This course has already ended.
The latest instance of the course can be found at: O1: 2023

Luet oppimateriaalin englanninkielistä versiota. Mainitsit kuitenkin taustakyselyssä osaavasi suomea. Siksi **suosittelemme, että käytät suomenkielistä versiota**, joka on testatumpi ja hieman laajempi ja muutenkin mukava.

Suomenkielinen materiaali kyllä esittelee englanninkielisetkin termit. Myös suomenkielisessä materiaalissa käytetään ohjelmien koodissa englanninkielisiä nimiä kurssin alkupään johdantoesimerkkejä lukuunottamatta.

Voit vaihtaa kieltä A+:n valikon yläreunassa olevasta painikkeesta. Tai tästä: **Vaihda suomeksi**.

Chapter 12.2: Robot Tournament

About This Page

Questions Answered: Which robot tribe is the best?

Topics: Designing your own algorithm. Programming in something that vaguely resembles machine language.

What Will I Do? Program.

Rough Estimate of Workload:? Depends entirely on how much effort you want to put in.

Points Available: C100. You'll get a minimum of fifty just for taking part; the other fifty comes from your placement in a tournament.

Related Modules: RobotTribes.

../_images/robot_fight1.png

Program Your Own Robot Tribe

This chapter consists of a single programming assignment. It continues where Chapter 11.2 left off.

Task description

Define your own robot tribe in RoboSpeak. Try to design it to be effective in duels against other tribes.

Instructions and hints

- For this assignment, you will submit two files:

    - `mytribe.tribe`, which contains the RoboSpeak for your tribe; and
    - `mytribe.png`, which is the image you've chosen to depict your tribe's robots.

    Note that you don't submit any Scala code.

- The general instructions for placing the RoboSpeak and image files within the module are in Chapter 11.2.
- You can use one of the given image files, or you can create or download some other image.
- It would be nice if you'd start the RoboSpeak program with a comment that introduces your tribe and its behavior. If you want to write other comments in your code, that's great too, but commenting isn't mandatory.
- You must use the RoboSpeak language as defined in Chapter 11.2. Even if you've made your own additions to the language, don't use them in this tribe.
- The tribe file mustn't be larger than a few dozen kilobytes. During the tournament (see below), a maximum of one thousand RoboSpeak instructions will be executed during a single robot turn.
- You may find writing programs in RoboSpeak to be arduous work. Sorry! ;-)

Grading

The only thing that A+ will automatically assess is whether your RoboSpeak program is compatible with the RobotTribes module. You'll need to pass that preliminary check to score any points for this assignment. If your program fails to run or ends up in an infinite loop during this check, A+ will give you zero points. Otherwise, A+ gives you one (1) point to signal that you passed the initial test. This is just the prologue to the actual assessment, though.

Immediately after the deadline, the course staff will check all the programs that passed the initial test. Assuming the staff member approves your program, you are certain to score a minimum of 50 points for it. We'll approve any and all programs that demonstrate an effort to write a new tribe. It doesn't need to be fancy and it doesn't need to be big, but it needs to be something that you created.

One of the teaching assistants will organize a tournament that pitches the approved tribes against each other. Your final score, between 50 and 100, is determined by how well your tribe does in that tournament.

We'll add the final scores in A+ after the deadline, as soon they're ready.

The Tournament

Duels and matches

The tribe submitted by each student or pair will fight one "match" against each other student-created tribe. A match is a sequence of duels. Each duel is set up as follows:

- The dueling ground is a 20-by-20 robot world with ten walls at random locations.
- The two dueling tribes each start with six robots. The robots are initially placed in random locations and face in random directions.
- In addition to the dueling tribes, there are 30 peaceful robots of the bunny tribe (see `bunny.tribe`). The bunnies are placed at random locations that aren't adjacent to any of the dueling robots.

That is, the settings are the same that you get by selecting Tribal → Basic Duel in the app's GUI.

If one of the tribes wipes its opponent off the map, it wins the duel. It can also happen that neither side is completely annihilated. In that case, the winner is the tribe with more robots after many rounds of play.

The winner of a match is the tribe that won more duels (against the same opponent). The winner of the tournament is the tribe that won the most matches (against different tribes). The more matches your tribe wins, the more points you'll score.

Results and awards

We publish the robot tournament's results soon after the Week 12 deadline. We'll also see if we can come up with small prizes for the creators of the top tribes.

Points C 60 / 100   My submissions 3 / 10 ▾   Deadline Wednesday, 9 December 2020, 12:00
To be submitted alone or in groups of 2

This course has been archived (Tuesday, 31 August 2021, 23:59).

Assignment 10 (My tribe)

Select your files for grading

mytribe.tribe

Choose File No file chosen

mytribe.png

Choose File No file chosen

Submit

Feedback

Please note that this section must be completed individually. Even if you worked on this chapter with a pair, each of you should submit the form separately.

Accepted   My submissions 1 ▾

This course has been archived (Tuesday, 31 August 2021, 23:59).

Time spent: (*) Required

Please estimate the total number of minutes you spent on this chapter (reading, assignments, etc.). You don't have to be exact, but if you can produce an estimate to within 15 minutes or half an hour, that would be great.

240

Future offerings of O1 will probably also feature RoboSpeak in some form or another, but it's nice to vary the theme occasionally. Do you have any ideas about how to change the language in the future?

We especially welcome ideas that might add strategic elements to the tournament while keeping the abilities of individual robots fairly modest. We may reward the best ideas by realizing them.

Work for O1?

O1's teaching assistants are Aalto students; the automatic tests of O1's assignments are also being developed by students. Every year, the CS department hires several students to work on O1, A+, and/or other programming courses; this is a nice way for the students to contribute, learn, and get work experience and a salary. If you want to hear more, indicate your interest via this form, and we'll e-mail you when there are jobs on offer. (Filling this form isn't a requirement for applying, but it's a way of making sure you're kept informed.)

Please note that you can one or more options.

- ☑ **I'd like to be notified by email when O1 is hiring teaching assistants for Fall 2021. (We'll do that in late spring or early summer. A teaching assistant needs to know the course contents well, but broader programming experience isn't a requirement.)**
- ☑ **I'd like to be notified by email about summer jobs related to O1 and/or A+. (We'll notify you if and when new initiatives are launched. The specific requirements will depend on the project, but most of our summer jobs are suitable for BSc students.)**
- ☐ **I have prior programming experience beyond O1 and I'd like to be notified about part-time jobs related to O1 and/or A+ between January 2021 and May 2021. (We'll notify you before the end of 2020 in case there are positions available.)**

Written comment or question:

You aren't required to give written feedback. Nevertheless, please do ask something, give feedback, or reflect on your learning! (However, the right place to ask urgent questions about programs that you're currently working on isn't this form but Piazza or the lab sessions. We can't guarantee that anyone will even see anything you type here before the weekly deadline.)

Submit an update

What now?

O1 ends here — almost. Here's what's left:

- Chapter 12.3, which contains optional material on the Swing GUI library.
- Week 13, which opens soon after the Week 12 deadline. It doesn't have any normal chapters, but it does contain **the mandatory end-of-course feedback form** in Chapter 13.1. You'll need to fill that in by December 16th, 2020.
- We'll also publish a final weekly bulletin as Chapter 13.0. It will contain results from the robot tournament, the TAs' favorite text adventures, follow-on courses, and more.

Credits

Thousands of students have given feedback that has contributed to this ebook's design. Thank you!

The ebook's chapters, programming assignments, and weekly bulletins have been written in Finnish and translated into English by Juha Sorva.

The appendices (glossary, Scala reference, FAQ, etc.) are by Juha Sorva unless otherwise specified on the page.

The automatic assessment of the assignments has been developed by: (in alphabetical order) Riku Autio, Nikolas Drosdek, Joonatan Honkamaa, Jaakko Kantojärvi, Niklas Kröger, Teemu Lehtinen, Strasdosky Otewa, Timi Seppälä, Teemu Sirkiä, and Aleksi Vartiainen.

The illustrations at the top of each chapter, and the similar drawings elsewhere in the ebook, are the work of Christina Lassheikki.

The animations that detail the execution Scala programs have been designed by Juha Sorva and Teemu Sirkiä. Teemu Sirkiä and Riku Autio did the technical implementation, relying on Teemu's Jsvee and Kelmu toolkits.

The other diagrams and interactive presentations in the ebook are by Juha Sorva.

The O1Library software has been developed by Aleksi Lukkarinen and Juha Sorva. Several of its key components are built upon Aleksi's SMCL library.

The pedagogy of using O1Library for simple graphical programming (such as `Pic`) is inspired by the textbooks *How to Design Programs* by Flatt, Felleisen, Findler, and Krishnamurthi and *Picturing Programs* by Stephen Bloch.

The course platform A+ was originally created at Aalto's LeTech research group as a student project. The open-source project is now shepherded by the Computer Science department's edu-tech team and hosted by the department's IT services. Markku Riekkinen is the current lead developer; dozens of Aalto students and others have also contributed.

The A+ Courses plugin, which supports A+ and O1 in IntelliJ IDEA, is another open-source project. It was created by Nikolai Denissov, Olli Kiljunen, and Nikolas Drosdek with input from Juha Sorva, Otto Seppälä, Arto Hellas, and others.

For O1's current teaching staff, please see Chapter 1.1.

Additional credits for this page

The notion of programmable "tribes" or "species" that fight each other on a grid comes from a programming assignment by Nick Parlante.

Viljami Nurminen and Rune Pönni contributed additional commands to the RoboSpeak language, drawing on student feedback.

Teemu Sirkiä wrote the code for running the robot tournament. Joonatan Honkamaa and Otto Seppälä suggested an efficiency improvement to `TribalBot` for tournament purposes.

a drop of ink