Project Proposal

Project name	*Online Platform for Conducting AI Competitions*
Abbreviation	*OPCAIC*
Supervisor	*Jakub Gemrot*
Consultants	
Annotation	*An online platform for conducting game AI tournaments consisting of user frontend, administrator frontend and tournaments runner backend. As a proof of concept, the result will feature integration of a few existing AI games.*

Motivation

A few lectures of our university program are using tournaments as edu-candies for spicing up the labs, e.g., competitions organized during Artificial Intelligence 1, Multi-agent Systems or Human-like Artificial Agents. However, running tournaments by hand is time consuming as well as developing new semi-automated solutions for them. This project aims to provide a standardized environment for conducting and judging of these tournaments. On a broader scale, such a software can be employed in context of AI competitions held at various academic conferences (such as CIG, AIIDE or IJCAI). Our primary competitor is http://theaigames.com/ webpage, whose solution is proprietary and therefore it cannot be adapted for our purposes.

Project description

The aim of the project is to implement a platform for conducting game AI tournaments.

The platform will support two types of use cases - ones for regular users and another for administrators. Regular users will be able to participate in tournaments by submitting a program that tries to solve a given problems. Administrators will be able to create, run and manage tournaments.

In particular, regular user will be able to:

- list past and opened (ongoing and/or future) tournaments;
- register;
- join open tournaments;
- submit their solutions in programming languages given tournament supports;
- see the results of their solutions, results of others (depending on a tournament configuration) and an ability to export the results;
- download a replay of all the games that the user participated in;
- manage their profile screen.

Administrators will be able to:

- create and configure tournaments;
- setup a webpage for tournaments, including descriptions, rules, results, et cetera;
- install a competition-runner program, in a form of a black box, that will simulate the game, evaluate submitted solutions and output results;
- manage and run a tournament;
- see and publish the results of a tournament.

Our goal is to make the tournaments highly customizable. The platform should provide support for all the tournament types that are frequently used in computer games tournaments.

Examples of tournament features:

- whether submissions will be evaluated right after they are submitted or only after the deadline of the tournament;
- mode of the tournament - round robin, single elimination, double elimination, table, ELO;
- support for the different number of players per tournament game;
- support solutions in multiple programming languages given the capabilities of tournament runner.

To ensure that the platform is able to evaluate a large number of submissions, it will support distributing workloads across multiple computing resources. The platform must also provide sandboxing of the user code in order to prevent the programs from cheating (creating connections to the internet, etc.) or damaging the server environment.

The platform will contain a few example tournaments, which will be drawn from the pool of competitions already existing for the Artificial Intelligence I labs (e.g., Super-mario, Pac-Man, Warlight, Minesweeper and others).

Platform, technologies

Target platform: Linux/Windows
Frontend: React, Typescript
Backend: ASP .NET Core, Entity Framework Core, some database (e.g. PostgreSQL)
Other: Git, GitHub
Methodology: initial software project decomposition will be done in a waterfall style, actual development will be agile-based using Scrum

Team

The team consists of 4 MFF UK students:

- Ondřej Nepožitek <ondra@nepozitek.cz>
- Michal Lehončák <m.lehoncak.12@gmail.com>

- Šimon Stachura <ghort@seznam.cz>
- Radek Zikmund <r.zikmund.rz@gmail.com>

Time estimation / Roadmap

Months (0-based) mean project start date deltas.

- Detailed specification - 0.-2. month
- Frontend - 1.-7. month
- Backend - 1.-7. month
- Tournament runner - 1.-7. month
- Proof of concept tournaments - 6.-8. month
- Polishing - 7.-8. month
- Documentation - 8. month

Project Scope

Discrete models and algorithms	
	discrete mathematics and algorithms
	geometry and mathematics structures in computer science
	optimizations
Theoretical computer science	
	theoretical computer science
Software and data engineering	
x	software engineering
x	software development
x	web engineering
x	database systems
	analysis and processing of large data sets
Software systems	
x	system programming
	reliable systems
x	performance systems
Mathematical linguistic	
	computer and formal linguistic
	statistical methods and machine learning in computer linguistic

Artificial intelligence	
x	intelligent agents
	machine learning
	robotics
Computer graphics and computer games development	
	computer graphics
x	computer games development