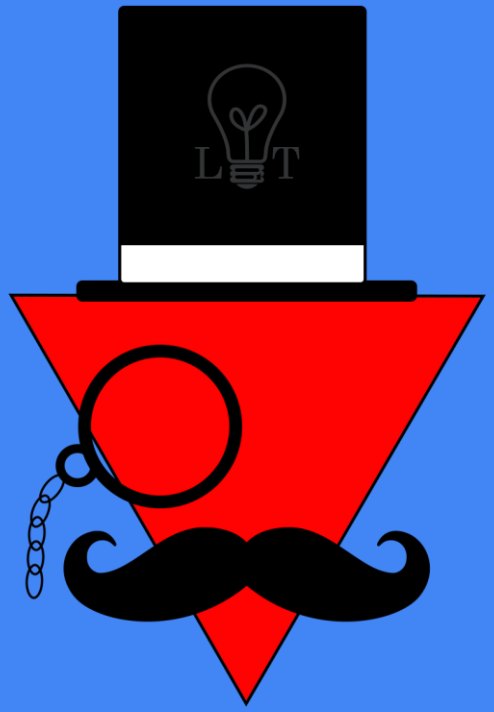
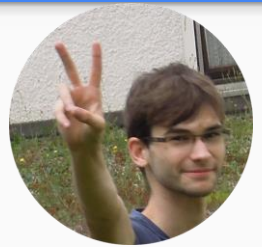
Welcome

Presentation of The Learning Triangle

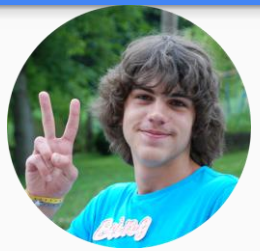


Team

Steven Kovacs



Marco Müller



Project Manager	Implementer	Graphic Artist
Tool Specialist	Designer	Business Designer
Configuration Manager	Test Designer / Tester	Requirements Specifier

We are using the Rational Unified Process for our roles

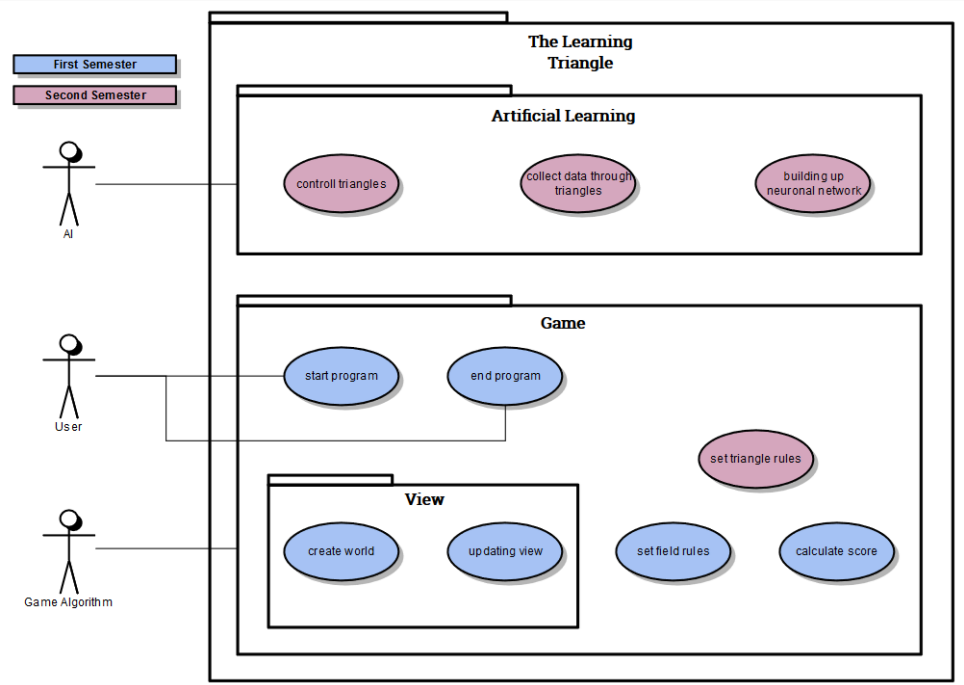
Our vision

- create a game with no user-interaction at all
- look how an artificial intelligence lives in an artificial world with clearly defined rules
- just an algorithm, no player

Purpose of this project

- non-profit
- research in an upcoming area of computer science
- do something we had never done before

Use Cases & Scope



- project parts splitted into smaller parts → Use Cases

First Sem. : game rules

Second Sem. : Neuronal Network

Project Management

How to document the specifications and Use Cases?

- *Software Architecture Document* and *Software Requirements Specification*
- each Use Case <-> Use Case Document

The way we develop

- We followed the idea of an *agile* project

→ you have to welcome a change while you are working on your project

- more contact with customer to follow his wishes

→ don’t develop over some months just to realize at the end that the customer wanted a program completely different

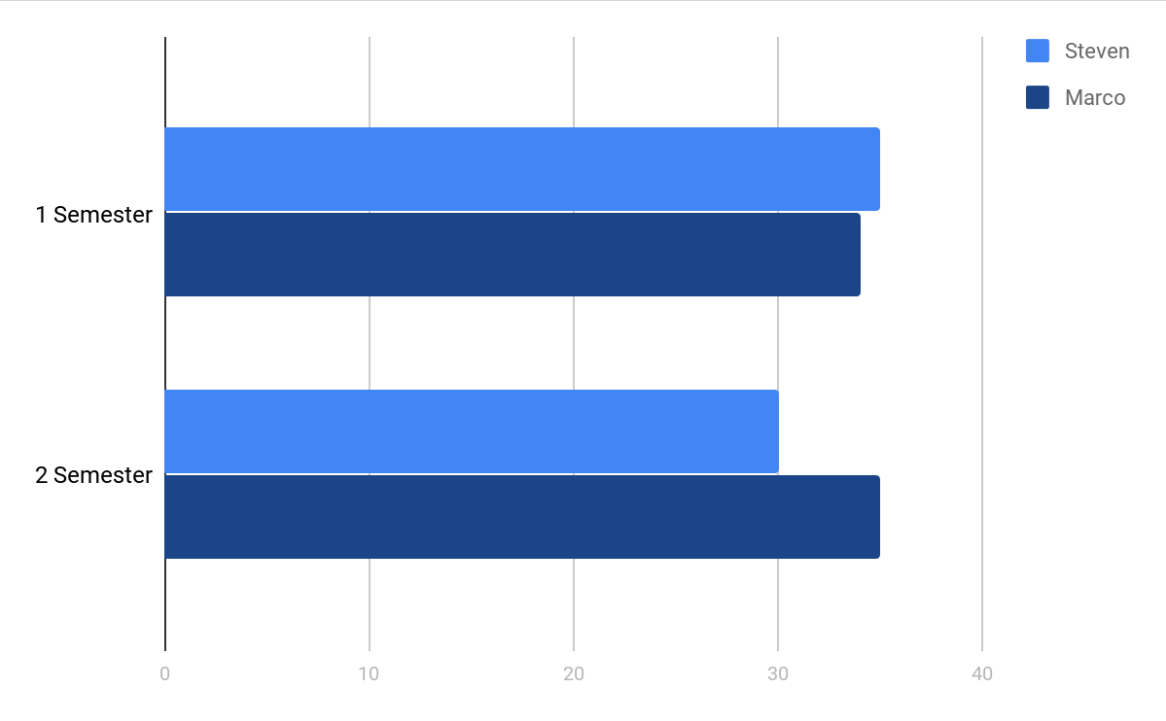
Scrumming

- realising an agile project needs a good overview over state of the project and remaining time → we use JIRA

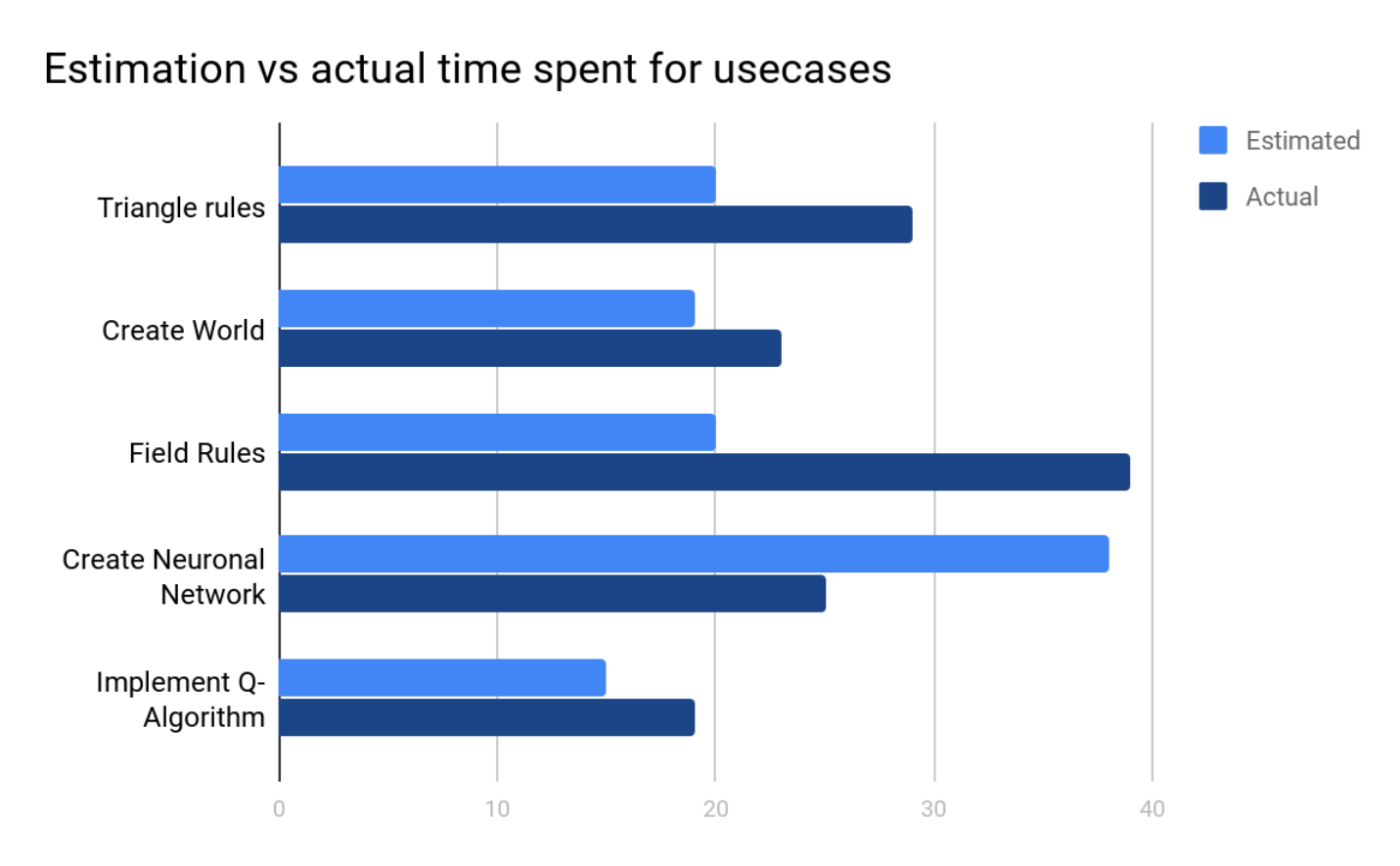
JIRA is a

- Platform to create and edit tasks
- Associate tasks with people
- Log worktime
- Estimate operating expenses

Time spend

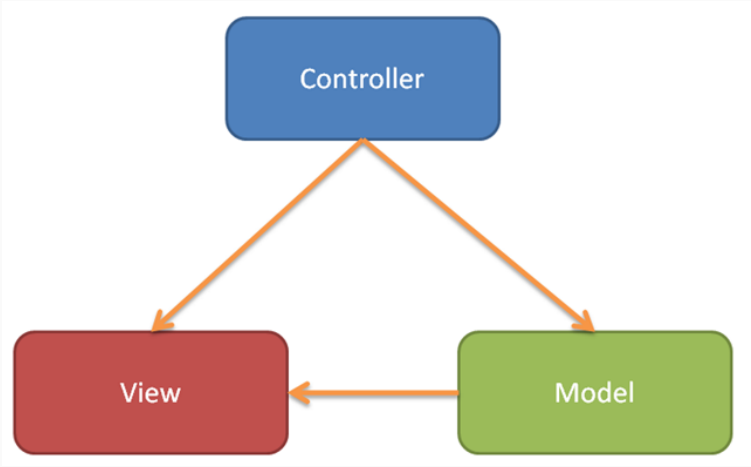


Estimation vs real time spent



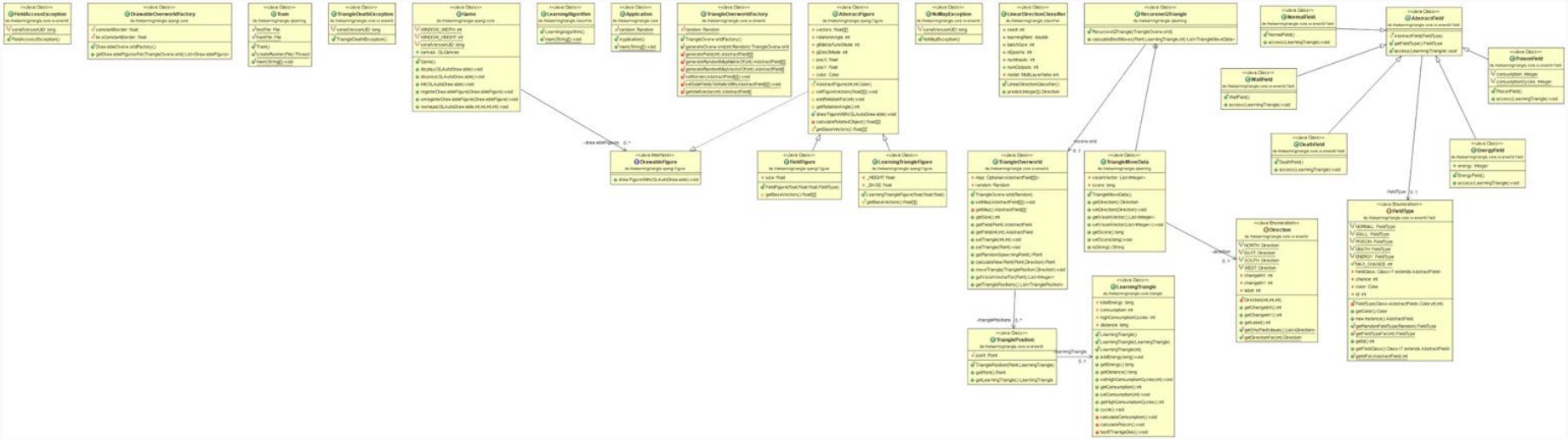
Patterns

- we use the MVC pattern in our project



Model	contains the data
Controller	contains the algorithms
View for showing the UI	contains the methods

Class Diagram



Application testing

- since we barely have any overview we decided to just use JUnit for unit-testing
- We can run all tests within one to two clicks

Metrics

Eclipse-Addon Metrics 1.3.6

→ rates the results

→ shows you the critical points of your project

Metric	Total	Mean	Std. Dev.	Maxim...	Resource causing Maximum	Method
Number of Parameters (avg/max per method)		0,689	0,903	5	/TheLearningTriangle/src/de/thelearningtriangle/ope...	reshape
Number of Static Attributes (avg/max per type)	12	0,429	0,776	3	/TheLearningTriangle/src/de/thelearningtriangle/ope...	
Efferent Coupling (avg/max per packageFragment)		2,4	1,562	6	/TheLearningTriangle/src/de/thelearningtriangle/cor...	
Specialization Index (avg/max per type)		0,005	0,027	0,143	/TheLearningTriangle/src/de/thelearningtriangle/qlea...	
Number of Classes (avg/max per packageFragment)	28	2,8	1,778	6	/TheLearningTriangle/src/de/thelearningtriangle/cor...	
Number of Attributes (avg/max per type)	37	1,321	1,928	7	/TheLearningTriangle/src/de/thelearningtriangle/clas...	
Abstractness (avg/max per packageFragment)		0,064	0,151	0,5	/TheLearningTriangle/src/de/thelearningtriangle/ope...	
Normalized Distance (avg/max per packageFragment)		0,301	0,327	0,929	/TheLearningTriangle/src/de/thelearningtriangle/cor...	
Number of Static Methods (avg/max per type)	11	0,393	1,345	7	/TheLearningTriangle/src/de/thelearningtriangle/cor...	
Number of Interfaces (avg/max per packageFragment)	1	0,1	0,3	1	/TheLearningTriangle/src/de/thelearningtriangle/ope...	
Total Lines of Code	1194					
Weighted methods per Class (avg/max per type)	134	4,786	4,64	20	/TheLearningTriangle/src/de/thelearningtriangle/cor...	
Number of Methods (avg/max per type)	92	3,286	3,644	14	/TheLearningTriangle/src/de/thelearningtriangle/cor...	
Depth of Inheritance Tree (avg/max per type)		1,643	1,076	6	/TheLearningTriangle/src/de/thelearningtriangle/ope...	
Number of Packages	10					
Instability (avg/max per packageFragment)		0,649	0,326	1	/TheLearningTriangle/src/de/thelearningtriangle/core	
McCabe Cyclomatic Complexity (avg/max per method)		1,301	0,761	5	/TheLearningTriangle/src/de/thelearningtriangle/qlea...	createRunner
Nested Block Depth (avg/max per method)		1,262	0,775	6	/TheLearningTriangle/src/de/thelearningtriangle/qlea...	createRunner
Lack of Cohesion of Methods (avg/max per type)		0,166	0,291	0,771	/TheLearningTriangle/src/de/thelearningtriangle/ope...	
Method Lines of Code (avg/max per method)	533	5,175	8,703	56	/TheLearningTriangle/src/de/thelearningtriangle/clas...	LinearDirectionClassifier
Number of Overridden Methods (avg/max per type)	1	0,036	0,186	1	/TheLearningTriangle/src/de/thelearningtriangle/qlea...	
Afferent Coupling (avg/max per packageFragment)		3,5	4,5	13	/TheLearningTriangle/src/de/thelearningtriangle/cor...	
Number of Children (avg/max per type)	7	0,25	0,987	5	/TheLearningTriangle/src/de/thelearningtriangle/cor...	

Risk Management

identified	prob of occurrence	damage	mitigation strategy	person in charge	risk factor
neuronal network doesnt behave as expected	25,00%	100,00%	include more machine learning frameworks for usage but tensorflow	Steven & Marco	25%
not enough time to get engaged with machine learning	10,00%	100,00%	start early enough with reading documentations	Steven & Marco	10%
TDD takes to much time and work	15,00%	30,00%	priotize working solutions over tested but unworking code	Marco	5%
untested funcionality doesnt work on first try	50,00%	5,00%	recode funtionality	Steven & Marco	3%
project architecture becomes unclear	10,00%	15,00%	refactoring, could influence deadlines	Marco	2%
private computer/server isn't strong enough to train the network	5,00%	15,00%	use computer from DHBW	Steven	1%

Life cycle management

- We don’t use life cycle management because
  - We don’t have a product to use, just to research on
  - We don’t have a web-platform to deploy our project on, it’s mostly a console application

The technology we are using

- OpenGL (JOGL-Framework)
- JUnit-Framework
- Deeplearning4J

- GitHub
- Jira

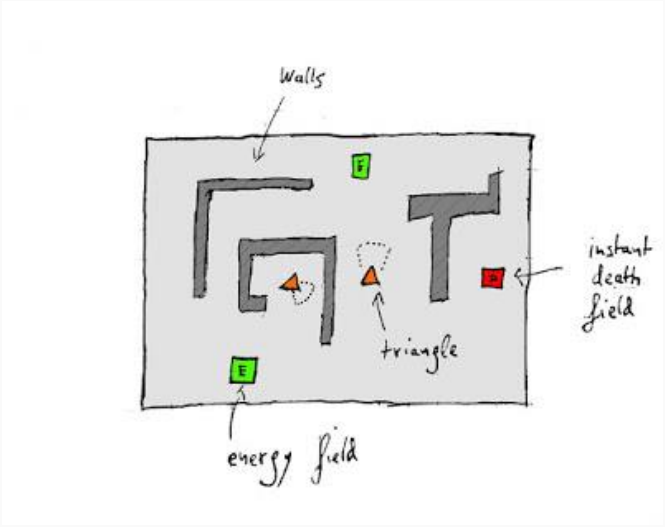
What does machine learning mean?

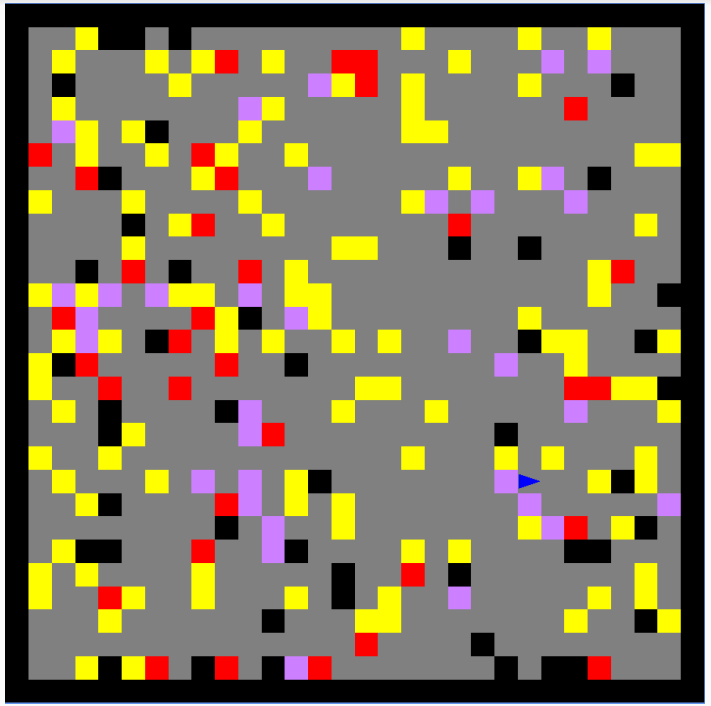
- abstract and classify situations
- learn specific tasks
- automate complex behavior

What is The Learning Triangle?

- a neuronal network
- approach brute force problem solving
- artificial creature living in an artificial environment

Concept to implementation





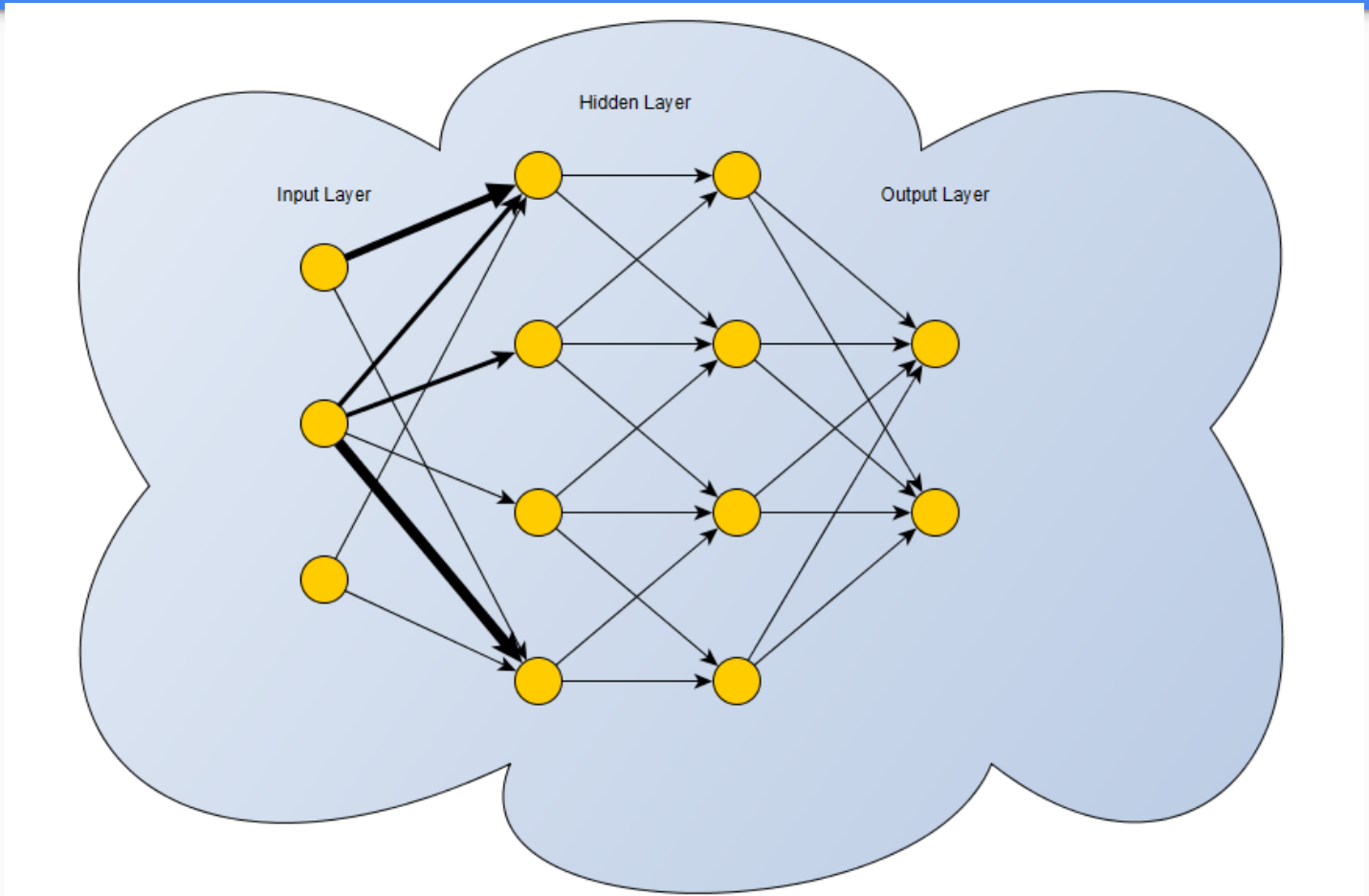
How do we learn?

- We implemented an Q-Algorithm
  - Reward function
  - Very expensive calculation
  - O(4^n)
- Write the first 5 results in a csv-file

How does such a csv-file look like?

- One line is one set of data
- Lines are separated by “\n”
- Fields are separated by “,”
- First field is the “label” (the route the q-algorithm would take)
- Other 36 fields are the vision

How does our neuronal network looks like?



Demo of our project