

|  |
| --- |
| Ian, Dennis, Joey, HarrietMay 5th, 2018 Connect4 Design Decisions  Machine Learning |
|  |

|  |  |  |
| --- | --- | --- |
| Design Decisions | | **Key Question**: How do we produce an end product that can efficiently interface with its software elements and hardware components? |
| 01What programming language do we use?02What neural network framework do we use?03What breadboard hardware do we use?04What hardware components do we use? | Programming Language One question we had to find an answer to initially was, what programming language do we use? The answer mainly came down to either using Python or Java. Python had some benefits, including more neural network libraries and machine learning documentation. However, we eventually decided to use Java because we were more familiar with programming in it, and the language still had machine learning libraries we could take advantage of. Neural Network Framework One question we were struggling with for a while was, what neural network framework do we use? Since we decided to use Java as our programming language, our main options were DeepLearning4J or Burlap. We tried to use Burlap initially, but the implementation was just too difficult. Eventually we were able to use DeepLearning4J despite little documentation because Dennis was able to get in contact with the developers and use past examples to learn from. Breadboard Another question we had starting out was what breadboard do we use to accommodate our project? Our main options were between Raspberry PI and Arduino boards. We decided to use an Arduino board because it had more USB ports and was easier to interface with Java as well as outside world hardware components.  Hardware Components  Our final major decision was what hardware components to use. It mainly came down to 3D printing vs. using wood. In the end, we decided to use wood because with what time we had, 3D printing would have taken too long and too many resources. Wood allowed us to make a working product given our timeframe. | |
| **What would we have added given more time?**  If we had more time to work on this project, we would have further adapted the neural network to make even smarter decisions based on the Connect4 input. On the hardware side, we would have used a larger breadboard (an Arduino mega). We would have also added an LCD display to interface with the user and possibly show a custom built GUI. We likely would have went with 3D printed parts as well if more time was given. We also would have further worked on our infrared sensor so the dropper knew when to stop (rather than manually hardcoding those conditions in).  Rubric Breakdown:  Use of Java API:   * Our project included several classes from the Java API aside from Collections. For example, Joey’s Arduino components had to interface with our software components, and one of the Java classes we had to take advantage of was the I/O Print writer. Our final algorithm that the neural network trained against also included linked lists. Our board class also used sets and hash sets from the Java library to help store and modify information.   Use of External API’s:   * Our project included a very large amount of external APIs and libraries. In fact, without external libraries our project would not have existed in the first place. External libraries were used to create our neural network and to interface with Joey’s Arduino hardware. Our neural network external API was DeepLearning4J, and one of the external APIs we used to successfully interface with Joey’s hardware was jSerialComm, which allowed Joey to interface with our project through a serial port. In addition, Raul Gonzalez’ Connect Four Solver (<https://github.com/raulgonzalezcz/Connect4-AI-Java>) provided a competent opponent for our model to train against.   GUI Elements:   * Our project did not use a GUI element because it was hardware based instead. However, given more time, we likely would have made a basic GUI to display on a small LCD screen for users so they could better understand how the computer was receiving input from the board. There is a console text representation for basic keyboard gameplay.   Documentation:   * Our project includes well-commented code as well as all necessary JavaDoc pages and a UML diagram which shows all related classes.   Video Presentation:   * Our project includes a video presentation that showcases both our project’s hardware and software components, including classes and the code construction for our neural network. | | |
|  | | |