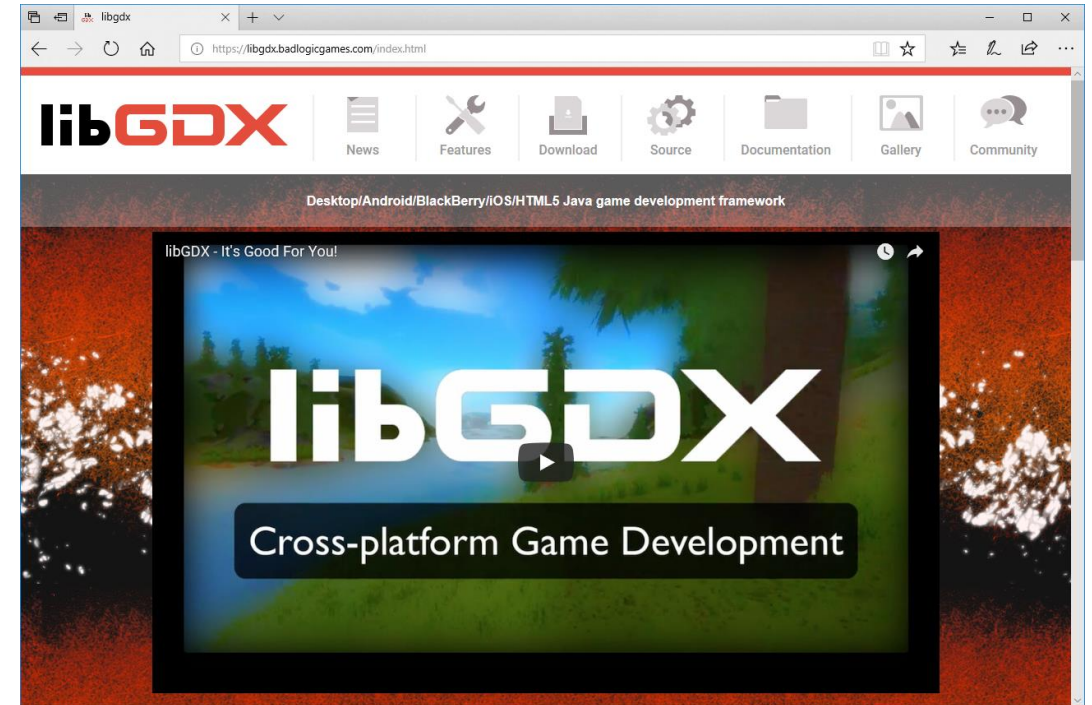


Úvod do LibGdx

Herný multiplatformový Java framework

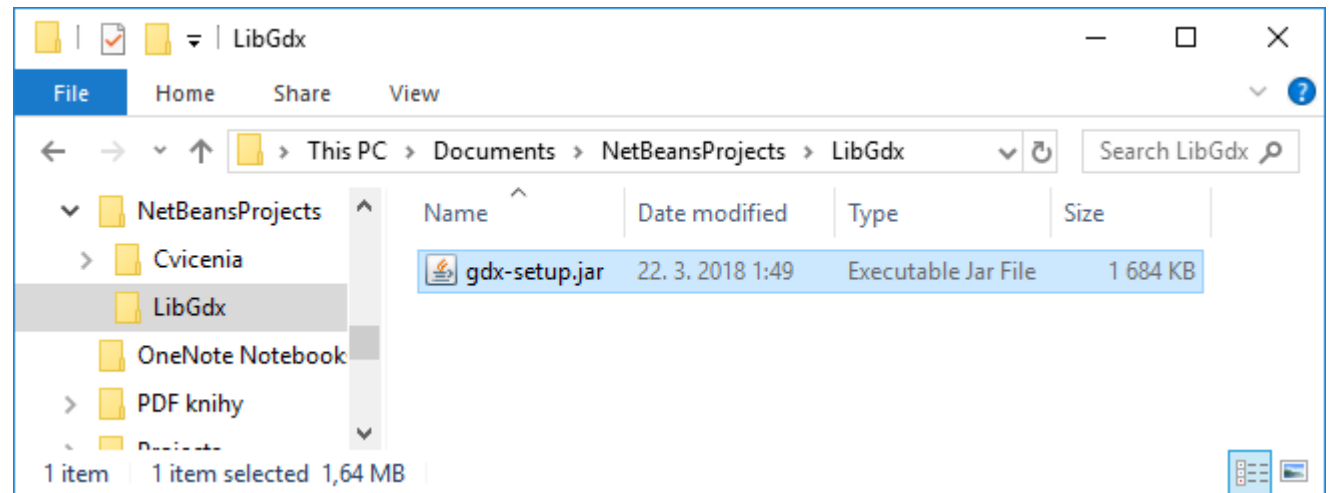
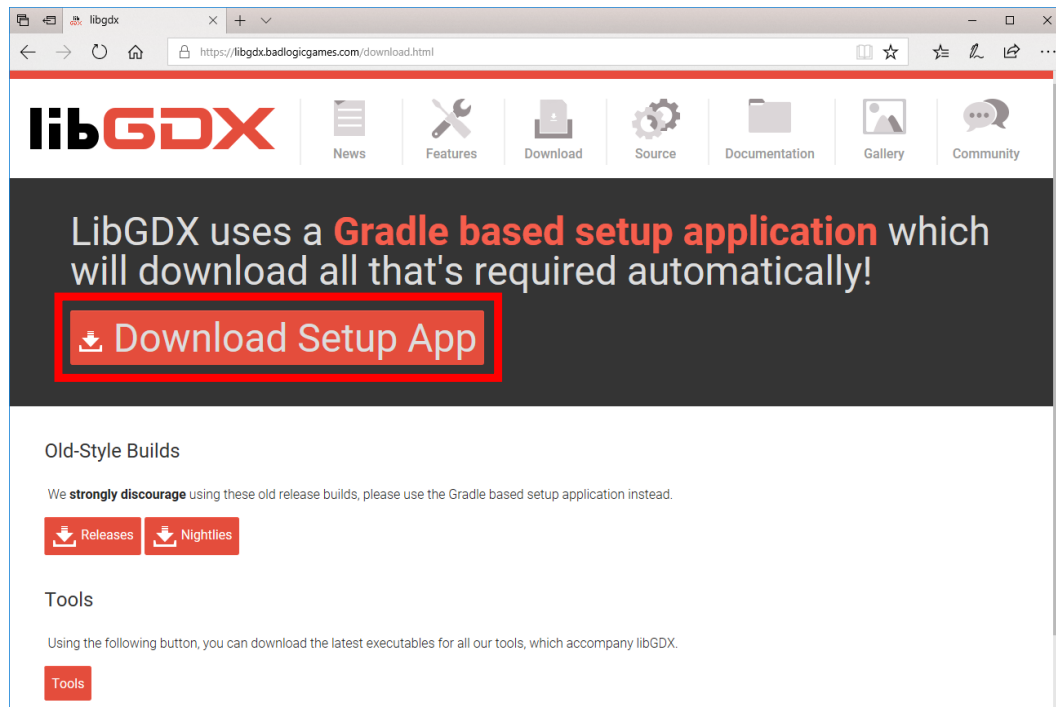
LibGDX

- <https://libgdx.badlogicgames.com/>
- **Herný open-source framework** pre **desktop** (Windows, Linux, Mac), **Android**, **BlackBerry**, **iOS**, **HTML5**
- Dokumentácia, príklady a tutoriály:
 - <http://libgdx.badlogicgames.com/nightlies/docs/api/overview-summary.html>
 - <https://github.com/libgdx/libgdx/wiki/A-simple-game> - tutoriál prvej hry
 - <https://libgdx.info/> - rôzne príklady
 - <https://www.youtube.com/channel/UCO9JvZ75Usyzgd1puurLF6A> - videonávody
 - <https://github.com/libgdx/libgdx/wiki/External-tutorials> - veľa ďalších tutoriálov



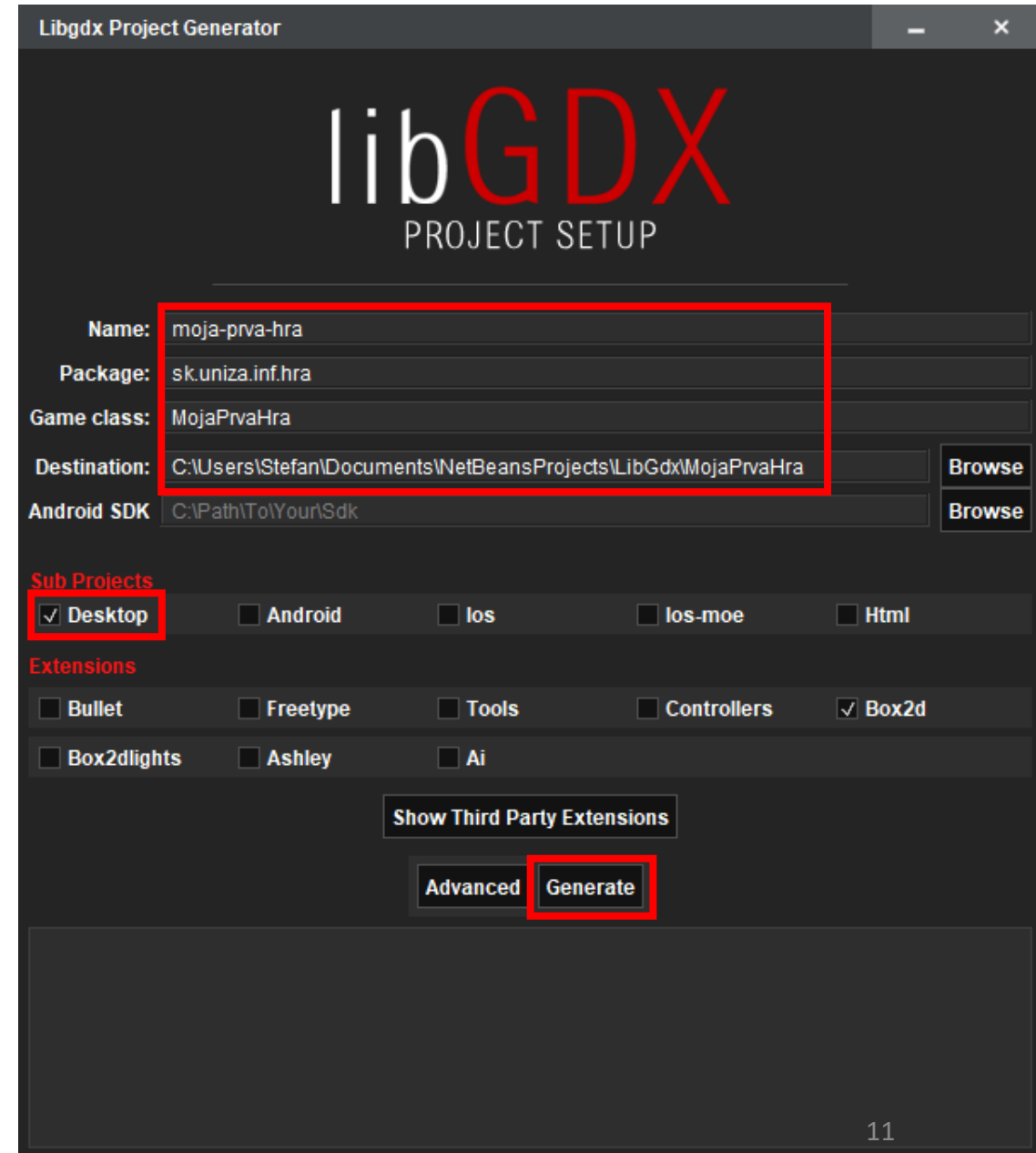
Stiahnutie generátora projektov

- **Stiahneme si aplikáciu Libgdx Project Generator (**gdx-setup.jar**):**
 - <https://libgdx.badlogicgames.com/download.html>
 - Klikneme na **Download Setup App**



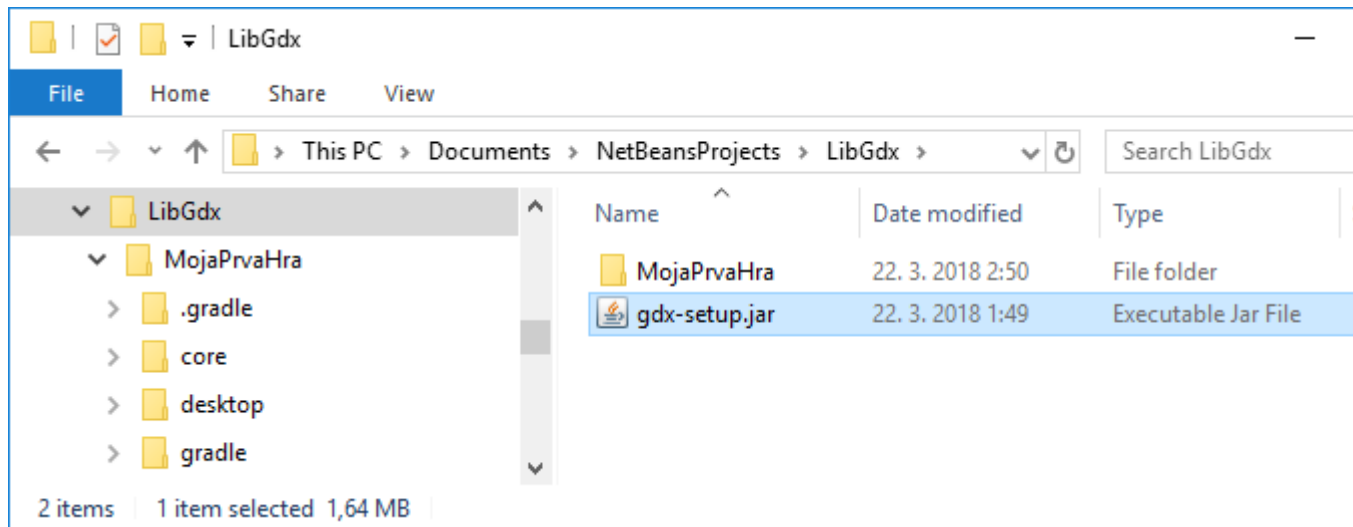
Spustenie generátora

- Spustíme **gdx-setup.jar**
- **Vyplníme polia** a zvolíme si, aké podprojekty požadujeme súčasne vygenerovať, t. j. na akých platformách chceme, aby naša hra bežala:
 - *Desktop* – Windows, Linux, MacOS
 - *Android* – telefóny s Androidom
 - *ios / ios-moe* – Apple telefóny a tablety s iOS (moe = Intel Multi-OS Engine)
 - *Html* – web pomocou GWT (Google Web Toolkit)
- Klikneme na **Generate**



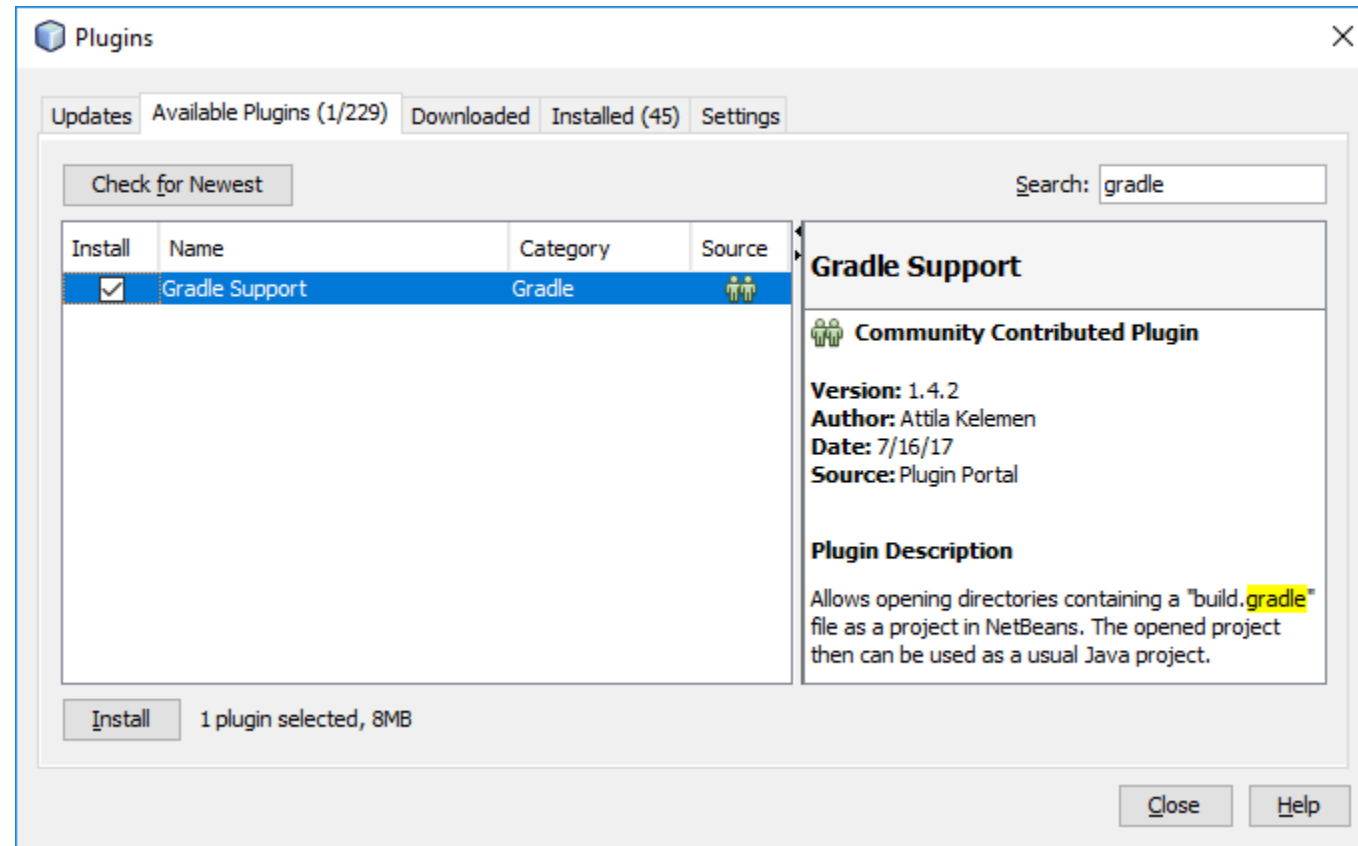
Po vygenerovaní

- Po vygenerovaní nás program informuje, že **projekt môžeme otvoriť** v Netbeanse, avšak to ide iba v prípade, že máme nainštalovaný **plugin Gradle Support** – ak ho nemáme, musíme najskôr nainštalovať

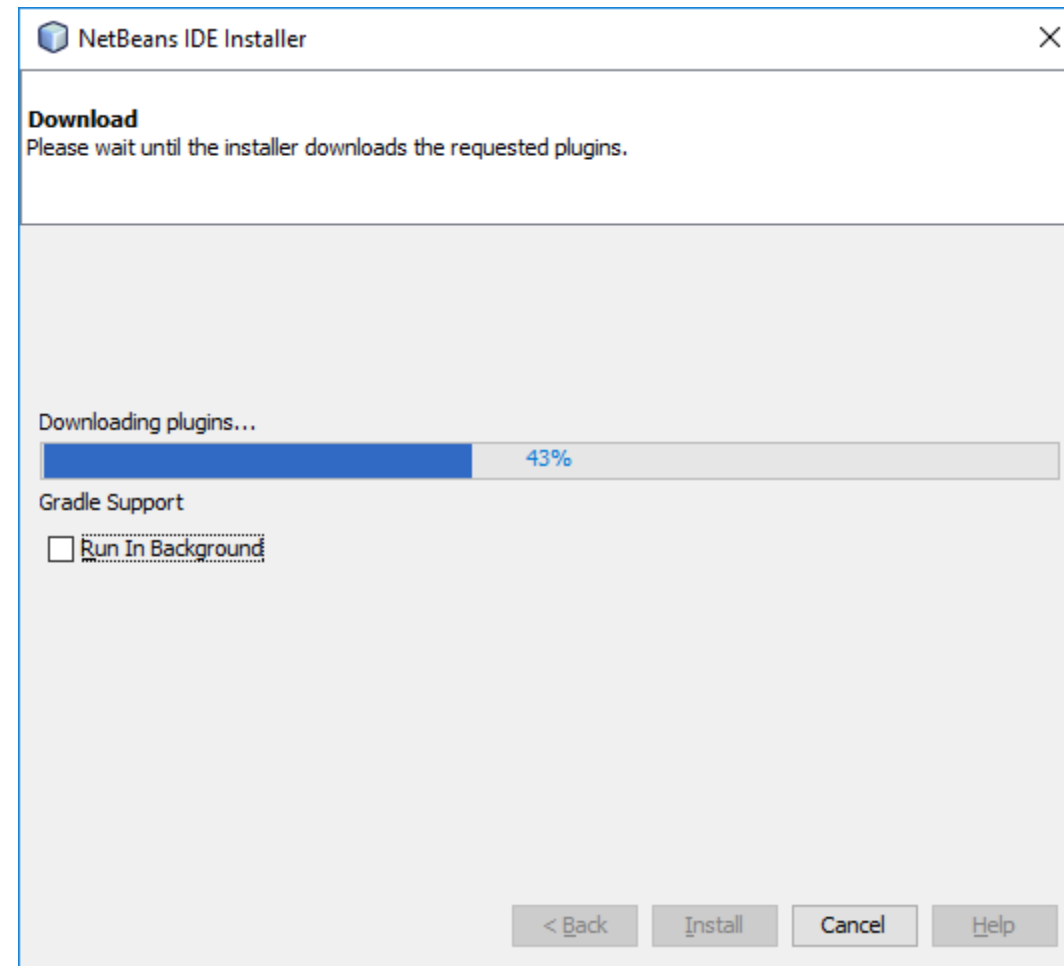
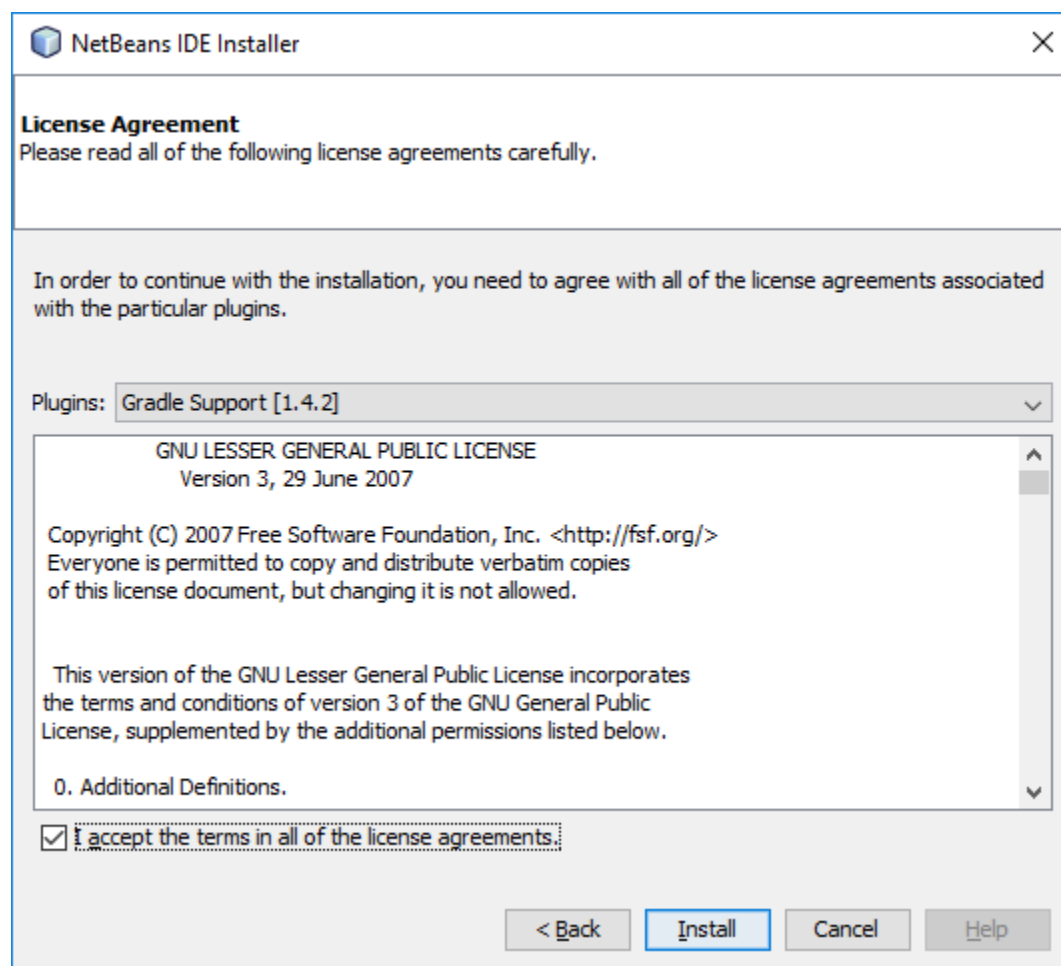


Inštalácia pluginu Graddle Support

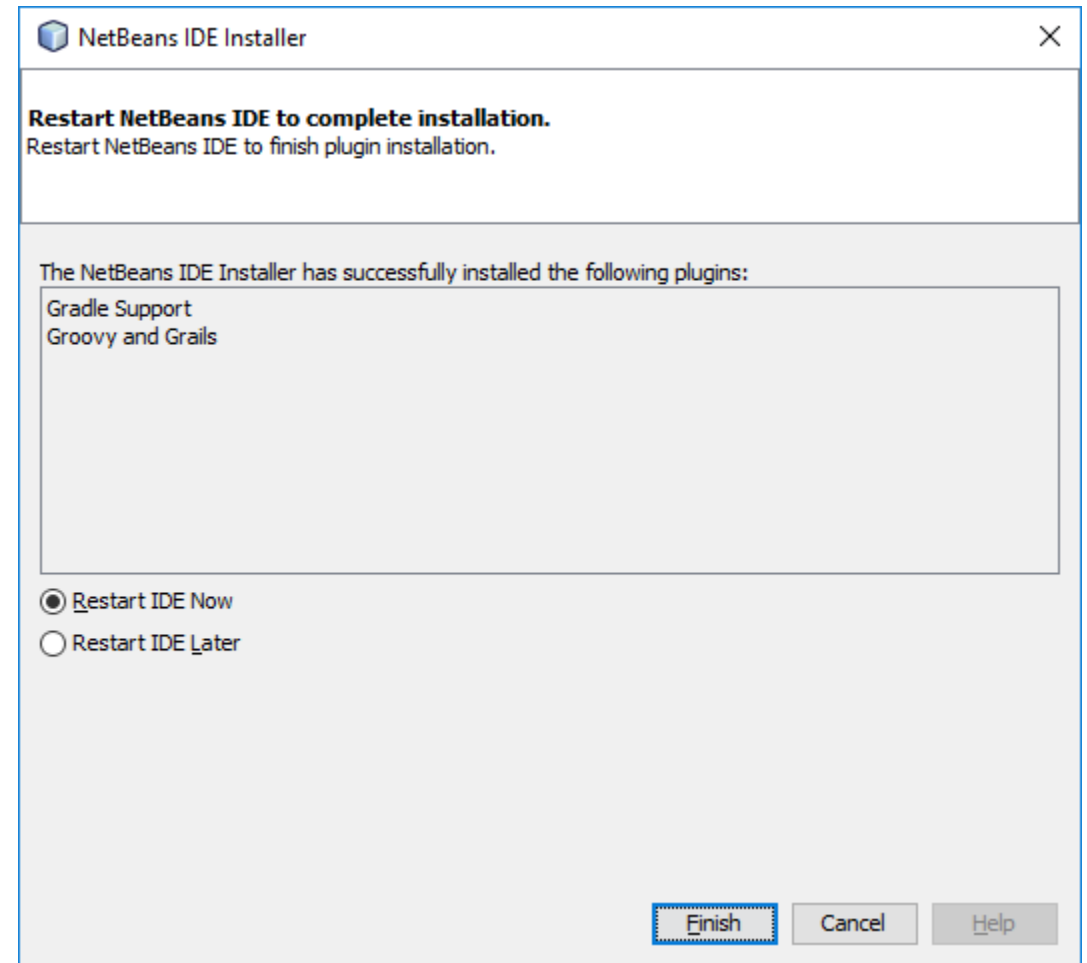
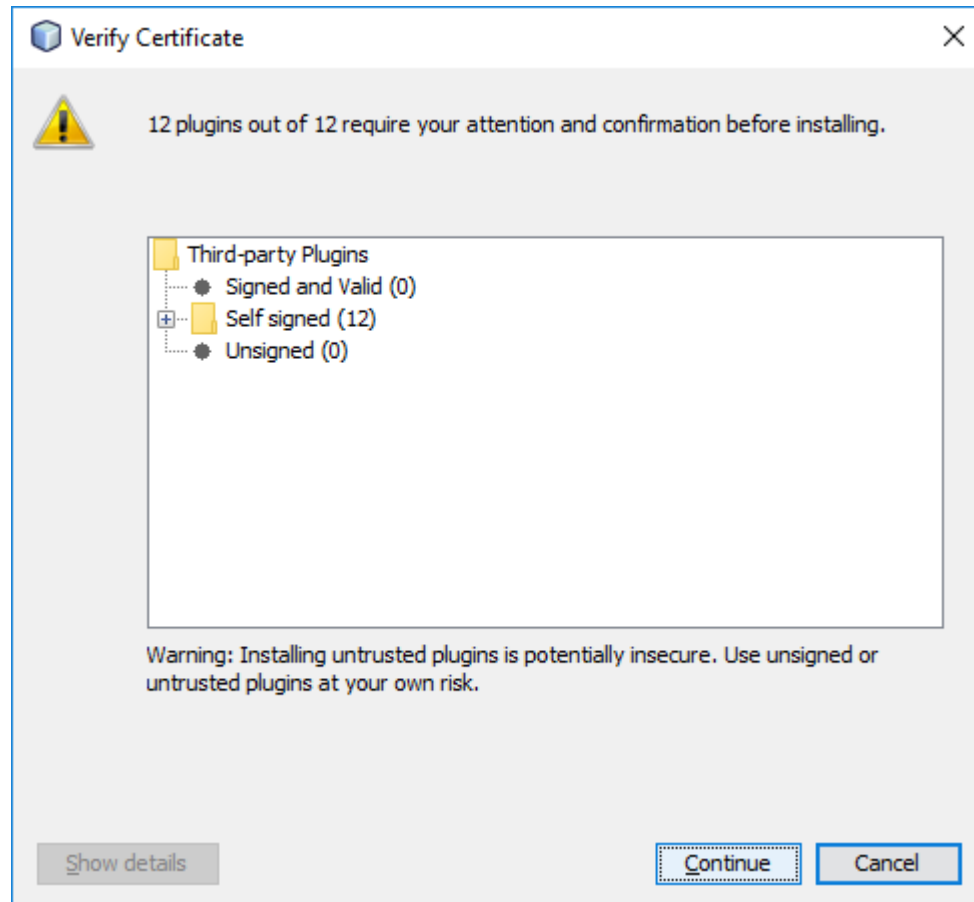
- V Netbeanse menu **Tools – Plugins – Available Plugins**:



Odklikáme sprievodcu

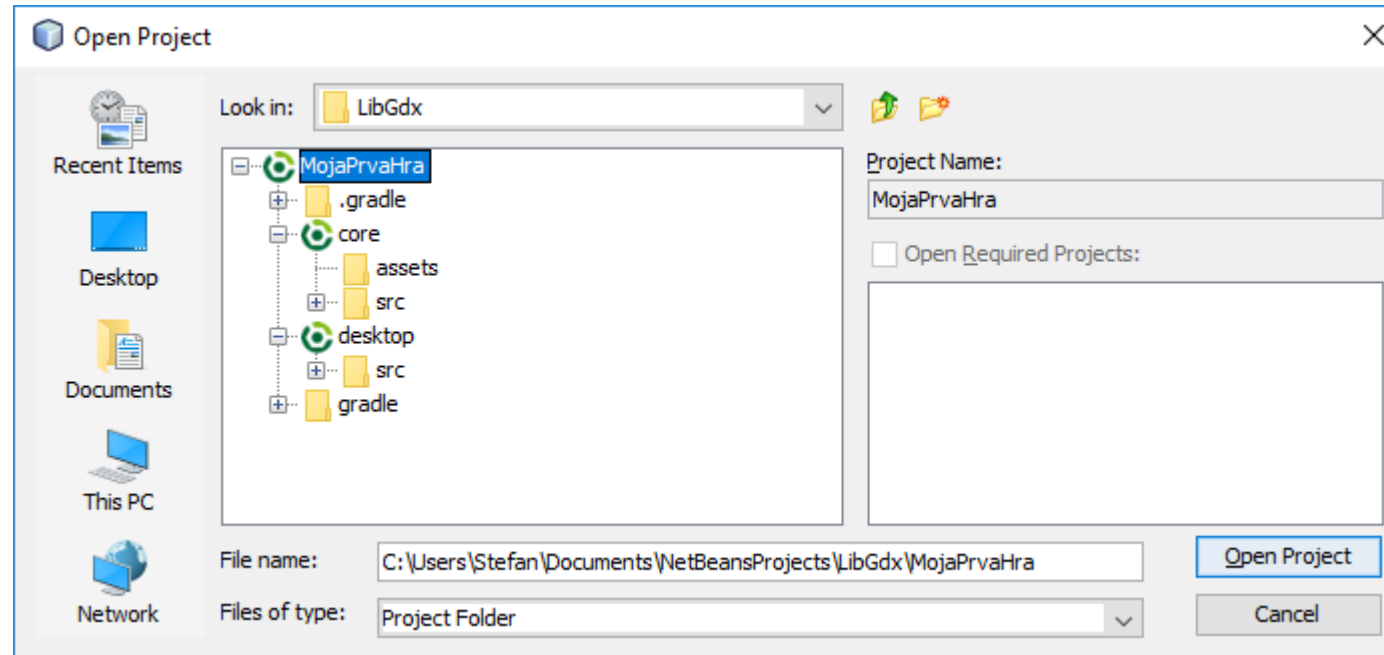


Odklikáme sprievodcu a reštartujeme NetBeans



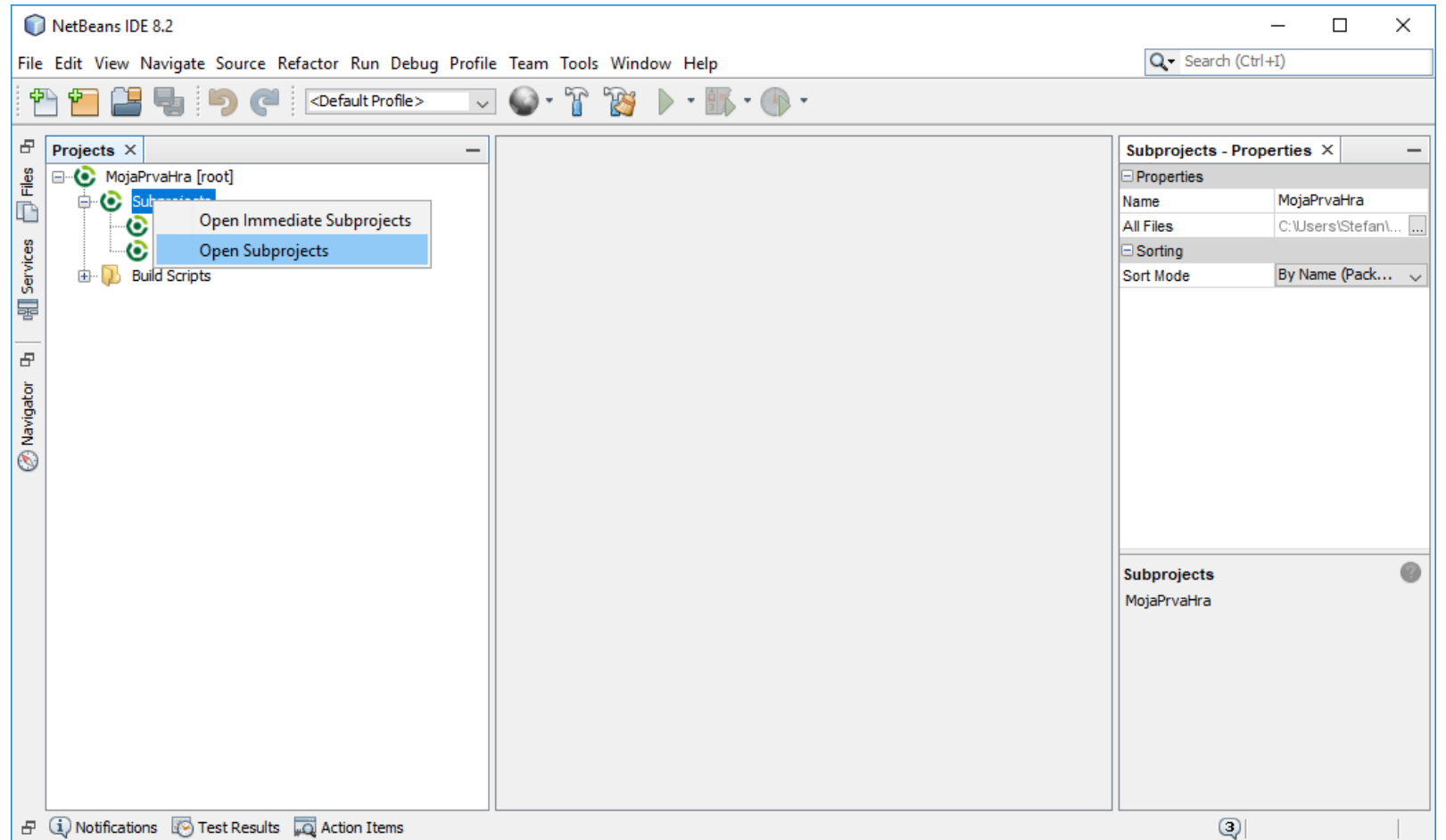
Otvoríme projekt

- V Netbeanse menu **File – Open Project...**

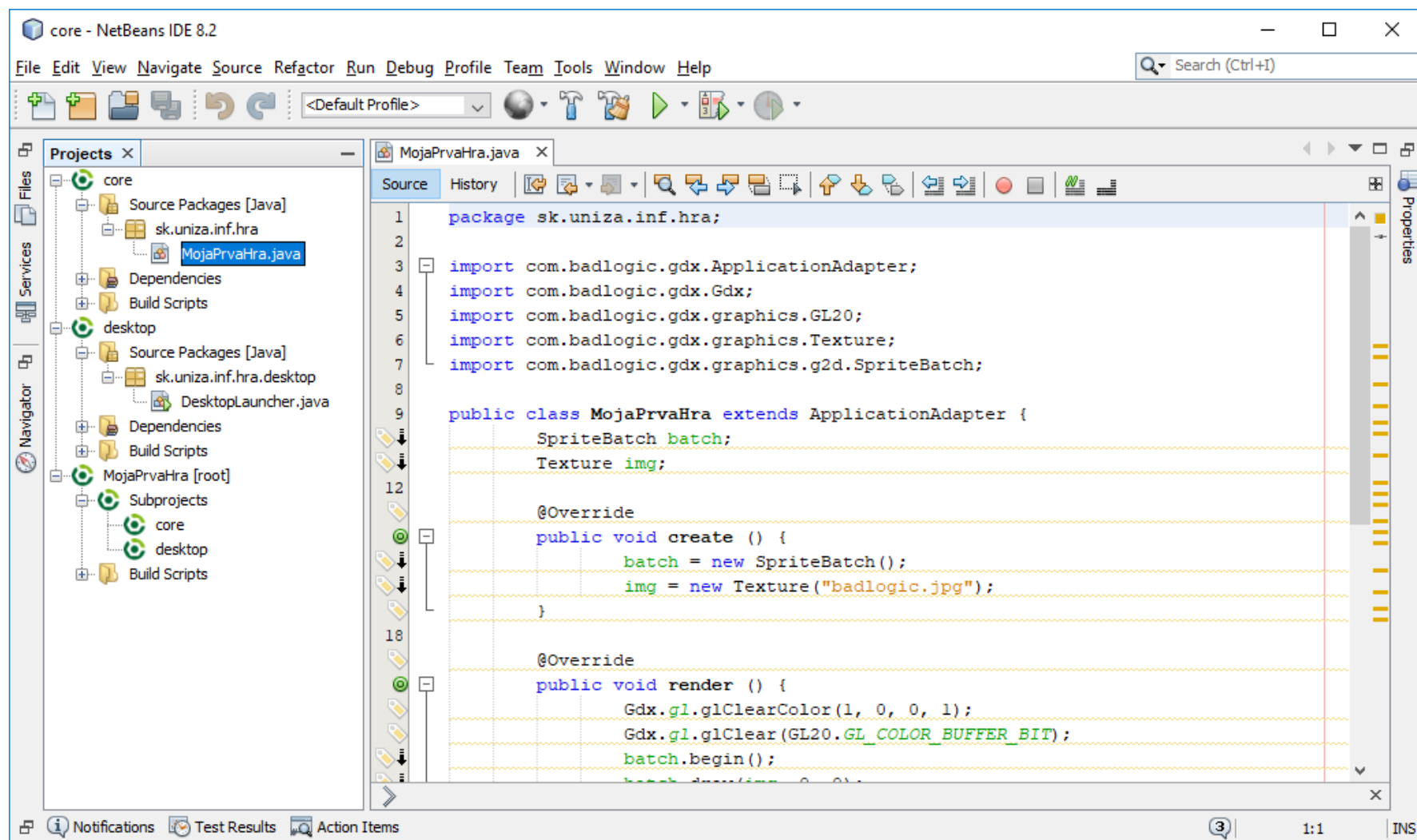


Otvoríme podprojekty

- **Otvoriť** projekty môžeme cez **kontextové menu nad Subprojects** (cez pravé tlačidlo myši), kde zvolíme **Open subprojects**
- Alebo **dvojkliknutím** na projekt core a desktop



Preskúmame vygenerované súbory



Zdrojový kód

```
public class MojaPrvaHra extends ApplicationAdapter {
    private SpriteBatch batch;
    private Texture textura;

    @Override
    public void create() {
        // Inicializujeme objekty, načítame obrázok
        this.batch = new SpriteBatch();
        this.textura = new Texture("badlogic.jpg");
    }

    @Override
    public void render() {
        // Vymažeme a prekreslíme celé okno červenou farbou
        Gdx.gl.glClearColor(1, 0, 0, 1);
        Gdx.gl.glClear(GL20.GL_COLOR_BUFFER_BIT);

        // Vykreslíme textúru obrázka na pozíciu (X:0, Y:0)
        this.batch.begin();
        this.batch.draw(this.textura, 0, 0);
        this.batch.end();
    }

    @Override
    public void dispose() {
        // Uvoľníme alokované objekty
        this.batch.dispose();
        this.textura.dispose();
    }
}
```

```
public class DesktopLauncher {
    public static void main(String[] arg) {
        LwjglApplicationConfiguration config = new LwjglApplicationConfiguration();
        new LwjglApplication(new MojaPrvaHra(), config);
    }
}

public abstract class ApplicationAdapter implements ApplicationListener {
    @Override
    public void create () { // Vytvorenie aplikácie - načítame objekty do pamäte
    }

    @Override
    public void resize (int width, int height) { // Vyvolá sa po zmene veľkosti okna
    }

    @Override
    public void render () { // „Herná slučka“ - cyklické vykresľovanie grafiky na obrazovku
    }

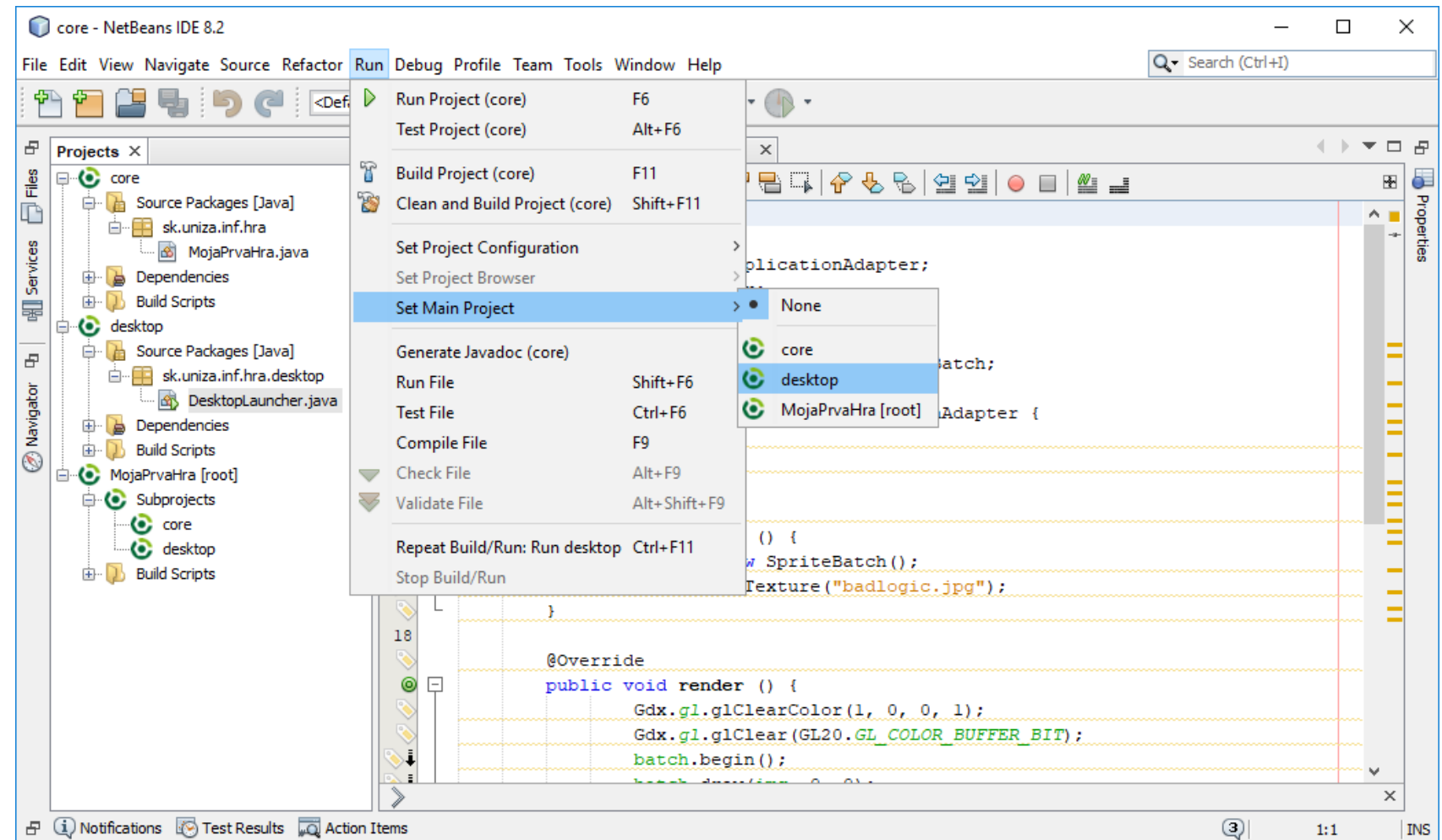
    @Override
    public void pause () { // Vyvolá sa pri pozastavení aplikácie (dostupné v Androide)
    }

    @Override
    public void resume () { // Vyvolá sa pri znovu spustení aplikácie (dostupné v Androide)
    }

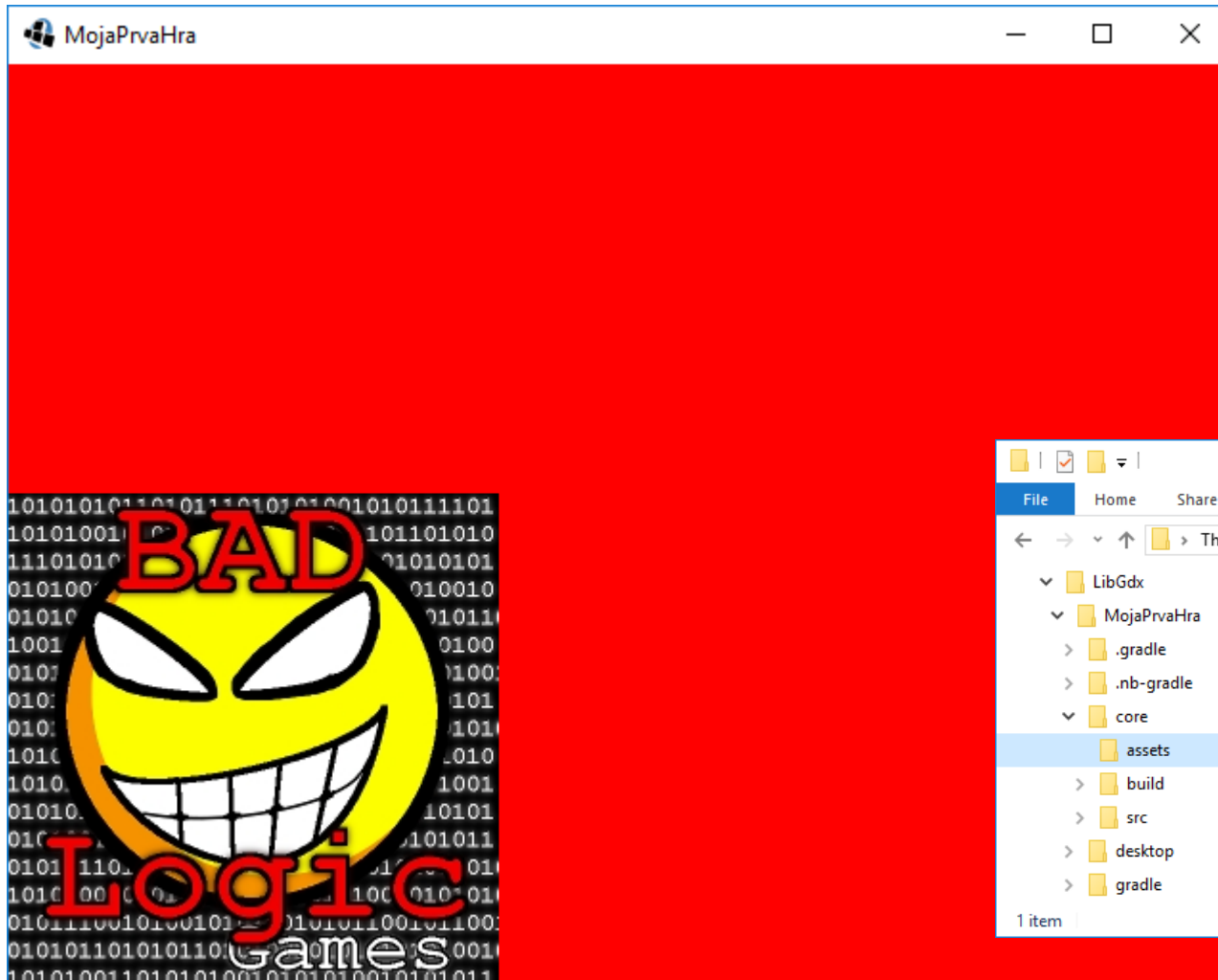
    @Override
    public void dispose () { // Uvoľníme objekty z pamäte
    }
}
```

Nastavenie hlavné projektu

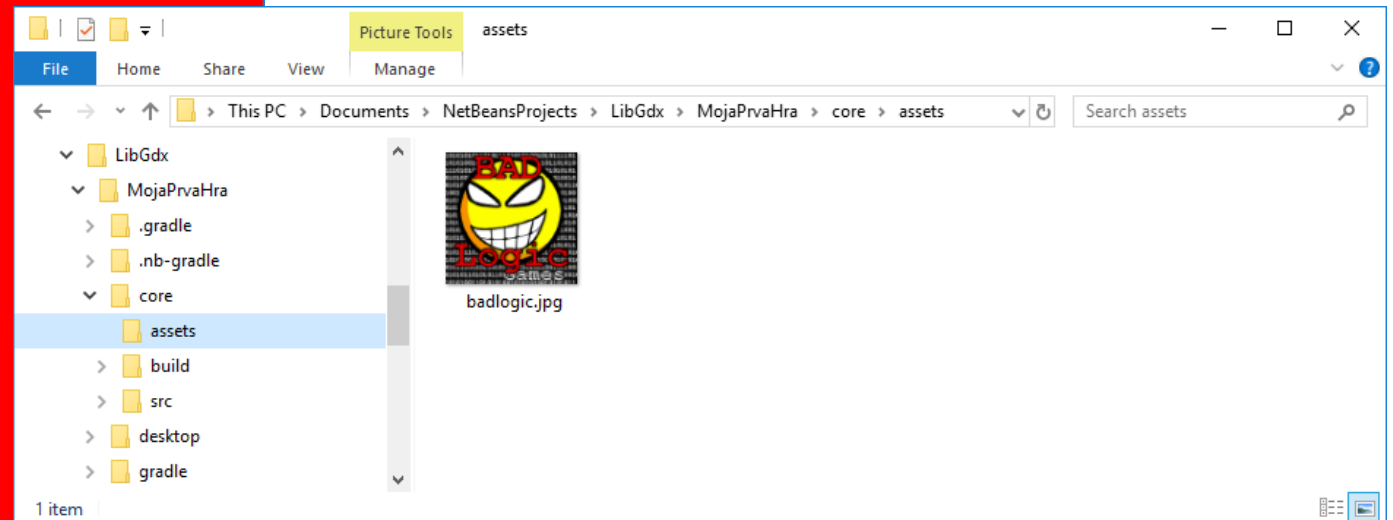
- Cez menu **Run – Set Main Project** si môžeme nastaviť projekt **desktop**, aby sa vždy spúšťal, aj keď budeme upravovať súbory patriace do iných otvorených projektov Netbeansu



Po spustení – F6

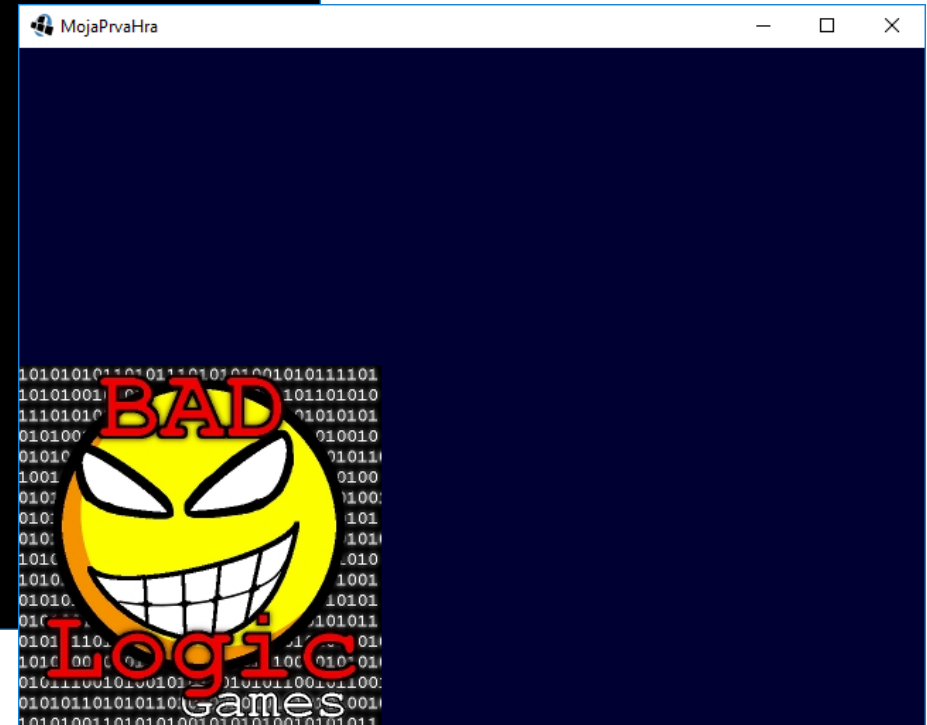
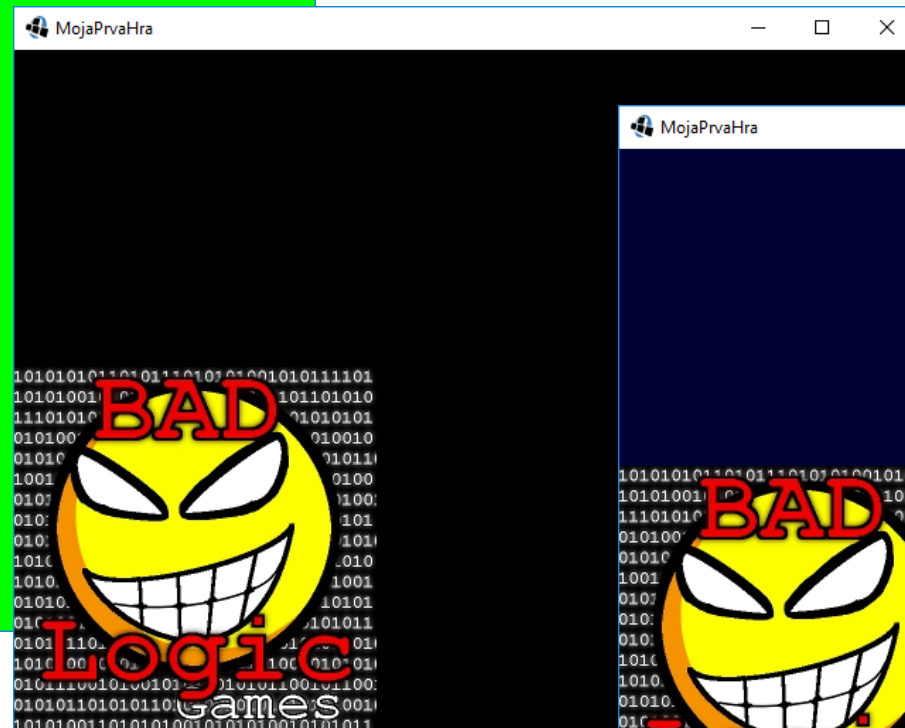
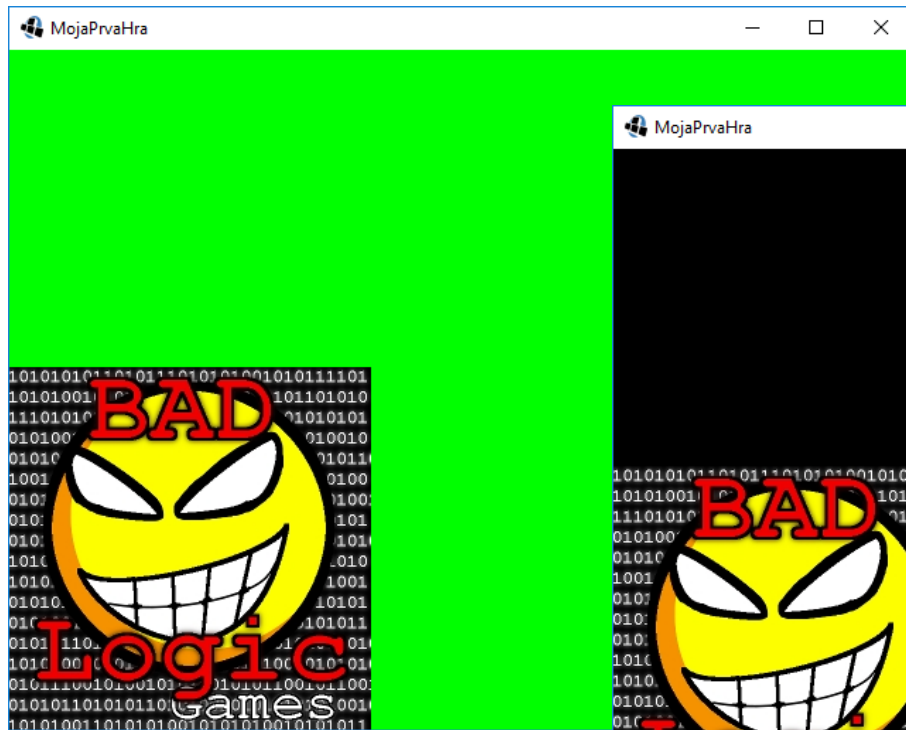


- Obrázok je získaný z priečinka **assets** – do neho môžete pridať ďalšie súbory, ktoré chcete používať v hre (zvuky, obrázky, fonty, ...)



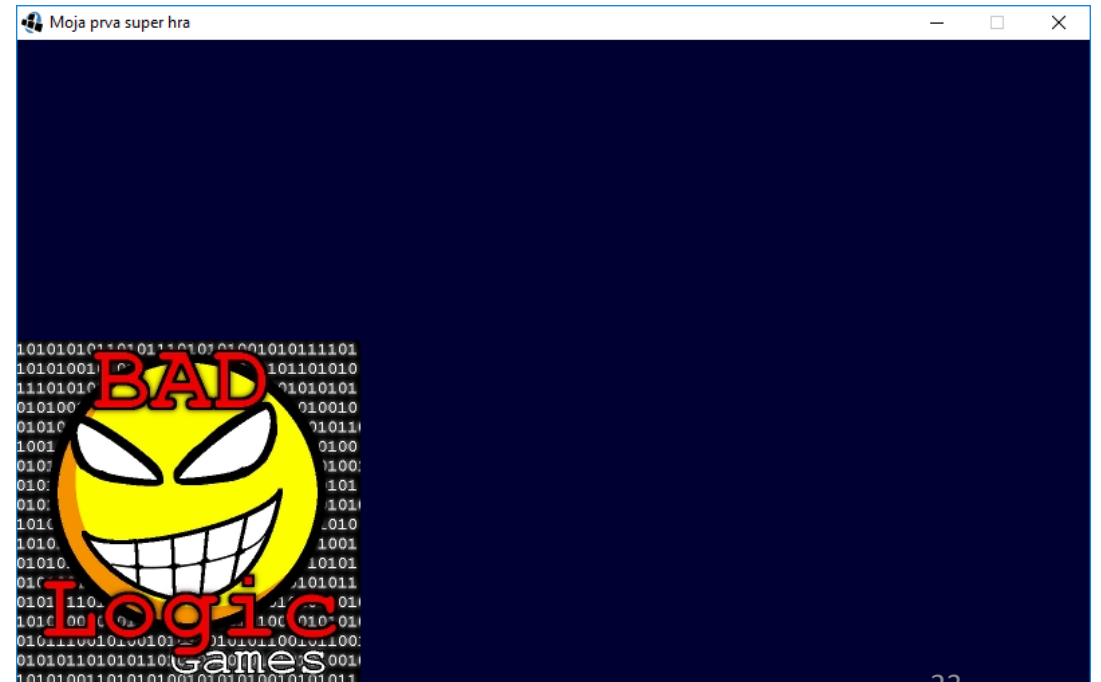
Úloha – zmeňte farbu pozadia

- Zmeňte farbu pozadia na **zelenú**, **čiernu** alebo **tmavomodrú**



Úloha – zmeňte konfiguráciu okna

```
public class DesktopLauncher {  
  
    public static void main(String[] arg) {  
        LwjglApplicationConfiguration config = new LwjglApplicationConfiguration();  
        config.title = "Moja prva super hra";  
        config.width = 800;  
        config.height = 480;  
        config.resizable = false;  
        config.fullscreen = false;  
  
        new LwjglApplication(new MojaPrvaHra(), config);  
    }  
}
```



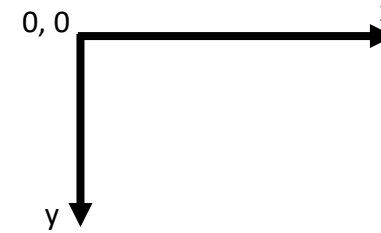
Úloha – zmeňte umiestnenie obrázka

- Vykreslite obrázok na **stred okna**

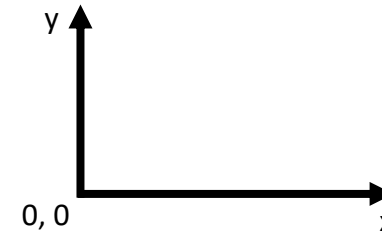


Súradnicové systémy

- Súradnice **dotykov** alebo **myšky**
 - začínajú v ľavom **hornom rohu (y-down)**



- Súradnice **obrazovky** alebo **obrázka**
 - začínajú v ľavom **dolnom rohu (y-up)**
 - $y = \text{Gdx.graphics.getHeight()} - 1 - \text{poziciaMysiY}$;



Úloha – vykreslite obrázok viackrát



Úloha – vykreslite obrázok viackrát

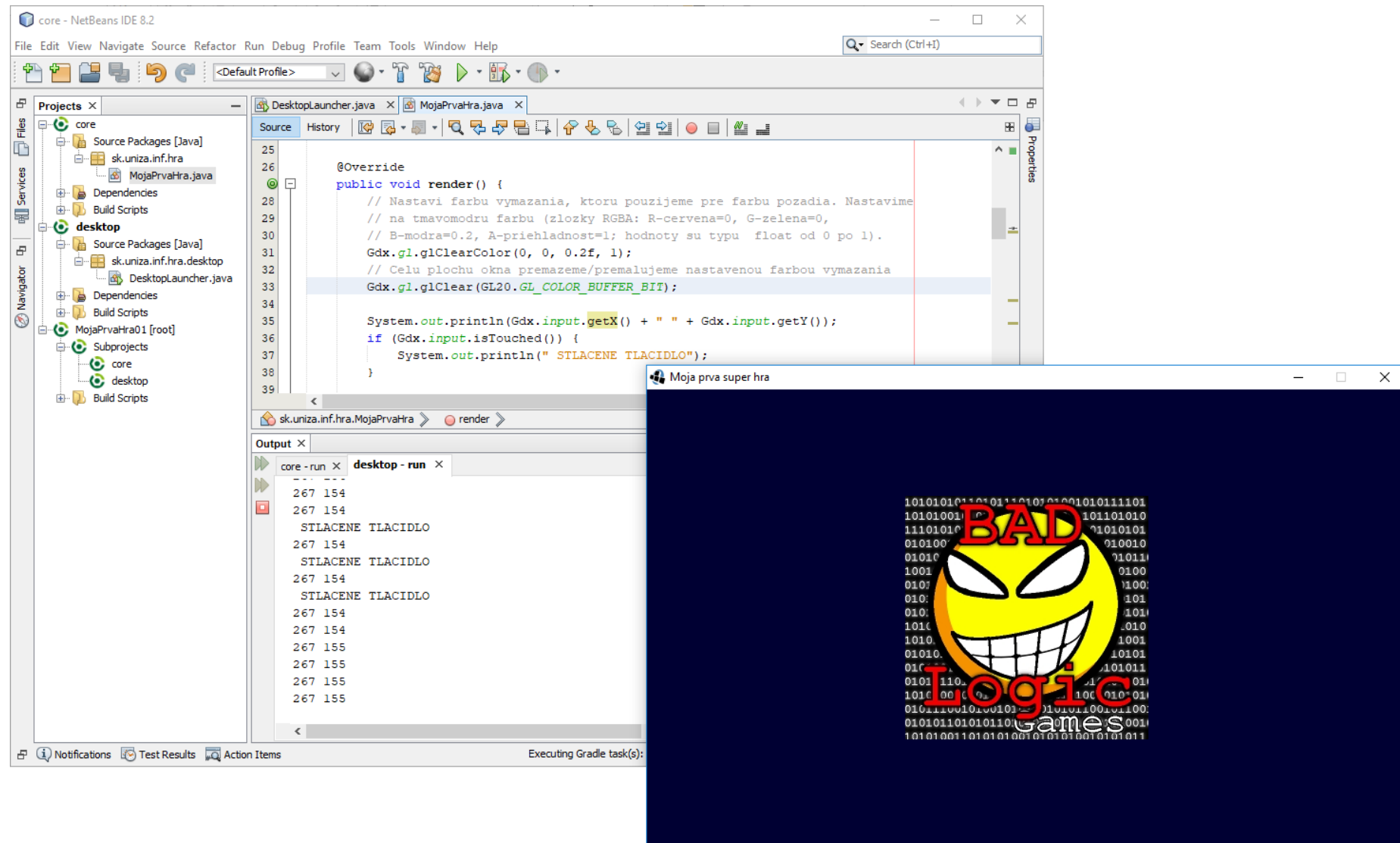


Obsluha myši, dotykov a klávesnice

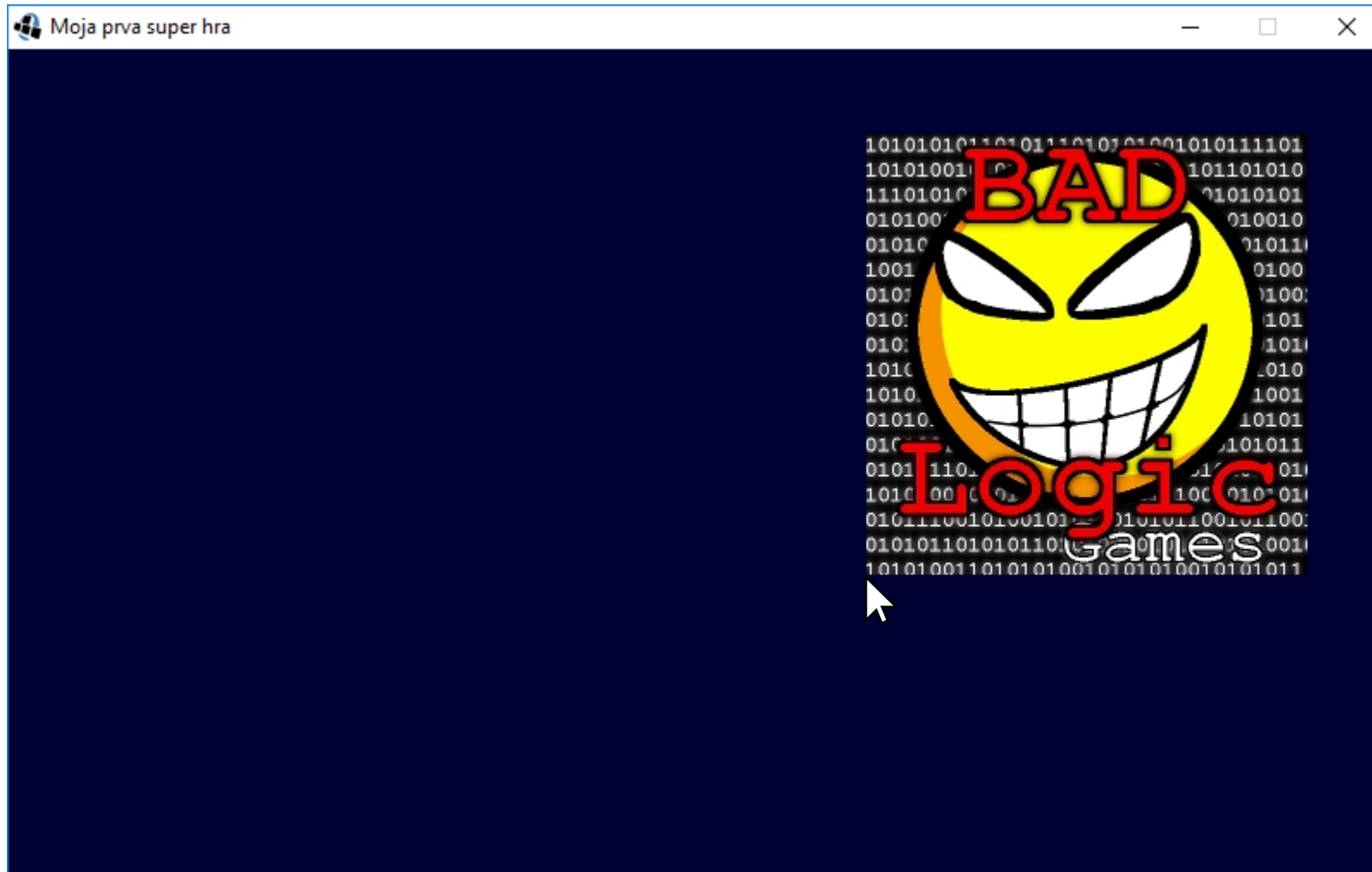
- `Gdx.input.justTouched()` – vráti `true`, ak bol vykonaný dotyk alebo kliknutie myšou na obrazovku
- `Gdx.input.isTouched()` – vráti `true`, ak bol a stále je dotyk, alebo stlačené tlačidlo myši na obrazovke
- `Gdx.input.isButtonPressed(Input.Buttons.RIGHT)` – vráti `true`, ak je stlačené pravé tlačidlo na myške
- `Gdx.input.getX()`, `Gdx.input.getY()` – súradnice pozície myšky alebo dotyku
- `Gdx.input.isKeyJustPressed(Input.Keys.UP)` - vráti `true`, ak bola práve stlačená klávesa (šípka hore)
- `Gdx.input.isKeyPressed(Input.Keys.LEFT)` – vráti `true`, ak bola a je stlačená klávesa (šípka vľavo)
- ...

Práca s myšou

- isTouched()
- getX()
- getY()

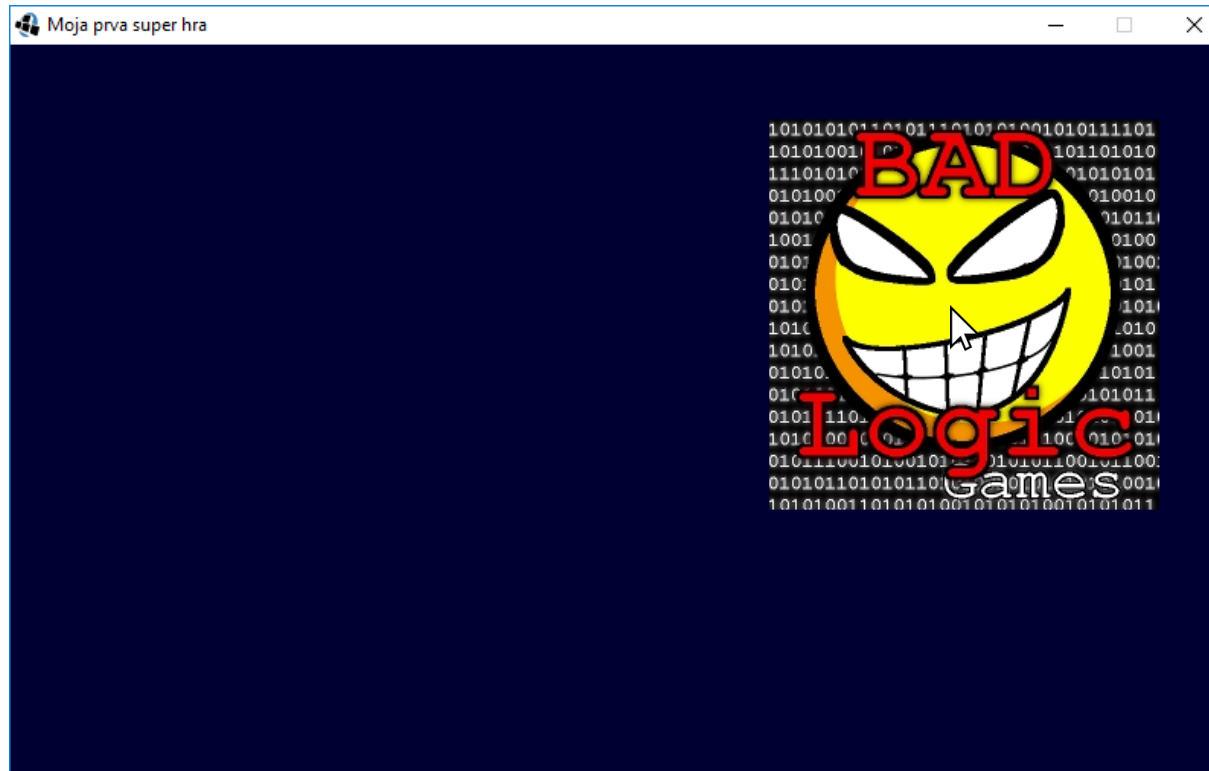


Úloha (myš) – zmeňte pozíciu obrázka na miesto kliknutia pravým tlačidlom myši



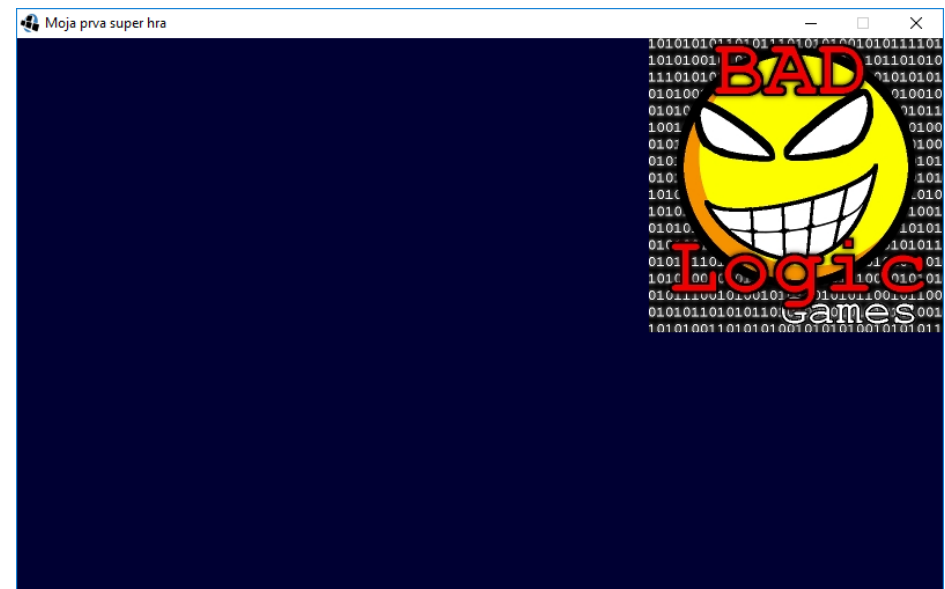
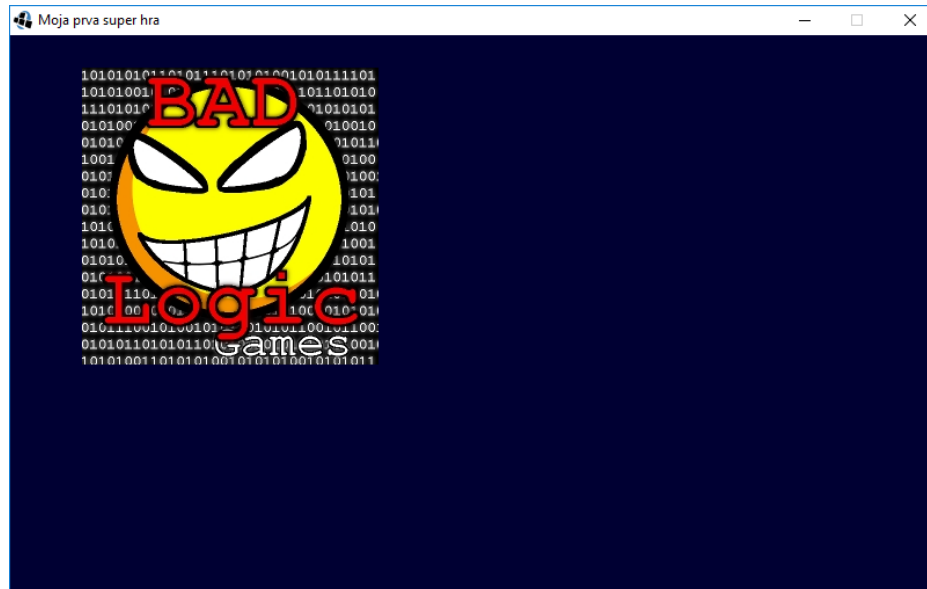
Úloha (myš) – zmeňte pozíciu obrázka na miesto kliknutia pravým tlačidlom myši

- Miesto kliknutia bude stredom obrázka
- **Zabezpečte, aby obrázok nevyšiel mimo obrazovky** (kontrolujte hranice)



Úloha (klávesnica) – meňte pozíciu obrázka pomocou klávesov

- **Zmeňte pozíciu obrázka aj po stlačení kláves hore, dole, vpravo a vľavo** tak, aby sa obrázok posúval vyššie, nižšie, doprava a doľava
- **Zabezpečte, aby obrázok nevyšiel mimo obrazovky** (kontrolujte hranice)

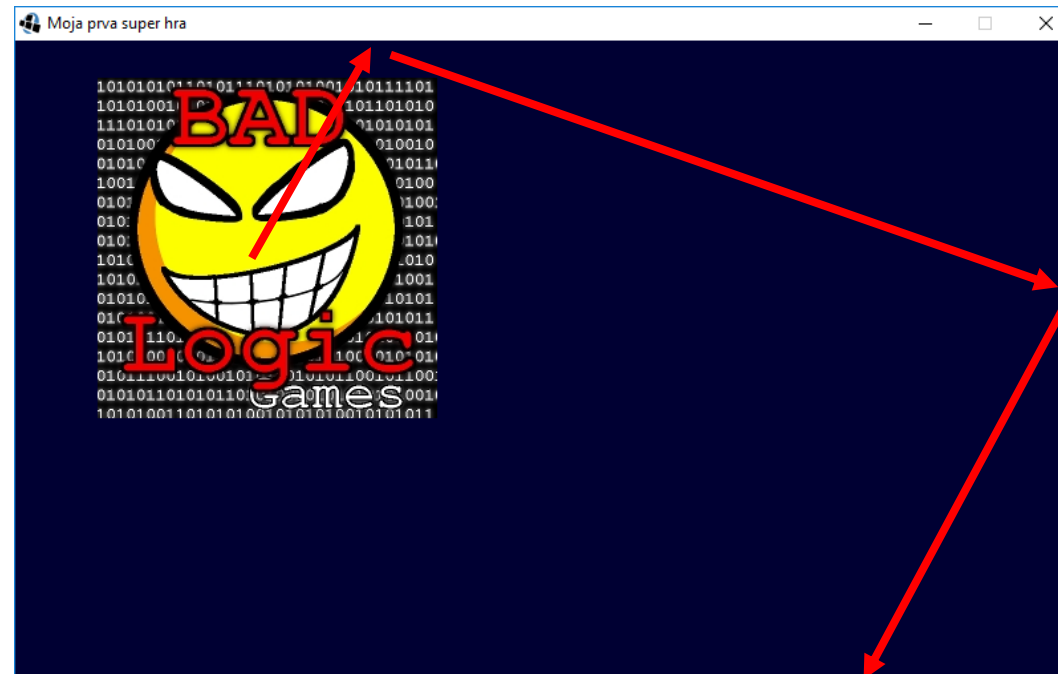


FPS a herný čas

- `Gdx.graphics.getFramesPerSecond()` – vráti priemerný počet snímkov za sekundu (FPS – frames per second), t. j. **počet vykonaní metódy `render()` za sekundu**
- `Gdx.graphics.getDeltaTime()` – vráti čas v sekundách medzi aktuálnym vykreslovaným snímkom a predchádzajúcim snímkom
 - Vďaka tomu môžeme meniť pozíciu objektov nezávisle od FPS
 - Príklad: zmena pozície o 200 jednotiek za jednu sekundu
`this.aktualnaPoziciaX += 200 * Gdx.graphics.getDeltaTime();`

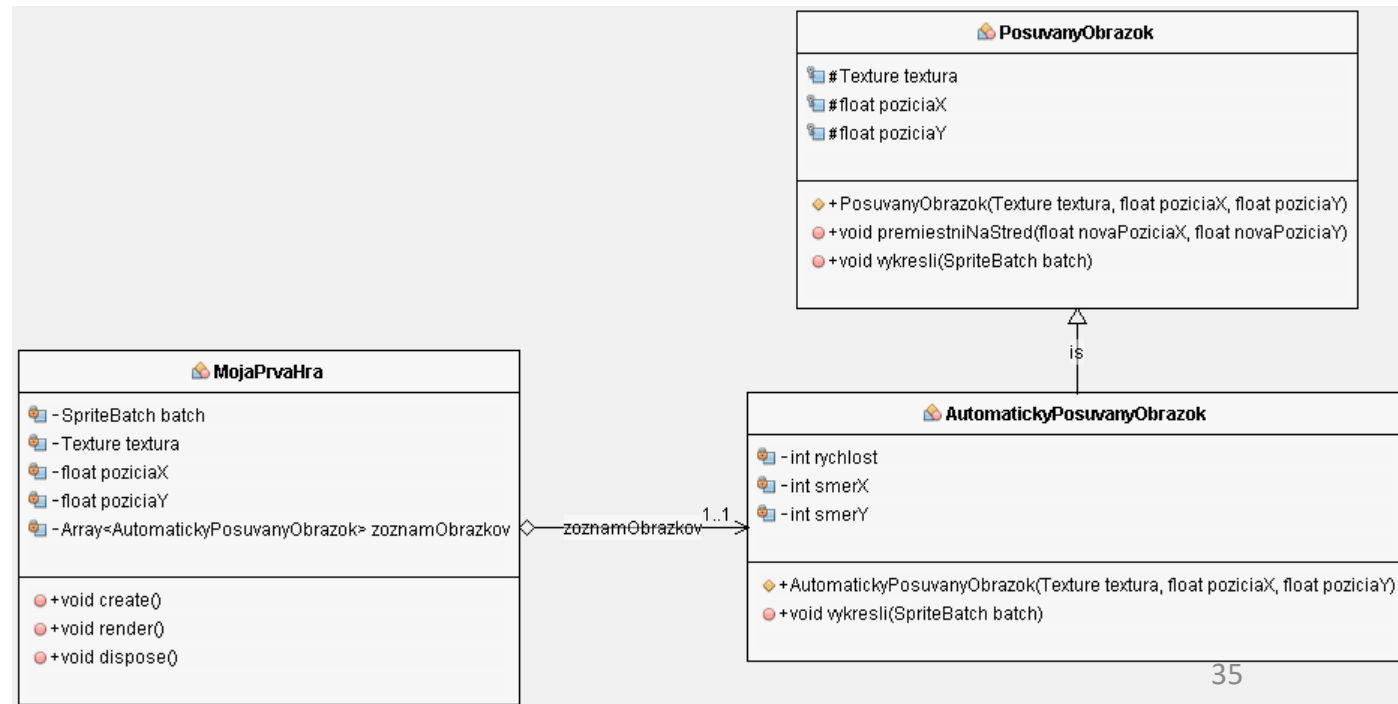
Úloha – automatické posúvanie a odrážanie obrázka od stien

- Umožnite, aby sa **obrázok posúval automaticky** s vopred definovaným smerom a rýchlosťou, **až dokiaľ nedosiahne hranice** okraja okna – vtedy sa „odrazí“ na opačnú stranu (hore – dole, dole – hore, vpravo – vľavo, vľavo – vpravo)



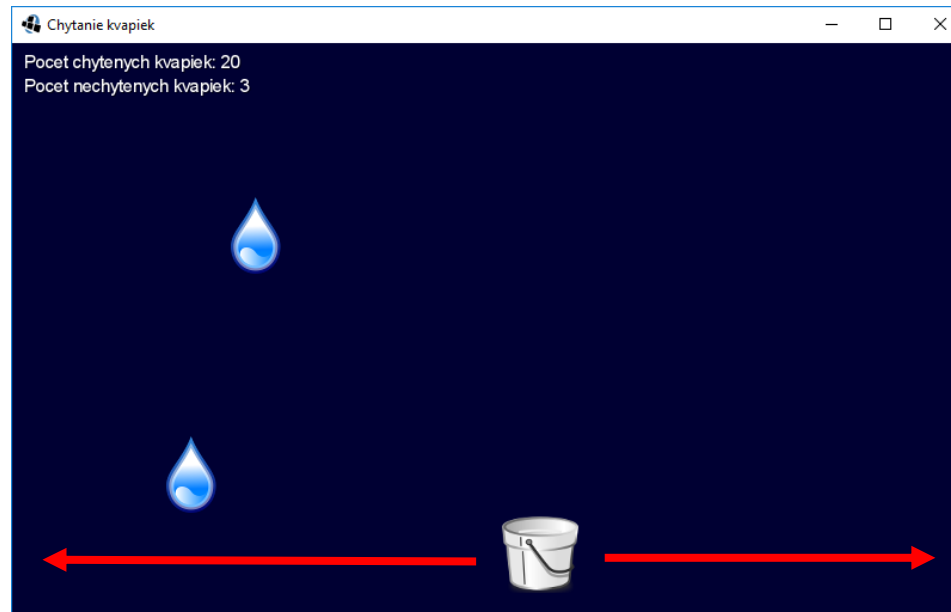
Úloha – vytvorenie ďalších automaticky premiestňovaných obrázkov

- **Vykreslite ďalší obrázok,** ktorý sa vytvorí na mieste kliknutia ľavým tlačidlom myši (alebo dotyku) a ktorý sa od tohto miesta bude **automaticky posúvať** náhodne **vygenerovanou rýchlosťou a smerom, až dokiaľ nedosiahne hranice okraja okna**



Úloha – nový projekt hry

- Vygenerujte nový projekt a novú hru, ktorej cieľom bude chytať padajúce kvapky vody vedrom posúvajúcim sa po X-ovej súradnici



- Postup:
 - <https://github.com/libgdx/libgdx/wiki/A-simple-game>
 - <https://github.com/libgdx/libgdx/wiki/Extending-the-simple-game>

Použitie kamery na prepočet súradníc

- Kamera a jednoduchšia práca so súradnicami – matice:

```
private OrthographicCamera camera;
...
camera = new OrthographicCamera();
camera.setToOrtho(false, 800, 480);
...
batch.setProjectionMatrix(camera.combined);
batch.begin();
...
if (Gdx.input.isTouched()) {
    Vector3 touchPos = new Vector3();
    touchPos.set(Gdx.input.getX(), Gdx.input.getY(), 0);
    camera.unproject(touchPos); // touchPos bude obsahovať prepočítané súradnice
    bucket.x = touchPos.x - 64 / 2;
}
```