




# PROJEKT ASTEROIDY

## OBJEKTOVÝ PRÍSTUP K RIEŠENIU PROBLÉMOV

### [ZBIERKA ÚLOH]

TERÉZIA SLIACKA  
JÁN KUČERA  
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

 **it** **AKADÉMIA**

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

[www.minedu.sk](http://www.minedu.sk) [www.employment.gov.sk/sk/esf/](http://www.employment.gov.sk/sk/esf/) [www.itakademia.sk](http://www.itakademia.sk)

**Projekt Asteroidy**  
**Objektový prístup k riešeniu problémov**  
**[Zbierka úloh]**

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

**Projekt Asteroidy - Objektový prístup k riešeniu problémov [Zbierka úloh]**

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

Autori: Terézia Sliacka, Ján Kučera, 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

---

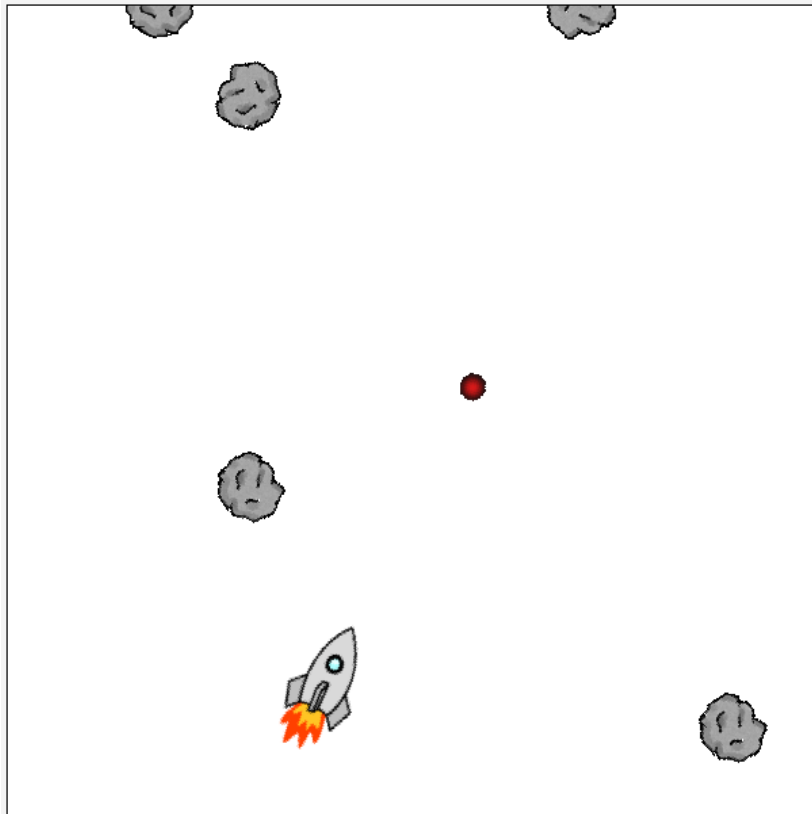
ÚVOD.....	5
1 Vesmírna Lod'.....	6
2 Pohyb vesmírnej lode vpred.....	10
3 Otáčanie vesmírnej lode za myšou.....	11
4 Pridanie asteroidov do hry.....	12
5 Pohyb asteroidov.....	14
6 Presun asteroidov, ktoré odídu z okna.....	17
7 Strelba z vesmírnej lode.....	18
8 Zničenie zasiahnutého asteroidu.....	21
9 Neúspešný koniec hry.....	22
10 Úspešný koniec hry.....	24
Index obrázkov.....	25
Bibliografia.....	26

# ÚVOD

---

## POPIS A PRAVIDLÁ HRY

Hra *Asteroidy* je určená pre jedného hráča. Hráč ovláda jednu vesmírnu loď okolo ktorej lietajú asteroidy. Cieľom hry je zabrániť nárazu vesmírnej lode s asteroidom a zároveň ničiť asteroidy pomocou striel. Hráč hru vyhrá, ak sa mu podarí zostreliť všetky asteroidy. V prípade, že dôjde ku kolízii vesmírnej lode s niektorým z asteroidov, hra končí neúspechom.

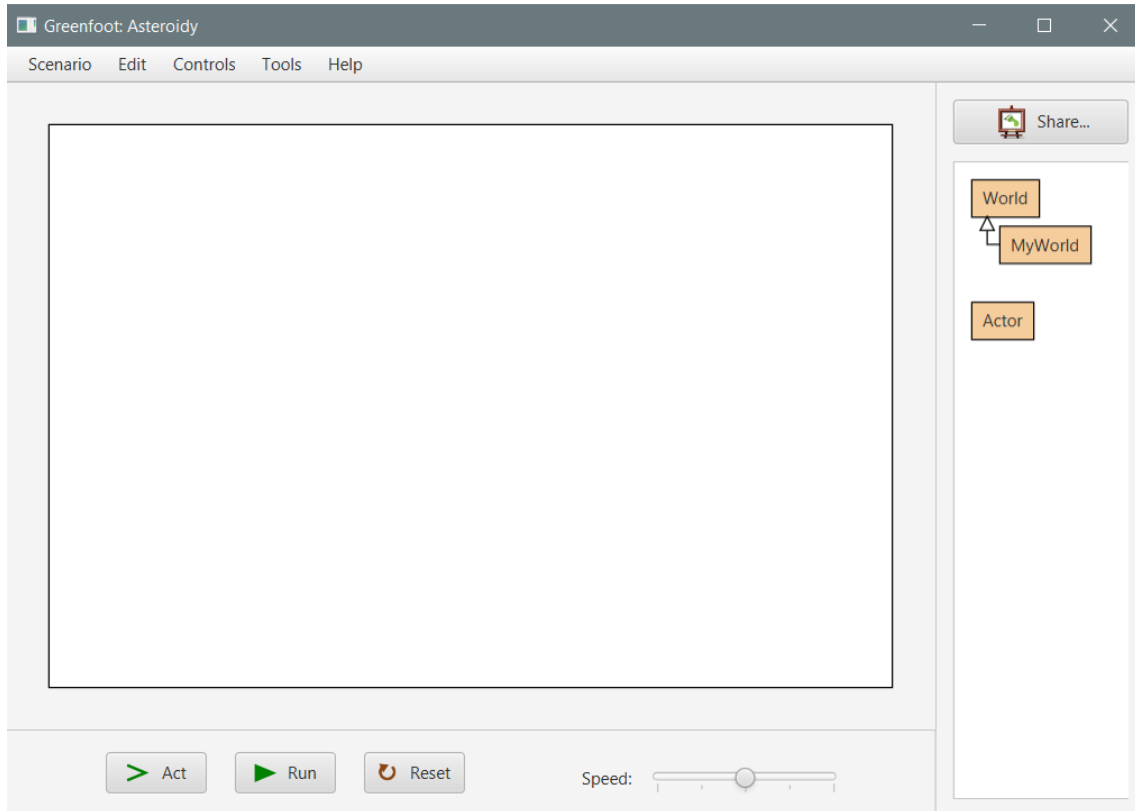


Obrázok 1 - Ukážka hry Asteroidy v prostredí Greenfoot

# 1 VESMÍRNA LOĎ

## ÚLOHA 1.1

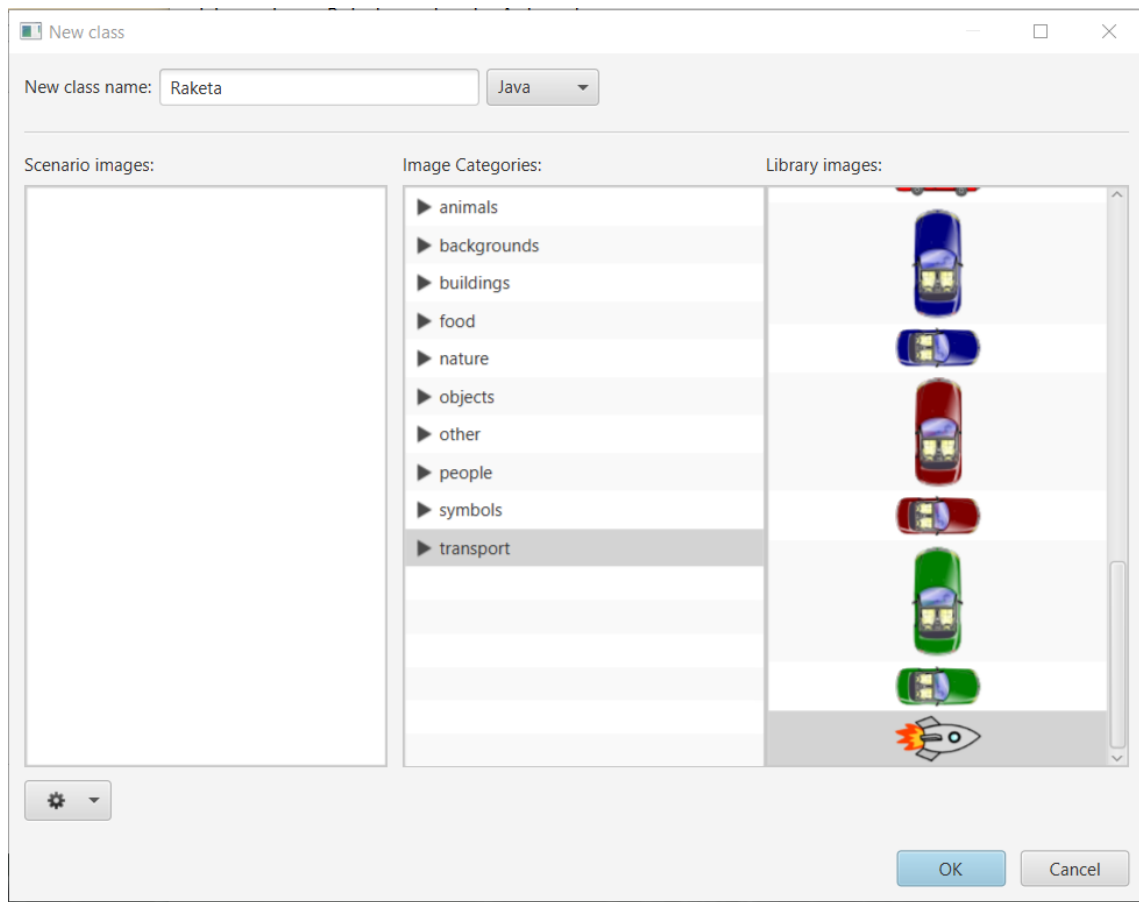
Vytvorte nový **Java Scenár**, ktorý bude zastrešovať našu hru. Pomenujte ho *Asteroidy*, aby jeho názov jasne identifikoval hru, ktorú vytvárate. Následne zvolte vhodnú zložku, do ktorej projekt uložíte. Po vytvorení bude projekt obsahovať triedy **World**, **Actor** a **MyWorld**.



Obrázok 2 - Projekt Asteroidy po vytvorení nového Java Scenaria

## ÚLOHA 1.2

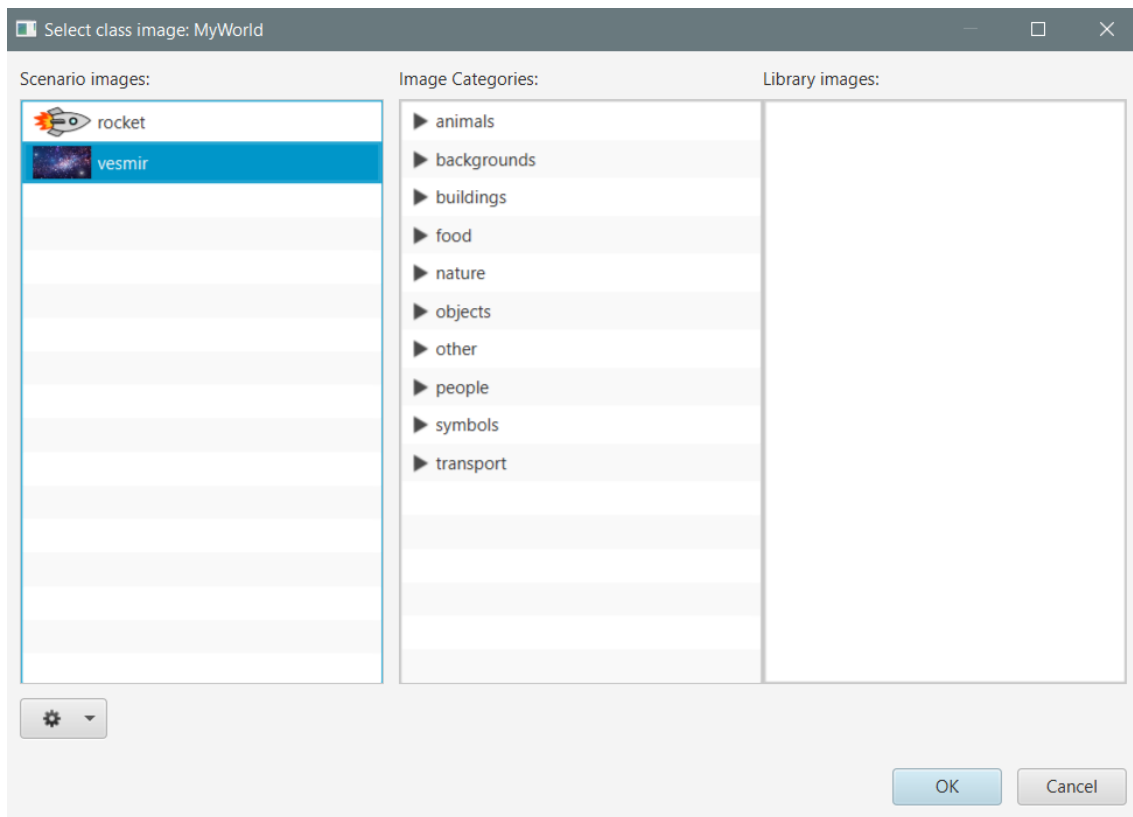
Prvým aktorm, ktorý pribudne na vesmírnu scénu je vesmírna loď (Raketa). Vytvorte triedu **Raketa** a nastavte jej obrázok zo skupiny **Transport**.



Obrázok 3 - Vytvorenie triedy a nájdenie obrázku Raketa

### ÚLOHA 1.3

Trieda **MyWorld** bude reprezentovať hracie pole hry. Nastavte pozadie hracieho poľa na obrázok vesmíru. Obrázok stiahnite z internetu.



Obrázok 4 - Nastavenie pozadia hracieho poľa

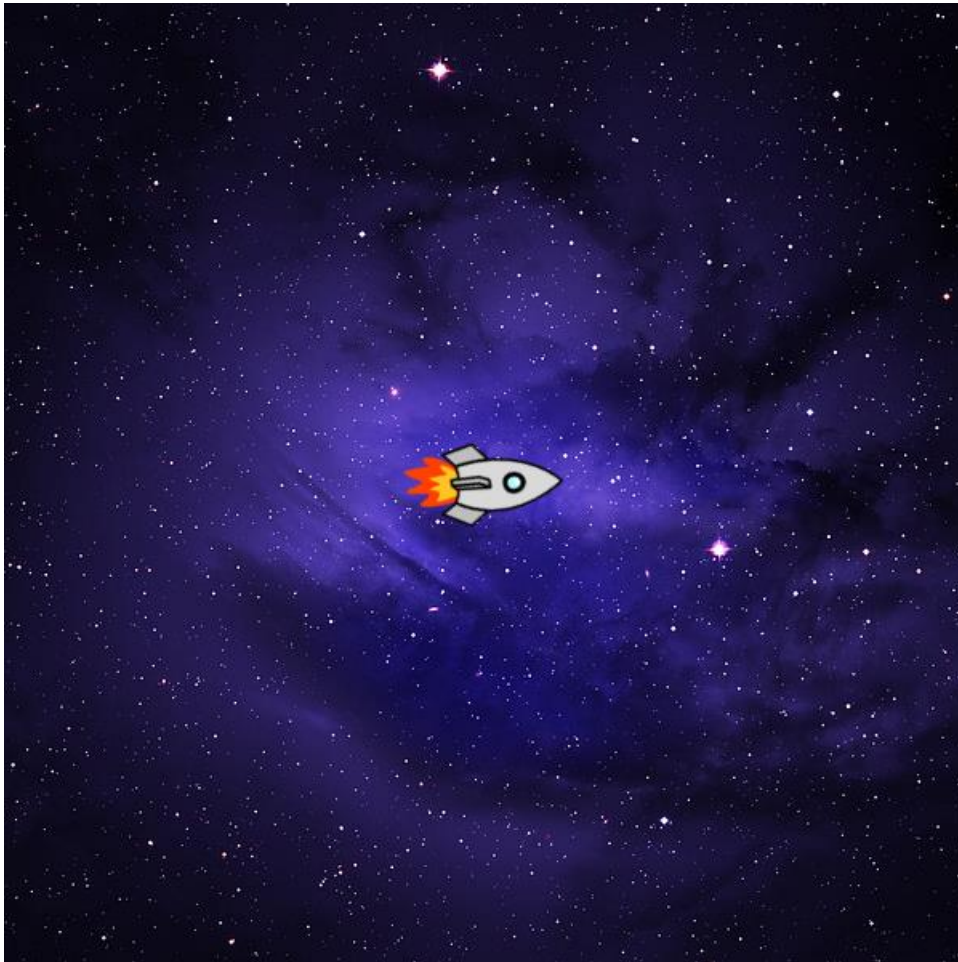
#### ÚLOHA 1.4

Zmeňte hraciu plochu tak, aby jej veľkosť bola 600x600 buniek, pričom každá bunka bude veľká jeden pixel.

#### ÚLOHA 1.5

Vytvorte inštanciu vesmírnej lode tak, aby bola zobrazená v hracom poli (vo vesmíre).





Obrázok 5 - Vesmírna loď v hracom poli [1]

## 2 POHYB VESMÍRNEJ LODE VPRED

### ÚLOHA 2.1

Pridajte vesmírnej lodi atribút, ktorý bude reprezentovať jej rýchlosť. Rýchlosť vesmírnej lode zadefinujte ako celočíselnú premennú, ktorej hodnotu nastavte v konštruktoze na 5.

### ÚLOHA 2.2

Vesmírna loď už má určenú svoju rýchlosť, avšak stále dokáže stáť len na jednom mieste, pretože nemá schopnosť *pohnúť sa*. Naučte vesmírnu loď, aby sa pohla o toľko bodov, koľko určuje jej rýchlosť a pohyb vykonala po každom stlačení medzerníka. Keďže chceme, aby sa v hre neustále kontrolovalo, či nedošlo k stlačeniu medzerníka – vytvorte podmienku v metóde `act()`, po ktorej splnení vesmírna loď vykoná pohyb vpred.

Na jednoznačné vymedzenie ovládania použite `Greenfoot.isKeyDown("space")` – čo znamená, že bola stlačená klávesa space (medzerník). Loď posuniete vpred zavolaním `this.move(this.rychlost)`.

## 3 OTÁČANIE VESMÍRNEJ LODE ZA MYŠOU

### ÚLOHA 3.1

Pre otočenie rakety za kurzorom potrebujete vedieť jeho pozíciu na hracej ploche. Informácie o myši a kurzore získate príkazom `Greenfoot.getMouseInfo()`.

Tieto informácie o myši a kurzore uložte do premennej typu `MouseInfo`. Dobre si premyslite, v ktorej časti kódu by sme mali programovať otáčanie vesmírnej lode.

### ÚLOHA 3.2

V prechádzajúcej úlohe sa nám podarilo získať informácie o kurzore a myši. Teraz ich budeme potrebovať, aby sme vedeli povedať rakete, ktorým smerom sa má otočiť.

Otočte raketu za kurzorom. Pozíciu kurzora myši získate funkciami `getX()` a `getY()`. Raketu k týmto koordinátom otočíte príkazom `this.turnTowards(x, y)`.

### ÚLOHA 3.3

Už viete otočiť myš za kurzorom a zároveň s využitím medzerníka a pohybu naprogramovaného v úlohe 2.1 dokážete raketu pohnúť ktorýmkoľvek smerom. Môže sa však stať, že myš sa nenachádza na hracom plátne a preto informácie o nej nie sú dostupné. Ak by sme sa snažili raketu otočiť za myšou, ktorú program nepozná (je mimo hracieho plátna), program by spadol a hra by nemohla ďalej pokračovať. Takýto prípad je potrebné vopred ošetriť.

Vytvorte podmienku, ktorá zabezpečí, že raketa sa bude otáčať za myšou len v prípade, že ju program dokáže rozpoznať. (Môžete využiť porovnanie premennej `mys` s hodnotou `null`).

## 4 PRIDANIE ASTEROIDOV DO HRY

### ÚLOHA 4.1

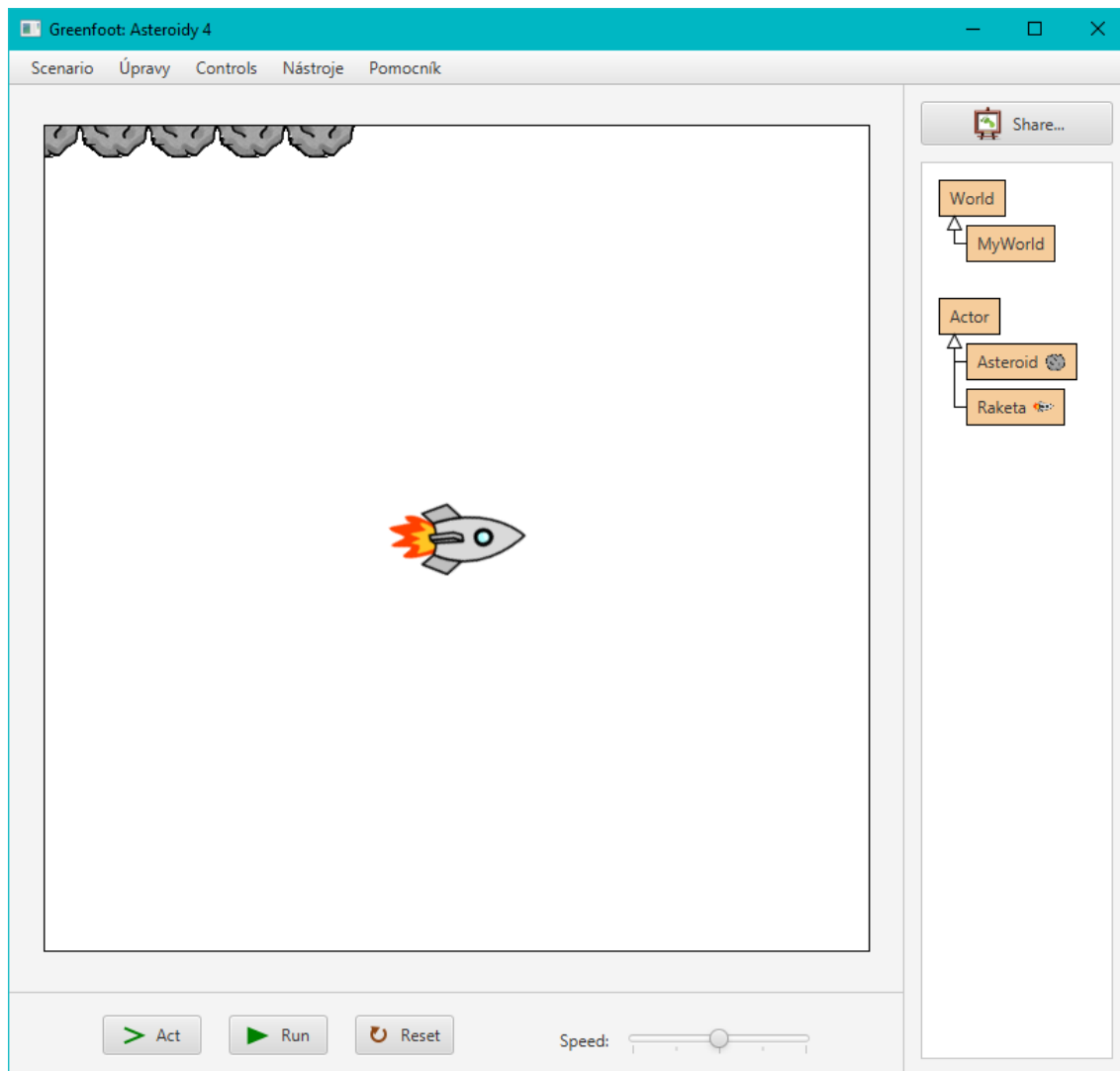
Vytvorte triedu `Asteroid` a nastavte jej obrázok asteroidu, ktorý nájdete v skupine `Nature`.

### ÚLOHA 4.2

Ak by sme chceli asteroidy do sveta pridávať rovnakým spôsobom, ako sme si ukázali v úlohe 1.5, kód by bol príliš dlhý a v počte asteroidov by sa dalo ľahko stratiť. Preto pomocou cyklu `for` vytvorte viacero asteroidov naraz s použitím jednoduchého kódu. Asteroidy vzniknú spolu so svetom, preto dopíšte kód v konšuktore triedy `MyWorld`.

### ÚLOHA 4.3

Použite číslo v premennej `i` na vytvorenie asteroidov na rôznych pozíciách, aby ste sa presvedčili, že ich je naozaj viac. Využite fakt, že číslo `i` sa zvyšuje pri každom novom asteroide.



Obrázok 6 - Vytvorenie asteroidov pomocou cyklu for

## 5 POHYB ASTEROIDOV

### ÚLOHA 5.1

Pridajte asteroidom atribút pre uloženie rýchlosti, rovnako ako pri rakete v úlohe 2.1 a použite túto rýchlosť v metóde `act()` na jeho pohyb.

### ÚLOHA 5.2

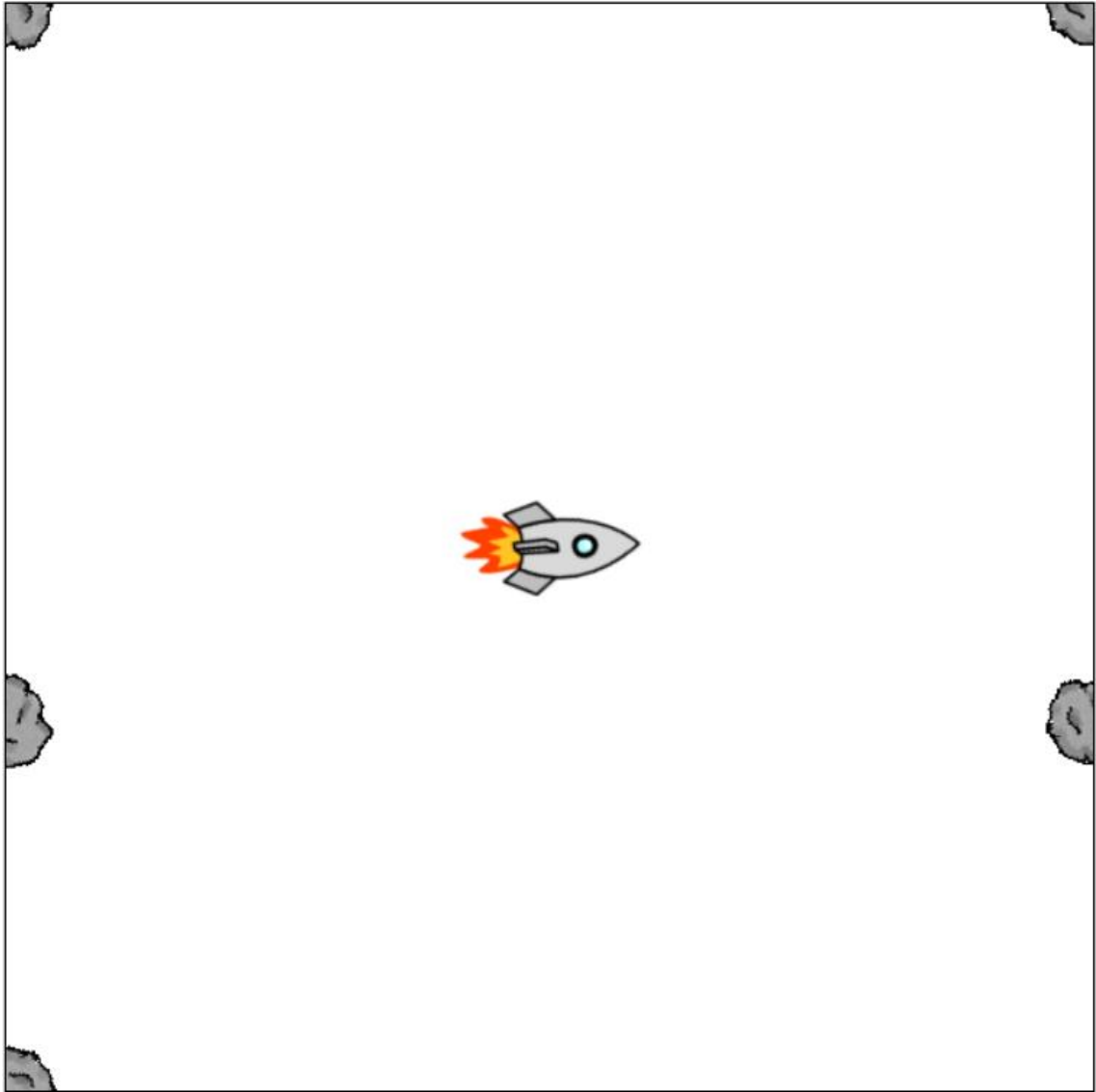
Aby sa asteroidy pohybovali rôznymi smermi, musíme ich potočiť. Na to budeme využívať generovanie náhodných čísel, ktoré budú predstavovať uhol, o ktorý sa pri vytvorení asteroid otočí. Asteroid chceme otočiť iba raz, preto ho otočíme rovno pri jeho vytvorení – v konštruktore.

### ÚLOHA 5.3

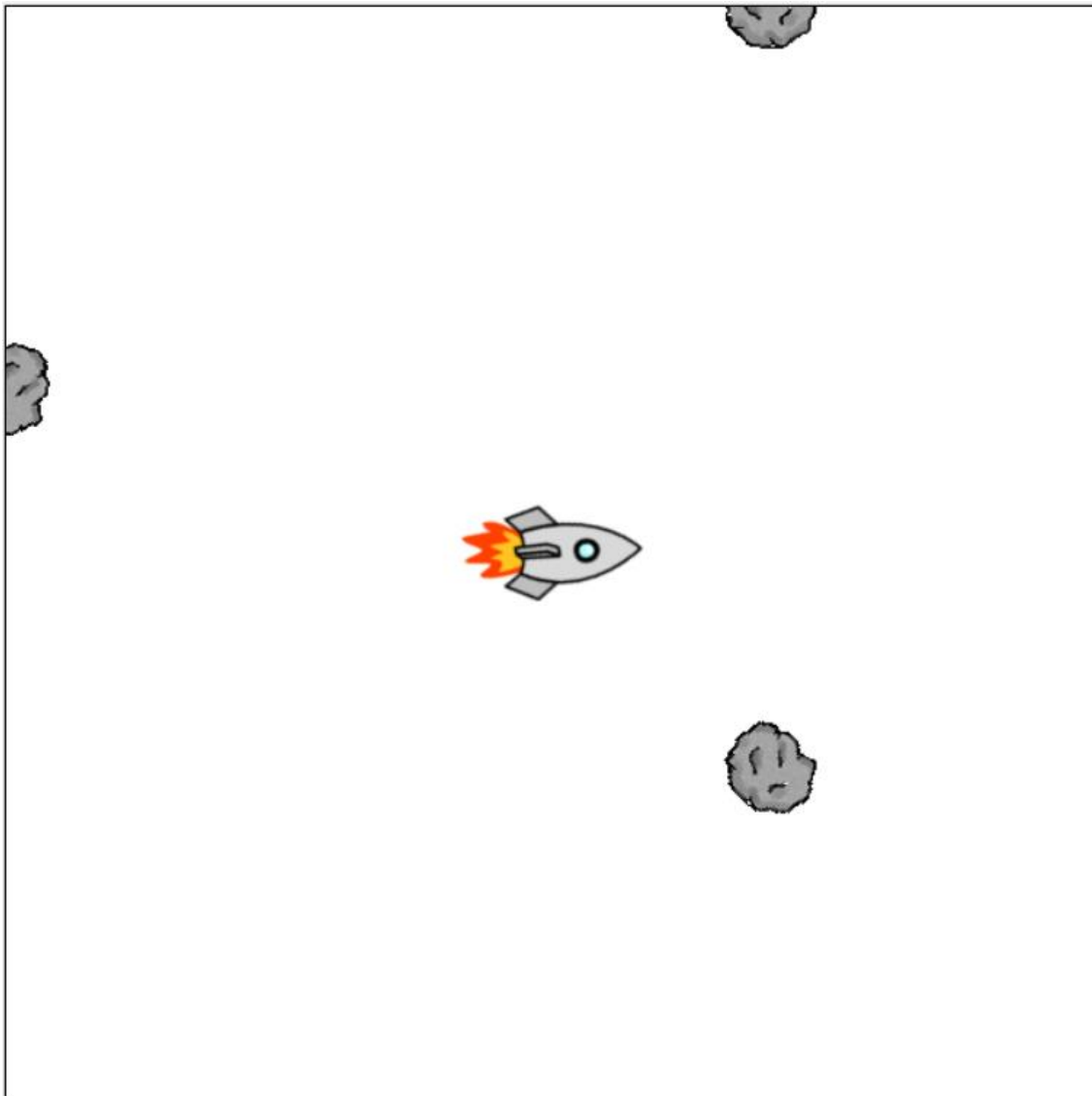
Upravte rýchlosť asteroidov na náhodné číslo, aby sa nepohybovali všetky rovnakou rýchlosťou. Aby sme sa vyhli vygenerovaniu čísla 0, vygenerované číslo môžeme navýšiť o 1.

### ÚLOHA 5.4

Nastavte, aby bol možný pohyb aktorov mimo hracej plochy. Ohraničenosť sveta sa určuje štvrtým parametrom v konštruktore triedy `MyWorld` po parametroch šírka okna, výška okna a veľkosť buniek. Tento štvrtý parameter je voliteľný a ak ho nepoužijete, bude automaticky nastavený na hodnotu `true`, čo znamená, že vychádzanie aktorov mimo okna bude zakázané. To by znamenalo, že svet bude mať vytvorené hranice za ktoré nemožno prejsť tak, ako to bolo doteraz (túto situáciu môžeme vidieť aj na obrázku 7).



Obrázok 7 - Asteroidy zaseknuté na hraniciach sveta



Obrázok 8 - Vychádzanie aktorov za hranice sveta



## 6 PRESUN ASTEROIDOV, KTORÉ ODÍDU Z OKNA

### ÚLOHA 6.1

Zadefinujte premenné  $x$  a  $y$  do ktorých uložíte aktuálne koordináty (pozíciu) asteroidu v rámci hracieho okna.

### ÚLOHA 6.2

Získajte informácie o veľkosti hracieho okna. Zistite šírku aj výšku hracieho poľa a uložte ich do premenných `maxX` a `maxY`. Tieto hodnoty budú reprezentovať maximálne pozície v smere  $x$  a  $y$ , ktoré môže asteroid dosiahnuť.

### ÚLOHA 6.3

Vypočítajte pomocou funkcie modulo (%) novú pozíciu pre asteroid. Keď asteroid presiahne hranicu okna, zvyšok po delení jeho koordinátov veľkosťou okna nám určí jeho novú pozíciu na opačnej strane.

## 7 STREĽBA Z VESMÍRNEJ LODE

### ÚLOHA 7.1

Do projektu pridajte nového aktora **Strela** a zvolte vhodný obrázok. My sme použili obrázok zo sekcie **Other**.

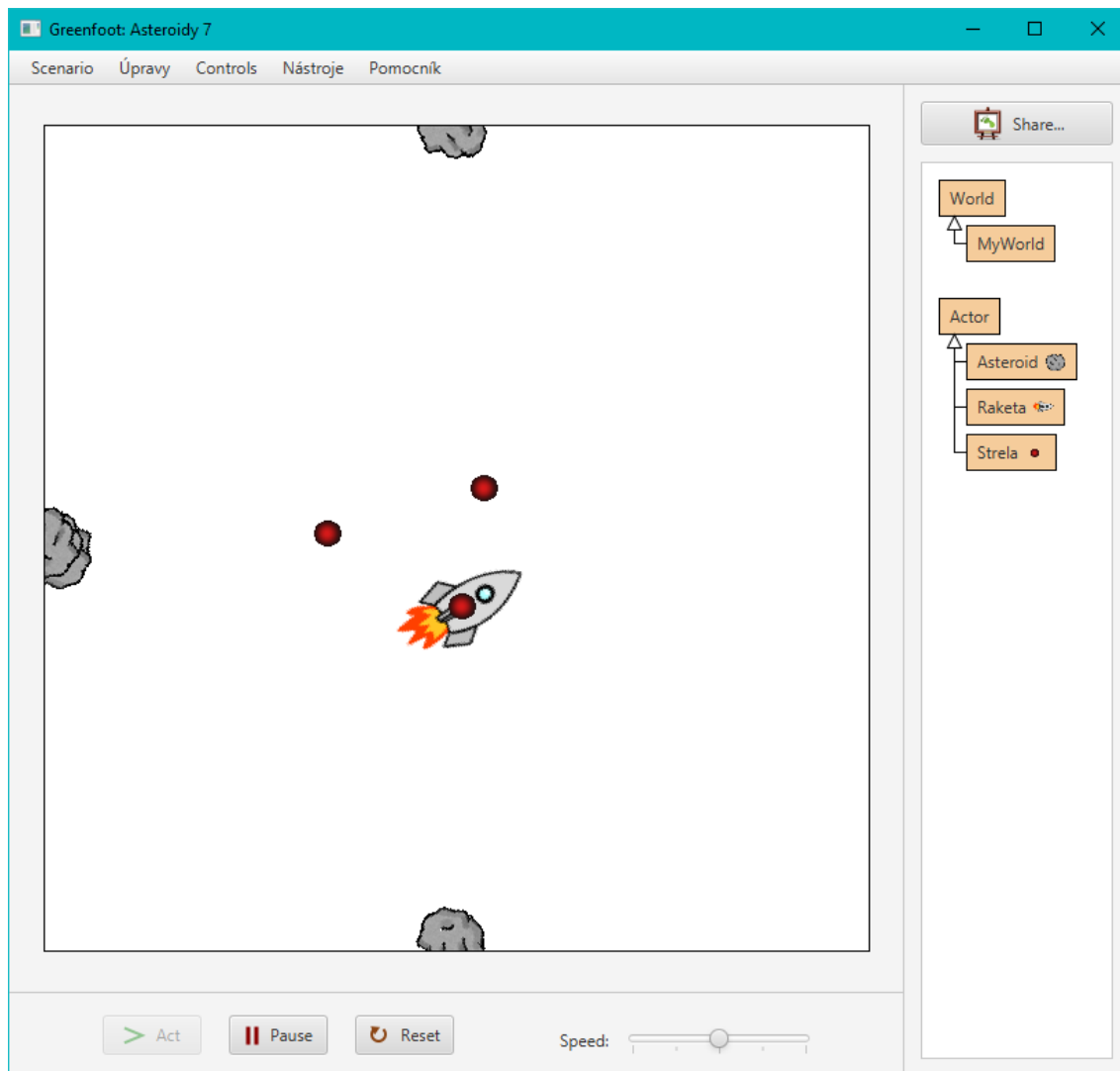
### ÚLOHA 7.2

V triede **Raketa** vytvorte parametrickú metódu **mozeStrielat(MouseInfo mys)** v ktorej skontrolujeme, či raketa môže vystreliť. V tejto úlohe opäť využijeme funkcie triedy **MouseInfo**, ktoré sme prvýkrát použili v úlohe 3.1 pri zisťovaní aktuálnej polohy kurzora.

Podmienky pre vystrelenie rakety sú 2:

1. Bolo stlačené ľavé tlačítko myši (`mys.getButton() == 1`)
2. `Greenfoot.mouseClicked(null) == true`, aby sa tlačidlo nedalo držať (podmienka slúži na identifikáciu kliknutia). Ak by sme túto podmienku nepoužili, stačilo by držať ľavé tlačidlo myši a raketa by neustále strieľala (ako samopál). Týmto spôsobom by však bolo príliš jednoduché hru vyhrať a preto chceme, aby na jeden klik raketa vystrelila len jednu strelu.

Po úspešnej kontrole podmienok metóda **mozeStrielat()** vráti hodnotu **true**, inak **false**, preto jej návratová hodnota musí byť typu **boolean**.



Obrázok 9 - Vytvorenie nehybných striel po kliknutí

### ÚLOHA 7.3

V triede **Raketa** vytvorte metódu **strielaj()**, ktorá vytvorí novú inštanciu triedy **Strela** a vloží ju do hry na pozíciu rakety. Rozmyšľajte si, akú návratovú hodnotu bude mať táto metóda a taktiež či je potrebné, aby mala nejaké parametre.

### ÚLOHA 7.4

V metóde **act()** triedy **Raketa** zavolajte metódu **mozeStrielat(MouseInfo mys)** z úlohy 7.2. Ak sú podmienky vystrelenia splnené, zavolajte metódu **strielaj()**, ktorú ste definovali v úlohe 7.3. Keďže pracujeme s myšou, kód súvisiaci so strieľaním nezapodíť písať na miesto, ktoré sme vytvorili v úlohe 3.3 – musí platiť, že premenná **mys** nie je rovná **null**.

### ÚLOHA 7.5

Rovnako ako v úlohe 5.1 rozhýbte strelu.

### ÚLOHA 7.6

Otočte strelu rovnakým smerom, akým bola otočená raketa v momente kliknutia. Zamyslite sa nad tým, kde je správne miesto na nastavenie rotácie.

## 8 ZNIČENIE ZASIAHNUTÉHO ASTEROIDU

Strela počas svojho letu bude kontrolovať, či nie je dostatočne blízko nejakého asteroidu, ktorý by mala zničiť. Pretože asteroidov môže strela prekryvať naraz aj viac, budeme pracovať so zoznamami (listami) asteroidov. V knihe sa na podobné účely používa v úlohe 10.12 funkcia `getOneIntersectingObject()`. V našom prípade sa však hra skladá z okrúhlych objektov (asteroid, strela, čiastočne aj loď) a preto použijeme presnejší spôsob zisťovania kolízií ako v uvedenej úlohe. Docielime tým vizuálne lepší výsledok pri hraní hry a zároveň spresníme ovládanie.

### ÚLOHA 8.1

Triede, v ktorej chceme zoznam asteroidov použiť, musíme najskôr oznámiť, že zoznam budeme používať. Zoznam (`List`) je samostatná trieda, ktorá už je naprogramovaná a vy si ju musíte importovať pomocou príkazu `import java.util.List`.

### ÚLOHA 8.2

Získajte zoznam asteroidov, ku ktorým je strela dostatočne blízko na ich zničenie. Pomôže vám veľkosť obrázku strely a funkcia `this.getObjectsInRange(vzdialenost, typ)`. Výsledok uložte do premennej typu `List<Asteroid>`, čiže zoznam asteroidov. Funkcia hľadá objekty v kruhovej oblasti zo stredu strely. Preto hľadáme tie asteroidy, ktoré sa od strely nachádzajú do vzdialenosti rovnjej *šírka obrázku strely / 2*.

### ÚLOHA 8.3

Ak zoznam asteroidov nie je prázdny, znamená to, že strela zasiahla cieľ a má ho zničiť spolu so sebou. Ak teda zoznam asteroidov nejaký obsahuje, zvolte prvý z nich a vymažte ho zo sveta spolu so strelou.

## 9 NEÚSPEŠNÝ KONIEC HRY

V predošlej kapitole sme naučili strely ničiť asteroidy. Ďalším krokom pre funkčnosť hry je, aby sme naučili loď zistiť kolíziu s asteroidom – v takom prípade má totiž hra skončiť. Použitím rovnakých príkazov ako pri strele získame zoznam asteroidov, ktoré sú príliš blízko rakety. Pokiaľ nejaké sú, hru ukončíme príkazom `Greenfoot.stop()`.

V knihe v kapitole 7.2 Identifikácia zasiahnutých hráčov nájdete iný spôsob riešenia rovnakého problému. V úlohe 7.7 (v knihe) každá bomba pomocou metódy `act()` neustále zisťuje, či nezasiahla niektorého z hráčov. Následne môžete v úlohe 7.9 vidieť, že po zásahu hráča bombou je v triede `Hrac` vyvolaná metóda `zasah()`. V prípade hry Asteroidy by sme mohli použiť rovnaké riešenie a v triede `Asteroid` v metóde `act()` kontrolovať, či nedošlo ku kolízii s raketou. Zasiahnutú raketu môžeme v triede `Asteroid` získať napríklad pomocou metódy `getOneIntersectingObject()`. Keďže hráča (raketu) máme len jedného, bolo by zbytočné použiť metódu `getObjectsInRange()` pretože tá namiesto jedného objektu získava zoznam objektov. Následne skontrolujeme podmienku, či raketa vôbec bola zasiahnutá (nie je `null`) a ak áno, vyvoláme jej metódu `zasah()`.

```
Raketa zasiahnutáRaketa =
(Raketa)this.getOneIntersectingObject(Raketa.class);
if(zasiahnutáRaketa!= null) {
    zasiahnutáRaketa.zasah();
}
```

V triede `Raketa` by sme museli vytvoriť public metódu `zasah()` a v nej zavolať všetko, čo chceme, aby sa udialo po zásahu rakety (napísali by sme rovnaké príkazy ako v úlohe 9.2 alebo 9.3).

V predošlej kapitole sme vysvetľovali, že v našom prípade nie je použitie metódy `getOneIntersectingObject()` príliš vhodné, pretože objekty sú ohraničené skôr kružnicou ako obdĺžnikom. Metóda `getOneIntersectingObject()` sa pozerá na actorov ako na obdĺžniky a preto zvolíme iný spôsob kontrolovania kolízií medzi asteroidmi a raketou ako sme ukázali v predchádzajúcom texte a tiež aj v knihe.

### ÚLOHA 9.1

Získajte zoznam asteroidov, ku ktorým je raketa príliš blízko, podobne ako v úlohe 9.1

### ÚLOHA 9.2

Ak raketa zasiahla nejaký asteroid, ukončíte hru.

Pomocou metódy `this.getWorld().showText()` vypíšte informáciu o prehre.

Chceme, aby text bol zobrazený uprostred hracieho poľa. Vytvorte si lokálne premenné `stredX` a `stredY` typu `int`, do ktorých si uložíte `x` a `y` pozíciu stredu hracej plochy (sveta).

Celkovú šírku sveta získate pomocou metódy `this.getWorld().getWidth()` a rovnako výšku zavolaním `this.getWorld().getHeight()`.

Parametre metódy `showText()` sú nasledovné: text, ktorý sa má zobraziť; x – pozícia zobrazeného textu; y – pozícia zobrazeného textu.

Následne odstráňte zničenú raketu z vesmíru zavolaním metódy `this.getWorld().removeObject(this)` – tento riadok zabezpečí, že `MyWorld` vymaže objekt, ktorý sme určili.

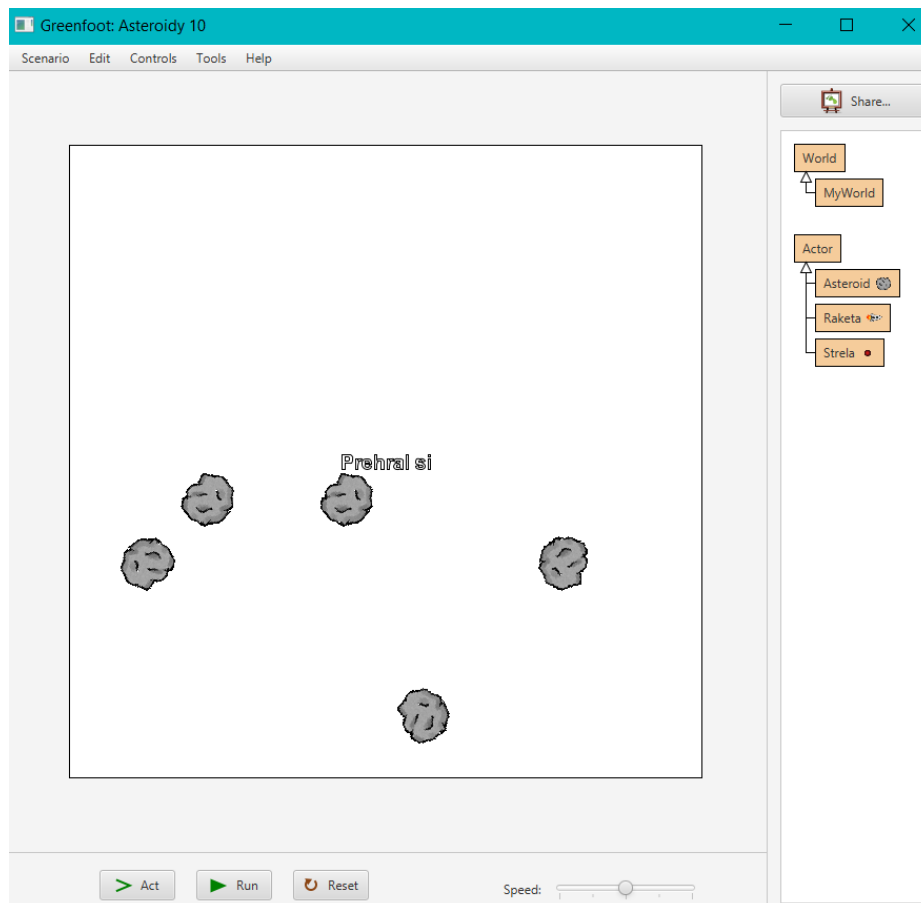
Hru ukončíte zavolaním príkazu `Greenfoot.stop()`.

### ÚLOHA 9.3

Do prechádzajúcej úlohy doplňte riadok, ktorý zabezpečí zvukový efekt výbuchu po zasiahnutí vesmírnej lode asteroidom.

Do priečinka `Asteroidy\sounds` uložte zvuk znázorňujúci výbuch. Typ súboru musí byť `wav`.

Na prehratie zvuku zavolajte metódu `Greenfoot.playSound("nazov.wav")`.



Obrázok 10 - Neúspešný koniec hry

## 10 ÚSPEŠNÝ KONIEC HRY

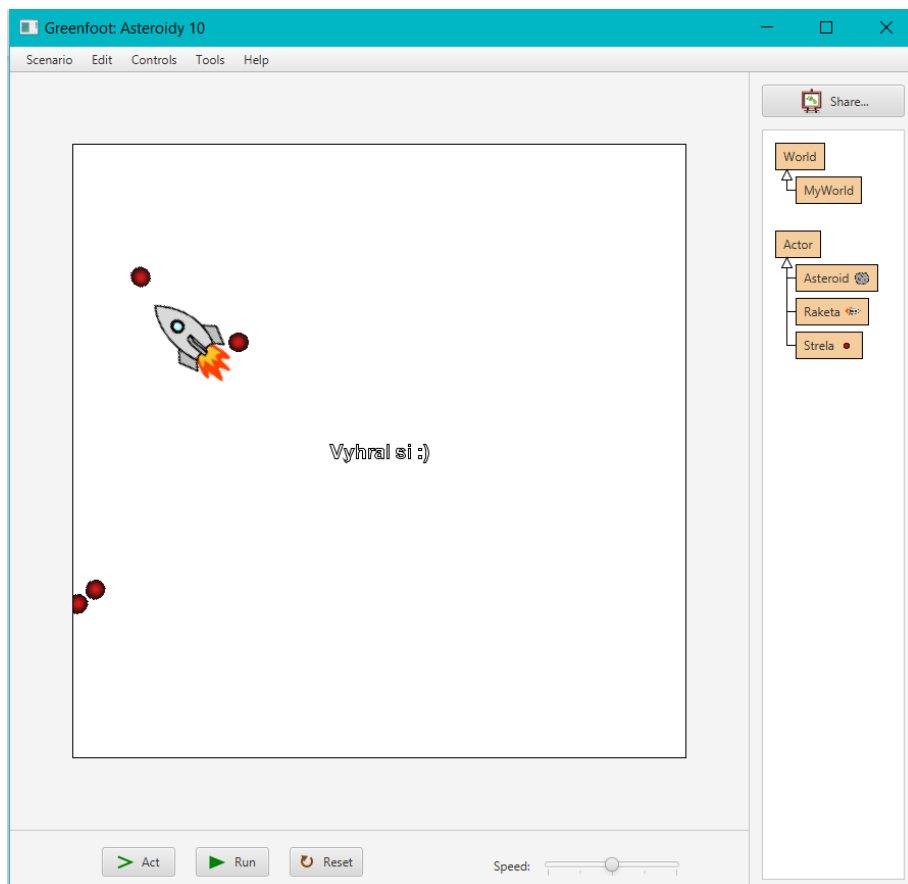
Výhra nastáva vtedy, keď sa vo svete nebudú nachádzať žiadne asteroidy. Túto kontrolu môžeme vykonávať v triede `MyWorld`, do ktorej pridáme metódu `act()`. Zoznam všetkých asteroidov vo svete získame príkazom `this.getObjects(Asteroid.class)`. Pokiaľ získaný zoznam neobsahuje žiadne asteroidy, opäť hráčovi oznámime výsledok hry ako v predošlej úlohe a ukončíme hru.

### ÚLOHA 10.1

Získajte zoznam asteroidov vo svete, ktoré ešte neboli zničené.

### ÚLOHA 10.2

Ak už neexistujú žiadne asteroidy, oznámte hráčovi že vyhral a ukončíte hru. Rovnako ako v úlohe 9.2, aj tu vypíšete informáciu o výhre do stredu hracej plochy.



Obrázok 11 - Úspešný koniec hry



## INDEX OBRÁZKOV

---

Obrázok 1 - Ukážka hry Asteroidy v prostredí Greenfoot.....	5
Obrázok 2 - Projekt Asteroidy po vytvorení nového Java Scenaria .....	6
Obrázok 3 - Vytvorenie triedy a nájdenie obrázku Raketa .....	7
Obrázok 4 - Nastavenie pozadia hracieho poľa .....	8
Obrázok 5 - Vesmírna loď v hracom poli [1] .....	9
Obrázok 6 - Vytvorenie asteroidov pomocou cyklu for .....	13
Obrázok 7 - Asteroidy zaseknuté na hraniciach sveta .....	15
Obrázok 8 - Vychádzanie aktorov za hranice sveta.....	16
Obrázok 9 - Vytvorenie nehybných striel po kliknutí.....	19
Obrázok 10 - Neúspešný koniec hry.....	23
Obrázok 11 - Úspešný koniec hry.....	24

## BIBLIOGRAFIA

---

- [1] Thorslotsdal, „Pixabay,“ 2017. [Online]. Available: <https://pixabay.com/photos/space-nasa-purple-2901400/>.