Samuel Řeřicha
AIS ID 103110

Dokumentácia projektu

Semestrálny projekt

VAVA

Samuel Řeřicha

AIS ID 103110

Samuel Řeřicha
AIS ID 103110

Zámer projektu:

Čo je to za aplikáciu?

Aplikácia bude iteráciou už overenej formuly, ktorej prvý a nemenný základ vznikol už takmer pred 40 rokmi, ešte na stroji s text-only displejom a veľmi obmedzenými výpočtovými schopnosťami.
Implementácia bude inšpirovaná hlavne mechanikami prvotných implementácií, ale bude obsahovať aj zaujímavé prvky z novších hier a prípadne nové mechaniky alebo ich kombinácie.

Kto bude aplikáciu používať?

Hocikto, kto má rád puzzle games a/alebo špecificky Tetris. Vďaka rôznym úrovniam rýchlosti, rôznym nastaveniam a herným módom bude hra dostupná a zaujímavá pre úplných nováčikov, ako aj pre skúsených hráčov.

Kde ju bude používať?

Aplikácia bude vyvýjaná najmä pre PC, avšak by bola možná implementácia pre iné platformy podporované Javou, pre ne by bolo potrebné iba upraviť alebo doimplementovať platform-specific I/O handling.

Kedy a prečo ju používať?

Tetris je zábavná, pohlcujúca a možno prekvapivo komplexná hra. Pre jej hranie nie je potrebný žiadny tutoriál, komplexná registrácia alebo čakanie a jednotlivé hry trvajú rádovo len minúty, takže je veľmi dobrým spôsobom, ako sa rýchlo odreagovať a zabaviť bez časového záväzku. Avšak mimo časovej tiesne je možné pri nej stráviť omnoho viac času.

Koľko bude aplikácia stáť?

Jedná sa o “passion project”, aplikácia bude zadarmo. Keďže bude bežať lokálne, nie je potrebné starať sa o servery ani o nijaké iné prostriedky, ktoré by bolo treba manažovať alebo financovať. Aplikácia nebude obsahovať žiadne mikrotransakcie alebo unlockable content, ktorý by mohol byť kúpený.

Ako sa aplikácia bude používať?

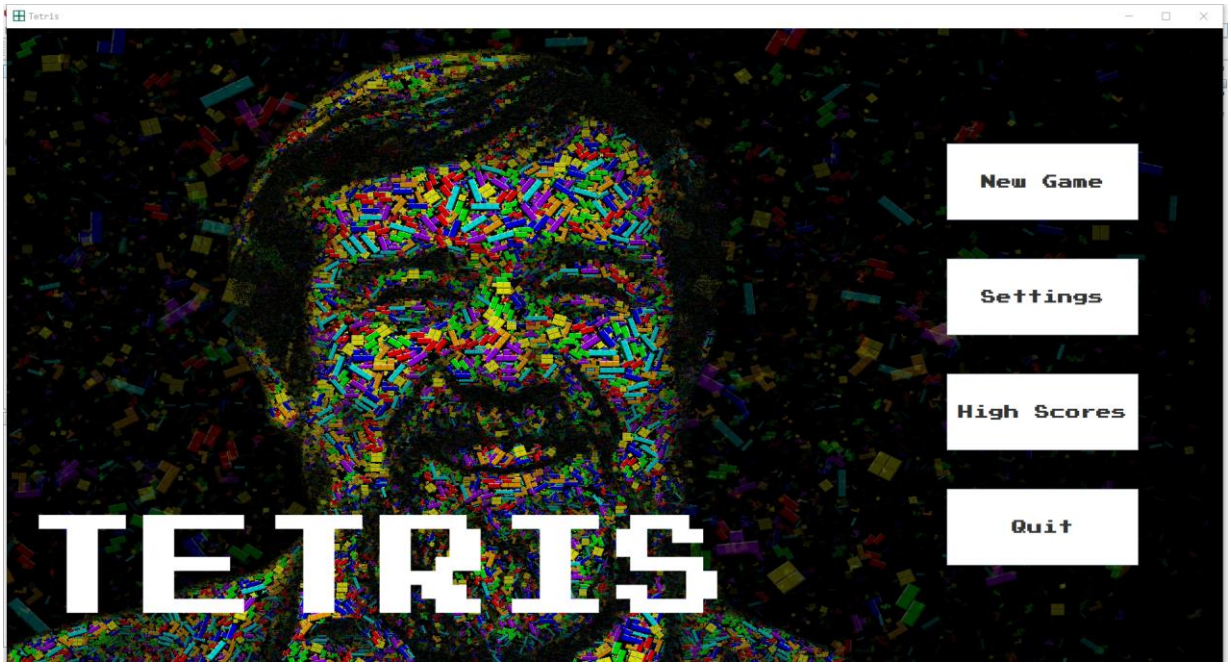
Aplikácia bude obsahovať hlavnú hernú časť, ale aj nastavenia (týkajúce sa výzoru, ale aj mechaník hry) a high score leaderboardy pre rôzne herné módy a nastavenia. Je plánovaný aj log-in alebo account systém, takže viacerí hráči na tej istej aplikácii budú môcť používať rôzne nastavenia a budú mať vlastné leaderboardy (logovanie, XML – uložené settings a high scores pre rôzne účty).
Nastavenia budú obsahovať aj výber jazyka, čím bude dosiahnutá feature internacionalizácia. I/O bude samozrejme použité pri vstupe a výstupe, kolekcie budú pri implementácii použité tam, kde budú potrebné alebo vhodné.

Samuel Řeřicha
AIS ID 103110

Výsledná implementácia:

Funkcionalita:

Aplikácia umožnuje hranie hry Tetris v rôznych módoch. Používateľ má vďaka nastaveniam možnosť si prispôsobiť skúsenosť. Vďaka použitiu custom retro fontu, hudby, zvukov a textúr a wallpaperov zo starších hier má hra celkom nostalgický nádych. Bola snaha spraviť okná resizable, avšak pri menších alebo "nepravidelných" veľkostiach okna je výzor horší.



Po zapnutí hry má používateľ možnosť začať novú hru, zmeniť nastavenia, prezrieť si rebríčky a samozrejme ukončiť aplikáciu.
Hra obsahuje tri herné módy s rôznymi pravidlami: ***Classic***, ***Modern*** a ***Wonky***.

Classic využíva staršiu (viac náhodnú) generáciu kúskov a nedovoľuje použitie novších features ako sú "držanie" kúsku alebo insta-drop.

Modern využíva novšiu (7-bag, počet všetkých kúskov je vždy takmer rovnaký) generáciu kúskov a povoľuje využitie novších features.

Wonky využíva novšiu generáciu kúskov a novšie features, šírka plochy jej však 20 a sú generované aj "čudné" kúsky tvorené 5 tiles, ktoré značne sťažujú hru.

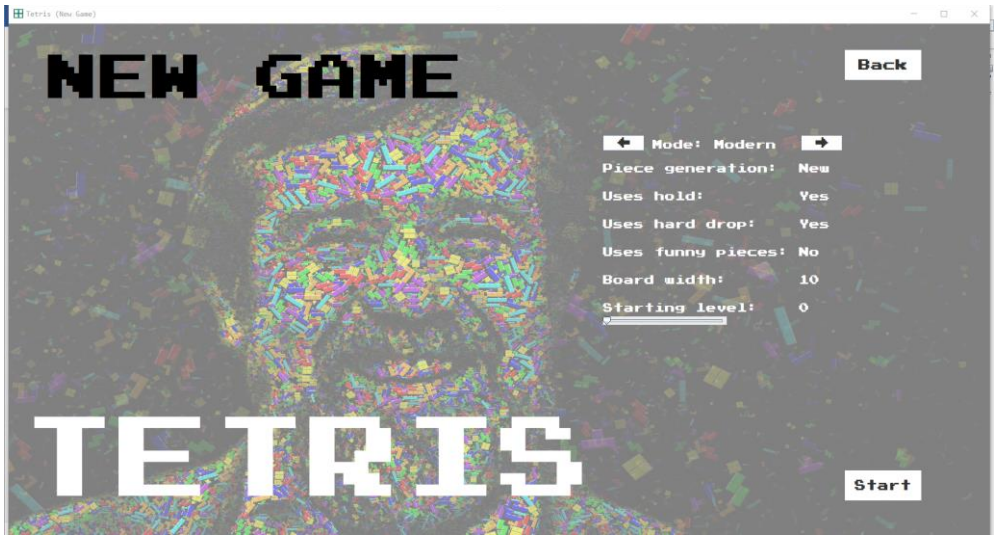
Počas tvorenia aplikácie bola plánovaná možnosť tvorenia nových módov používateľom kombináciou existujúcich pravidiel a implementácia tam čiastočne je (constructor, XML serializácia, implementácia fixovaná na pravidlá, nie na módy), avšak z časových dôvodov GUI pre tvorbu módov nebolo implementované.

Samuel Řeřicha
AIS ID 103110

New game:

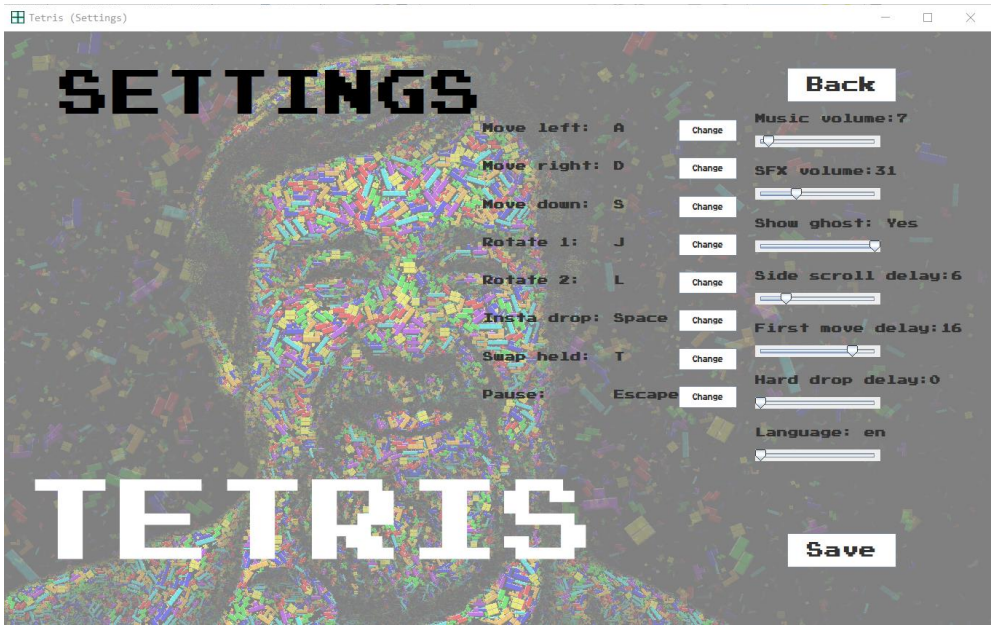
V časti **new game** je používateľovi umožnené vybrať si herný mód a taktiež štartovací level. Level progression v hre je viazaná na lines cleared a na starting level.

```
current_level = Math.max(current_level,
                         line_count / GameConsts.LINES_PER_LEVEL);
```



Settings:

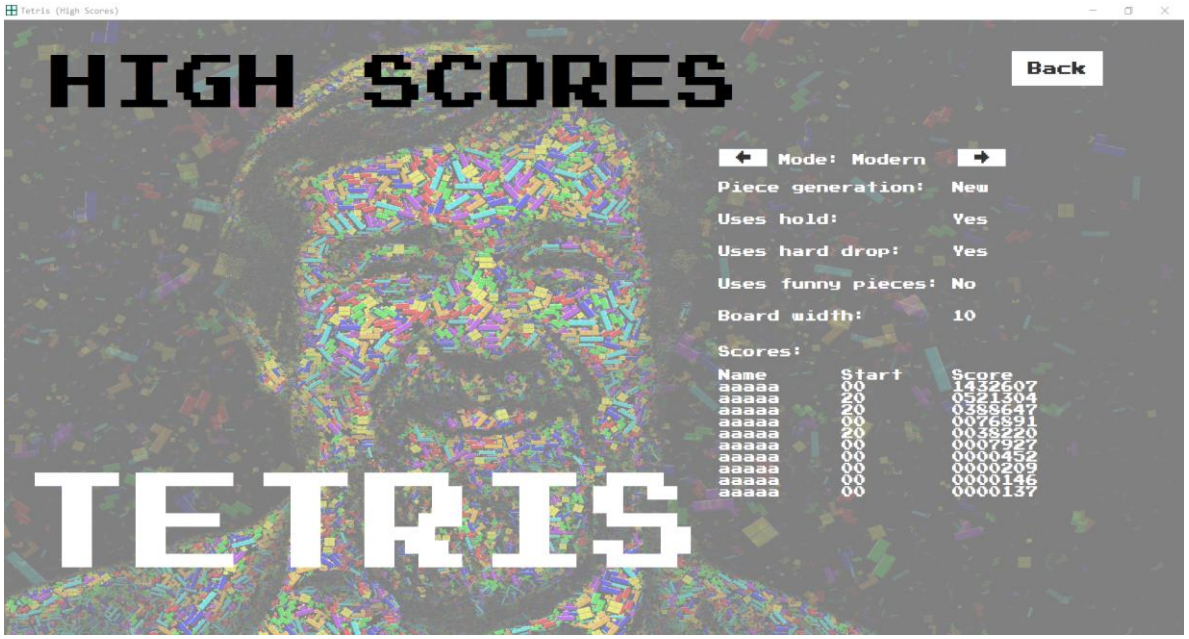
V nastaveniach je používateľovi umožnené zmeniť keybindy pre hru, music a sound effect volume a tiež aj určité parametre týkajúce sa rýchlosti a pohybu kúskov v hre.



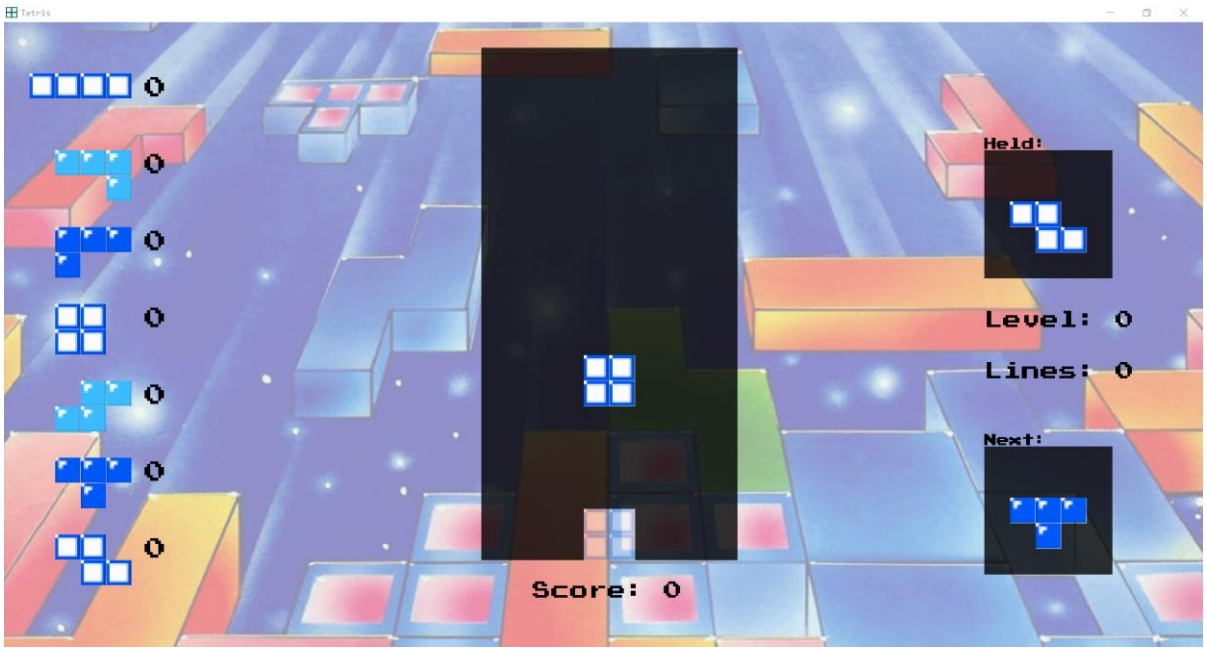
Samuel Řeřicha
AIS ID 103110

High scores:

V tejto časti je používateľovi umožnené prezerať si vysoké skóre pre existujúce herné módy.



Game screen:



Herná obrazovka obsahuje informácie o počte doteraz padnutých kúskov (užitočné pri staršej generácii kúskov, ponechané aj pri novšej generácii a pri použití "čudných kúskov" z estetických dôvodov. Tiež obsahuje informáciu o nasledujúcom kúsku, držanom kúsku, skóre, leveli a počte vyčistených riadkov. Tetris kúsky tiež menia výzor v závislosti od súčasného levelu.

Samuel Řeřicha
AIS ID 103110

Dôležité / zaujímavé časti logiky:

Všetky Tetris kúsky dedia od triedy **Piece**, ktorá implementuje všetky metódy potrebné pre logiku, ako sú pohyb do strán, rotácia alebo position checky. Tvar a rotácie špecifických kúskov sú definované v konkrétnych triedach veľmi efektívne, čo tiež umožnilo rýchlu implementáciu "čudných" kúskov.

Príklad – definícia jedného z kúskov.

```
public class PieceJ extends Piece {

    public PieceJ (int board_w) {
        this.tilecount = 4;
        this.rotationcount = 4;
        this.start_x = board_w/2;
        this.start_y = 0;
        this.current_state = this.start_state = 3;
        this.piece_id = GameConsts.ID_PIECE_J;
        initPieceStates();
    }

    @Override
    public void pieceSetStates() {
        this.states[0].setTiles(0, 0, 0, -1, 0,  1, -1,  1);
        this.states[1].setTiles(0, 0,  1, 0, -1, 0, -1, -1);
        this.states[2].setTiles(0, 0,  0, 1, 0, -1,  1, -1);
        this.states[3].setTiles(0, 0, -1, 0, 1,  0,  1,  1);
    }

}
```

Herný field je definovaný prostredníctvom triedy **Field,** ktorá manažuje input, pohyb, držanie informácií o ďalšom a držanom kúsku a podobne.

Viď atribúty triedy Field.

```
public class Field extends Thread {
    private int [][] tile_matrix;           // representation of the playfie
    private int width;                      // dimensions of the playfield
    private int height;
    private Piece current_piece;            // piece currently in the field
    private Piece next_piece;               // next piece (is shown in the n
    private Piece held_piece;               // currently held piece, can be
    private int current_level;              // speed level the game's curren
    private int start_level;                // starting level, used when sav
    private int score;                      // current score, used for savin
    private int line_count;                 // lines cleared, used for level
    private PieceFactory piece_gen;         // object responsible for servin
    private PlayScreen parent;              // parent JFrame, provides key i
    private boolean can_swap_held;

    private boolean can_pause;
    private boolean can_soft_drop;          // used for piece movement code
    private boolean can_hard_drop;          // responsive
    private boolean can_rotate_left;        // so it doesnt spin like crazy
    private boolean can_rotate_right;
    private boolean has_moved = false;      // needed to make tucks convenie
    private int move_timer = 0;             // makes sure it doesnt go turbo
    private int hard_drop_timer = 0;        // used to make sure you dont ac
    private int move_timer2 = 0;            // makes sure you dont move twic
    private int fall_timer = 0;             // so it only drops the piece ev

    private int soft_drop_points = 0;       // when soft dropping a piece, y
    private Boolean paused;
    private Boolean game_ended;

    int counters[];
    int piece_count;
```

Samuel Řeřicha
AIS ID 103110

Požiadavky:

Kolekcie:

Pre implementáciu nebolo potrebné značné využitie kolekcií, avšak sú využité prostredníctvom napríklad ArrayList-ov.

Príklad 1 ArrayList keys, ktorý v hernej obrazovke slúži na ukladanie práve stlačených kláves, ktoré sú hernou logikou manažované pre lepší vstup (použitie iba metód typu keyPressed a keyReleased viedlo k ignorovaniu vstupov napr. pri stlačení 3 a viac kláves).

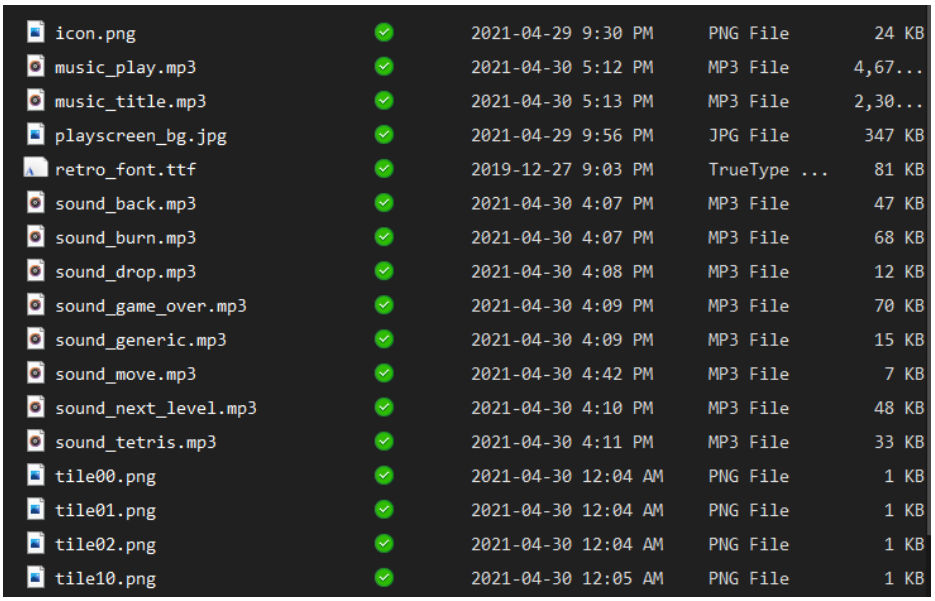
```
*/
public class PlayScreen extends javax.swing.JFrame {

    private Image bg, bg_temp;
    private ArrayList keys;
```

Príklad 2 ArrayList bag, ktorý v PieceFactoryBag obsahuje hodnoty generovaných Tetris kúskov (vedie k rovnomernému, ale stále náhodnému generovaniu Tetris kúskov).

```
public class PieceFactoryBag extends PieceFactory {

    private int width;
    private ArrayList bag;
    private int piece_count;
```

Input / Output:

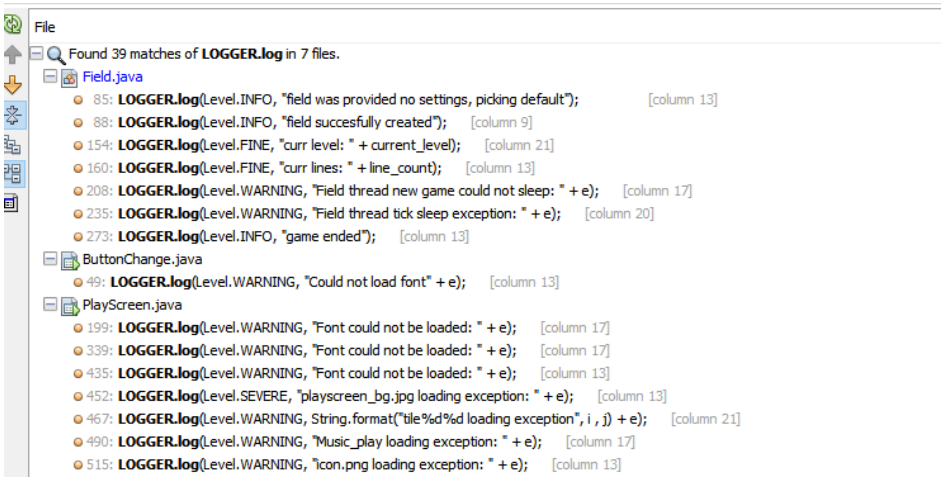
Program využíva množstvo súborov pre grafiku a zvuk (nachádzajúcich sa v priečinku **resources**), ale aj niekoľko XML súborov tvorených programom slúžiacim na uloženie nastavení, vytvorených herných módov a ich rebríčkov (nachádzajúcich sa v priečinku **saves**).



Samuel Řeřicha
AIS ID 103110

Logovanie:

Logovanie bolo použité najmä pri situáciách, keď sa čítalo zo súborov. Taktiež bolo použité aj pri dôležitých a/alebo úspešných eventoch, ako je uloženie skóre, začanie alebo ukončenie hry, prechod do nového levelu a podobne.




```
run:
May 01, 2021 4:52:21 PM Screens.PlayScreen load_tile_images
WARNING: tile00 loading exceptionjavax.imageio.IIOException: Can't read input file!
May 01, 2021 4:52:21 PM Field.Field <init>
INFO: field succesfully created
May 01, 2021 4:52:21 PM Screens.PlayScreen <init>
INFO: New playscreen created successfully
```

Internacionalizácia:

Celý program, s výnimkou stringov nachádzajúcich sa v oknách ako sú JoptionPane, je možné používať v slovenčine a angličtine, pričom je možné v nastaveniach jazyk zmeniť. Internacionalizácia bola implementovaná prostredníctvom bundlov. V prípade potreby by bolo bez veľkého zásahu možné do programu pridať ďalšie jazyky.

Samuel Řeřicha
AIS ID 103110

XML:

Ukladanie a načítavanie nastavení a súborov definujúcich herné módy je realizované prostredníctvom XML.
Triedy, ktorých sa XML serializácia/deserializácia (resp. marshalling/unmarshalling) týka majú dátové typy špecificky anotované XML keywordami.

Nastavenia uložené prostredníctvom XML:

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<gameSettings>
    <after_first_move_delay>16</after_first_move_delay>
    <after_hard_drop_delay>0</after_hard_drop_delay>
    <bg_image>path</bg_image>
    <key_down>83</key_down>
    <key_hard_drop>32</key_hard_drop>
    <key_left>65</key_left>
    <key_pause>27</key_pause>
    <key_right>68</key_right>
    <key_rotate1>74</key_rotate1>
    <key_rotate2>76</key_rotate2>
    <key_swap_held>84</key_swap_held>
    <language>en</language>
    <mus_volume>7</mus_volume>
    <sfx_volume>31</sfx_volume>
    <show_ghost>true</show_ghost>
    <side_scroll_speed>6</side_scroll_speed>
</gameSettings>
```

XML anotácia v triede GameRules:

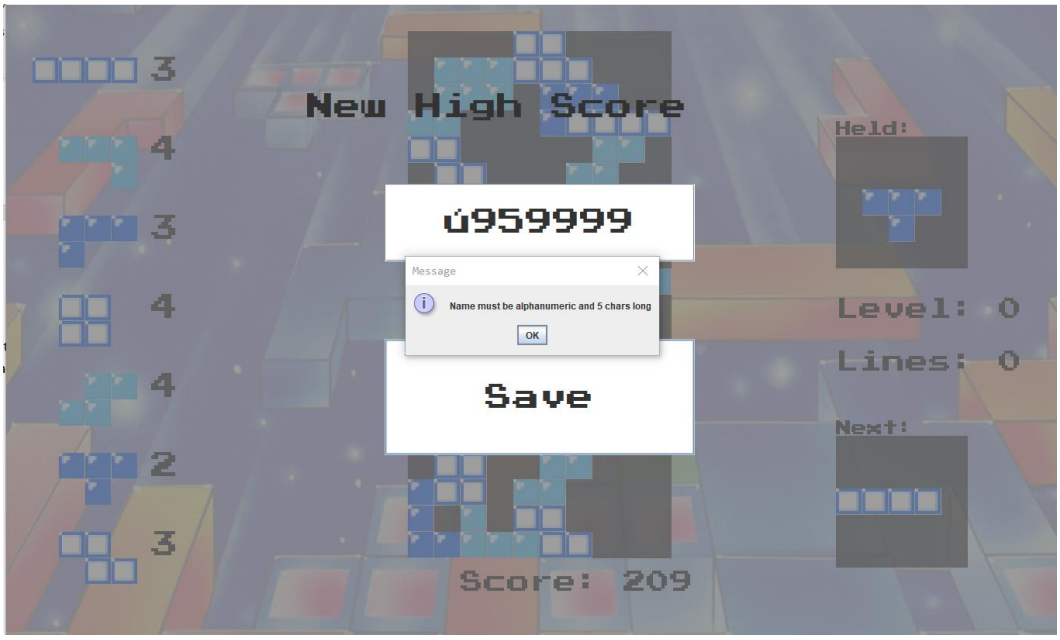
```java
@XmlElement(name = "board_width")
public void setBoard_width(int board_width) {
    this.board_width = board_width;
}

@XmlElement(name = "name")
public void setName(String name) {
    this.name = name;
}

@XmlElementWrapper(name = "scores")
@XmlElement(name = "score")
public void setScores(HighScore[] scores) {
    this.scores = scores;
}

@XmlElement(name = "use_funny_pieces")
public void setUse_funny_pieces(Boolean use_funny_pieces) {
    this.use_funny_pieces = use_funny_pieces;
}
```

Samuel Řeřicha
AIS ID 103110

Regulárne výrazy:

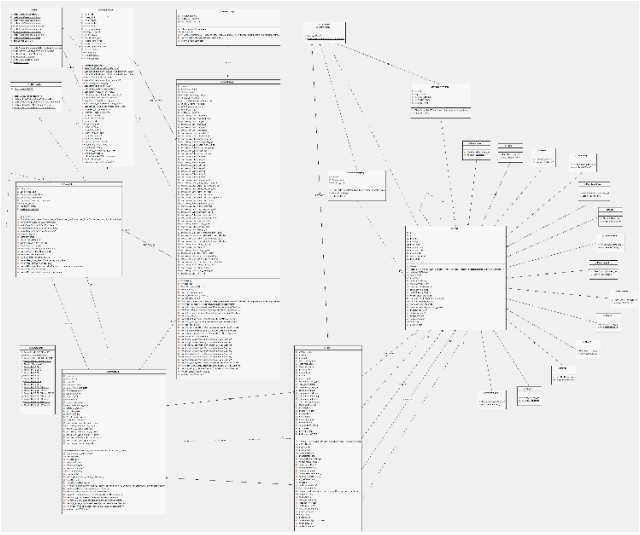
Vzhľadom na pomerne nízke množstvo text inputu používateľa nebolo možné do väčšej miery využiť regulárne výrazy, sú použité iba pri vyberaní mena pre vysoké skóre.

```
String new_name = high_score_name.getText();
String regex = "^[a-zA-Z0-9]{5}$";
Pattern pattern = Pattern.compile(regex);
Matcher matcher = pattern.matcher(new_name);
boolean result = matcher.matches();
```



UML diagram tried:

V rámci tohto dokumentu nie je možné umiestniť full resolution diagram.
Full resolution verzia UML diagramu sa nachádza v odovzdaných materiáloch.



Samuel Řeřicha
AIS ID 103110

Zhrnutie:

Vzhľadom na implementáciu projektu samostatne som s výsledkami a s implementovanými features spokojný. Aplikáciu som niekoľkým kolegom a kamarátom poskytol na testovanie a aj vďaka nim boli implementované quality of life features ako sú pauza, zmena hlasitosti, zmena keybindov a iné. Pohyb kúskov a input sú vcelku príjemné a je možné aj hranie na vyšších rýchlostiach s celkom veľkou mierou dôvery, nie je potrebné "bojovať" s hrou, tiež následkom niekoľkých nastaviteľných parametrov. Implementácia tiež obsahuje všetky požiadavky projektu v dostupnej miere.

Predpokladané features projektu obsiahnuté v zámere boli vzhľadom na požiadavky a rozhodnutia počas vývoja čiastočne pozmenené, avšak projekt obsahuje všetky požiadavky a pri jeho implementácii boli pridané features ako sú hudba, zmena keybindov, sound effects a pod.

Zdroje:

Program využíva niekoľko audiovizuálych materiálov, ktoré neboli vytvorené mnou. Boli použité v snahe spraviť aplikáciu príjemnejšiu a ľahšiu na používanie.

Pozadie hlavného menu:
https://hipwallpaper.com/view/4GgiCz

Pozadie hernej obrazovky:
Cover art hry Tetris pre NES (1989).

Retro/NES font:
https://www.fontspace.com/press-start-2p-font-f11591

SFX a hudba:
Hra NES Tetris (1989)

Tetris tile graphics:
Hra NES Tetris (1989)