The Learning Triangle

Software Architecture Document

Version 1.2

Date	Version	Description	Author
24.11.2016	1.0	First set up	LearningTriangleTeam
12.12.2016	1.1	Added new Class Diagram	LearningTriangleTeam
01.07.2017	1.2	Added architecture description	LearningTriangleTeam

Introduction

Purpose

This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

Scope

This document will describe, how the world around with the artificial creatures behave. The architecture will show, how this world is build up.

Definitions, Acronyms and Abbreviations

None.

References

None

Overview

This document provides information about the architecture of the software “TheLearningTriangle”.

Architectural Representation

Architectural Goals and Constraints

The goal of the architecture is to provide information about how the classes and interfaces of the program is working together.
We want to use the MVC pattern, but we don’t use any MVC pattern libraries because for our purposes we need to implement own ones.
For example, Models will have some logic, other models need to have view implementations.

Additionally we are using a the factory pattern to create fields and the overworld. This is implemented by, for exampe, the enumeration “Fieldtype.java” in our project. It can provide an AbstractField instance of any given Fieldtype by passing it the wanted value.
The factory pattern is used, when objects of the same interfaces or classes are needed on certain points in the program in condition of parameters. For example, if a parameter is a certain value we want to use another implementation of a certain interface. To avoid ugly switch-case statements we can use the factory pattern to clarify the purpose of the function itself.

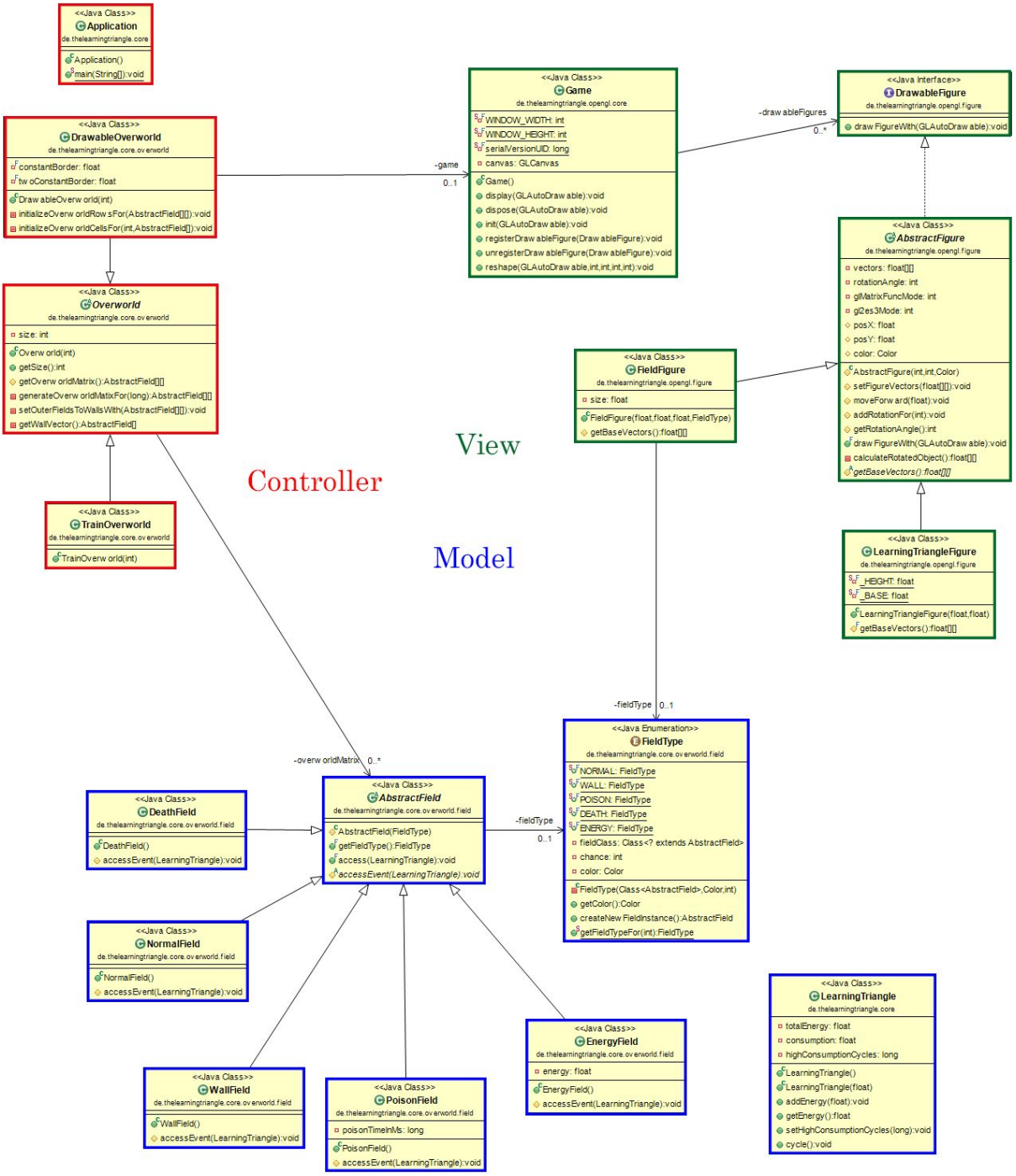
Use-Case View

1. Simulate View
   The simulate view will just provide functionality to train the neuronal network of our triangles. It does not have any kind of view, because it is performance killing.
2. Drawable View
   The drawable view will provide a nice look on how the triangles are moving around.

Use-Case Realizations

Logical View

Overview



Architecturally Significant Design Packages

Process View

Deployment View

Implementation View

Overview

Layers

Data View

There are saved versions of our triangles brain, but we can’t describe how it is ordered or saved. We just know that the files represent the binary-files of the neuronal network in a specific version.

Size and Performance

N.A.

Quality

N.A.