TEC | Tecnológico de Costa Rica

Knight's Tour

User Manual

Sebastián M. Chen Cerdas
Valerie M. Hernández Fernández
Oscar M. Soto Varela

Costa Rica Institute of Technology
Cartago Central Technology Campus
Computer Engineering Academic Area

CE 3104 Languages, Compilers, and Interpreters
Eng. Marco Rivera Meneses, MGP
Class 01

I Semester

May 19, 2023

Knight's Tour

User Manual

♘

To correctly run the program for the Knight's Tour Problem you need to meet some hardware and software requirements shown in this section.

Requirements

♘ Hardware

To run this program, you need at least 128Mb of RAM and 1.5Mb of storage on the device that you are using.

♘ Software

This program is made 100% on the Racket programming language, so to run it you need an environment where this language can run, it is preferred to use *DrRacket* to run the program properly. Another software requirement would be to run the program in windows 10 (there is no guarantee that it would run correctly in other OS)

Run

After downloading and extracting the source code from the *GitHub* repository **github/valeriehernandez-7/Knights-Tour** you have to go to the *src/app* directory and open the *kt_algorithm.rkt* file in the selected environment.

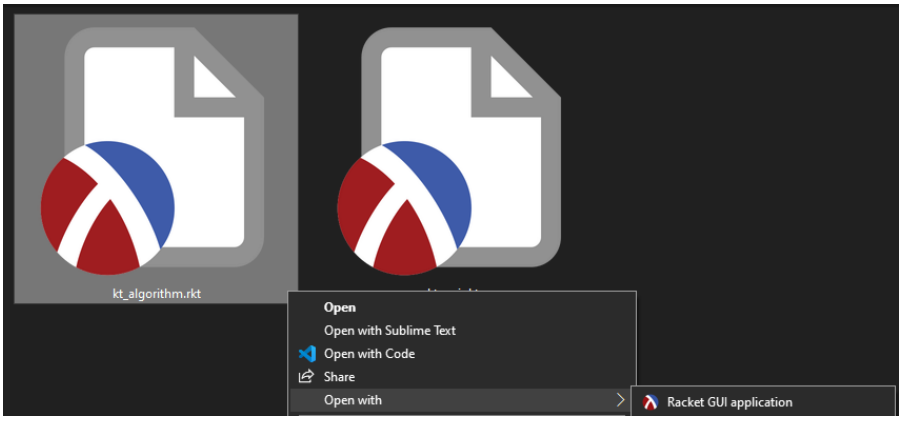

Figure 1. Loading the program in the environment.

After *DrRacket* finishes loading you can click the green run button that's placed on the top right of the window to run the code.

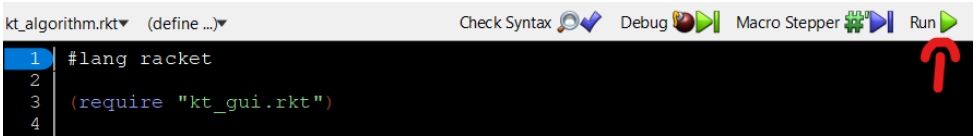

Figure 2. Running the program.

Clicking the run button should open a terminal on the bottom of the window where you can enter text.

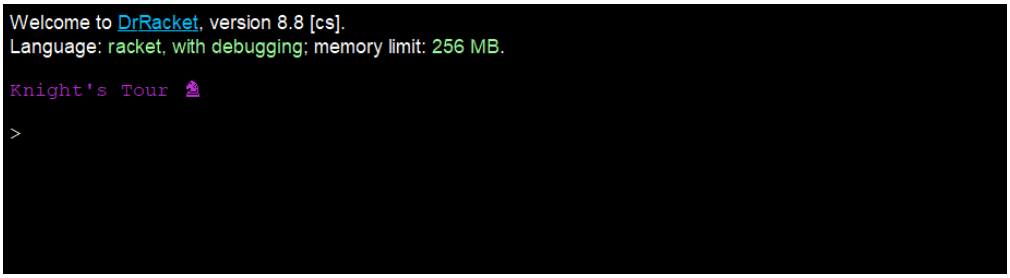

Figure 3. Terminal available after running the program.

In this terminal you can call all the functions of the program.

Features

Paint

Let's start with running the graphic interface, to call the graphic interface inside the terminal you can write the function (paint) with the parameters: **(paint [board size] [solution])**

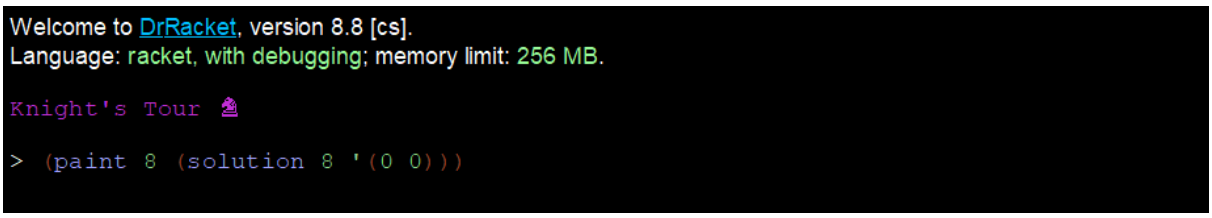

Figure 4. Starting the graphic interface.

It should open another window with a case for the knight's tour problem with a board size of 8 and a solution from the starting position (0 0) loaded in.

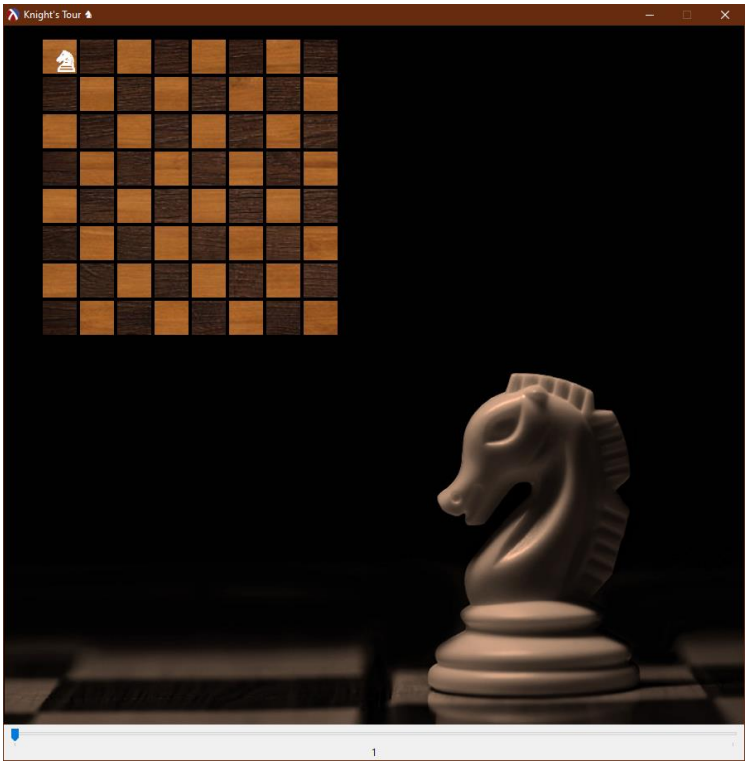

Figure 5. Graphic interface for the knight's tour problem.

To visualize the movements of the knight on the board you can use the slider on the bottom of the window, while you slide it to the right the horse on the board should start moving and leave the movement number on the square it visited.

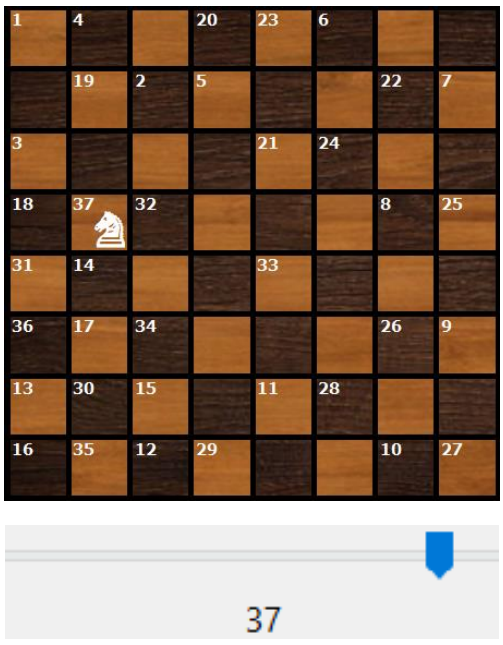

Figure 6. Knight's movement across the board through the slider.

Solution

This program also supports the creation of solutions to the knight's tour problem, to create them you need to use the same terminal as the one used to call the graphic interface (Figure 3) but calling a different function, in this case (solution) with the arguments: **(solution [board size] [initial position])**

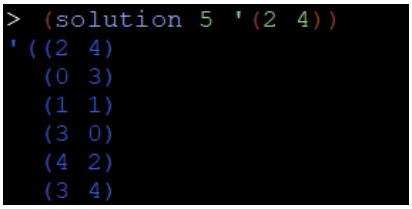

Figure 7. Creating a solution through the terminal.

Solutions

You can get n solutions for a board with the function (solutions) with the arguments: **(solutions [n-solutions] [board size] [initial position])**

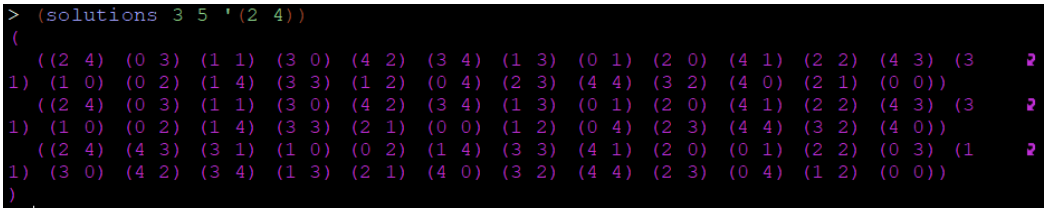

Figure 8. Creating 3 solutions for the same board and start.

Test

The last feature that can be called from the terminal is (test) this function transforms a solution into a matrix and prints the terminal with it you can call it with the arguments: **(test [board size] [solution])**

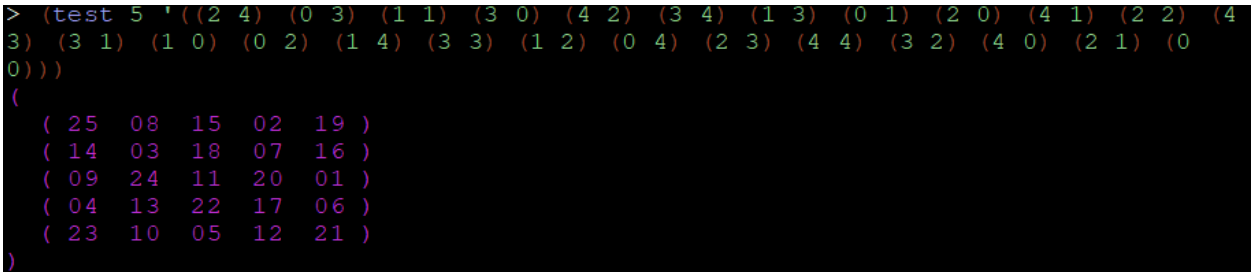

Figure 9. Matrix generated by test on the terminal.

Notes

If you need to know more about each function, you can go to this project documentation to see more details. Also, here are some samples for the Knight's Tour functions (Figure 10), you can rewrite them with the specific values that you want.

```
(define board-size 5) ; board size
(define knight-position '(2 2)) ; start position
(define n-sol (* board-size 2)) ; number of solutions


(displayln "\n>>> KT-Solution 💡 <<<\n")
(solution) ; solution - default
(solution board-size knight-position) ; solution - custom


(displayln "\n>>> KT-Solutions 📦 <<<\n")
(solutions) ; solutions - default
(solutions n-sol board-size knight-position) ; solutions - custom


(displayln "\n>>> KT-Test ✅ <<<\n")
(test) ; test - default
(test board-size (solution board-size knight-position)) ; test - custom


(displayln "\n>>> KT-Paint 🎨 <<<\n")
(paint) ; paint - default
(paint board-size (solution board-size knight-position)) ; paint - custom
```

Figure 10. Tests for the Knight's Tour.