

EPFU Carcassonne Project

Game description

Players are given a common set of squared tiles. Each tile is divided into 4 segments, i.e., north, east, south, and west. Each segment of a tile can be covered with one of the following symbols:

- city,
- road,
- plain.

Moreover, in the middle of a tile there can be a temple.

Players play in turns. A player in his turn choose one tile from the set of tiles, take it and place on his board. Starting from the second tile, a player must place next tile in contact w edge of at least one already placed tile (the whole map must be connected). Each player is building his own board. However, the players share the set of tiles. When a tile is taken by some player, it's no longer available for the second player.

Each player may rotate the tiles. The aim is to arrange a board using the tiles to maximize the total amount of points. You don't have to use all tiles. Two tiles must have the same symbol on neighboring segments. Examples of valid tiles arrangement:



And an invalid one:



For each city segment you are given 1 point. However, if whole city is surrounded, then the city is completed, and each city segment is worth 2 points. There are also city segments with bonus point, marked with a small shield, which gives +1 points for a tile.

Here are 3 cities, no one is completed, but the city to the left gives bonus point (it has blue shield), so the cities are worth 4 points (the blue marker does not matter in our game):

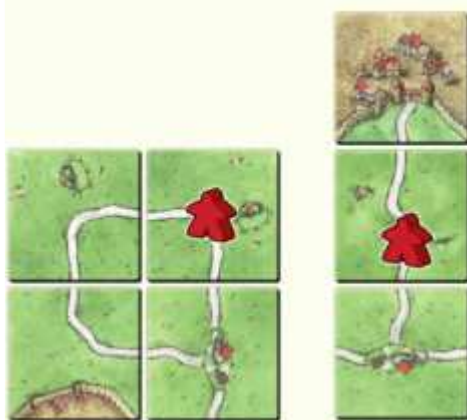


In the following picture:



there is one city in the middle which is completed, and one city which is incomplete. The scoring for the cities is $2+2$ (completed city) + $1=5$ points. Total scoring is 6 points due to the road (the red marker does not matter in our game).

For each road segment you are given 1 point. However, if a road is completed, then each road segment is worth 2 points. A road is completed if it starts and ends in: city, temple or crossroads. Below, there are two completed roads (and 3 road segments that are incomplete):



First pictures results in 11 points for roads, and second case counts to 8 points for roads.

For each temple you are given 1 point plus 1 point for each surrounding tile. So, maximum number of points for a temple is $1+8=9$.

The main idea of the game is based on Carcassonne board game, however it is simplified here.

Your task

Your aim is to deliver a program that will play the game on behalf of you (auto mode).

The set of tiles will be provided in an input file. Program must put final board (and score) into the output file. Program must accept some command line parameters, including the names of files, for instance:

```
carassone.exe tiles.txt board_input.txt board_output.txt
```

Complete set of command line parameters and format of input and output file are to be decided within the tutorial group.

Two modes of program are expected. In an **interactive mode** only one player is assumed. Once program is run it should lead a user through a game, i.e., ask user to choose a tile, let the user decide where to put a tile, calculate scoring, etc.

In an **auto mode** program is supposed to play automatically against other program. In a single program run it should do just one turn. So, it must read everything what is needed from files, choose and place a tile, save everything to file and quit. Therefore, a tournament between two players can be arranged by consecutive calls like this:

```
player1.exe tiles.txt board_input1.txt board_output1.txt
player2.exe tiles.txt board_input2.txt board_output2.txt
player1.exe tiles.txt board_output1.txt board_output3.txt
player2.exe tiles.txt board_output2.txt board_output4.txt
...
```

In the auto mode your program can do its job in no more than 10 seconds for 30 tiles.

It is group project, however, each student is expected to be able to explain any part of your code. Contribution of each student **must be clear** and **reflected in the source files**.

Schedule (7 meetings)

1. AIM OF THE MEETING:

Introduction, preliminary discussion, kick-off. Establish your team. Use flowcharts to sketch the general concept. Design a structure of input/output files. Organize your work (git recommended).

2. EXPECTATIONS:

- Each team will present proposition of file formats and program arguments. We will choose one solution, that must be supported by each team.
- Each team must have and use github account.
- You are expected to present preliminary source code for the interactive mode (main loop, first interaction with a user).

3. EXPECTATIONS:

- You should have general concept of your source code
 - Your project should be divided into several files
 - You should have set of functions listed, including parameters and resulting values. It does not have to be a final set of function, try to have complete vision assuming your current C knowledge.
- Assign team members to each file/function, so the responsibilities are clear for everyone

4. EXPECTATIONS:

- You are expected to be prepared with data structures. It should reflect your current state of knowledge and can be completely changed later in the project.
- Source code for interactive code should be more advanced, printing board on the screen, some user interaction with a game should be provided.

5. EXPECTATIONS:

- First attempt to reading and writing to files
- Interactive mode almost ready - it's possible to play a game

6. EXPECTATIONS:

- Interactive mode ready. It can be extremely simple, for instance you can choose first tile that can be placed on your board.
- Program must allow for arguments that are agreed and must read and write properly to files. We will test if programs are able to cooperate each other.

7. Final tournament, assessment.

Teams are requested to arrange folder REPORTS on github and put one page handouts before each tutorial. In the handouts each team member in separate paragraph must write what was his or her contribution since the last tutorial. Dates of file uploads are important.

During meetings from 2 to 6 each team should present their achievements. You are requested to nominate a spokesman for each meeting and each time it should be different team member. Spokesman has only 10 minutes and this time regime will enforced. To present source code you can either use your own computer or you must have the source on usb stick. Please, get prepared so the presentations will go smoothly.

Remember - it's better to have simple solution that will work, then the most sophisticated algorithm but not working. Try to work incrementally, be agile! Try to always have code that can be compiled.

Assessment

You can obtain up to 3 points for your work reported during meetings 2-6, in total 15 points. During the meeting #7 there will be final assessment. Elements of the assessment are as follows (but not limited to):

- quality of the code (use of proper data types, formatting, comments),
- logical structure of sources (division into files/functions),
- fulfillment of the requirements.

Other expectations for maximal points:

- sophisticated data types like structures, enums, ...,
- preprocessor macros,
- conditional compilation controlled by preprocessor macros,
- explicit dynamic memory allocation,
- must work under windows or linux control without any additional libraries (if you really need any then contact me).

You can obtain up to 11 points.

At the last tutorial we will organize a tournament. Your program will fight against programs prepared by other groups. The winner will obtain 4 points, second place 3 points, third place 2 points, and any other team that will participate in the tournament 1 point.

Totally, there is 30 points for tutorial (15+11+4).

Extra points

Tasks for extra points:

- tiles' set generator
- points calculator
- think how you can prevent cheating by your opponent

Anybody is welcome to do these additional tasks, but you are recommended to share your intentions and vision during the tutorials or via facebook.

Type of tiles in the game:

