



Cloud Address: <https://tinyurl.com/JAM-Code>

Topic: Nascent Machine Learning: Guessing Game

Learning Intention:

Develop a Guessing Game that uses Artificial Intelligence to make game decisions and Machine Learning to improve the algorithm with each play.

Success Criteria:

- Use knowledge of Linked List data structures to develop a decision tree.
- Code a guessing game that uses AI to determine a correct guess.
- Code a machine learning algorithm to teach the algorithm new nouns. (Add to the decision tree.)
- Serialize code so that the decision tree data is saved and reusable.
- Challenge Extension: Create a web app using Java and the Spring Framework.

Prior Knowledge:

- Students are comfortable with OOP and working with dependent classes.
- Student have experience working with Binary Tree Data Structure

CS Big Ideas:

- Creative Development
- Collaboration
- Abstraction
- Data and Information.
- Algorithms & Programming

Resources:

- [AI/ML Slide Deck](#)
- [Guessing Game Tutorial](#) Slide Deck & Guessing Game Code
- [Serialize Slide Deck](#) (linked inside the Guessing Game Tutorial)
- [Web Dev/Spring Framework Slide Deck](#)

Agenda:

- Students learn about Machine Learning (ML) and Artificial Intelligence (AI)
 - ◆ Slides were designed for the flipped-learning model. They can be used as part of a live demonstration OR independent from the teacher. The slides contain links to outside videos and resources.
 - ◆ [AI/ML Slide Deck](#)
Please read the speaker notes.

- Demonstrate Guessing Game & Follow with the Tutorial
 - ◆ Tutorial slides were once again designed so that they could be used live or with a flipped-model. Students can pair-program or parallel-program. While it is not explicitly stated in the tutorial, students should be using these protocols as they code.
 - ◆ [Guessing Game Tutorial](#)
Please read the speaker notes.
 - ◆ [Guessing Game Repo](#)
 - ◆ A segway into serializing data is included in the Guessing Game Tutorial. A rich set of tutorials and resources about serializing java code are [linked here](#).
- Extension: Develop a Guessing Game web app using Java for backend and the Spring Framework for front end
 - ◆ Introduce & Set Up STS4
 - ◆ Create Functional Base Class
 - ◆ Create JSP Files (and learn basic HTML)
 - ◆ Develop controller class
 - ◆ Update POM.XML
 - ◆ Update Driver Class and Connect to H2
 - ◆ Finalize Controller and JSP Files
 - ◆ Students develop original app further or develop their own
 - ◆ [Spring Slide Deck](#)

Differentiation:

- AI/ML Introduction: Students are provided with a variety of resources to explore these topics. Video, websites, images. They can choose the resources.
- Guessing Game: Scaffolding is built into the Guessing Game tutorial. Students who need help can opt to follow links to video tutorials. Students who are more advanced are challenged to extend their project to serialize data, be challenged to explore neural networks or even further to turn it into a web app.

Random Musings:

- The game is only as good as the people training it. It could be violent, abusive, racist or pornographic if the people training it have those traits. Consider the bias. For example, our beta testers were teenage boys. They developed a snarky database of questions that referenced family members. (Of course there were farts involved.) Your students may go there. Proceed with caution.
- You can end up with a long, single branch, decision tree. Writing good questions is key to a more effective guessing mechanism. It is also possible that a noun might live in more than one sub-tree. The runtime on a poorly constructed tree is likely to be $O(n)$.