Tetris – Dokumentace

*Zápočtový program, Tetris, je zábavný program, který lze využít pro zabití času hraním legendárních her i pro edukativní cíle. Program obsahuje klasickou hru tetris, s typickým bodovým systémem a pravidly, které jsou většině lidí na planetě Zemi známy. Mimo to program obsahuje 2 různé umělé inteligence, které (za vhodných podmínek) dokážou hrát tetris i hodiny. Nejenom herní pamětnící zde najdou další hru, kterou nejspíš ve svých mladých létech pařili na gameboyích.*

# Programátorská sekce

## Část I – Hra Tetris

Kapitola 1 – hrací deska, místo pro radost i rozruch

Hrací deska pro tetris a obě umělé inteligence v daném projektu je reprezentována pomocí objektu třídy GameBoard, který je v hlavní třídě Form1.cs uložen pod názvem gb. Při zapnutí (nebo i přepnutí) kteréhokoliv režimu se vždy zinicializuje nová hrací deska – **gb = new GameBoard();** –. Objekt této třídy obsahuje dvojrozměrné pole **public char[,] Board**, které je v konstruktoru defaultně naplněno nulovým charem ‘\0‘, což je chápano jako prázdné pole, místo, kam se může TetroBlock posunout. Hrací deska má rozměry 10x20, ale zobrazováno je jenom dolních 18 řad. Horní 2 sice nejsou zobrazovány, ale i přesto může hráč (například při rotaci) do nich zavítat. Tato třída kromě hracího pole také obsahuje celočíselné hodnoty **level, lines a score**. Nejpodstatnější z této trojice čísel je **level**, tato hodnota upravuje rychlost hry. Dochází ke zrychlení hry až do 10. levelu (kdy je rychlost hry 140ms), pak už je zrychlení nulové.

Třída **GameBoard** je mimo jiné používáná pro generaci TetroBlocků. Generace není přímo náhodná. Existuje 7 druhů TetroBlocků, se kterými můžeme hrát tetris. Generaci si můžeme představit následovně. Do měšce vložíme všech 7 existujících TetroBlocků a postupně je vytahujeme po jednom. V okamžiku, kdy je měšec prázdny, navrátíme všechny TetroBlocky zpět a začínáme odznova. Tím pádem je záruka, že každých 13 generací (v nejhorším prípadě) dostanu chtěný TetroBlock. Tuto úlohu řeší statická metoda **static public Shape GeneratePiece()**, která ukládá potřebné informace do statických proměnných **static int** **numOfPieces** a **static bool[] piecesDistribution**. Návratová hodnota této funkce je Shape, což je abstraktní mateřská třída všech 7 různých TetroBlocků, proto můžeme jednoduše dosadit a přetypovat proměnnou Shape na potřebný objekt. Další podstatné metody této třídy jsou:

* **public bool AddToBoard(Shape shp)**
* **public int[] FindFullLines()/static public int[] FindFullLines(ref char[,] deska)**
* **static public void MoveMap(ref char[,] deska, int[] lines)**
* **static public void ClearLines(ref GameBoard gb, int[] lines)**

První zmíněná metoda vrátí **true**, pokud se nám podaří vložit aktivní hrací TetroBlock do hrací desky, tím pádem patří ke stěžejním metodám, která nám prozradí, kdy je třeba hru ukončit, protože už není místo pro další.

kapitola 2 – malujeme kostičky

t

Kapitola 3 – třída Shape a jejich 7 dětí

Kapitola 4 – form1.cs aneb třída, co vládne všem a všem káže

## Část II – Umělá intelligence

kapitola 1 – všechny cesty vedou na hrací desku

kapitola 2 – dobrý tah, nebo zlý, to je, oč tu běží

kapitola 3 – když jde tetroblock na výlet

## Část III – Bonus

kapitola 1 – drtič zdí

# Uživatelská sekce