

Programozói dokumentáció

Tetris

UML Osztálydiagram.....	3
A Tetrisről általában.....	4
Játékszabályok.....	4
A program képernyői.....	4
Főmenü.....	4
Játéknézet	4
Leaderboard.....	4
Név bekérő képernyő.....	4
Osztályok.....	5
Program	5
Metódusok.....	5
Tetris.....	5
Tagváltozók.....	5
Metódusok.....	5
exit_game_listener.....	6
Metódusok.....	6
start_game_listener.....	6
Metódusok.....	6
leaderboard_listener	6
Metódusok.....	6
GameWindow.....	6
Tagváltozók.....	6
Metódusok.....	7
Keychecker.....	7
Metódusok.....	7
tick_listener	8
Metódusok.....	8
save_button_listener.....	8
Metódusok.....	8
load_button_listener	8
Metódusok.....	8
pause_resume_listener.....	8

Metódusok.....	8
back_listener.....	8
Metódusok.....	8
score_save_listener.....	8
Metódusok.....	8
Board	8
Tagváltozók.....	8
Metódusok.....	9
Leaderboard.....	10
Tagváltozók.....	11
Metódusok.....	11
NameScore	11
Tagváltozók.....	11
LeaderBoardPanel	11
Tagváltozók.....	11
Metódusok.....	11
back_listener.....	11
Metódusok.....	11
NextComponentPanel.....	11
Tagváltozók.....	11
Metódusok.....	12
Coordinate	12
Tagváltozók.....	12
Metódusok.....	12
Component	12
Tagváltozók.....	13
Metódusok.....	13
Enumerációk.....	14
BoardEvents	14
Colors	14
Components.....	14
Tesztelés.....	14
Tesztlefedettség.....	14

Pintér Bence Zsolt



A Tetrisről általában

Az első Tetris játékot 1984-ben adták ki. A játék lényege, hogy a 10x20 blokk méretű játéktér tetején megjelenő 4 blokkból álló formákat (tetromínók) kell a játékosnak megfelelő orientációval, megfelelő helyre illeszteni a játéktér alján, hogy teljes sorok alakuljanak ki.

Játékszabályok

- A pontszám növekedésével arányosan a játék sebessége növekszik.
- A játék végét új formák létrejöttének ellehetlenítése jelenti. (Betelik a képernyő)
- Az újonnan megjelenő tetromínók a játéktér tetején, középen jelennek meg.
- Amennyiben a játékosnak sikerül egy teljes sort kitölteni, a sor eltűnik, és pontot kap érte.
- Ha egy bejövő forma alja hozzáér egy már letett forma tetejéhez, azonnal „letevődik”.

A program képernyői

A program 4 képernyőből tevődik össze:

1. Főmenü
2. Játéknézet
3. Leaderboard
4. Név bekérő képernyő

Főmenü

A játék elindításakor a főmenü jelenik meg, innen indítható a játék, illetve innen lehet elérni a leaderboard képernyőt.

A főmenü 3 gombot tartalmaz:

- Play
- Leaderboard
- Exit

Játéknézet

A menüben a Play gombot kiválasztva megjelenik a játéknézet, mely az alábbi elemeket tartalmazza:

- Játéktér (10 blokk széles, 20 blokk magas)
- Következő elem (a soron következő tetromínót mutatja)
- Pontszám (a jelenleg elért pontszám)
- Pause gomb, a játék megállításához

A játékos a fentről lefelé mozgó formákat a következőképp tudja mozgatni:

- balra, jobbra forgatás
- balra, jobbra mozgatás
- azonnali letevés

Leaderboard

A leaderboard képernyő a nevének megfelelően a 10 legjobb pontszámot elérő ember nevét és pontszámát tartalmazza pontszám szerinti csökkenő sorrendben.

Név bekérő képernyő

A név bekérő képernyő a játék végetérte után jelenik meg. A képernyőn megtalálható a pontszám, illetve egy mező, mely a játékos nevét várja. A képernyőn a Save gombra kattintva a pontszám mentésre kerül a leaderboardba. (Üres mező esetén nem ment.)

Osztályok

Program

A Program osztály tartalmazza a program belépési pontját, mely feladata a játék elindítása egy új Tetris példány létrehozásával.

Metódusok

public static void main(String[] args)

A program futtatásakor ez a függvény hívódik meg, létrehoz egy Tetris példányt, és megjeleníti.

Tetris

Az osztály a JFrame osztály leszármazottja. Feladata az ablak létrehozása, illetve megjelenítése.

Tagváltozók

JPanel main_menu

A főmenü paneljét tárolja.

GameWindow gamePanel

Az éppen futó játék paneljét tárolja.

JBUTTON start_game_btn

A játékot elindító gomb.

JBUTTON exit_game_btn

Kilépés gomb.

JBUTTON leaderboard_btn

A leaderboardre átlépést végrehajtó gomb.

Leaderboard leaderboard

A legfrissebb leaderboardet tárolja.

LeaderboardPanel lbp

A leaderboardet megjelenítő panelt tartalmazza.

JLabel error_text

Hiba esetén a hiba szövegét tárolja.

Metódusok

public Tetris()

Létrehoz egy új Tetris objektumot, benne a főmenüvel.

private void start_game()

Létrehoz, és elindít egy új játékot (GameWindow).

public void back_to_menu()

Eltünteti a jelenleg megnyitott nézetet, és megjeleníti a főmenüt.

private void load_leaderboard()

Fájlból beolvassa a mentett leaderboardet. Ha a fájl nem létezik, vagy nem sikerül beolvasni, kiírja az error dobozba a hibát.

private void show_leaderboard()

Megjeleníti a betöltött leaderboardet.

private void close_window()

Bezárja az ablakot.

public JButton getStartGameButton()

Visszaadja az játék indító gombot.

public JButton getLeaderboardButton()

Visszaadja a leaderboardot mutató gombot.

exit_game_listener

A főmenüben hallgat a kilépés gombra.

Metódusok

public void actionPerformed(ActionEvent evt)

Bezárja az ablakot.

start_game_listener

A főmenüben hallgat az indítás gombra.

Metódusok

public void actionPerformed(ActionEvent evt)

Elindítja a játékot.

leaderboard_listener

A főmenüben hallgat a leaderboard gombra.

Metódusok

public void actionPerformed(ActionEvent evt)

Betölti és előhívja a leaderboardot.

GameWindow

Magát a játéknézetet tartalmazó panel. Leszármazottja a JPanelnek.

Tagváltozók

private int score

A jelenlegi pontszámot tartalmazza.

private final int initialSpeed

A játék sebességének alapértéke.

private final int maxSpeed

A játék sebességének maximális értéke.

private int currentSpeed

A játék jelenlegi sebessége.

private Timer timer

Az órajeleket biztosító időzítő.

private JLabel error_label

Ha van error, annak a szövegét tartalmazza.

TextField text_field

Név bekérése esetén a ebbe lehet a nevet beleírni.

private Board board

A játéktáblát tárolja.

private NextComponentPanel next_pan

A következő komponens ebben jelenik meg.

private JLabel score_label_score

A jelenlegi pontszám megjelenítéséhez kell.

JBUTTON pause_resume_btn

A játékot megállító gomb.

JBUTTON save_btn

A mentést elindító gomb.

JBUTTON load_btn

A betöltést elindító gomb.

JBUTTON back_btn

A menübe visszavivő gomb.

boolean key_listener_enabled

Igaz: a játék hallgat a gombnyomásra, Hamis: nem.

Metódusok**public GameWindow()**

Létrehoz egy új játéknézetet, benne egy új játéktáblával, majd elindítja a játékot.

private void save_game()

Elmenti a játéktér jelenlegi állását pontszámmal együtt.

private void save_score()

Hozzáad egy rekordot a leaderboardhoz, és elmenti. Ha nem létezik korábbi leaderboard, létrehoz egy újat.

private void render_name_input_box()

Kirajzolja a név bekérő képernyőt.

private void load_game()

Betölti a korábban elmentett állást.

private void back_to_menu()

Visszalép a főmenübe.

public JButton[] getAllButtons()

Visszatér a panelon található összes gombból álló tömbbel. (Tesztelési céllal)

public Board getBoard()

Visszatér a játéktérrel. (Tesztelési céllal)

Keychecker

A gombnyomásokat érzékeli. Implementálja a KeyAdapter interfészt.

Metódusok**public void keyPressed(KeyEvent event)**

A lenyomott gombtól függően elvégez egy műveletet a játéktéren.

tick listener

Az időzítő által keltett órajelekre reagál.

Metódusok

public void actionPerformed(ActionEvent evt)

Frissíti a játéktáblát, majd a tábla új állapota alapján cselekszik.

save button listener

A mentés gomb kattintására reagál.

Metódusok

public void actionPerformed(ActionEvent evt)

Elmenti a játéktér jelenlegi állapotát.

load button listener

A betöltés gomb kattintására reagál.

Metódusok

public void actionPerformed(ActionEvent evt)

Betölti az elmentett játékteret.

pause resume listener

Hallgat a pause gomb kattintására.

Metódusok

public void actionPerformed(ActionEvent evt)

Szünetelteti a játékot és megjeleníti a mentés, illetve a betöltés gombot.

back listener

Reagál a vissza gombra.

Metódusok

public void actionPerformed(ActionEvent evt)

Visszalép a főmenübe.

score save listener

Reagál a pontszám mentése gombra.

Metódusok

public void actionPerformed(ActionEvent evt)

Hozzáadja a játékos nevét és a pontszámot a leaderboardhoz, majd elmenti a frissített leaderboardot.

Board

A játékteret, illetve az azon végzett műveleteket tartalmazza.

Tagváltozók

public static final Map<Components, Colors> component_colors

Tárolja a tetromínókhoz tartozó színeket.

public static final Map<Colors, Color> color_map

Tárolja a tetromínó színekhez tartozó színekódokat.

public static final Components[] all_components

Tárolja a létező teromínókat.

private int startSquareX

A pálya bal felső sarkának X koordinátája.

private int startSquareY

A pálya bal felső sarkának Y koordinátája.

private int squareW

A pálya egy négyzetének szélessége.

private int squareH

A pálya egy négyzetének magassága.

private int board_height

A pálya magassága négyzetekben megadva.

private int board_width

A pálya szélessége négyzetekben megadva.

private Coordinate spawn_pos

Az új tetromímó spawn pozíciója.

private Random rand

Random szám generátor (Random);

private Colors[][] board

A pálya maga.

private int score

A jelenlegi pontszám.

Component current

A jelenleg eső tetromínó.

Components next

A következő tetromínó típusa.

Metódusok

public Component getCurrComponent()

Visszatér a jelenleg eső tetromínóval.

public Components getNextComponent()

Visszatér a következő komponens típusával.

public int getScore()

Visszatér a jelenlegi pontszámmal.

public BoardEvents updateBoard()

Frissíti a pályát.

private boolean checkCollision(Component comp)

Megnézi, hogy a jelenleg eső tetromínó ütközik-e valamivel.

private void calculateScore(int rowsDeleted)

A törölt teljes sorok száma alapján növeli a pontszámot.

private Colors getFieldColor(Coordinate a)

Egy pályán lévő négyzet színét adja vissza.

private boolean spawnComponent(Components a)

Új komponenst spawnol.

public void resetCurrentComponentPosition()

Visszahelyezi a jelenleg eső tetromínót a spawnolási helyére.

private void eraseCurrentComponent()

Törli a jelenleg esésben lévő tetromínót a pályáról.

private void renderCurrentComponent()

Rárajzolja a jelenleg esésben lévő tetromínót a pályára.

protected void paintComponent(Graphics g)

Kirajzolja a pályát.

private boolean checkIfNewPlaceIsEmpty(Component b)

Megnézi, hogy a komponens új pozíciója üres-e.

private void removeFullRows()

Törli a teli sorokat.

private boolean checkIfNewPlaceIsInBounds(Component b)

Megnézi, hogy a tetromínó új pozíciója nincs-e kint pályáról.

private void deleteRow(int k)

Töröl egy sort.

public void control_move_right()

A jelenlegi tetromínót jobbra mozgatja eggyel.

public void control_move_left()

A jelenlegi tetromínót balra mozgatja eggyel.

public void control_rotate_left()

A jelenlegi tetromínót balra forgatja eggyel.

public void control_rotate_right()

A jelenlegi tetromínót jobbra forgatja eggyel.

public void control_move_place()

A jelenlegi tetromínót leteszi a pálya aljára.

private int generate_placement_outline()

Leképezi a jelenlegi tetromínó jelenlegi pozícióját.

Leaderboard

A leaderboard kezeléséért felel.

Tagváltozók

List<NameScore> lista

A leaderboard bejegyzéseit tartalmazza.

Metódusok

public void add(String name, int score)

Hozzáad egy bejegyzést a leaderboardhoz.

public int len()

Visszatér a leaderbord bejegyzéseinek számával.

public List<Integer> getScores()

Visszatér az összes leaderbordban lévő pontszámmal.

public List<JPanel> getAllPanels(int width)

Visszatér a leaderboard összes rekordját tartalmazó JPanel listával.

NameScore

Egy leaderboard bejegyzés.

Tagváltozók

public String name

A bejegyzésben szereplő név.

public int score

A bejegyzésben szereplő pontszám.

LeaderBoardPanel

A leaderboardot és egy visszalépés gombot tartalmazó panel.

Tagváltozók

Leaderboard leaderboard;

A betöltött leaderboard.

Metódusok

public LeaderboardPanel(Leaderboard lead)

Létrehoz egy új leaderboard panelt.

public void back_to_menu()

Visszalép a menübe.

back_listener

A vissza gomb kattintására reagál.

Metódusok

public void actionPerformed(ActionEvent evt)

Visszalép a menübe.

NextComponentPanel

A következő komponenst mutató panel kirajzolásáért felelős.

Tagváltozók

Components contained;

A jelenleg kirajzolt komponens típusa.

Colors[][] field;

A rajzolási terület.

Metódusok

public NextComponentPanel(Components curr)

Konstruktor. Létrehoz egy új példányt.

public void switch_component(Components a)

Lecseréli a kirajzolt komponenst.

protected void paintComponent(Graphics g)

Kirajzolja a jelenleg eltárolt komponenst.

Coordinate

Egy koordináta tárolásáért felelős.

Tagváltozók

private int x;

Az X koordináta.

private int y;

Az Y koordináta.

Metódusok

public Coordinate()

Létrehoz egy új példányt.

public Coordinate(int a, int b)

Létrehoz egy új példányt.

public Coordinate(int[] array)

Létrehoz egy új példányt.

public Coordinate Add(Coordinate b)

Hozzáad a koordinátahoz egy másik koordinátát.

public int getX()

Visszatér a koordináta X komponensével.

public int getY()

Visszatér a koordináta Y komponensével.

public int[] getAsArray()

Visszatér a koordinátával egy kételemű tömbként.

public boolean isInSameRowOrColumn(Coordinate b)

Megnézi, hogy az adott koordináta egy másikkal egy sorban, vagy oszlopban van-e.

public boolean equals(Coordinate b)

Megnézi, hogy a két koordináta egyenlő-e.

Component

Egy tetromínó.

Tagváltozók

Components type

A tetromínó típusa.

Coordinate[] coords

A tetromínó blokkjainak abszolút koordinátái.

public static final Map<Components, int[][]> component_dims

Az összes tetromínó típushoz rendel relatív blokk koordinátákat.

Metódusok

public Component(Component t, Coordinate spwn_coords)

Létrehoz egy új komponenst.

public Component(Component b)

Létrehoz egy új komponenst.

public Coordinate[] getCoords()

Visszaadja a tetromínó abszolút koordinátáit.

public Components getType()

Visszaadja a tetromínó típusát.

public void move_down(int n)

Lejjebb mozgatja a tetromínót n-nel.

public Component shadow_move(Coordinate b)

Visszatér egy új tetromínóval, mely az eredeti elmozgatottja.

public void rotate_left()

Balra forgatja eggyel a tetromínót.

public Component shadow_rotate_left()

Visszatér egy új tetromínóval, mely az eredeti eggyel balra forgatottja.

public void rotate_right()

Jobbra forgatja eggyel a tetromínót.

public Component shadow_rotate_right()

Visszatér egy új tetromínóval, mely az eredeti eggyel jobbra forgatottja.

public void move_right(int n)

Jobbra mozgatja a tetromínót n-nel.

public void move_left(int n)

Balra mozgatja a tetromínót n-nel.

public boolean isAtBorder(Coordinate top_left, Coordinate bottom_right)

Megnézi, hogy a tetromínó egy megadott téglalap széléhez ér-e.

public boolean isAtBottomBorder(int board_height)

Megnézi, hogy a tetromínó eléri-e a pálya alját.

public boolean isPartOf(Coordinate b)

Megnézi, hogy egy adott koordináta része-e a tetromínónak.

Enumerációk

BoardEvents

A pálya lehetséges állapotai.

Colors

A tetromínók lehetséges színei.






















Components

A lehetséges tetromínó típusok.

Tesztelés

A program teszteléséhez a JUnit 4 tesztkörnyezetet használom.

Tesztlefedettség

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
▼ Tetris	 75,6 %	3 524	1 137	4 661
▼ src	 73,2 %	3 109	1 137	4 246
▼ windows	 52,1 %	711	654	1 365
> GameWindow.java	 51,2 %	534	508	1 042
> NextComponentPanel.java	 43,6 %	106	137	243
> LeaderboardPanel.java	 88,8 %	71	9	80
▼ board	 69,5 %	835	367	1 202
> Board.java	 69,5 %	835	367	1 202
▼ tetris	 83,2 %	323	65	388
> Tetris.java	 83,2 %	323	65	388
▼ component	 96,3 %	750	29	779
> Component.java	 96,3 %	750	29	779
▼ misc	 88,0 %	81	11	92
> Coordinate.java	 88,0 %	81	11	92
▼ program	0,0 %	0	11	11
> Program.java	 0,0 %	0	11	11
▼ enums	 100,0 %	202	0	202
> BoardEvents.java	 100,0 %	44	0	44
> Colors.java	 100,0 %	84	0	84
> Components.java	 100,0 %	74	0	74
▼ leaderboard	 100,0 %	207	0	207
> Leaderboard.java	 100,0 %	207	0	207