John Litborn

john.litborn@pm.me • Linköping, Sweden

Experience

- **Education group for Programming and Programming Didactics** **Linköping University**
  Department of Computer and Information Science
  Amanuensis, Course Assistant, Head Assistant *2013 – 2017*
  - Recruited my second year at university, I taught courses in Python, C++, Ada and Matlab.
  - Primarily introductory courses, where I held lessons with repetition and shared problem solving, lab assistant where I answered questions and helped solve problems, and correcting lab hand-ins and exams.
  - Later on I also held more advanced courses, teaching the C++standard library, git, object-oriented programming or unit-testing. I was also advisor on projects and grading documentation and hand-ins, and as head assistant taking on light administrative and managerial duties for students and other assistants.
- **Division for Artificial Intelligence & Integrated Computer Systems** **Linköping University**
  Department of Computer and Information Science
  Course Developer, Software Developer *Summer of 2014 & 2015*
  `https://github.com/h00701350103/XPilot-AI_LiU_fork` Python, C, HTML
  - After being unhappy with a course and it's software and giving feedback to the professor, I was offered a summer job to improve it. Where I remade the assignments and improved the Python API for XPilot, a 2D multiplayer space shooter, that was used in the course.
  - I overhauled the structure of the assignments, changed, removed and added several ones and wrote better and clearer instructions and documentation.
  - The XPilot-AI API in use was developed at Connecticut College, and after discussions with them I forked the project and begun modifying the C-Python interface to fix bugs and add features to suit our needs.
  - My second year working on it I also started modifying the client-server netcode to send more data so the client-side API for example didn't have to re-calculate the speed of objects.
- **Ericsson, HiQ** **Linköping**
  IT-Consultant *Mar – Oct 2017*
  - Hired by the consulting firm HiQ to work at Ericssons main offices in Linköping in a team with other consultants.
  - Our task was to updated 4G base-station unit tests written in Erlang to work in a virtualized environment. I was the git-master in the team, and helped the other team members when they encountered problems with Git or Linux.
  - I was also responsible for updating our sections on the internal Wiki, and wrote python and bash scripts for myself and other team members to simplify rote tasks.

Education

- **Linköping University, Faculty of Science and Engineering** *2012 – 2014, 2020 – 2021*
  Computer Engineering *120 Credits*
  - I have almost exclusively studied part-time, early on because of parallel work at the university, later on because of mental health issues.
  - $2/3$ of credits are in programming in a diverse set of languages (Python, C++, C, Java, Ada, VHDL, Prolog, Assembly, GNU MathProg), with a focus on algorithm construction, optimization and low-level code. $1/3$ of credits are in math courses.
  - My second year I won a programming competition in a course on Computer Hardware and Architecture, which consisted of writing a sorting algorithm in microcode for a low-level simulated computer. My solution also beat the professors best implementation, and outclassed the previous student record.

Core Technical Skills

Languages: Python, C++, C, Ada
Tools: Arch Linux, NeoVim, git, command line tools (gdb, pdb, mypy, linters)

Personal Projects

- **Necro Score Bot** *2015-present*
  `https://github.com/h00701350103/necro_score_bot` Python, 1500 lines of code
  - Lacking a good way to track scores on the leaderboards in the indie rhythm roguelike Crypt of the Necrodancer I decided to write a twitter bot that pulls leaderboards from the Steam API, parses them, and tweets out notable updates.
  - Players can add a link to their twitter in their steam profile, which lets the bot tag them and post less notable updates only to them and their followers. It auto-detects cheated or bugged scores and notifies the developers. It supports 16 different characters, 4 run types, and all permutations of 9 game modes for a total of over 400 leaderboards.
  - Uses the twitter API, queries both the official and legacy steam API as well as raw html requests. Discord support and both speedrunslive and NecroLab (a community power-ranking site) leaderboard support was implemented but never activated. Also interfaces with toofz, another community site.
  - Parses XML, json & pickle. Multithreaded, thorough error handling, config file and command line flags and modularly written with >120 functions across 10 files.
  - Initial development spurt during 2015, and a second spurt in 2017 to make it work with the DLC. Has multiple small pull request from other community members and is kept running and maintained to this day by them.
- **Seat Exchange Bot** *May-June 2019*
  `https://github.com/h00701350103/seat_exchange` Python, 3500 lines of code
  - One of several "small" discord bots I've whipped up. This one during a bout of interest in the Korean game show The Genius, where I adapted and implemented one of the games from the show.
  - The bot supports 40 different commands, different player counts, computer players, multiple simultaneous games and permissions.
  - The code is object-oriented, easily extendable, statically typed (using mypy), fully adheres to PEP8 and official python coding standards (passes pylint with very few disabled checks) and even has some comments.
- **Home Automation** *2016-present*
  >400 actions across >30 actions
  - Using the Android scripting application Tasker I've created a system for managing my life, notifying when to wake up, take medication & vitamins, eat, sleep and other reminders. Logs everything to the calendar, as well as time-to-fall-asleep and sleep duration, enabling statistics. Also assists with daily diary taking and evaluating drugs with self-blinded experiments.
  - Interfaces with my smartwatch, a Pebble Time, and widgets custom-created with KWGT. Planning to interface it to my self-built automatic roller blinds running on an Arduino.