GSoC ‘18 Proposal for Aimacode AIMA-java: Knowledge, Reasoning and Learning

(Link to the latest version of the Proposal)

Name and Contact Information

- **Name:** Samagra Sharma
- **Email**: samagra14@gmail.com
- **Phone Number**: +91 910957 8251
- **Github :** samagra14
- **Gitter nickname :** samagra14
- **Skype :** live:samagra14
- **Timezone :** UTC +05:30
- **Blog** (Non technical)

University and Current Enrollment

- **University**: Indian Institute of Technology Roorkee.
- **Field of Study**: Computer Science and Engineering (sophomore)
- **Expected Graduation Year:** 2020
- **Degree:** Bachelor of Technology

Meeting with mentors

- Before May 7:
  - Available anytime between 3:30 am to 7:30 pm UTC (9:00 am to 1:00 am IST) through Email/Gitter.
  - Pre Planned meetings if required, any time during weekends.

- After May 7:
  - Available most of the time except while sleeping.
  - Typical waking hours: 4:30 am to 8:30 pm UTC (10:00 am to 2:00 am IST)

Pre Gsoc Involvements With Aima-java

Links to pull requests: here

I was introduced to the field of Artificial Intelligence by an enthusiastic youtuber named Siraj Raval. After watching him, I was highly motivated to study AI and was introduced to the book “AI A Modern Approach”. I have been religiously reading the book since then. A few days later, I came across the online code repository and had started reading the codebase.
I am a well experienced Android Developer, so the choice of the language was inevitable. I started exploring aima-java. In order to get familiar to the organisational structure, I have been implementing algorithms from the book in Kotlin. After getting well-acquainted with the API structure and the organisational patterns followed by the developers, I have started contributing regularly to aima-java and plan to do so in the future as well. Here is a glimpse of a few of my pull requests :-

- #341 (merged): This PR adds Probability Model to the AIMA4e branch. This PR implements the ProbabilityModel interface with the following dependencies
  - Domain Interface
  - Proposition Interface
  - RandomVariable Interface
- #349 (merged): This pull request Implements ProbabilityDistribution interface with the following dependencies:
  - AbstractProposition class
  - AbstractTermProposition class
  - TermProposition Functional Interface
  - AssignmentProposition class
- #351 (merged): This PR adds Decision Theoretic Agent to the AIMA4e branch.
- #389 (merged): This PR was aimed at adding NQueensBoard to the set of Environments available.
- #390 (merged): This PR was aimed at modifying the existing tests for the NQueensBoard implementation so as to optimise the usage of the JUnit framework. These changes simplify the functions used for testing as well as reduce boilerplate code.
- #354 (not-merged): This PR aims at discrete probability models and discrete probability distribution. Hence, it adds the following classes:
  - FiniteProbabilityModel
  - CategoricalDistribution
- #392 (not-merged): This PR aims at:
  - Adding documentation to the nqueens environment

    - Adding relevant test cases for the environment
- #393 (not-merged): This PR adds the tests for the LabeledGraph environment for the AIMA4e branch.
- #396 (not-merged): This PR made minor modifications to the map2d package.
- #400 (not-merged): This PR makes minor changes to fix typos in method names as well as comments. This PR also removes unwanted explicit types of ArrayLists.
- #401 (not-merged): This PR adds the implementation of the FOL-FC-ASK algorithm.
- #402 (not-merged): This PR adds tests for FOL literals.

Currently (pre-GSoC), I will be working on the implementation of "**Uncertain Knowledge and Reasoning** " portion of the book. I will shift my attention towards "**Learning**" as soon as this portion gets finished.

Project Proposal and Timeline

Currently, aima-java is targeted towards the fourth edition of the book. As stated by the mentors, the project is focussed towards developing the aima4e branch and plans to make it the default branch. Currently, only initial few chapters (only up to the 7th chapter, to be precise) of the book are implemented. As a part of my GSoC proposal, I plan to implement the **Uncertain Knowledge and Reasoning** as well as **Learning** sections of the book . I also plan to introduce the relevant alternate implementations in the extras module.

Besides this, I also plan to develop the testing suite for the relevant sections of the project. Depending upon the scope of requirements, I would also be interested in working on some application demos of various algorithms that provide a clearer picture to the reader. I have experience in developing algorithm demonstrations and hence would love to contribute to the same.

Here is a list of algorithms I plan to implement as a part of my GSoC proposal along with the project timeline:

Proposed Timeline

Current - 15 Apr	● Chapter No 14 (Probabilistic Reasoning)
15 Apr - 07 May	● Hidden Markov Models ● Dynamic Bayesian Networks ● JUnit tests (This period involves my end term examinations. So, I will not be able to devote my full time.)
08 May - 14 May	● Start making APIs for the Learning section of the book. ● Decision List Learning. (My college break starts, can devote my entire time for GSoC)
08 May - 14 May	● Implement Decision Tree Learner ● Cross Validation Wrapper
15 May - 21 May	● Back-propagation (Single layer) ● Ada-Boost
22 May - 28 May	● Finish Chapter Number 18 with their tests (Chapter 19 will require certain APIs from the Logic part of the book) ● Add missing APIs from the Logic part of the book. ● Add Prior Sample and Rejection Sampling
29 May -11 Jun	● Graphplan (for both the branches) ● Hierarchic Search (for both the branches) ● Relevant APIs for testing
Evaluation No 1. (Above stated algorithms)	
11 Jun - 25 Jun	● Angelic Search (for both the branches) ● Value Iteration ● Policy Iteration
25 Jun - 09 Jul	● POMDP Value Iteration (For both the branches) ● Current Best Learning (for both the branches) ● Version Space Learning (for both the branches)
Evaluation No 2 (Above stated algorithms)	
10 Jul - 23 Jul	● Develop a small Java applet demonstrating the use of learning algorithms.

24 Jul - 06 Aug	- Consistent Determination - Alternate Implementations of a few of the above algorithms for the extras module - Completing the testing suite for the above algorithms
24 Jul - 06 Aug	- Submission of code and final evaluations.

Wishlist: Can be implemented if all the above stated Goals are completed successfully and the programme is yet to end, also desired to be taken up post-GSoC.

- Targeting Reinforcement Learning and NLP portions of the book.
- Owing to the popularity of Deep Learning, add support for creating neural nets with hidden layers and activation functions in the extras module.
- A logic-based demo agent (for the extras module) for performing symbolic Integration (Strictly Post GSoC) to demonstrate the power of simple logic. (Idea inspired from this lecture)

About Me

I am a sophomore at the **Indian Institute of Technology Roorkee**. I am majoring in Computer Science and Engineering. My areas of interest include **Math, Creative Programming, Android Development, Artificial Intelligence** and **Dramatics**. I enjoy **teaching** too. Currently, I am an active member of Mobile Development Group IIT Roorkee where I contribute to some training and software development activities. Besides this, I am also a part of the Dramatics Section at IIT Roorkee.

Ever since I was young I was awed by the mathematical nature of the universe. The simplicity with which the complex cosmic and biological events can be explained mathematically really inspires me. I am very excited and happy to have been born at a time when the secrets of intelligence are being decrypted and artificial consciousness is being aimed. My love for mathematics has inspired me to win several national science and math competitions. (KVPY, NTSE, NSEP, RMO to name a few).

I also do a little bit of creative writing. I have written a few articles on my blog. I also tried to be a youtuber once. This was my first attempt at youtubing.

My first contribution to open source was about a year ago. I have looked over to repositories of the products I use/come across, trying to give my part of contribution back to the organization whose product has been an asset to me. My contribution activity can be viewed on my Github profile.

I have realised Artificial Intelligence as my passion and Open Source as my hobby. AIMA is a perfect combination of the two. I am really excited to be a part of this organisation and to contribute to the noble work of the authors. Hope this summer proves to be my first few steps towards AI research. Here is a glimpse of a few **projects and demonstrations** I have worked on :

Anomaly detection using Machine Learning (Project Certificate)

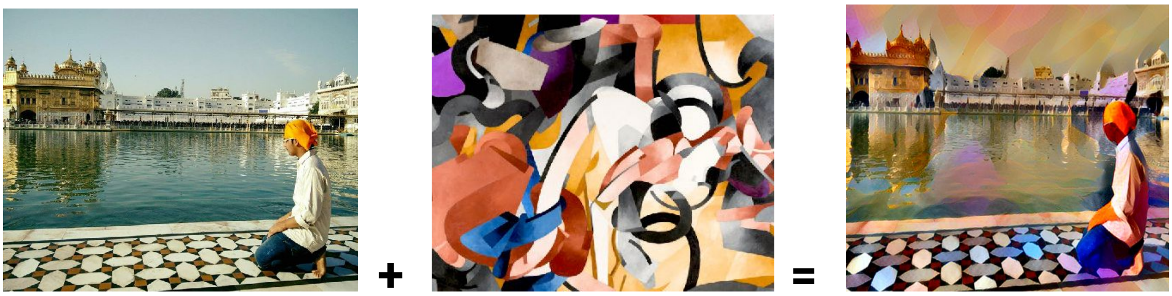
This was a research project as a part of my vocational training. Hence, I was not allowed to claim the source code. However, I have linked the project certificate. In this project, I worked on **“Anomaly Detection using Machine Learning in Industrial Environments.”**

Detailed achievements:

- Successfully demonstrated and verified that viruses and other malwares share statistical similarities;
- Applied a collection of machine learning algorithms to features extracted from around 138000 PE (Portable Executable) files with some algorithms showing around 99% accuracy

Artistic Style Transfer (Github)

This project was an attempt to reproduce the famous paper “A neural algorithm for artistic style”. This was a relatively easier and simpler implementation.



Parallel processing for an OCR (Github)

This project was built during a hackathon. In this project, we applied Android specific parallel computation techniques in order to increase the efficiency of an Optical Character Reader to read characters from a live Camera feed.

Machine Learning Projects (Github)

This is a toy repository for all of my small projects related to machine learning. Most of them are the solutions to the exercises found on the book Hands-On Machine Learning with Scikit-Learn and TensorFlow.

Cognizance (2017, 2018)

Cognizance is the tech-fest of IIT Roorkee. It is the largest tech-fest in Asia. I have been among the lead developers for the official app for Cognizance for the last two years. The app for Cognizance 2017 is open source and can be viewed here.

Calling Text (Github)

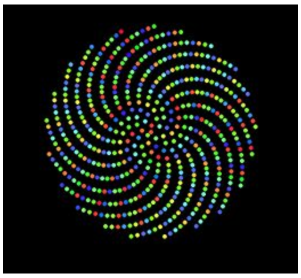
Calling Text is a smart calling assistant to enhance the calling experience of users. It helps you know the context of a call even before picking up. It is currently a work in progress.

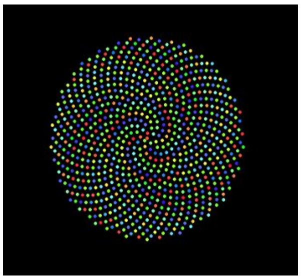
Trianglify (Github)

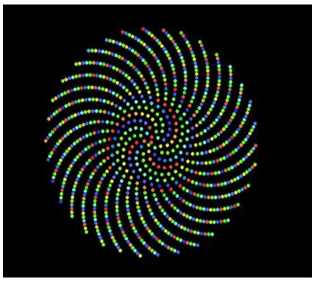
Trianglify is an Android library that helps creates views with beautiful patterns. It uses Delaunay Triangulation under the hood to generate triangular patterns. I had relatively minor contributions to this project.

Phyllotaxis (Website , Github)

In botany, phyllotaxy is the arrangement of leaves on a plant stem. Phyllotactic spirals form a distinctive class of patterns in nature. This is just a mathematical implementation of the pattern to demonstrate variations in patterns due to the variations in divergence angle.







Shush (Android)

Shush is an intelligent manager for your phone's ringer. It manages the status of your ringer based on your location and/or your schedule. Now, gone are the days when you were caught by your teacher because of your phone. This app uses geolocation and geofences to manage your ringer.

AI Stories (Website, Github):

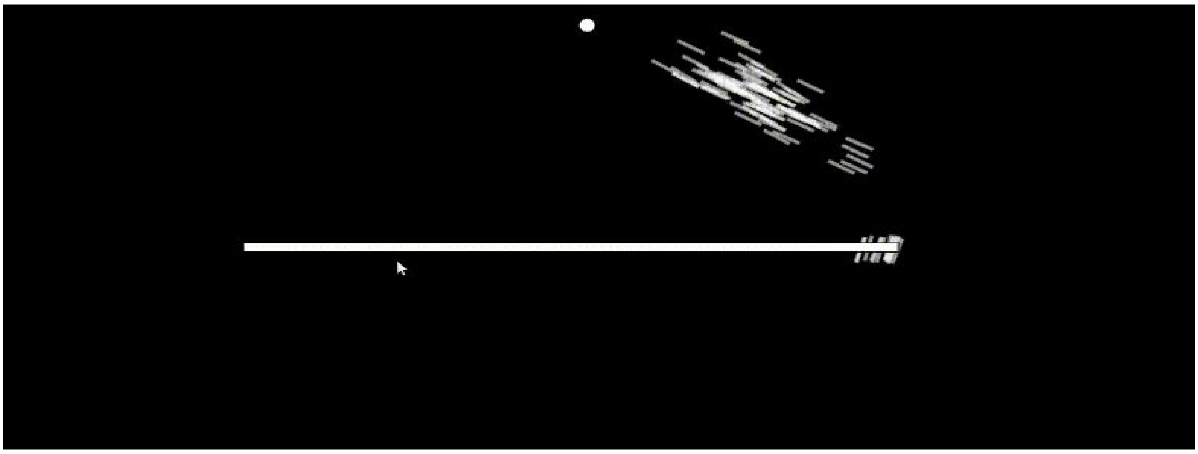
A toy story writer built to understand the basics of NLP and Markov Chains. This algorithm analyses text from a *.txt file* and then creates a new story..

Samsh (Github, Youtube)

SAMSH is a simple implementation of a shell in C. It demonstrates the basics of how a shell works. That is: **read, parse, fork, exec**, and **wait**. Since its purpose was learning (not feature completeness or even fitness for casual use), it has many limitations.

Genetic Rockets (Website, Github) :

An implementation of **genetic algorithms** to a population of spacecrafts to enhance their steering behavior. This was built for a teaching demonstration. *(Please don't mind the shape of the rockets)*



Fireworks (Website, Github):

This was an attempt to code a **simple Physics engine**. What better way to demonstrate it than fireworks!! This engine included basic forces.

Sherlocked (Github)

This was my first programming/development project. This app was a compilation of resources and information for Sherlock fans.

Teaching

I have a passion for teaching. I have been involved in teaching activities at the Mobile Development Group IITR. Here is a list of a few of the lecture notes on lectures delivered by me:

- Linux Notes
- Networking in Android
- Genetic Algorithms (Demo)

Why me for the project?

I have been studying AI for almost six months now. I am deeply attracted towards the field and have decided to pursue it in the future. I have attempted a few AI-related projects in the past and hence have had sufficient introductory knowledge so as to contribute to AIMA.

Besides this, I have been developing for Android for almost **two years** now. As a result, I am fairly acquainted with the specifics of Java. I am fairly experienced in Software Development techniques and have had sufficient coursework regarding the same. This makes me eligible to apply for the AIMA-java project as this project is a unique combination of software with math. Some of the modules can be directly imported from the project for their further use owing to their **elegant architectural design** (credits to all the previous developers).

I can easily devote **around 50 hours** per week during my timeline. I believe that the allotted work per week is completely doable for me and neither overloaded nor slacked. I make atomic commits with clean commit messages and well structured PRs. **I shalt not leaveth anything undocumented and seldom shalt I deviate from the pseudocode.**

Other Commitments

I will be having my semester examinations between **25 April - 5 May** so I will be **slightly** inactive between 11 April to 5 May for exam preparations, which is why I have kept the timeline relatively light for this period.

During vacations (7 May - 18 July) I have **no other commitments** and I seldom get tired of programming. I can easily devote more than 50 hours per week **until 18th July.** After that my institute reopens and I will have to spare some time for regular academics, hence I could then devote about **35-40 hours** a week (4-5 hours on weekdays and 10 hrs on weekends), but according to my timeline, the primary goals would be almost complete by that time, so everything is manageable throughout the timeline.