Machine Translated by Google

Java Programming — Project

L3 Computer Science – 1st semester – fall 2024 – Version 1, dated October 11, 2024

Subject: Making the Cascadia game in Java

The goal of this project is to make an offline PC version of a board game: Cascadia. This is a strategy game where one to four players seek to create the most welcoming environment for local wildlife in the Cascadia region.
Bosphore Jug´e, 11, who loves this game, asked Santa Claus to receive an electronic version; as he is in no hurry, he is ready to wait until January 12. Help him!

The game

The Cascadia game rules are available online. We invite you to read it carefully, since this is the game that you will have to program.

Advice

ÿ Read the entire statement before even starting anything! ÿ The goal of this
project is to show us that you know how to program “object”. You will therefore be penalized if you do not sufficiently exploit the concepts seen in class.

ÿ Read the rules of the game carefully and take the time to think about the best way to implement a particular game mechanism before you start coding. If you start coding before you have thought about it enough, you will, in practice, be forced to go back, which will require much more time and effort.

ÿ The subject is evolving: respect the phases of realization, but keep in mind what you will need to do in the following phases when making implementation choices!

The program `realize

You will build your game in five phases. Each phase must be completed and functional before moving on to the next phase. However, when building a given phase, it is of course essential to maintain a modular code that will allow easy integration of changes made during subsequent phases.

Phase 1: The base

First, you need to make a simplified version of the game, in command line, in which

ÿ instead of having hexagonal tiles, or only 1×1 square tiles, two tiles being considered neighbors if they have a common edge; this will make geometric aspects easier to handle: difficult to display a hexagon on the command line. . . ;

ÿ as with hexagonal tiles, the starting tile is made up of three contiguous 1×1 tiles; you are left to choose which ones, and what shape (L or bar) the starting tile will be; ÿ each 1×1 tile contains a single type of
landscape (so you never need to rotate them) and allows the placement of exactly two animal species: you are left to choose which ones at your leisure;

Machine Translated by Google

ÿ there are no Nature tokens; ÿ there

are exactly two players; ÿ when the

game is launched, the user is given the choice between the family and intermediate variants.

Phase 2: Graphical display

Once phase 1 is completed, you are asked to set up a simple graphical interface.
You will need to find a suitable way to let users choose between the GUI and the command line interface.

Attention! It is not the quality of the graphical interface that is evaluated in this project, but your skills in design and object programming.

ÿ You must use the zen GUI library provided with this topic (zen-6.0.jar file).

To add a jar to a project in Eclipse, you must:

ÿ Add a lib folder in the project directory and place the .jar file there. ÿ In Eclipse, right-click on
the .jar file and choose Build Path > Add to Build
Path.

ÿ We also provide you with a mini-example (very incomplete) of code using this library and following a classic development model called MVC (Model-View-Controller) in your code:

ÿ A SimpleGameData class is used to manage the game data (the model) as well as all possible actions, following the rules of the game.
ÿ A GameView interface implemented by the SimpleGameView record allows to manage the display graphic (the view).
ÿ The controller implements the game loop and the management of user events: clicks, keys, . . .

Phase 3: The game almost complete

Once Phase 2 is complete, you must make the full game, allowing two to four human players to play, with the exception of the Cascadia achievements (pages 12 to 14). You will therefore need to find a suitable way to let users choose

ÿ the number of players (between two and four); ÿ

between the command line version with square tiles, the graphical version with square tiles, and the graphical version with official hexagonal tiles;

ÿ scorecards used: family variant, intermediate variant, or a card (between A and D) for each animal species.

Phase 4: Single Player and Cascadia Achievements

Once Phase 3 is complete, you need to include a single-player mode, as well as a mechanism to implement Cascadia's achievements from one game to the next, including saving your progress from one game to another. It's up to you to find a reasonable way to do this!

Rendering instructions

ÿ This project is to be done in pairs (i.e. exactly two people) from the same practical work group. You will have until Monday , November 4 at 23:59 to indicate, on e-learning, the name of your project partner. If you have difficulty finding a project partner, contact your practical work supervisor without delay, so that he or she can help you with this.

Machine Translated by Google

If you have not indicated your choice of partner by the requested date

ÿ and if you have not participated in the graded practical work on Monday, November 4, you will automatically be considered as failing the project (therefore in the L3 year); ÿ and if
you have participated in the graded practical work on Monday, November 4, a project partner will be assigned to you assigned automatically, arbitrarily.

We strongly recommend that you do not wait until November 4th to find a project partner. If you have reported to your project manager by Thursday, October 24th at 23:59, we will try to find you a partner by October 25th, which will allow you to coordinate a minimum before the All Saints' Day holidays.

ÿ A ÿ defense will be organized on Monday 25, Tuesday 26 and Wednesday 27 November. During this defense, you will do a demonstration on a machine in the practical work rooms and will be questioned on the project. The purpose of this defense is, among other things, to allow your teacher to encourage you if you are on the right track, and to put you back on the right path if not.

It is expected that at this stage of the project, Phase 1 will be completed, and Phase 2 will be well underway, if not completed; indeed, Phase 2 is usually one of the most difficult for students, but if you have nothing to show us, we will have nothing to suggest to you.

ÿ A first intermediate work, which will be presented during the defense ÿ, must be deposited on e-learning before Sunday November 24 at 11:59 p.m.

In addition, by 24 November at 23:59 at the latest, self-assessment form no. 1 must be completed by each group. ÿ The final submission deadline is

Sunday , January 12 at 23:59. After this deadline, the submission area will be closed, and we will not consider any submission that may be sent to us otherwise (for example by email). However, it is possible to submit your submission multiple times within the allotted time; only the latest version will then be taken into account.

Finally, by 12 January at 23:59 at the latest, self-assessment form no. 2 must be completed by each group; it is the same form as form no. 1, with two additional questions concerning your consideration of the remarks made during the thesis defense ÿ.

Intermediate rendering

Your intermediate output will have to consist of a zip archive: any rar, tar.gz, 7z or other will not be opened. This archive will contain:

ÿ a src directory containing the project sources; ÿ a docs

directory containing a user manual (user.pdf) and a manual explaining the architecture you have chosen (dev.pdf); the user.pdf manual must be readable by Bosphore, 11 years old, who knows the rules of Cascadia and will try to use your program without having read any other document, not even the project statement; ÿ a docs/doc directory containing the javadoc, written in English; ÿ a lib directory

containing the libraries on which the application depends; ÿ an executable jar

Cascadia.jar, which works with the command java -jar Cascadia.jar.

This archive will be named Name1 Name2 Cascadia.zip, where the names are those of the members of the pair in alphabetical order. Extracting this archive should create a directory named Name1 Name2 Cascadia and containing all the elements requested above.

Final rendering

Your final output should consist of a zip archive: any rar, tar.gz, 7z or other will not be opened. This archive will contain:

Machine Translated by Google

ÿ a src directory containing the project sources; ÿ a docs directory

containing a user manual (user.pdf) and a manual explaining the architecture you have chosen (dev.pdf); the user.pdf manual must be readable by Bosphore, 11 years old, who knows the rules of Cascadia and will try to use your program without having read any other document, not even the project statement; the dev.pdf manual must notably include a section dedicated to improvements and corrections made since the ÿ defense; ÿ a classes directory, empty in the archive, and which will contain the classes once compiled; ÿ a lib directory containing the libraries on which the application depends; ÿ an executable jar

Cascadia.jar, which works with the java -jar Cascadia.jar command, and which therefore has an adequate manifest file; ÿ a

build.xml file, written by hand, which allows

ÿ compile sources (target compile); ÿ create the executable

jar (target jar); this should be the default target; ÿ generate the javadoc, written in English, in the docs/doc

directory (target javadoc); ÿ clean the project so that only the requested elements remain (target clean).

This archive will be named Name1 Name2 Cascadia.zip, where the names are those of the members of the pair in alphabetical order. Extracting this archive should create a directory named Name1 Name2 Cascadia and containing all the elements requested above.

Scoring criteria

ÿ Bosphore, 11 years old, must be able to play your game and enjoy it; ÿ the cleanliness and

readability of the code will have a very important weight in the grade; ÿ the architecture that you have defined

(interfaces, classes, etc.) must be given in the PDF documents and will also have a very important weight in the grade; thus, your code must be modular, so that adding other extensions (for example more cards, one more level, . . .) is as easy as possible; ÿ your code must not contain methods longer than 20 lines; ÿ no code duplication, and respect the principles of object-oriented programming; ÿ no global variables; ÿ no useless code;

ÿ presence of the different reports and, consequently, correct spelling! ÿ taking into

consideration the remarks made during the defense ÿ for the final rendering.

Generally speaking, if everything is perfect according to your self-assessment form, and if it accurately reflects the reality of your project, your grade should be excellent.

Rules to be followed imperatively – Sudden death

Here is a list of rules that you must respect absolutely. If you do not respect even one of these rules, and depending on the situation and your explanations, the consequence for you could range from the removal of a few points on your grade to a failing status for the project, therefore for the L3.

ÿ You must participate in the ÿ defense on November 25, 26 or 27 ; if only one of the two members of a pair participates in the ÿ defense, the absent member will be considered as failing for the project.

ÿ You must submit a first assignment on e-learning before November 24 at 23:59.

Machine Translated by Google

ÿ You must submit your final version on e-learning before January 12th at 23:59. ÿ In both cases (intermediate version and final version), your code must compile. ÿ You must include a javadoc written in English and user.pdf and dev.pdf files with your final version.

ÿ Your archive will need to have the correct name, be a .zip archive, and produce a directory that has the good name.

ÿ The project must not use or include any external library other than those indicated in the subject.

ÿ The project must not contain code copied and pasted from the net. The presence of such code will be interpreted as an attempt at cheating, and will therefore be accompanied by a summons before the IGM disciplinary council. ÿ The project must manage inputs/outputs in accordance with the course; for example, it must not do not use java.io.File.

ÿ The project should not contain any fields with visibility other than private, and any method with public visibility should first check that its arguments are reasonable. For example, if a function throws a NullPointerException, it should be caused by a call to Objects.requireNonNull that detected that the proposed argument was null.

References

1. Ant Manual for building the build.xml file;
2. How to create an executable jar?
3. The JavaDoc ;
4. File inputs/outputs ;
5. The Zen graphic library , its sources and its documentation ;
6. An example of code using the Model-View-Controller development model to program with Zen;
7. A video showing how to integrate Zen into a project with Eclipse;
8. The photographs that Bosphorus included in his letter to Santa Claus: the list of tiles, the list of cards, the cards to use for the family variants and intermediate, and an overview of the Nature tokens, animal tokens, tile backs, and scorecard.