Object Oriented Programming and Design
Exam 2010-2011
Martin Nicklas Jørgensen, DIKU
Examination nr.: 160

20-01-2011

Object Oriented Programming and Design

The Tron Game

Exam project 2010-2011 (January 20, 2011)

Table of content

Test plan for WallHuggerStrategy (Assignment 2.2.3)

To test the wall hugger we must make sure it have the following reactions.

Expected reactions

1. If it encounters a wall right in front of it, it may not continue ahead.
2. If a wall is right in front of it, it must turn to either side to avoid collision.
3. If it is parallel to a wall it must continue straight along it and not leave.
4. At any turn it shall try to avoid driving into walls.
5. If it ever gets trapped it must continue straight ahead as a form of suicide move.

It do not have to look down “alleys” to see if they’re dead-ends, in that case it’s okay if it drives down the allay and then executes reaction number 5 from above.

CRC-Card for class: Game() (Assignment 5.1.1)

Responsibilities

The Game class must represent the state of a game. This includes containing the Board the game is played on, and the list of Players currently on the board (including dead ones.)

The Game class is therefore responsible for handling:

- Board field occupation
- Game state
- Player positions

Note that the actual playerlist is not available from this class, nor is the actual Board!

Collaborators

- Board
- Player

Game vs. Board

Game and Board are just two parts that the GameManager holds together in order to keep all the information together. One could very well merge Game and Board together but very little would be gained. Since they both contained in the GameManager, so is all their information and it’s therefore still available from a single class.

But keeping them separated into different classes gives more clarity as to the specific role of a class and allows for easier understanding of each class and an easier time coding them.

So with that I will conclude it with a personal opinion, I prefer the separated way since small neat classes are way better than big mushy classes.

model.Board() data structure & implementation (Assignment 5.1.2)

Since the Board class needs only represent the board itself (not the players) it means it largely static, except when walls are added to it, meaning it's mostly a storage class (or database if you wish) that holds the current state of the board.

Combined with the knowledge that a position is either free or occupied (player or wall,) we can make certain assumptions.

Assumptions about the board class'

- Every positions state is binary (either free or not)
- The boards only job is to manage these positions
- Once a position is occupied it cannot become free
- If we decide that all positions are initially free, we need only manage the occupied ones

Many of the above assumptions actually translate very well into design decisions and functional requirements.

Predefined requisites for the Board class

The Board() class was delivered with some initial coding.:

- A class constructor requiring nothing but width and height as integer.
- Accessor methods for width and height.
- An empty method for occupying a position.
- An empty method for seeing whether a given position was already occupied.

Meaning that the two last ones was required of us.

Chosen design and implementation

In accordance with the assumptions I chose to implement it like this:

- The board is initially blank, no occupied positions.
- The occupied positions are stored in a list.
- When isOccupied is called, it must look for a given position in this array.
- When occupyField is called it must add the given position to the array.

I have but one regret in the way I implemented isOccupied, that the fact that the array is searched from one end to another when looking for a given position. On modern computers and with tracks that fit on screens on/below 1920x1080 resolutions this will not be an actual issue.

If I had more time to polish the design I would have implemented a method using the x-coordinate of the position as index. This would result in faster searching when looking for something at the end of the array.

Other Design choices (Not compulsory assignment)

In this section I will briefly describe my thoughts when designing/implementing classes and solutions in the project.

model.strategies.WallHuggerStrategy()

When a WallHugger is created it creates a list of directions by using the Direction.shuffledValues() method to get a “random” unsorted list of direction. This list is used for searching directions.

I chose to implement this inconsistency because if they all for instance always sought the same way, they would get an uneven chance of completing solely based on their starting position making each strategy irrelevant.

Also by using a “random” direction, each players chance of winning is not determined by any strategy but they all have a 25% chance of getting a direction that will be optimum. Of course making an AI that’s smarter and actually decides the best direction would be best, but the time wasn’t present so I lowered the scale of AI intelligence.

Also, it’s more fun to look at the games because they’re different now!

Looking back I think I could have made better use of the Position.getNeighbours() method in this strategy.

view.NewGameDialog().PlayerPanel()

I choose do add a function to return the selected index of the combox. The let me skip the whole convert from enum to strategy thing and made it easier to have “disabled” players.

I decided to interpret empty comoboxes an unused players, with the exception of Player #1 who will remain active if the “Use human player?” is selected.

APPENDIX A (FILELIST):

The following is a list of files/folders needed/used/made as part of this project.:

(NOTE: all files/folder within" oopd-eksamen-2010\" is files in the BlueJ project JAR the first 4 files are added manually to the Jar afterwards.)

```
EksamensOplæg.pdf
oopd-eksamen-2010.jar
Jørgensen.MartinNicklas.Report.pdf (THIS FILE)
Compiled.Game.jar
oopd-eksamen-2010\.classpath
oopd-eksamen-2010\.project
oopd-eksamen-2010\advanced.tron
oopd-eksamen-2010\broken.tron
oopd-eksamen-2010\crosshair.tron
oopd-eksamen-2010\package.bluej
oopd-eksamen-2010\README.TXT
oopd-eksamen-2010\simple.tron
oopd-eksamen-2010\control\GUIController.class
oopd-eksamen-2010\control\GUIController.ctxt
oopd-eksamen-2010\control\GUIController.java
oopd-eksamen-2010\control\package.bluej
oopd-eksamen-2010\control\TronLauncher.class
oopd-eksamen-2010\control\TronLauncher.ctxt
oopd-eksamen-2010\control\TronLauncher.java
oopd-eksamen-2010\doc\allclasses-frame.html
oopd-eksamen-2010\doc\allclasses-noframe.html
oopd-eksamen-2010\doc\constant-values.html
oopd-eksamen-2010\doc\help-doc.html
oopd-eksamen-2010\doc\index-all.html
oopd-eksamen-2010\doc\index.html
oopd-eksamen-2010\doc\logfile.txt
oopd-eksamen-2010\doc\overview-frame.html
oopd-eksamen-2010\doc\overview-summary.html
oopd-eksamen-2010\doc\overview-tree.html
oopd-eksamen-2010\doc\package-list
oopd-eksamen-2010\doc\serialized-form.html
oopd-eksamen-2010\doc\stylesheet.css
oopd-eksamen-2010\doc\control\GUIController.html
oopd-eksamen-2010\doc\control\package-frame.html
oopd-eksamen-2010\doc\control\package-summary.html
oopd-eksamen-2010\doc\control\package-tree.html
oopd-eksamen-2010\doc\control\TronLauncher.html
oopd-eksamen-2010\doc\model\Board.html
oopd-eksamen-2010\doc\model\ComputerPlayer.html
oopd-eksamen-2010\doc\model\Direction.html
oopd-eksamen-2010\doc\model\Game.html
oopd-eksamen-2010\doc\model\GameManager.html
oopd-eksamen-2010\doc\model\HumanPlayer.html
oopd-eksamen-2010\doc\model\package-frame.html
oopd-eksamen-2010\doc\model\package-summary.html
oopd-eksamen-2010\doc\model\package-tree.html
oopd-eksamen-2010\doc\model\Player.html
oopd-eksamen-2010\doc\model\Position.html
oopd-eksamen-2010\doc\model\config\Configuration.html
oopd-eksamen-2010\doc\model\config\IParser.html
oopd-eksamen-2010\doc\model\config\LineSegment.html
oopd-eksamen-2010\doc\model\config\package-frame.html
oopd-eksamen-2010\doc\model\config\package-summary.html
oopd-eksamen-2010\doc\model\config\package-tree.html
oopd-eksamen-2010\doc\model\config\ParserException.html
oopd-eksamen-2010\doc\model\config\TextParser.html
oopd-eksamen-2010\doc\model\config\TextParserTest.html
oopd-eksamen-2010\doc\model\strategies\IStrategy.html
oopd-eksamen-2010\doc\model\strategies\package-frame.html
```

```
oopd-eksamen-2010\doc\model\strategies\package-summary.html
oopd-eksamen-2010\doc\model\strategies\package-tree.html
oopd-eksamen-2010\doc\model\strategies\RandomStrategy.html
oopd-eksamen-2010\doc\model\strategies\StrategyFactory.html
oopd-eksamen-2010\doc\model\strategies\StrategyFactory.Type.html
oopd-eksamen-2010\doc\model\strategies\StrategyFactory.Types.html
oopd-eksamen-2010\doc\model\strategies\WallHuggerStrategy.html
oopd-eksamen-2010\doc\model\strategies\WallHuggerStrategyTest.html
oopd-eksamen-2010\doc\resources\inherit.gif
oopd-eksamen-2010\doc\view\MainWindowFrame.html
oopd-eksamen-2010\doc\view\NewGameDialog.html
oopd-eksamen-2010\doc\view\package-frame.html
oopd-eksamen-2010\doc\view\package-summary.html
oopd-eksamen-2010\doc\view\package-tree.html
oopd-eksamen-2010\doc\view\TronColors.html
oopd-eksamen-2010\doc\view\TronDisplayPanel.html
oopd-eksamen-2010\doc\view\tronFilter.html
oopd-eksamen-2010\model\Board.class
oopd-eksamen-2010\model\Board.ctxt
oopd-eksamen-2010\model\Board.java
oopd-eksamen-2010\model\ComputerPlayer.class
oopd-eksamen-2010\model\ComputerPlayer.ctxt
oopd-eksamen-2010\model\ComputerPlayer.java
oopd-eksamen-2010\model\Direction$1.class
oopd-eksamen-2010\model\Direction$2.class
oopd-eksamen-2010\model\Direction$3.class
oopd-eksamen-2010\model\Direction$4.class
oopd-eksamen-2010\model\Direction.class
oopd-eksamen-2010\model\Direction.ctxt
oopd-eksamen-2010\model\Direction.java
oopd-eksamen-2010\model\Game.class
oopd-eksamen-2010\model\Game.ctxt
oopd-eksamen-2010\model\Game.java
oopd-eksamen-2010\model\GameManager.class
oopd-eksamen-2010\model\GameManager.ctxt
oopd-eksamen-2010\model\GameManager.java
oopd-eksamen-2010\model\HumanPlayer.class
oopd-eksamen-2010\model\HumanPlayer.ctxt
oopd-eksamen-2010\model\HumanPlayer.java
oopd-eksamen-2010\model\package.bluej
oopd-eksamen-2010\model\Player.class
oopd-eksamen-2010\model\Player.ctxt
oopd-eksamen-2010\model\Player.java
oopd-eksamen-2010\model\Position$1.class
oopd-eksamen-2010\model\Position.class
oopd-eksamen-2010\model\Position.ctxt
oopd-eksamen-2010\model\Position.java
oopd-eksamen-2010\model\config\Configuration.class
oopd-eksamen-2010\model\config\Configuration.ctxt
oopd-eksamen-2010\model\config\Configuration.java
oopd-eksamen-2010\model\config\IParser.class
oopd-eksamen-2010\model\config\IParser.ctxt
oopd-eksamen-2010\model\config\IParser.java
oopd-eksamen-2010\model\config\LineSegment.class
oopd-eksamen-2010\model\config\LineSegment.ctxt
oopd-eksamen-2010\model\config\LineSegment.java
oopd-eksamen-2010\model\config\package.bluej
oopd-eksamen-2010\model\config\ParserException.class
oopd-eksamen-2010\model\config\ParserException.ctxt
oopd-eksamen-2010\model\config\ParserException.java
oopd-eksamen-2010\model\config\TextParser.class
oopd-eksamen-2010\model\config\TextParser.ctxt
oopd-eksamen-2010\model\config\TextParser.java
oopd-eksamen-2010\model\config\TextParserTest.class
oopd-eksamen-2010\model\config\TextParserTest.ctxt
oopd-eksamen-2010\model\config\TextParserTest.java
oopd-eksamen-2010\model\config\__SHELL0$1.class
oopd-eksamen-2010\model\config\__SHELL1$1.class
```

```
oopd-eksamen-2010\model\strategies\IStrategy.class
oopd-eksamen-2010\model\strategies\IStrategy.ctxt
oopd-eksamen-2010\model\strategies\IStrategy.java
oopd-eksamen-2010\model\strategies\package.bluej
oopd-eksamen-2010\model\strategies\RandomStrategy.class
oopd-eksamen-2010\model\strategies\RandomStrategy.ctxt
oopd-eksamen-2010\model\strategies\RandomStrategy.java
oopd-eksamen-2010\model\strategies\StrategyFactory$Type.class
oopd-eksamen-2010\model\strategies\StrategyFactory.class
oopd-eksamen-2010\model\strategies\StrategyFactory.ctxt
oopd-eksamen-2010\model\strategies\StrategyFactory.java
oopd-eksamen-2010\model\strategies\WallHuggerStrategy.class
oopd-eksamen-2010\model\strategies\WallHuggerStrategy.ctxt
oopd-eksamen-2010\model\strategies\WallHuggerStrategy.java
oopd-eksamen-2010\model\strategies\WallHuggerStrategyTest.class
oopd-eksamen-2010\model\strategies\WallHuggerStrategyTest.ctxt
oopd-eksamen-2010\model\strategies\WallHuggerStrategyTest.java
oopd-eksamen-2010\view\MainWindowFrame$1.class
oopd-eksamen-2010\view\MainWindowFrame$GameInput.class
oopd-eksamen-2010\view\MainWindowFrame.class
oopd-eksamen-2010\view\MainWindowFrame.ctxt
oopd-eksamen-2010\view\MainWindowFrame.java
oopd-eksamen-2010\view\NewGameDialog$PlayerPanel.class
oopd-eksamen-2010\view\NewGameDialog.class
oopd-eksamen-2010\view\NewGameDialog.ctxt
oopd-eksamen-2010\view\NewGameDialog.java
oopd-eksamen-2010\view\package.bluej
oopd-eksamen-2010\view\README.TXT
oopd-eksamen-2010\view\TronColors.class
oopd-eksamen-2010\view\TronColors.ctxt
oopd-eksamen-2010\view\TronColors.java
oopd-eksamen-2010\view\TronDisplayPanel.class
oopd-eksamen-2010\view\TronDisplayPanel.ctxt
oopd-eksamen-2010\view\TronDisplayPanel.java
```