👤 Binh Nguyen 🔻 CS-A1110 O1 ▼ v1.20.4 Course This course has already ended. The latest instance of the course can be found at: O1: 2023 **↑** CS-A1110 Course materials Chapter 10.2: Programming Paradigms » « Week 10 Course materials Your points CS-A1110 / Week 10 / Chapter 10.1: Your Text Adventure **Form a group** H Code Vault 2 Lab Queue 2 Luet oppimateriaalin englanninkielistä versiota. Mainitsit kuitenkin taustakyselyssä osaavasi suomea. Siksi suosittelemme, että käytät suomenkielistä versiota, joka Telegram chat on testatumpi ja hieman laajempi ja muutenkin mukava. Lab sessions Suomenkielinen materiaali kyllä esittelee englanninkielisetkin termit. Myös suomenkielisessä materiaalissa käytetään ohjelmien koodissa englanninkielisiä nimiä kurssin alkupään johdantoesimerkkejä lukuunottamatta. Glossary Voit vaihtaa kieltä A+:n valikon yläreunassa olevasta painikkeesta. Tai tästä: Vaihda suomeksi. Scala reference O1Library docs **FAQ** Chapter 10.1: Your Text Adventure IntelliJ installation Learning goals About This Page Style guide Questions Answered: What will my own text adventure be like? Debugger Topics: Practice on designing a program independently. Explore other programming topics as you choose. You can practice on pretty much any topic covered so far in O1 Resources or even something we haven't covered. For the reader What Will I Do? Program. Make design decisions about which sort of program to create and how. Rough Estimate of Workload:? Even though you can work from the given starter code, deciding on your goals, refining them, and coming up with the needed solutions can be very difficult for a beginner. Even if you happen to be a seasoned programmer already, you can turn up the difficulty for a greater challenge. The length of the assignment depends on your own ambitions, but this will be the most time-consuming program in O1 for nearly all students. We recommend that you reserve an absolute minimum of 10 hours, and preferably at least 15. There's no upper limit — you can sink weeks into this if you're so inclined. Points Available: C200. Related Modules: Adventure. .../\_images/person11.png Task Description Forget about the forest, the remote, the batteries, and the other elements of the given game world and write a new text adventure game. Use any elements you like. Let your hair down and create! Here are some ideas that you can develop further if you like. But feel free to ignore this list and come up with something entirely of your own instead. • Classic adventure. The player could explore an ancient pyramid, for instance, or venture into a ruined castle. The game world consists of rooms, passages, and the

• Mystery adventure. The player has to recover a lost item, find out Who Dunnit, or something of the sort.

• Literary adventure. The player takes the role of Alice, captain Ahab, Travis Bickle, Dolores Abernathy, or whoever.

while the game is in full swing, that's okay, too, as long as it doesn't take a disproportionate effort to discover the goal.

o This map should be in PNG, GIF, or PDF format or (failing that) as a character-based diagram in a plain text file.

than that unless the assistants discover serious flaws such as game-breaking bugs or completely illegible source code.

getting more points is not worth it. You can very well score full marks with a small game of just a few well-functioning areas.

• Write the walkthrough in the same file as the map or in a separate walkthrough.txt (or similar) at the module root.

• The teaching assistants may use the walkthrough as they assess your game. A less frustrated TA may go easier on the marking. :-)

• A help command that prints out instructions for playing. Include any commands you added that the player should know about.

• Action adventure. For instance, the player might need to escape from somewhere.

• Challenges or puzzles that player needs to deal with in order to win the game.

include loss conditions (e.g., the player character dying), but that's optional.

• At least a few new textual commands that the player can use for doing something meaningful.

1. A file that contains a "creater's map" of the game world (cf. the given forest\_map.gif).

• Modern adventure. For instance, the player could be a first-year student at university who needs to accomplish one thing or another. The game world consists of

• Inhuman adventure. The player character is a dog, an insect, a robot, a microbe, a planet-eating space creature, a sentient kumquat, or anything you like.

• Something completely different. Anything that makes a game. For instance, the areas of the game aren't spatial locations but states of mind, stages of life, or

• A new victory condition that the player must reach to win the game. It needs to be something other than simply walking over to another location. You can also

• A welcome message that indicates what the game is about and what the player should try to accomplish. If you prefer that the goal of the game is revealed only

1. A command of the form *use itemname*, which makes the player character use an item. What happens as a consequence should depend on the item.

2. Non-player characters — human or non-human — that move about in the game world and are somehow useful, dangerous, or otherwise meaningful.

The marking won't be very strict. If your game fulfills the above requirements, you can expect to get at least half the available points. Indeed, you can expect to get more

The game does not need to be a mind-blowing spectacle for you to the full points. For that, it's enough that you submit a reasonably high-quality program that works and

Your code will be read by a human, so write it clearly. Remember O1's style guide. You don't need to obey that guide or any other specific set of guidelines slavishly; the

The size of the game is not a significant factor in grading. Although a larger game world may result in a more interesting game, adding more areas merely in the hopes

o However, you are allowed to change the user interfaces if you want to. Indeed, you are completely free to change anything in the given code and add whatever

• The game does not need to be compatible with both the given AdventureTextUI and AdventureGUI. One user interface is enough (and you can even create

o Try to work on one feature of the game at a time, finishing it before moving on to additional features. That way, you'll have something submissible even if you

• Since your game will be played by an actual human, don't forget the help command or the walkthrough. Try to ensure that your game is understandable. For

instance, if the game has a bug that unexpectedly and imperceptibly teleports the player to a different location, the game may be a pain to play and assess.

If an IntelliJ module depends on the classes in another module, you need to indicate the dependency. If you don't, your import s from the other module won't find their

As given, Adventure is an independent module. If you want to use O1Library in your game, find the above setting and click the plus icon on the very right and select

This assignment cannot be submitted for auto-grading via IntelliJ's A+ Courses plugin. Instead, package your program as per the instructions below and submit it via the

There are a few students in O1 who prefer to use programming tools other than IntelliJ. If you're one of them, read the instructions for IntelliJ in this section and use

The zip file should contain a folder that holds your program's Scala source code, a compiled, runnable version of the program as .class files, and the other files

2. Make doubly sure you're looking at the exact version of your program that you intend to submit and that all the files are saved. Then make triply sure.

If you did the assignment with a pair, please make sure you've selected a pair submission in A+. Only one member of the pair should submit the assignment!

© Deadline Wednesday, 25 November 2020, 12:00

Great work! Because this message is prerecorded, any observations regarding your performance are speculation on our part. Please disregard any undeserved

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

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

■ To be submitted alone or in groups of 2

• If you want to use O1Library's tools in your game, go ahead and do so but read the box below first. In fact, you may want to read the box in any case.

The assistants may award additional points at their discretion. You can score them for technical quality but also for interesting and fun details in your game.

important thing is that you get the basics right so that an unknown person can read your code with reasonable effort. Indent your code, pick sensible names for variables,

## We'll publish the scores two weeks after the deadline, at the latest. Each of the assistants picks their own favorite among the games in their packet. In making that decision, the assistant may use any criteria, such as entertainment value, originality, or technical aspects. We'll reward the assistants' favorite games with movie vouchers and make those games available for other O1 students to see in A+, unless

Have fun.

you wish.

Dependencies between IntelliJ modules

Module Dependency when prompted.

Submitting the Assignment

whichever other tools you prefer for creating a similar zip file.

Select Plugins in the settings dialog.

entire module, source code and all.

• We accept games in English, Finnish, and Swedish. • You can build on the given Adventure module. You'll need to edit some or all of the classes. The class Adventure, especially, needs a complete makeover or could

even be replaced by a different class altogether.

**Further Instructions and Hints** 

you specifically request that we don't do that.

buildings and rooms on campus.

Areas that the player can enter.

Task requirements in more detail

Within the Adventure module you submit, you should also include:

O1's teaching assistants will manually assign a mark to your game.

etc. If your code is very hard to read, you're unlikely to score full marks.

You aren't required to write comments in your code but you may do so at your own discretion.

• You may very well focus on building the game logic and leave the user interface(s) untouched.

run out of time before you can implement every feature that you had planned.

2. A walkthrough that describes, step by step, how to win the game.

• Name it map.png or something similar.

• Place it in the module's root folder.

something else.

The game should have:

• One or both of:

On Marking

fulfills the listed requirements.

- your own if you choose). In any case, it's good to follow the principle of keeping the game's inner logic independent of the user interface.
- Hopefully, inspiration will hit you and you'll create something fun. But don't bite more than you can chew: there's only a limited amount of time until the deadline. You can ask the teaching assistants to help you estimate how time-consuming it might be to implement a feature that you have envisioned. Ultimately, though, you are personally responsible for your own schedule.
- o If you wish to write additional comments or instructions for the teaching assistant, you can also include a readme.txt or readme.pdf file. • Additions to the game mechanics are welcome but optional. For instance, you may wish to consider: new directions to move in; vehicles; attributes or abilities for the player character; weights for items that limit what the player can carry; complex commands that consist of multiple words; etc. etc. • In Aalto IT's Linux labs, you can use Inkscape, Dia, or GIMP to draw the game map. There are many alternative graphics programs, too; use whatever is available in
- Most O1 modules already define such a dependency to O1Library. You can see this in the settings: select File → Project Structure in the menu, then pick one of the earlier modules and open the Dependencies tab on the right. (On a Mac, the menu item is :menu:`IntelliJ IDEA → Preferences.)

target.

browser. **Preparations** 

If you don't use IntelliJ

In the readme file, explain that this is not an IntelliJ module and include clear instructions for the teaching assistant to launch your game. 1. Update IntelliJ's A+ Courses plugin to version 1.8: Select File → Settings in the menu. (On a Mac, IntelliJ IDEA → Preferences.)

Your attention, please!

Points C 200 / 200

Submit file

Submit

compliments.

Choose File No file chosen

mentioned above on this page.

3. In IntelliJ's menu, select A+ → Export Module. Select Adventure. Press OK. 4. IntelliJ will create a zip package of your Adventure module. It prompts you for a location and a name. Pick some suitable folder on your computer and a descriptive name such as 654321\_TitleOfMyGame.zip. Please start the name with the student ID(s) or surname(s) of the author(s). 5. Locate the zip file on your computer, in the folder you chose earlier. Examine its contents to make sure it contains everything you want it to. It should contain the

• Find the Installed tab at the top and A+ Courses within that tab. Update to 1.8 and restart to activate the update.

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

My submissions 2 / 10 -

Submitting your program for marking

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

300

My submissions 1

within 15 minutes or half an hour, that would be great.

anything you type here before the weekly deadline.)

—Aperture Announcer

This course has been archived (Tuesday, 31 August 2021, 23:59). Time spent: (\*) Required

Written comment or question:

Submit an update

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.

Credits

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.

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 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

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

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

with input from Juha Sorva, Otto Seppälä, Arto Hellas, and others. For O1's current teaching staff, please see Chapter 1.1.

Chapter 10.2: Programming Paradigms »

a drop of ink

**Privacy Notice** 

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

Additional credits for this page

Course materials

This chapter draws inspiration from classic adventure games by Infocom and the work of David Barnes and Michael Kölling.

« Week 10 Feedback C **Accessibility Statement** A+ v1.20.4 Support