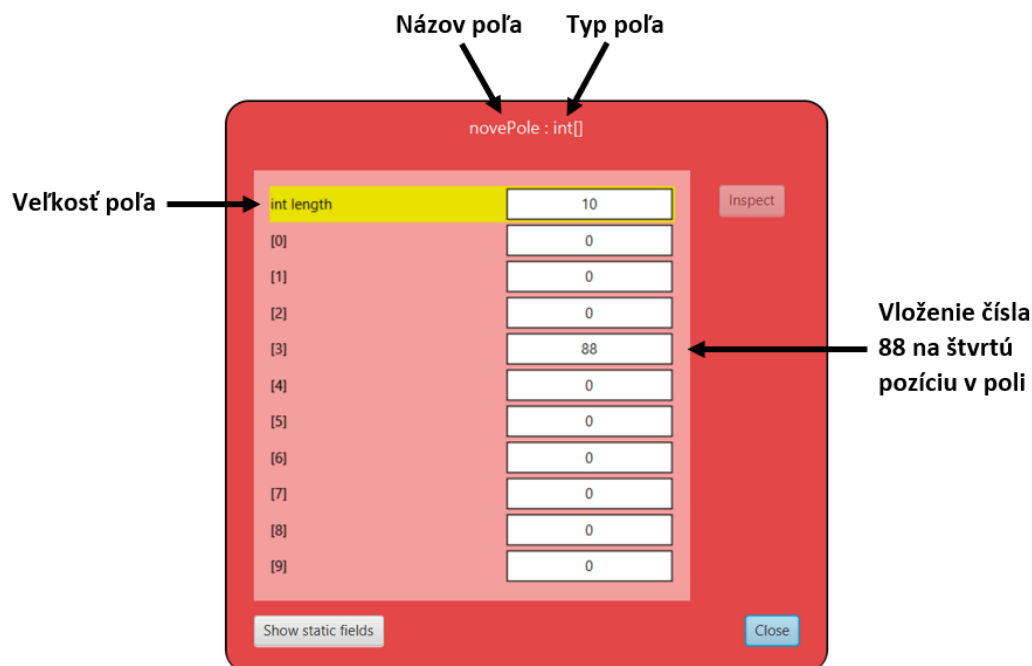
Pole sa dá definovať nasledujúcimi spôsobmi:

1) Zadaním veľkosti poľa.

```
Deklarácia: int[] nazovPola;
```

```
Inicializácia: nazovPola = new int[10];
```

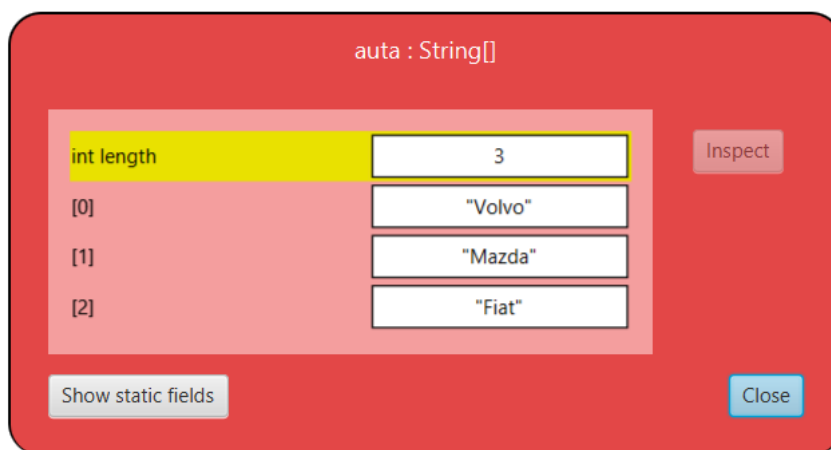
```
Vloženie hodnoty: nazovPola[3] = 88;
```



Obrázok 22: Zobrazenie poľa čísel cez Inspect

2) Vymenovaním prvkov poľa.

Deklarácia spojená s inicializáciou: `String[] auta = {"Volvo", "Mazda", "Fiat"};`



Obrázok 23: Zobrazenie poľa reťazcov cez Inspect

Jednotlivé prvky poľa sú prístupné cez index – pozíciu, pričom index začína **od nuly**. Napríklad, tretí prvok z poľa môžete získať pomocou kódu:

```
int tretiPrvok = nazovPola[2];
```

Trieda je objekt, takže môže mať svoje atribúty a metódy. Ak chceme, aby atribúty alebo metódy patrili triede, tak ich označíme kľúčovým slovom **static**. Atribúty a metódy triedy sú spoločné pre

všetky inštancie danej triedy. K atribútom triedy je možné prísúpiť pomocou príkazu `NazovTriedy.nazovAtributu` a analogicky k metódam `NazovTriedy.nazovMetody()`.

Atribúty a metódy môžu mať aj modifikátor **final**, ktorý určuje, že atribút je možné inicializovať iba raz a jeho hodnotu nie je možné ďalej počas behu programu meniť.

Konvencia jazyka Java je taká, že identifikátory atribútov, ktoré obsahujú modifikátory **static** a **final** sú písané veľkými písmenami (napr. `BOJISKA`) a prípadne sú slová oddelené podčiarkovníkom (`_`).

Príkladom **static** a **final** vlastností sú napríklad:

```
private static final String PRVY_DEN_V_TYZDNI = „PONDELOK“;
private static final String DRUHY_DEN_V_TYZDNI = „UTOROK“;
private static final String TRETI_DEN_V_TYZDNI = „STREDA“;
.
.
.
```

ÚLOHA 7.4

V tomto kroku si naprogramujete to, aby bola po zničení jedného z tankov vytvorená ďalšia úroveň. Budete na to používať pole jednotlivých úrovní, ktoré si uložíte ako atribút triedy (**static**) **Bojisko**, pretože trieda **Bojisko** je zodpovedná za organizáciu úrovní. Toto pole bude inicializované iba raz a nebude sa dať počas hry meniť. Pole zadefinujte vymenovaním jeho prvkov, teda ho naplňte novými inštanciami jednotlivých typov bojísk.

V jednotlivých úrovniach hry môžu hráči zničiť prekážky alebo inak zmeniť základný stav bojiska. Preto pri prechode na novú úroveň pokazené bojisko zahodíme a vytvoríme „klon“ nasledujúcej úrovne. Týmto spôsobom nie je potrebné zisťovať, ktoré prekážky boli zničené a opätovne ich obnovovať.

Jednotlivé typy bojísk budú prototypy, ktoré budú predstavovať jednotlivé úrovne, na ktorých sa bude odohrávať hra *Tanky*. Vždy sa bude vedieť, ktoré je aktuálne bojisko. Pre vytvorenie nového typu bojiska sa využije prototyp na klonovanie. Aby nebolo potrebné zisťovať akého typu je trieda aktuálneho bojiska, využijeme abstraktnú metódu **clone**, v ktorej každému potomkovi zadefinujeme aby vrátil svoju inštanciu. Názov metódy **clone** je zaužívaný identifikátor pre klonovaciu metódu.

ÚLOHA 7.5

Ďalej bude bojisko potrebovať nový atribút triedy (**static**), ktorý bude predstavovať aktuálnu úroveň hry. Pre striedanie úrovní je potrebné po skončení úrovne vytvoriť kópiu nasledujúceho typu bojiska. Preto v triede **Bojisko** vytvorte abstraktnú metódu **clone** s návratovou hodnotou typu **Bojisko**. Vo všetkých typoch bojísk prekryte túto metódu tak, aby vytvorila novú inštanciu daného bojiska a vrátila ju.

ÚLOHA 7.6

Do triedy **Bojisko** pridajte novú verejnú (**public**) metódu s názvom **ukoncenieUrovne** bez návratovej hodnoty na ukončenie aktuálnej úrovne. Vstupným parametrom tejto metódy bude zničený tank. Vymažte v nej zničený tank zo sveta a zastavte hru.

Ďalej si zdefinujete akciu po zasiahnutí tanku strelou. V triede **Tank** prekryte metódu **zasah** triedy **PrvokBojiska**. Odčítajte v nej jeden život tanku. Po minutí všetkých životov tanku (teda po jeho zničení) oznámte bojisku, že je možné prejsť na ďalšiu úroveň. To urobíte tak, že získate inštanciu bojiska pomocou príkazu **this.getWorldOfType(Bojisko.class)** a zavoláte metódu bojiska na ukončenie úrovne, kde ako vstupný parameter pošlite tento tank.

ÚLOHA 7.7

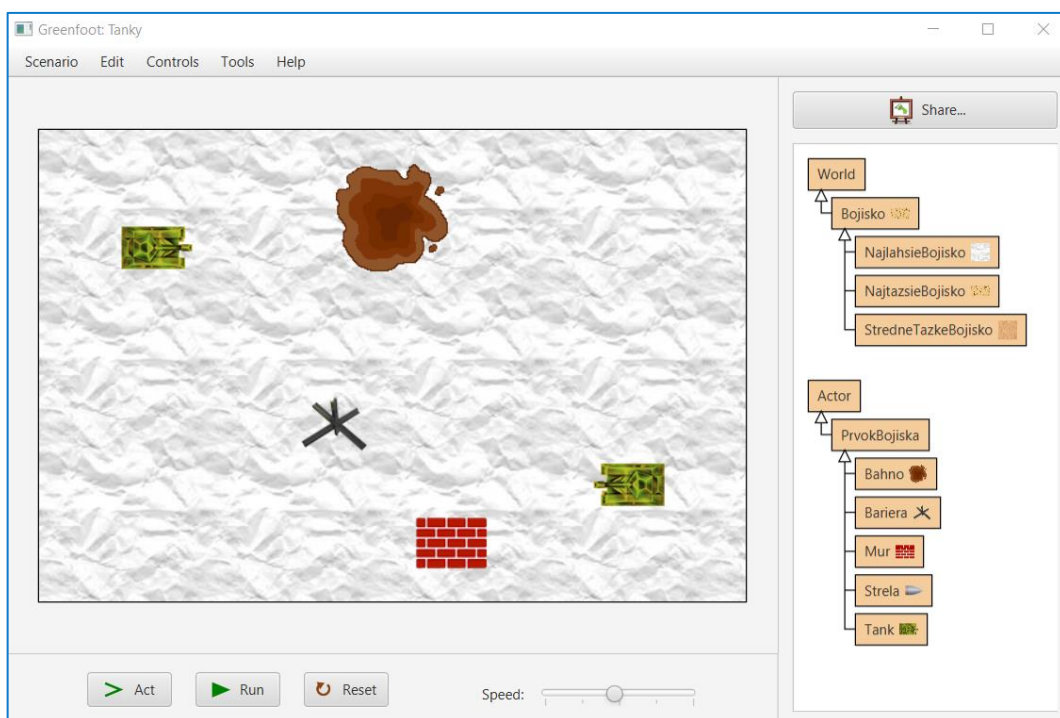
Po odstránení tanku z bojiska nastavte, aby sa hra na chvíľu zastavila. Následne vyberte ďalší typ bojiska z poľa bojísk, vytvorte ho pomocou príkazu **setWorld** a metódy **clone**. Zapnite opäť beh hry.

8 BAHNO A SKÓRE HRÁČOV

V ôsmej kapitole si vyskúšate cyklus s podmienkou na začiatku. Vytvoríte nový prvok hry bahno. Tank sa po prejení cez bahno spomalí a zašpiní. Aby ste mohli vidieť, ktorý z tankov má vyššie skóre, budete zobrazovať počet vyhraných úrovní na bojisku.

ÚLOHA 8.1

Vytvorte nový prvok bojiska, bahno. Keď tank prejde cez bahno, tak sa mu zníži rýchlosť a zašpiní sa na určitú dobu. V tejto úlohe priradte bahnu obrázok, spravte ho nezasiahnutelné strelou, nastavte mu jeden život a umiestnite ho na bojisko.



Obrázok 24: Umiestnenie bahna do bojiska

ÚLOHA 8.2

Pridajte tanku atribúty reprezentujúce pôvodnú rýchlosť a trvanie spomalenia jeho rýchlosti. Hru *Tanky* môžete rozšíriť o ďalšie prekážky, ktoré môžu spomaľovať tank, preto tento atribút pomenujte všeobecne **trvanieZmenyRychlosti**. Inicializujte atribút trvania zmeny rýchlosti tanku na hodnotu nula a pôvodnú rýchlosť na hodnotu atribútu reprezentujúceho rýchlosť tanku.

ÚLOHA 8.3

Vytvorte v triede **Tank** metódu, ktorá spomalí pohyb tanku pri dotyku s bahnom. Pomenujte túto metódu ako **spomaSa** a definujte jej vstupné parametre typu **int** - **novaRychlostTanku** a **trvanieSpomalenia**. Táto metóda zmení tanku jeho rýchlosť pomocou vstupného parametra **novaRychlostTanku** a nastaví atribút **trvanieZmenyRychlosti** vstupným parametrom **trvanieSpomalenia**.

ÚLOHA 8.4

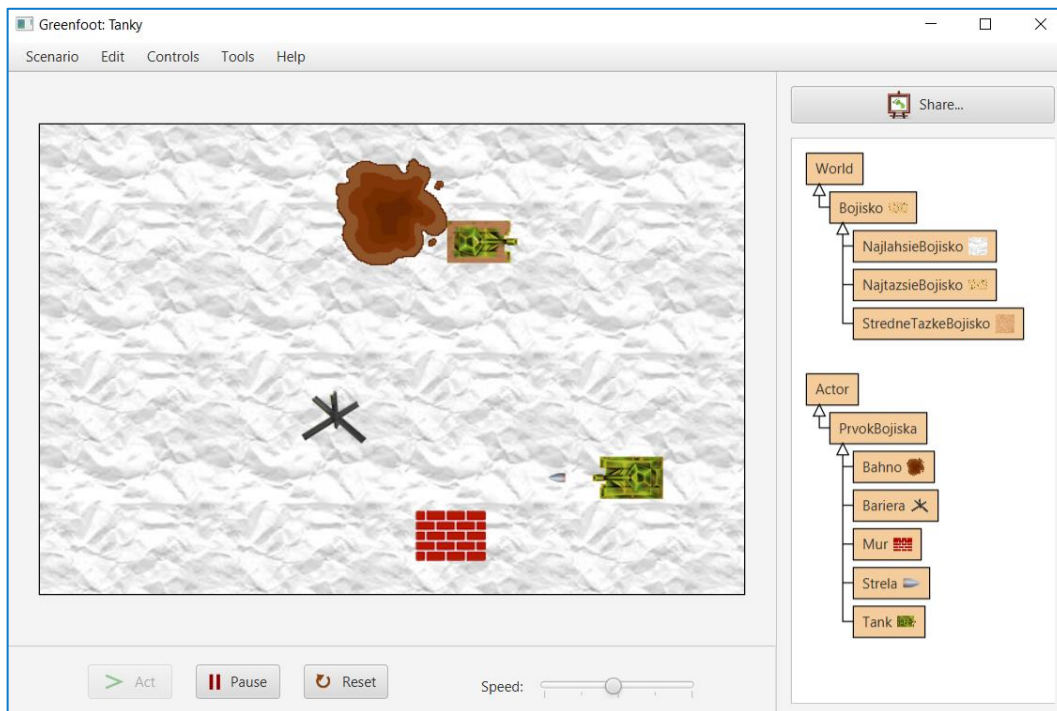
V triede **Bahno** v metóde **act** zistíte všetky tanky, ktoré sa dotýkajú bahna a ukladajte ich do zoznamu tankov. Metódu **spomaSa**, ktorú ste vytvorili v predchádzajúcej úlohe, zavolajte v metóde **act** triedy **Bahno** tak, aby sa vykonala pre každý tank v zozname tankov dotýkajúcich sa bahna.

ÚLOHA 8.5

V metóde **act** triedy **Tank** znižujte hodnotu trvania spomalenia bahnom, ak je väčšie ako nula. Ak trvanie vypršalo, nastavte tanku jeho pôvodnú rýchlosť.

ÚLOHA 8.6

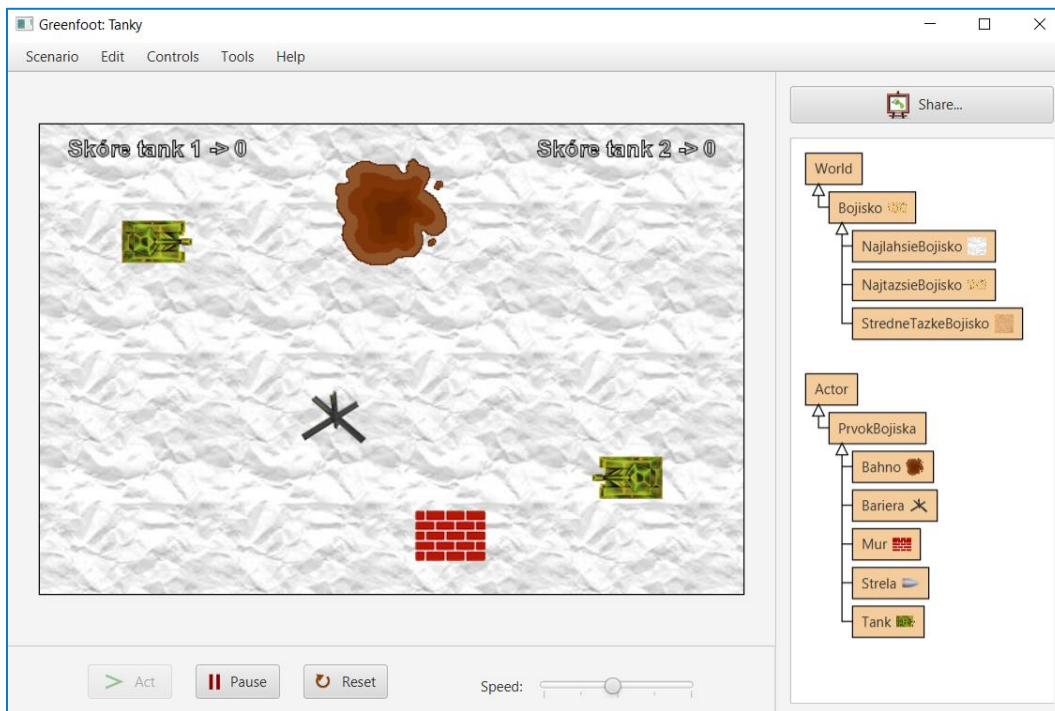
Pripravte si štyri obrázky zašpineného tanku. Skúste nájsť spôsob ako postupne meniť obrázok tanku tak, aby sa zdalo, že je čím ďalej, tým menej špinavý. Po implementácii tejto vlastnosti bude nový prvok bojiska hotový.



Obrázok 25: Zašpinenie tanku po prejení cez bahno

ÚLOHA 8.7

V nasledujúcich úlohách pridáte skóre hráčov do hry *Tanky*. Skóre sa bude pripočítavať hráčovi, ktorý vyhrá úroveň. V triede **Bojisko** vytvorte dva nové atribúty tak, aby sa ich hodnoty nestratili pri vytvorení nového bojiska. Tieto atribúty budú predstavovať skóre jednotlivých tankov a inicializujete ich už pri ich deklarácii. Aktuálne skóre zobrazujte na bojisku pomocou metódy sveta **showText** podobne, ako je to zobrazené na nasledujúcom obrázku. Po zničení jedného z tankov navýšte jeho súperovi skóre.



Obrázok 26: Zobrazenie skóre tankov

9 BONUSY

V ďalšej kapitole vytvoríte bonus náboje a bonus rýchlosť, ktoré budú môcť tanky získať. Tieto bonusy budú mať rôzne účinky na tank, ktorý ich zodvihne.

ÚLOHA 9.1

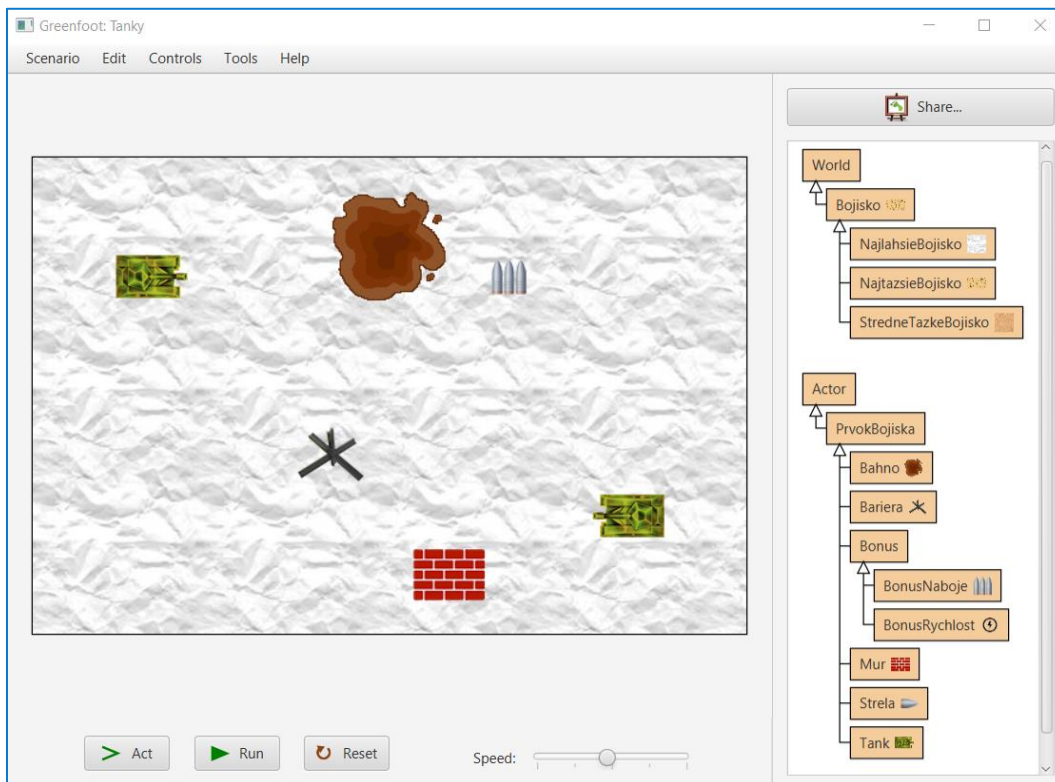
Vytvorte predka pre všetky bonusy, ktoré budú v hre. Nastavte, aby bol nezasiahnutelný strelou, aby mal jeden život a aby bol potomkom triedy **PrvokBojiska**. Pridajte predkovi všetkých bonusov abstraktnú metódu na aplikovanie bonusu so vstupným parametrom typu **Tank**.

ÚLOHA 9.2

Pri dotyku tanku s bonusom vymažte bonus zo sveta a aplikujte jeho účinok na tank.

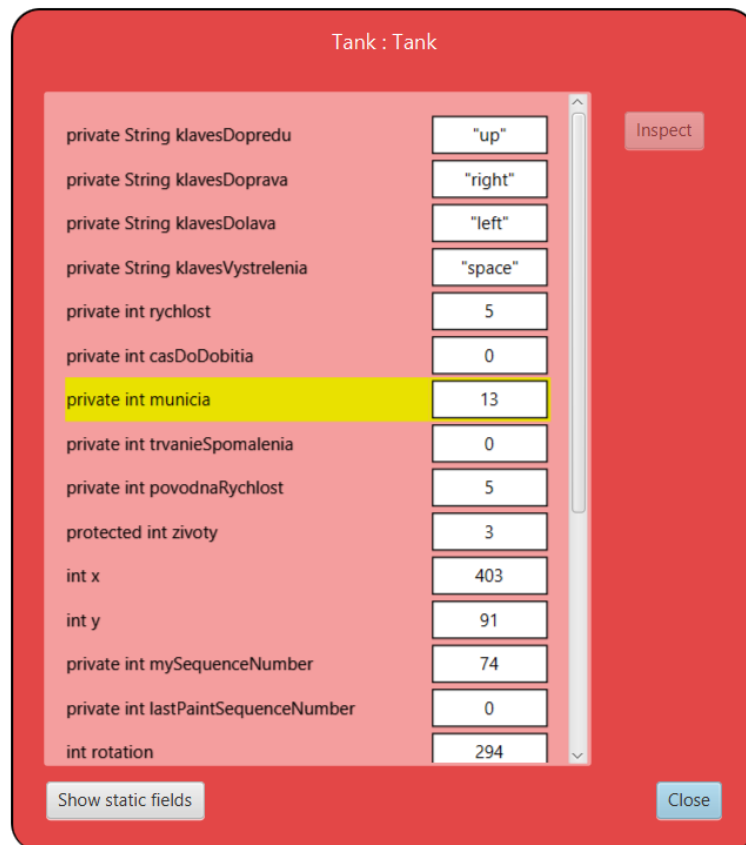
ÚLOHA 9.3

Vytvorte nový bonus náboje, po získaní ktorého sa tanku zvýši množstvo jeho munície. Pridajte tanku metódu **pridajNaboje**, ktorá zvýši jeho muníciu. Prekryte metódu na aplikovanie bonusu, v ktorej pridáte tanku náboje. Následne si v bojisku vytvorte inštanciu tohto bonusu a vyskúšajte implementované riešenie. Bojisko spolu s bonusom náboje bude vyzerať podobne ako na nasledujúcom obrázku.



Obrázok 27: Bojisko s bonusom náboje

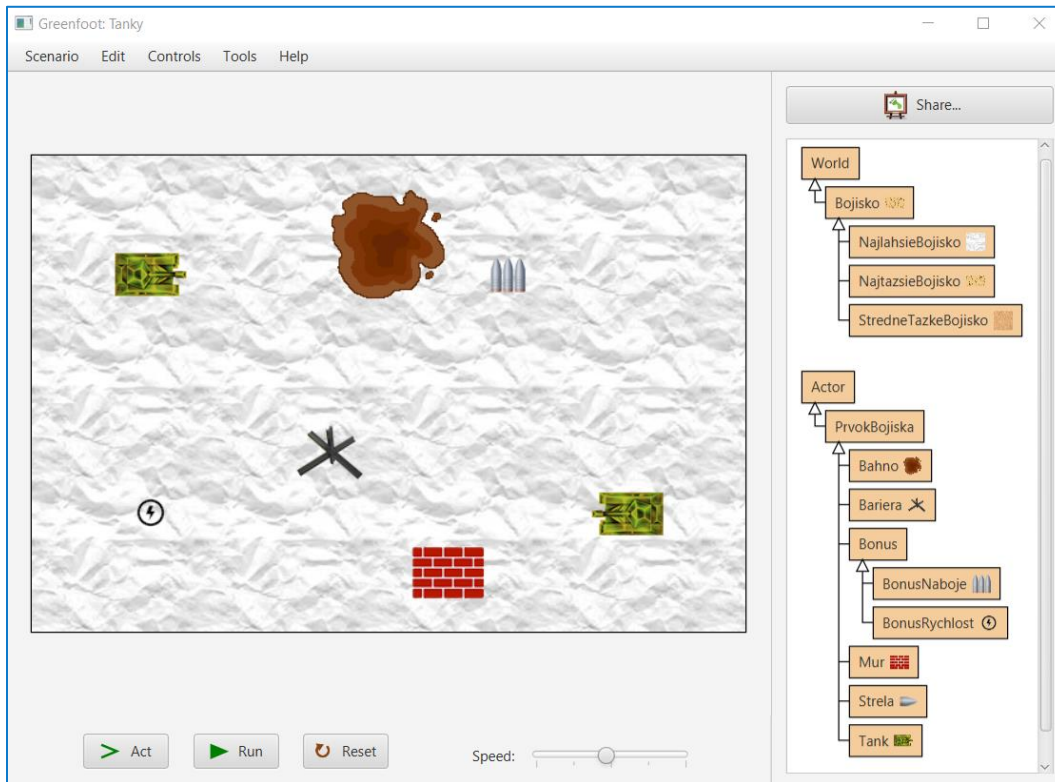
Správnosť daného riešenia si dokážete overiť pozastavením hry a kliknutím na **Inspect** tanku, kde uvidíte, že počet nábojov sa zvýši podobne ako na nasledujúcom obrázku.



Obrázok 28: Vnútorý stav tanku

ÚLOHA 9.4

Vytvorte nový bonus rýchlosť. Keď tank získa tento bonus, jeho rýchlosť sa zvýši. Doplňte tanku metódu **pridajRychlost** na zvýšenie jeho rýchlosti. Podobne ako v predchádzajúcej úlohe, prekryte metódu na aplikovanie bonusu, v ktorej zvýšite tanku jeho rýchlosť. V bojisku vytvorte inštanciu bonusu rýchlosť a vyskúšajte jeho aplikovanie. Bojisko s bonusom rýchlosť bude vyzeráť nasledujúco.



Obrázok 29: Bojisko s bonusom rýchlosť

10 GENEROVANIE BONUSOV

Ďalšou časťou našej hry bude automatické generovanie bonusov na bojisku, ktoré si v nasledujúcej kapitole implementujete.

ÚLOHA 10.1

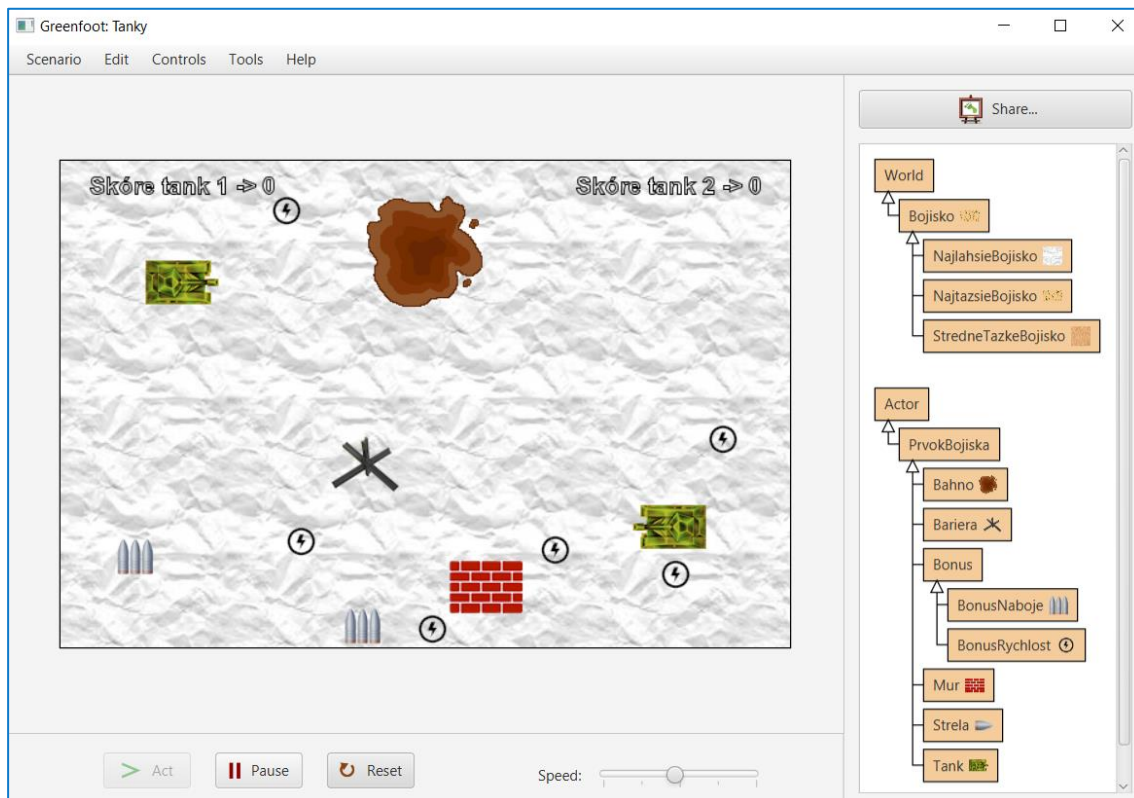
Vytvorte abstraktnú metódu v triede **Bonus**, ktorá bude klonovať daný bonus. V jednotlivých bonusoch hry prekryte túto metódu tak, aby sa po jej zavolaní vytvoril klon danej triedy.

ÚLOHA 10.2

Pre klonovanie je potrebné si niekde uložiť vzory bonusov, ktoré budete klonovať. Uložte vzory bonusov do poľa v triede **Bonus**, ktoré sa nebude môcť meniť (**final**), bude patriť danej triede (**static**) a bude spoločné pre všetky jej inštancie. Vytvorte nový atribút **casDoBonusu** predstavujúci čas do vygenerovania ďalšieho bonusu, ktorý sa bude postupne odrátavať. Nezabudnite vymazať kód na vytvorenie bonusov v konštruktoze, pretože bonusy sa už budú vytvárať automaticky.

ÚLOHA 10.3

Z poľa bonusov náhodne vyberte jeden z bonusov a vytvorte jeho klon. Následne náhodne vygenerujte súradnice klonu, tak aby bol umiestnený v bojisku a pridajte ho doň. Ďalej vygenerujte náhodný čas do ďalšieho bonusu.



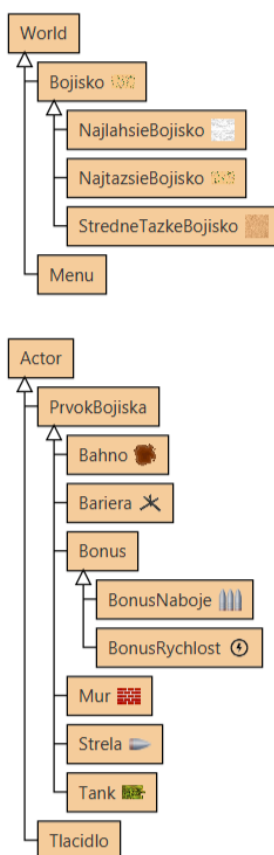
Obrázok 30: Automatické generovanie bonusov

11 MENU

Ako každá správna hra aj hra *Tanky* by mala mať vytvorené menu a ukončenie hry. V nasledujúcej kapitole si vytvoríte úvodné menu. Po skončení všetkých typov bojísk zobrazíte ukončenie hry, kde zobrazíte skóre oboch hráčov.

ÚLOHA 11.1

Vašou prvou úlohou bude vytvoriť predka **Menu** pre začiatkové a koncové menu, ktorý bude potomkom triedy **World**. Keďže trieda **Menu** je potomkom triedy **World** potrebuje ako vstupné parametre informácie o veľkosti sveta. Spravte tiež predka pre tlačidlá, ktoré sa budú v menu vyskytovať, ako potomka triedy **Actor**.



Obrázok 31: Pozícia triedy **Menu** a **TlacidloHrat** v hierarchii tried

ÚLOHA 11.2

Vytvorte chránenú (**protected**) abstraktnú metódu **klik** v triede **Tlacidlo**, ktorá bude vykonávať rôzne akcie tlačidiel. Abstraktnú metódu **klik** zavolajte v metóde **act** triedy **Tlacidlo** po jeho kliknutí.

ÚLOHA 11.3

V triede **Bojisko** vytvorte metódu triedy (**static**) s názvom **zacatieUrovne**, ktorá bude mať ako vstupný parameter číslo úrovne. V tejto metóde nastavte atribút **indexBojiska** triedy **Bojisko** na hodnotu vstupného parametra. Z poľa bojísk vyťahnite bojisko na pozícií hodnoty aktuálnej úrovne a vytvorte jeho klon. Tento klon nastavte ako aktuálny svet pomocou metódy **Greenfoot.setWorld** a spustite hru pomocou metódy **Greenfoot.start**.

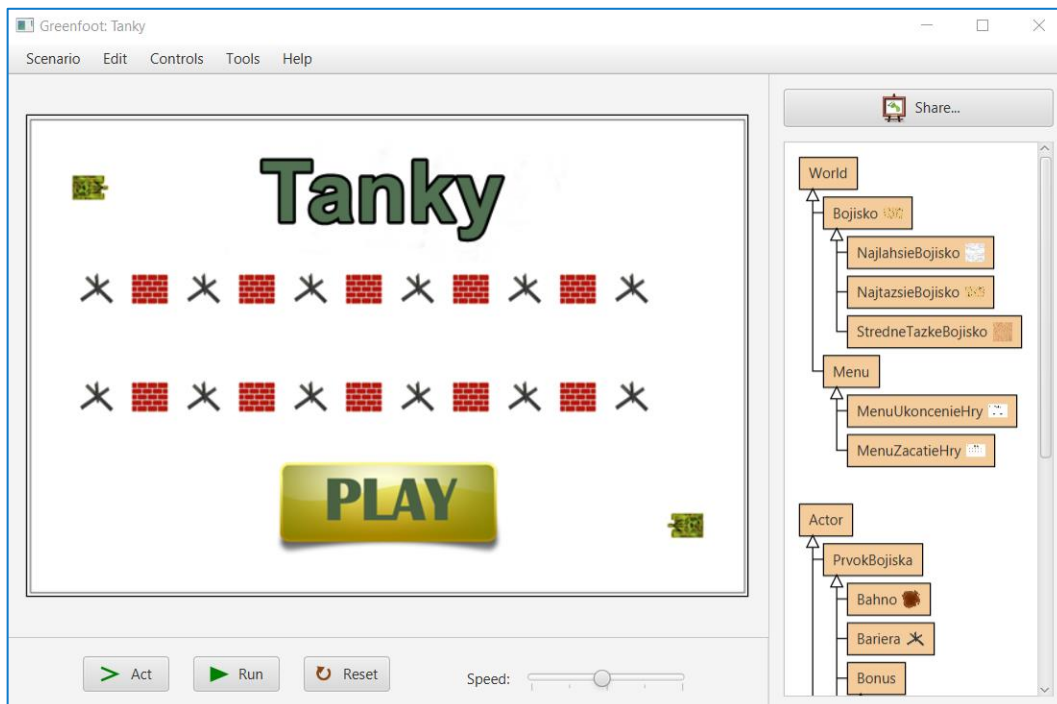
Vytvorte novú triedu **TlacidloHrat**, ktorá bude predstavovať tlačidlo na spustenie hry, čiže po jeho stlačení sa vytvorí najľahší typ bojiska. Nastavte tlačidlu prislúchajúci obrázok. V prekrytej metóde **klik** vytvorte nový najľahší typ bojiska pomocou metódy **zacatieUrovne** triedy **Bojisko**.

```
public class TlacidloHrat extends Tlacidlo
{
    /**
     * Akcia po kliknutí myšou na toto tlačidlo.
     */
    protected void klik()
    {
        Bojisko.zacatieUrovne(0);
    }
}
```

Vytvorte novú triedu **MenuZacatieHry**, ktorá je potomkom triedy **Menu** a pridajte do nej toto tlačidlo. Nastavte menu výstižný obrázok ako pozadie.

```
public MenuZacatieHry()
{
    super(600, 400);
    this.addObject(new TlacidloHrat(), 300, 320);
}
```

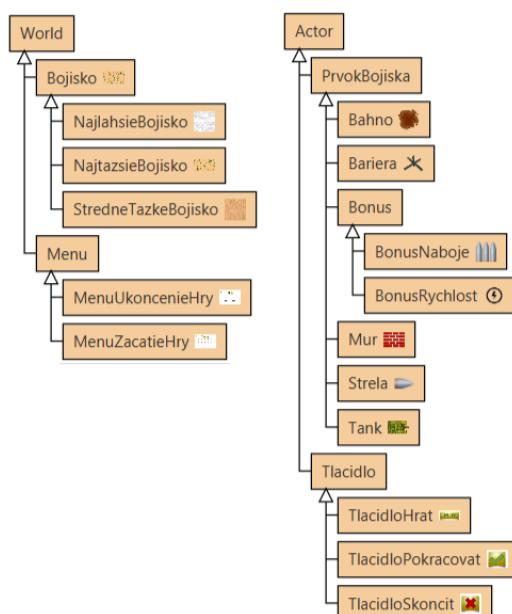
Je potrebné vytvoriť novú inštanciu triedy **MenuZacatieHry** kliknutím pravého tlačidla myši na **MenuZacatieHry** a následným kliknutím na príkaz **new Menu()**, aby sa ako prvé zobrazovalo menu začatia hry pri spustení hry *Tanky*.



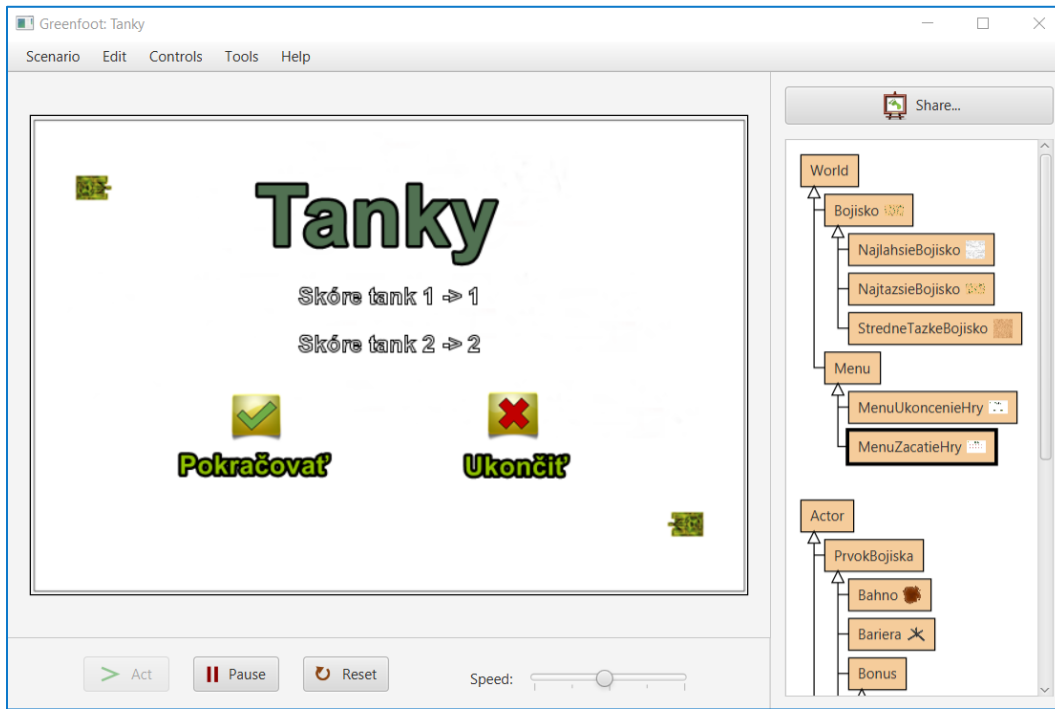
Obrázok 32: Menu začatie hry

ÚLOHA 11.4

Vytvorte nové tlačidlá ako potomkov triedy **Tlacidlo** – jedno na pokračovanie hry, po prejdení všetkých typov bojísk a druhé na vypnutie hry. Prekryte abstraktnú metódu **klik**, v ktorej zadefinujte akciu spojenú s kliknutím konkrétnych tlačidiel. Vytvorte záverečné menu ako potomka triedy **Menu**, na ktorom zobrazte tieto tlačidlá a vypíšte skóre obidvoch hráčov po prejdení všetkých úrovní.



Obrázok 33: Pozícia triedy **MenuUkoncenieHry**, **TlacidloPokracovat** a **TlacidloSkoncit** v hierarchii tried



Obrázok 34: Menu ukončenie hry

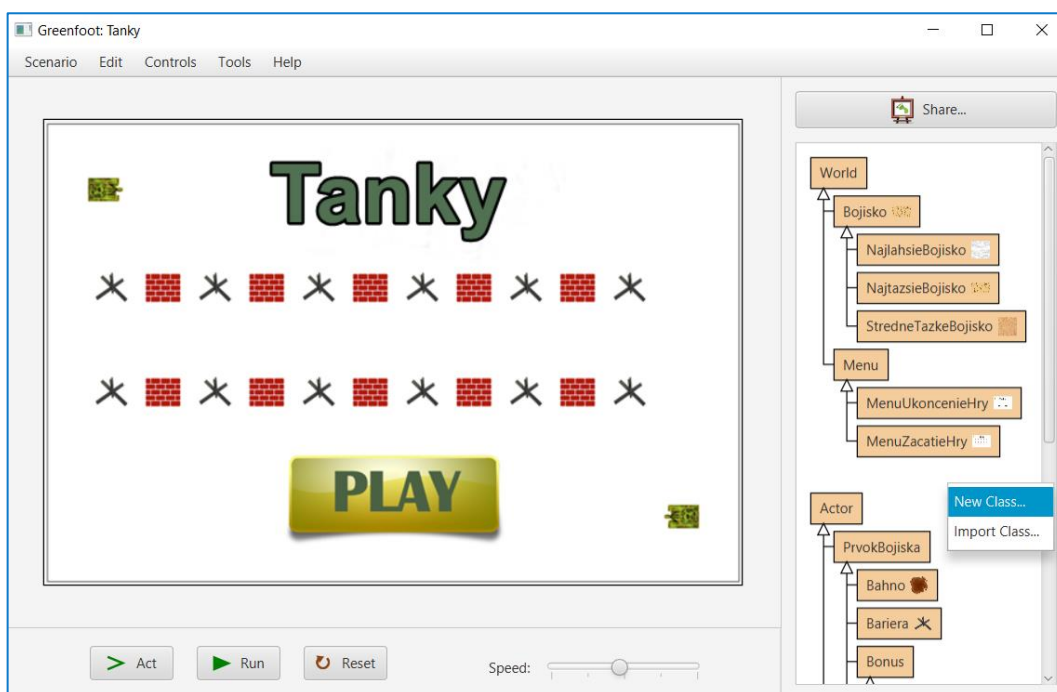
12 OVLÁDANIE

V kapitole Ovládanie osamostatníte ovládanie tankov do samostatnej triedy, čím spravíte triedu **Tank** prehľadnejšou a váš kód viac objektovo orientovaný. Podstatou objektovo orientovaného programovania sú objekty a ich vzájomná spolupráca. Ovládanie tak môžete definovať ako samostatnú triedu, ktorej inštancie budú mať určené rozličné klávesy na ovládanie tankov.

ÚLOHA 12.1

Prvou úlohou bude osamostatnenie ovládania tanku do samostatnej triedy, ktorá nebude potomkom tried **Actor** a **World**. Vytvorte preto novú samostatnú triedu **Ovladanie**.

Pre vytvorenie osamostatnenej triedy kliknite pravým tlačidlom do oblasti diagramov tried a vyberte z kontextového menu možnosť **New Class...**



Obrázok 35: Vytvorenie osamostatnenej triedy

ÚLOHA 12.2

Aby bolo opäť možné pre každý tank definovať iné klávesy pre pohyb je potrebné poslať názvy klávesov ako vstupné parametre do konštruktora triedy na ovládanie. Tieto parametre si uložte ako atribúty.

Pridajte metódu, v ktorej bude vykonávané ovládanie prvkov, kde vstupným parametrom bude inštancia triedy **Tank**. V tejto metóde po stlačení konkrétnych kláves zavolajte metódy tanku na vykonanie pohybu a na vystrelenie strely. Nezapudnite kontrolovať, či môže tank vystreliť. Čo je potrebné urobiť s metódami tanku, ktoré chceme použiť v triede na ovládanie?

13 NEPRIATEĽSKÁ VEŽIČKA

V ďalšej časti si vytvoríte nepriateľskú vežičku, ktorá sa bude točiť a v náhodných chvíľach bude strieľať. Vežu bude možné zničiť strelou.

ÚLOHA 13.1

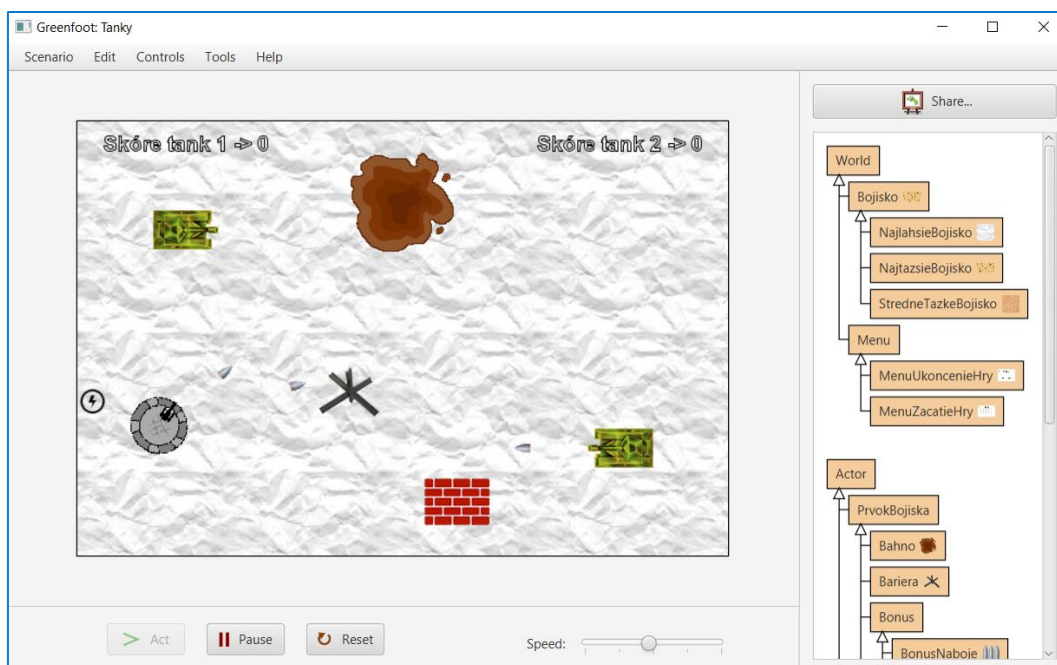
Ďalšou úlohou bude vytvorenie nového prvku bojiska, nepriateľskej veže, ktorá bude mať jeden život a bude zasiahnuteľná. Priradte jej obrázok a dopíšte kód, ktorý bude vežu točiť stále dokola proti smeru pohybu hodinových ručičiek.

ÚLOHA 13.2

Nepriateľská vežička bude strieľať v náhodných okamihoch. Dopíšte kód v triede **NepriatelaskaVeza** v metóde **act**, ktorý náhodne vytvorí strelu pred vežou.

ÚLOHA 13.3

Pridajte nepriateľskú vežu do stredne ťažkej a najťažšej úrovne.



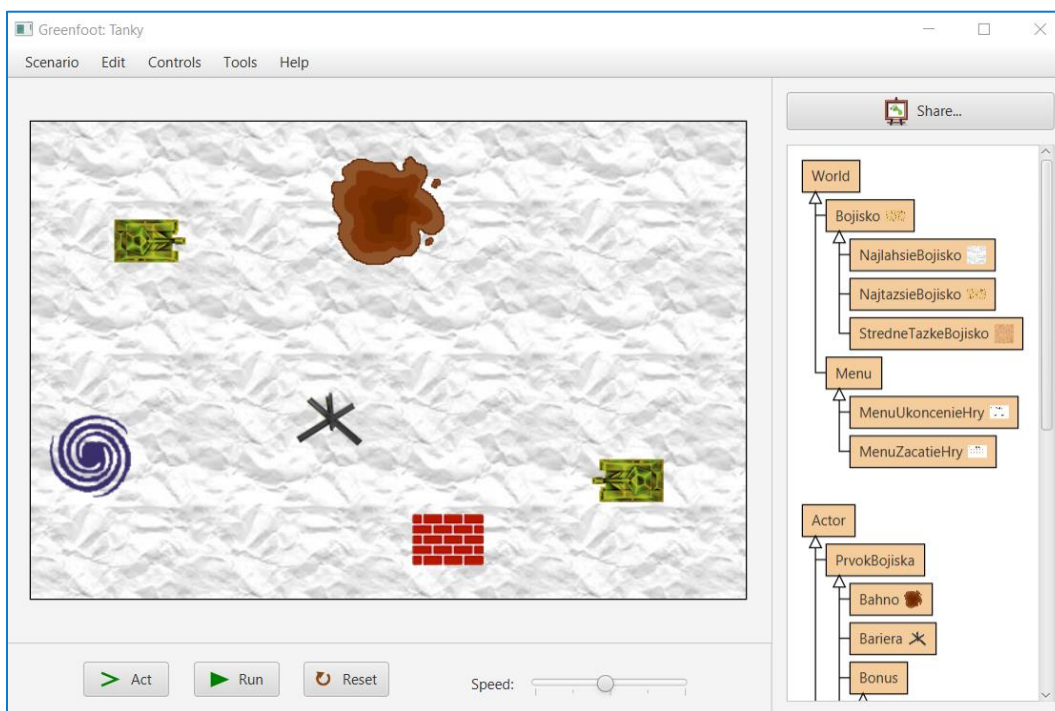
Obrázok 37: Nepriateľská veža v bojisku

14 ČIERNA DIERA

V ďalšej kapitole vytvoríte nový prvok bojiska, čiernu dieru, ktorá bude priťahovať tanky vo svojom okolí. Ak sa tank dostane do jej stredu, tak ho čierna diera zničí.

ÚLOHA 14.1

Vytvorte novú triedu čierna diera, ktorá je potomkom triedy **PrvokBojiska** a má jeden život. Priradte jej vhodný obrázok a pridajte ju na najťažšiu úroveň. Nastavte čiernej diere aby bola nezasiahnutelná strelou.



Obrázok 38: Čierna diera v bojisku

ÚLOHA 14.2

V metóde **act** vyhľadávajte tanky v okolí čiernej diery. Pomocou metódy **getObjectsInRange** zistíte, ktoré tanky sa nachádzajú v okolí 150 pixelov od čiernej diery a ukladajte ich do zoznamu tankov.

ÚLOHA 14.3

V triede **CiernaDiera** doplňte metódu, ktorej vstupným parametrom je tank, ktorý sa priblíži ku čiernej diere. Úlohou tejto metódy je posunúť tank maximálne o jeden bod smerom ku čiernej diere.

ÚLOHA 14.4

Predchádzajúcu metódu zavolajte v metóde na zistenie tankov v okolí čiernej diery tak, aby sa vykonala pre každý tank v zozname tankov. Následne vždy po zavolaní tejto metódy zistite, či sa nejaký tank nachádza v čiernej diere. Ak áno, tak zistite, či je v strede čiernej diery a v prípade, že áno, tak následne ho zničte.

INDEX OBRÁZKOV

Obrázok 1: Hra Tanky	5
Obrázok 2: Vytvorenie nového projektu	6
Obrázok 3: Pozadie v hre Tanky	7
Obrázok 4: Vloženie tanku do sveta	8
Obrázok 5: Zmena pozície tanku	8
Obrázok 6: Vnútorňý stav tanku.....	9
Obrázok 7: Strely umiestnené vo svete.....	11
Obrázok 8: Pridanie bariér do bojiska	13
Obrázok 9: Vytvorenie inštancie tanku	15
Obrázok 10: Nesprávne nastavenie kláves.....	15
Obrázok 11: Pridanie tankov do bojiska.....	16
Obrázok 12: Vloženie prekážky múr do bojiska	17
Obrázok 13: Pridanie prekážok do bojiska	18
Obrázok 14: Strieľanie tanku pomocou určeného klávesu	19
Obrázok 15: Strieľanie s dobíjaním munície.....	21
Obrázok 16: Nezasiahnuteľnosť bariéry.....	23
Obrázok 17: Umiestnenie prekážok do riadku	25
Obrázok 18: Umiestnenie prekážok do obdĺžnika.....	25
Obrázok 19: Ukážka najľahšej úrovne	26
Obrázok 20: Ukážka stredne ťažkej úrovne.....	26
Obrázok 21: Ukážka najťažšej úrovne	27
Obrázok 22: Zobrazenie poľa čísel cez Inspect.....	28
Obrázok 23: Zobrazenie poľa reťazcov cez Inspect.....	28
Obrázok 24: Umiestnenie bahna do bojiska	31
Obrázok 25: Zašpinenie tanku po prejdení cez bahno	33
Obrázok 26: Zobrazenie skóre tankov.....	34
Obrázok 27: Bojisko s bonusom náboje	36

Obrázok 28: Vnútorný stav tanku	37
Obrázok 29: Bojisko s bonusom rýchlosť	38
Obrázok 30: Automatické generovanie bonusov	40
Obrázok 31: Pozícia triedy Menu a TlacidloHrat v hierarchii tried	41
Obrázok 32: Menu začatie hry	43
Obrázok 33: Pozícia triedy MenuUkoncenieHry, TlacidloPokracovat a TlacidloSkoncit v hierarchii tried	43
Obrázok 34: Menu ukončenie hry	44
Obrázok 35: Vytvorenie osamostatnenej triedy	45
Obrázok 36: Pozícia triedy Ovladanie v hierarchii tried	46
Obrázok 37: Nepriateľská veža v bojisku	47
Obrázok 38: Čierna diera v bojisku	48

BIBLIOGRAFIA

- [1] freesvg, „Stone,“ [Online]. Available: <https://freesvg.org/stone-noshadow>. [Cit. 28.6.2021].
- [2] freesvg, „Palm top view,“ [Online]. Available: <https://freesvg.org/palm-04>. [Cit. 28.6.2021].
- [3] freesvg, „Tree top view,“ [Online]. Available: <https://freesvg.org/tree-32>. [Cit. 28.6.2021].
- [4] freepik, „Set of wooden buttons play and quit,“ [Online]. Available: https://www.freepik.com/premium-vector/set-wooden-buttons-play-quit-isolated-icons-check-mark-cross-games-menu_12872983.htm#&position=4. [Cit. 28.6.2021].
- [5] freepik, „Blank banner vector,“ [Online]. Available: <https://freesvg.org/blank-banner-vector>. [Cit. 28.6.2021].