

# G12 Computer Science

## Final Course Project

Most of the topics listed below are from our main reference on JavaScript during this course, namely *Eloquent JavaScript: A Modern Introduction to Programming* from Marijn Haverbeke.

The last two projects listed are, actually, general topics rather than specific projects. Like open problems, you'd need to dive in a bit more, define your idea of a project and see it through. I would help all the steps along the way, of course.

The last topic on Linguistics with Haskell is way more technical. I would provide you with notes. Nevertheless, there is a reference I'd be using as a guide, which is *Computational Semantics with Functional Programming*, from Jan van Eijck and Christina Unger. It would serve two purposes at the same time, namely, introducing you to one of the coolest functional languages out there nowadays, Haskell, and exploring ideas on Linguistics and Logic and have a glance on how all that may lead to quite advanced mathematical methods.

Along the way we will learn about things like regular expressions, parsing or finite automata, besides some elementary topics in linguistics. From the point of view of Haskell, we will about its powerful implementation of abstract data types, its lazy evaluation or its use of pattern-matching definitions. Some pointers on Haskell are [the Haskell wiki](#).

Alternatively, this topic may be addressed through Python with the NLTK module. This has a more applied emphasis. The reference is *Natural Language Processing with Python* by Steven Bird et al. You can find an online version [here](#)

Finally, the last topic on Deep Learning, is an invitation to further explore the topic of neural networks and build one with the use of standard software, either with Google's Tensor Flow or Python's SciKit-Learn library (or alternatively, Anaconda). Simple ideas that are feasible to explore in the time we have are, e.g., character recognition or image classification.

## References

Besides the notes and explanation I will provide you, googling for a topic might give you some pointer.

Now that we are at it, here a question on the interaction of CS and Society: the Library Genesis. There you will find **all textbooks for free in electronic form**. Its use though is not exempt of criticisms and potential problems. While their server is likely illegal in most countries, its ethical status is far from being clear cut: Can learning on your own the fundamentals of human knowledge be reprovable, specially in the digital age? Can accessing such information be reprovable? Should it be left only to the few wealthy or lucky enough? Or rather shouldn't the creator of something new and original be able to claim any right on her/his creation? or should the author get stripped off a chance for such profit instead? Is this an ethical question (e.g., like not killing for a smartphone, or stealing the food of those starving) or just an administrative, legal question (e.g., like a speed limit or the convention to drive on the right)? For more details, search for it in Google or visit the [Wikipedia article on it](#)

## List of Project topics

Tentative List of Course Project Ideas:

### Closed Problems

1. Implement own programming language. Ref: Eloquent JS, Chap 11. See exercises and compiler suggestion
2. A platform Game. Ref: Ibidem., Chap 15. See Exer.
3. Drawing on Canvas: Bouncing Ball. Ref: Ibidem., Chap. 16. See Exer
4. Conway's Game of Life. Ref: Ibidem. Chap 18, exer. 2
5. A painting program. Ref: Ibidem. Chap 19
6. Web site. Ref. Ibidem. Chaps 20/21.
7. JS & Performance (Graph layout, Dijkstra, Profiling,...). Ref: Ibidem, Chap 22
8. Bioinformatics I: Intro to Dynamic Programming. Multiple Sequence Alignment

### Open Problems

9. Dynamic Web Site with Ember
10. GUI Design and Programming (Non web-based)
11. Concurrent and Parallel programming
12. Deep Learning w/ Google Tensor Flow
12. Linguistics with Haskell
13. Bioinformatics II: Analyzing Protein structure with Python and PyMol
14. Bioinformatics III: Determining Protein Structure via Homology Modeling in Python
15. Programming Geometry with Geogebra API