U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Formal Modeling of a Tetris Game

Mestrado Integrado em Engenharia Informática e Computação

Métodos Formais em Engenharia de Software

Grupo 1 Turma 4MIEIC02
Ângela Cardoso - up200204375
Tiago Galvão - up201500034
Nuno Valente - up200204376

January 3, 2017

Contents

1 Informal system description and list of requirements **3**
- 1.1 Informal system description 3
- 1.2 List of requirements 3

2 Visual UML model **4**
- 2.1 Use case model 4
- 2.2 Class model 5

3 Formal VDM++ model **5**
- 3.1 Class Game 5
- 3.2 Class Board 5
- 3.3 Class Tetromino 6
  - 3.3.1 Class Tetromino-"type" 6

4 Model validation **6**
- 4.1 Class MyTestCase 6
- 4.2 Class TestGame 6

5 Model verification **6**
- 5.1 Example of domain verification 6
- 5.2 Example of invariant verification 6

6 Conclusions **6**

7 References **6**

A Source Code **7**

B Formal indispensable rules **7**

C The 7 tetrominoes **8**

D The position of each mino inside respective tetromino **8**

E All possible orientations of each tetromino **9**

1 Informal system description and list of requirements

1.1 Informal system description

Tetris game it's a puzzle game and one of the most recognizable and influential video game brands in the world. It's no wonder why there are hundreds of millions of Tetris products being played, worn, and enjoyed by fans in their everyday lives. The game was born in 1984 and it's living proof of a game that have truly transcended the barriers of culture and language.

A meritorious reference to Alexey Pajitnov because he his the person who developed this popular game. He is a russian video game designer and computer engineer and in his spare time, he drew inspiration from his favorite puzzle board game, pentominoes, and decided to create a computer game for himself. Pajitnov envisioned an electronic game that let players arrange puzzle pieces in real time as they fell from the top of the playing field. The resulting design was a game that used seven distinctive geometric playing pieces (appendixC), each made up of four squares. Pajitnov called this game "Tetris," a combination of "tetra" (the Greek word meaning "four") and "tennis" (his favorite sport).

The rules to play the game are very simple. Tetris game requires players to strategically rotate and drop a chaining of tetrominoes that fall into the rectangular board at increasing speeds. Players attempt to clear as many lines as possible by completing horizontal rows of blocks without empty space, but if the tetrominoes surpass the skyline(top of the board) the game is over! Speed and consequent level advance can make the game ally to strategy more enthusiastic. Formal details about other rules are presented in appendixB.

put some image here

1.2 List of requirements

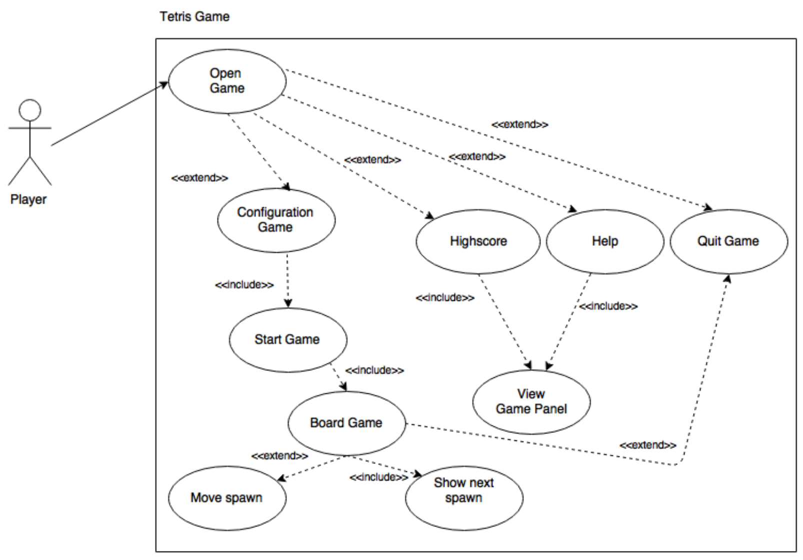
Id	**Priority**	**Description**
R1	Mandatory	The player can configure the game with his nickname
R2	Optional	The player can view best highscores
R3	Mandatory	The game allows two movements - rotation and drop
R4	Mandatory	The player should be able to leave the game when he wants
R5	Mandatory	The player has access to help menu where are the rules to play the game
R6	Mandatory	The spawns appears in a random order to be played
R7	Mandatory	When a row is full of blocks it must be cleared
R8	Mandatory	If tetraminoes surpass the skyline the game is over
R9	Opcional	Time to time the level advance one degree and the game became more difficult
R10	Mandatory	Each tetromino is formed by four squares named minos by us

These requirements are directly translated onto use cases as shown next.

2 Visual UML model

2.1 Use case model

In all use cases we've made an assumption that all keyboard game keys are functioning properly.



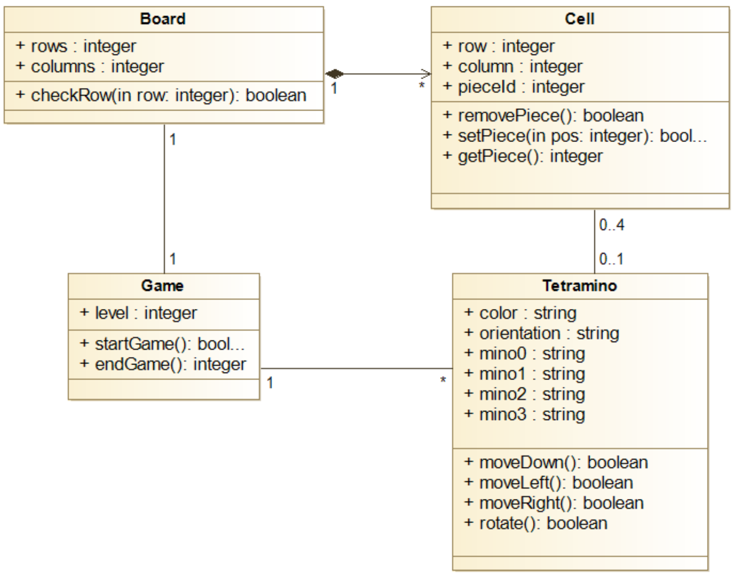
The main use cases are described below:

Scenario	Configure Game
Description	The user can write his nickname and difficulty level
Pre-conditions	The game is in the configuration menu
Post-conditions	The player became associated with that nickname and game with difficulty level
Steps	(none)
Exceptions	(none)

Scenario	Rotation Movement
Description	Rotate a tetromino when is falling
Pre-conditions	Tetromino must not be frozen in place
Post-conditions	Tetromino position has changed
Steps	The player must click on a key that allows rotation
Exceptions	Tetromino type O doesn't have other form when try rotation; when, simultaneously, try to rotate and the spawn reached one final position; the stack of tetrominoes reached the penult line of the board

Scenario	**Drop Movement**
Description	Accelerates the tetromino falling and position it as it was before reach the final position
Pre-conditions	Tetromino must not be frozen in place
Post-conditions	Tetromino position has changed
Steps	The player must click on a key that allows falling spawn
Exceptions	(none)

2.2 Class model



Class	Description
Game	Core model; defines the state variables and operations available to the players.
Board	Defines a game environment for playing and where eachhpiece of tetraminoes can stay
Tetromino	Defines one piece to play with.

Table 1: Description of each class

3 Formal VDM++ model

paste code

3.1 Class Game

paste code

3.2 Class Board

paste code

3.3 Class Tetromino

paste code

3.3.1 Class Tetromino-"type"

4 Model validation

todo

4.1 Class MyTestCase

todo

4.2 Class TestGame

todo

5 Model verification

todo

5.1 Example of domain verification

todo

5.2 Example of invariant verification

todo

6 Conclusions

The model that was developed by us covers all the requirements included implicitly on the theme project and the list of requirements descripted in section 1.2. In the final and after model verifications, we all see the game developed in VDM++ like one of the projects more consistent and safer that we have ever developed during the course. In addition it is noticed that all the elements of the group have already had contact in the past with the game and continue feeling enthuse with this version developed by us. Maybe in the future we can all add more features to this game make it appears near the original version. This project took approximately 16 hours to develop.

7 References

- https://en.wikipedia.org/wiki/Tetris
- http://tetris.com/
- https://tetris.wiki/Tetris_Guideline
- http://overturetool.org/

A Source Code

maybe not necessary because of section3

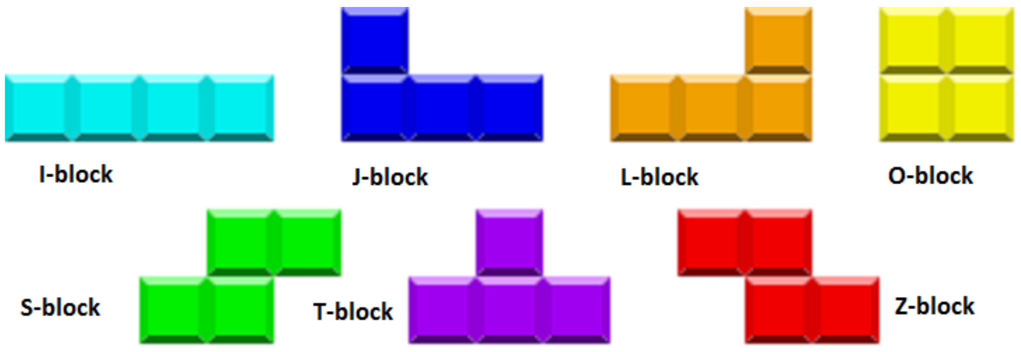
B Formal indispensable rules

In this section we present the formal rules that Tetris game must follow. We already present other rules, named informal and in a player view way in section 1.1.

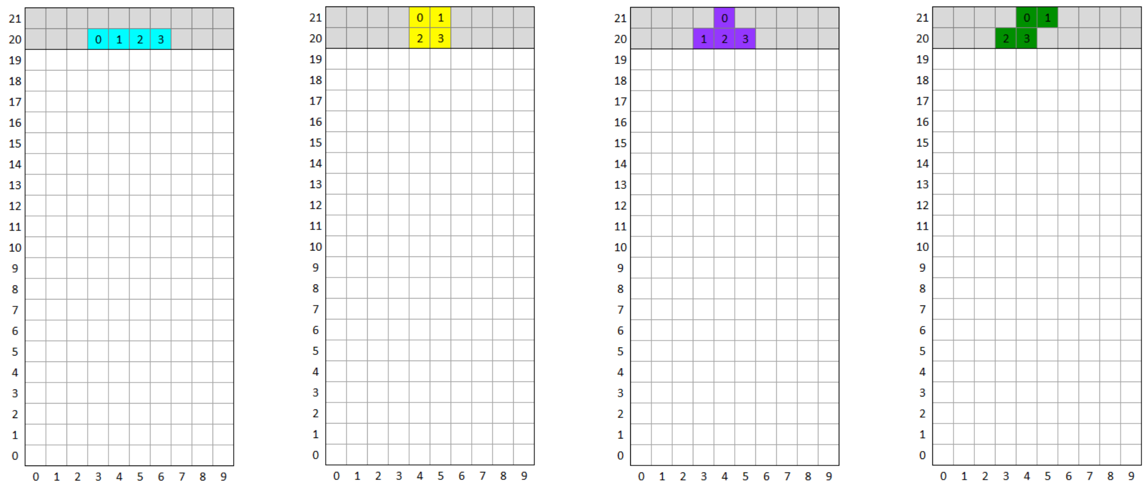
- Playfield is 10 cells wide and at least 22 cells tall, where rows above 20 are hidden or obstructed by the field frame. We follow in our case 10 cells wide and 22 cells tall where the player can see the first 20 rows and the 2 last cells are invisible to the player.
- The tetromino colors are:
  Cyan I;
  Yellow O;
  Purple T;
  Green S;
  Red Z;
  Blue J;
  Orange L.
- Each tetromino appear on these exactly locations:
  The I and O spawn in the middle columns;
  The rest spawn in the left-middle columns;
  The tetrominoes spawn horizontally and with their flat side pointed down.
- Super Rotation System (SRS) specifies tetromino rotation.
- Standard mappings for console and handheld gamepads: Up, Down, Left, Right on joystick perform locking hard drop, non-locking soft drop (except first frame locking in some games), left shift, and right shift respectively. Left fire button rotates 90 degrees counterclockwise, and right fire button rotates 90 degrees clockwise.
- Standard mappings different from console/handheld gamepads for computer keyboards
- So-called Random Generator (also called "random bag" or "7 system")
- "Hold piece": The player can press a button to send the falling tetromino to the hold box, and any tetromino that had been in the hold box moves to the top of the screen and begins falling. Hold cannot be used again until after the piece locks down. Games on platforms with fewer than eight usable buttons (such as the version on iPod) may skip this feature. The combination of hold piece and Random Generator would appear to allow the player to play forever.
- Game must have ghost piece function.
- Terms used in the user manual: "Tetriminos" not "tetrominoes" or "tetrads" or "pieces", letter names not "square" or "stick", etc.

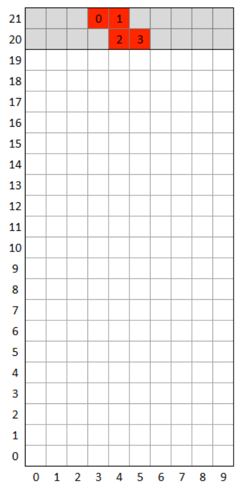
- Designated soft drop speed. Details vary between guideline versions.
- Player may only level up by clearing lines or performing T-Spin. Required lines depends in the game.
- The game must use a variant of Roger Dean's Tetris logo, although this was true from around 2000 - before the guidelines emerged.
- Game must include a song called Korobeiniki. (Guideline 2005)
- The player tops out when a piece is spawned overlapping at least one block, or a piece locks completely above the visible portion of the playfield.

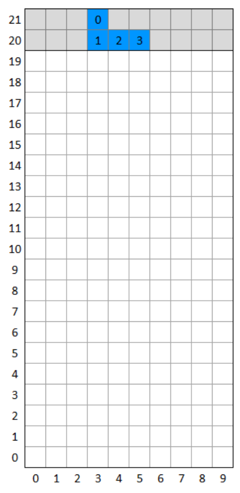
C The 7 tetrominoes

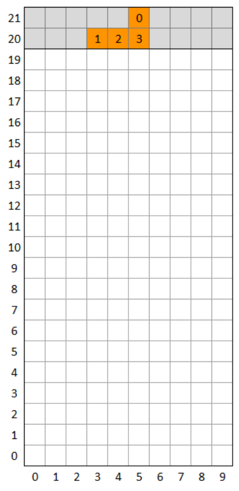


D The position of each mino inside respective tetromino

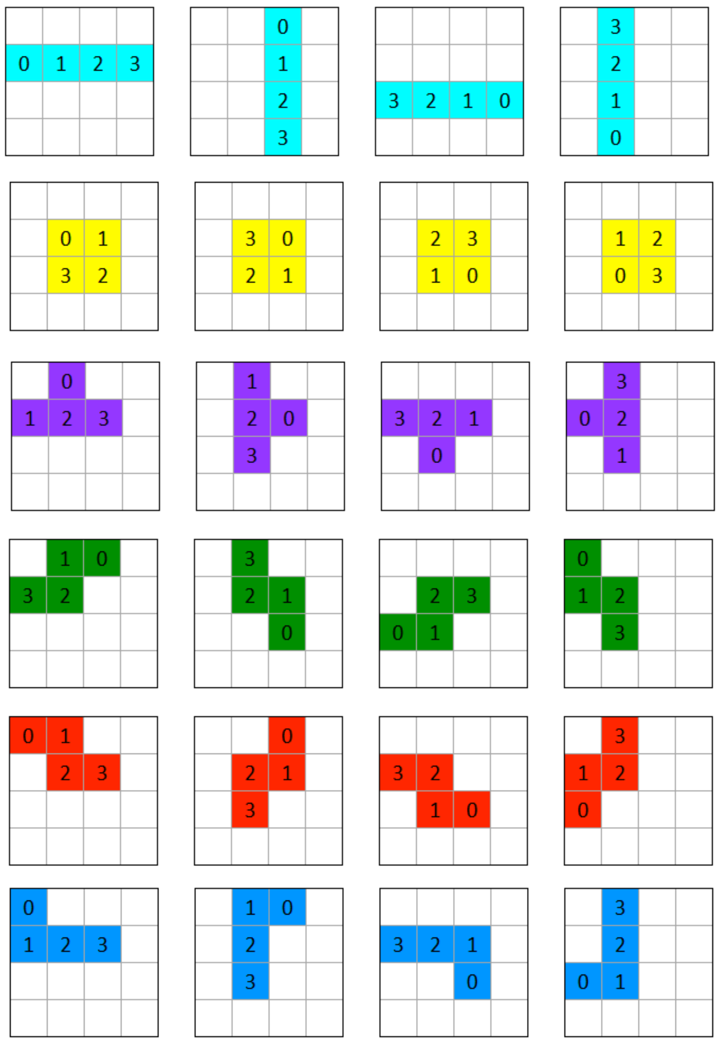


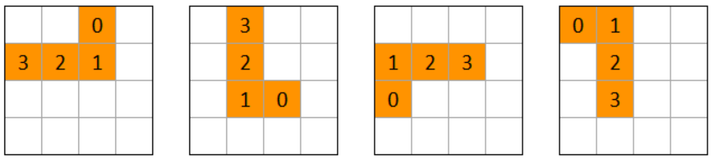






E All possible orientations of each tetromino




0
3 2 1
3
2
1 0
1 2 3
0
0 1
2
3