Artificial Intelligence 4 Games

Short Introduction to CodinGame

2020

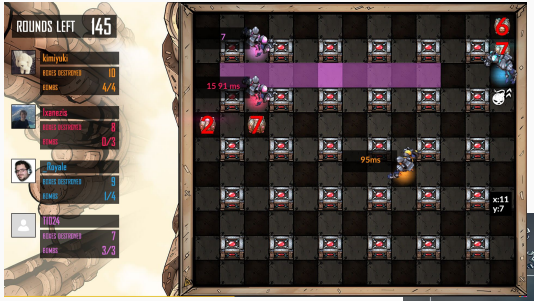
Jakub Kowalski, University of Wrocław

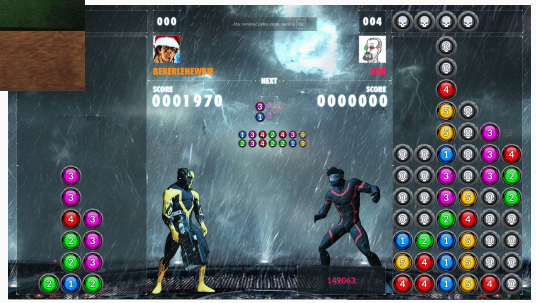
CodinGame

CodinGame (CG) is an online platform for competitive/selflearn programming, focused mostly on AI approaches.

It contains a number of problems to solve, and number of programming games for which you have to write an agent program to play with other users' programs.

All this, with really nice visual interface, online programming IDE, supporting ~30 programming languages, friendly community, and it's still developing.





Role of CodinGame in the Lecture

What

Every lab exercise will require (at least partially) writing some code to CG-based task.

In particular we will participate in CG contest(s).

Optionally, some of the project propositions will be CG-related.

Why

Learning to have fun to learn.

I want you to code some more advanced tasks “in real environment” and existing a context of worldwide community.

Thanks to the community contributions, CG allows me to create course-specific tasks and ease the task of running and comparing your solutions.

And it’s visually pleasant :-).

Main Activities

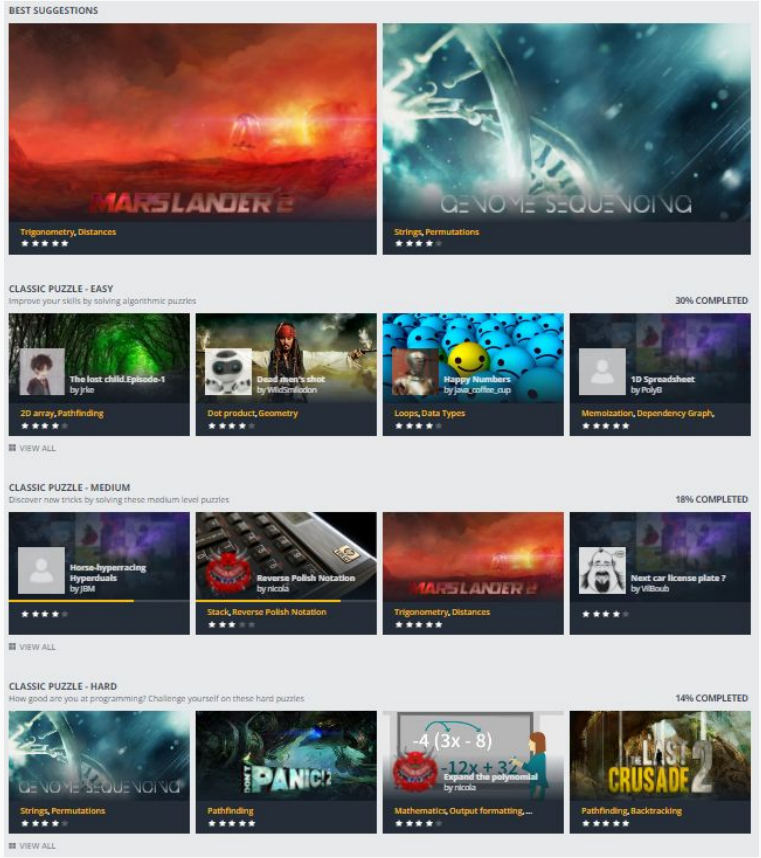
- ★ Puzzles
- ★ Bot Programming
- ★ Optimization Games
- ★ Contests
- ★ Clash of Code
- ★ Code Golf
- ★ User Contributions

Puzzles

Codingame has a large of problems to solve, splitted by the difficulty (easy, medium, hard, very hard). Each puzzle has a number of visible and hidden (so called validators) tests that your program has to pass.

Some puzzles have form of simple input-output tasks with no graphical interface, and one valid answer. Usually these are user puzzles (without the author avatar visible).

Other (usually made by CodinGame team) are more game-like, requiring the program to iteratively send responses, with multiple ways to pass, and interesting graphic.

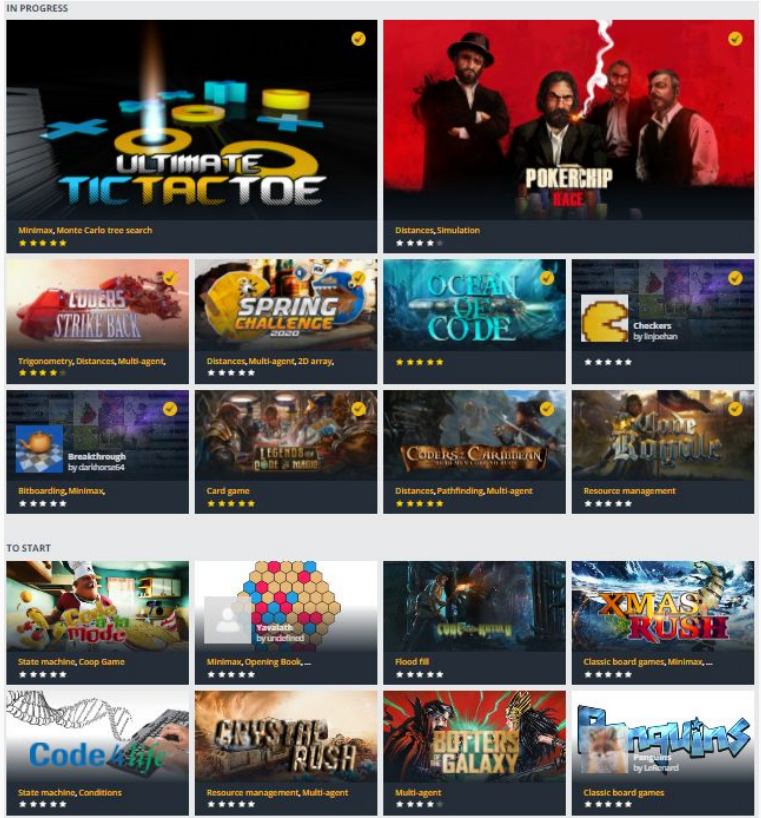


Bot Programming

The main multiplayer activity. Your task is to write a program that plays a given game against the programs of other users.

The games work in a *league system*, that is the agents are divided by the chosen agents called *bosses*. If the player's agent is better than the poss it advances to the next league.

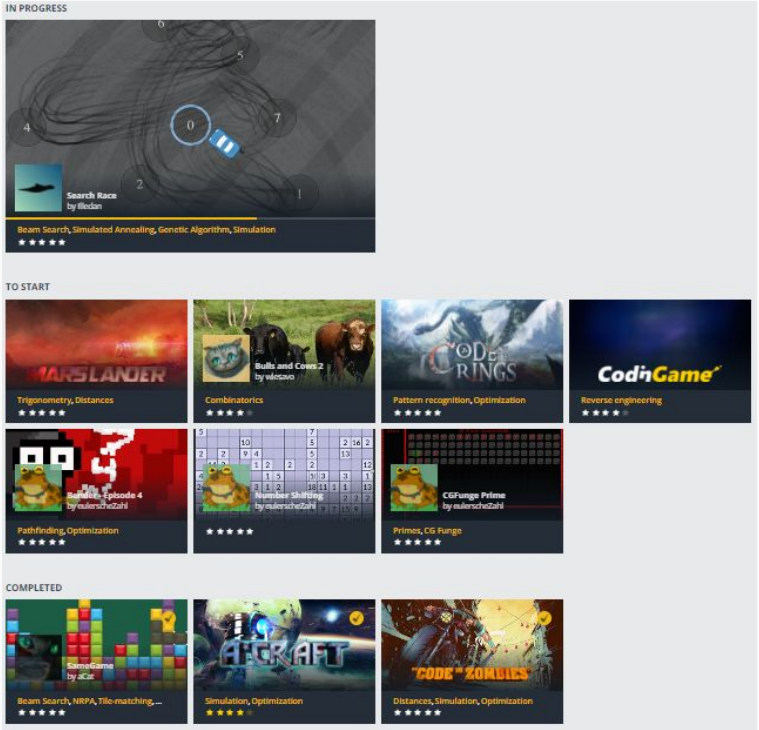
The highest league is *Legend*, then *Gold*, *Silver*, and *Bronze*. Below, there is a number of *Wood* leagues, with simplified rules, with a goal to teach the game.



Optimization Games

These are single player games, that work somewhat like puzzles (they have test cases that can be passed and failed), but the goal is to maximize/minimize some kind of score.

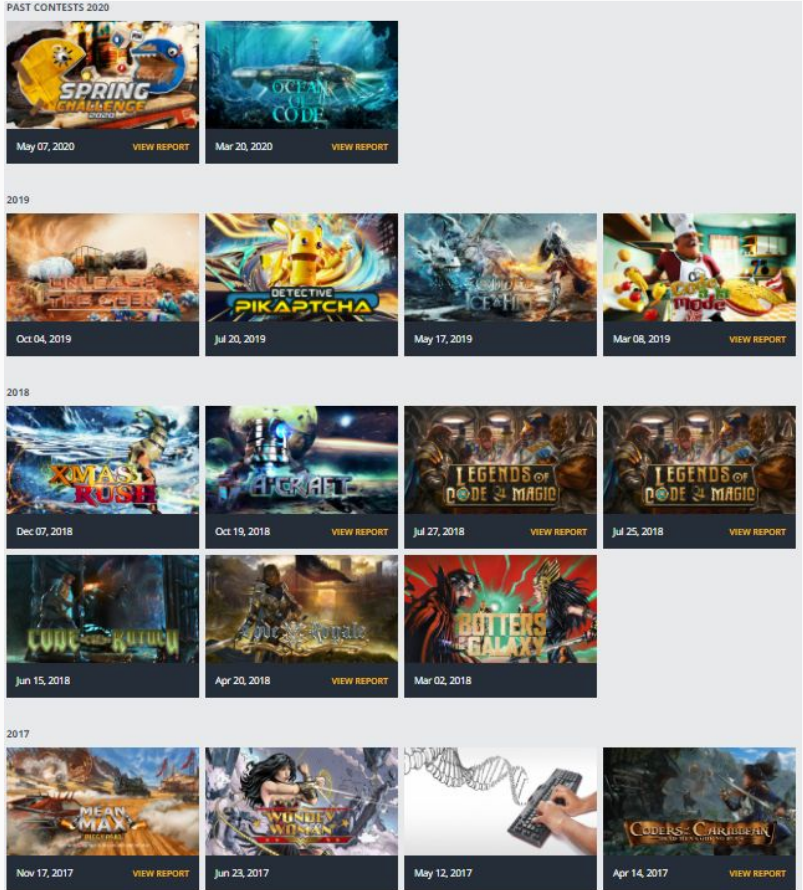
Thus, the program is not evaluated by directly playing against other programs, but its strength is depends on the achieved score.



Contests

Few-times-per-year, limited-time events that gather lot of attention. A new, previously unknown game (usually multiplayer, more rarely optimization) is released and everyone tries to write the best program in a given time. Leagues are opened gradually.

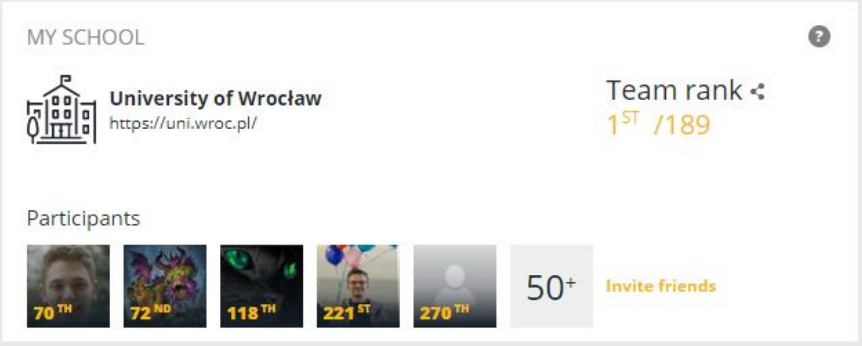
Contests are a very fun experience, and an occasion to gather community together. Although they are competitive by nature, there is usually a lot of knowledge sharing and helping other people, even programming streaming.

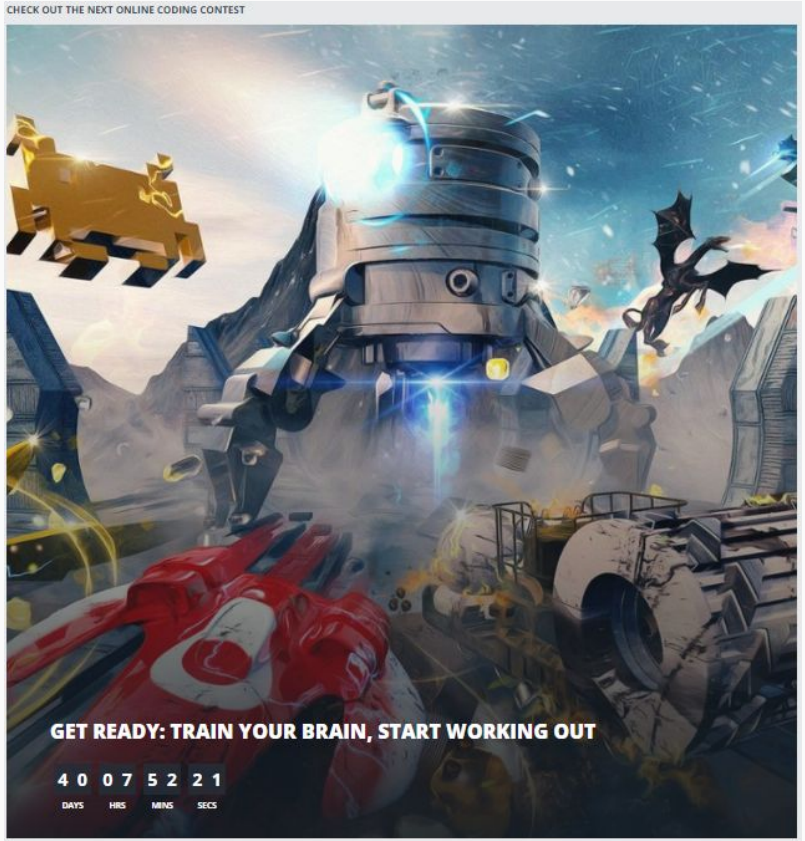


Contests: Fall Challenge 2020

Recently, the new leaderboard category for the contests was introduced: top schools. As you might have heard, we had quite a nice results in this category so far :-).

Writing an agent for next contest will be one of the exercises for the course. The competition will take place 12.11 - 23.11.





Clash of Code

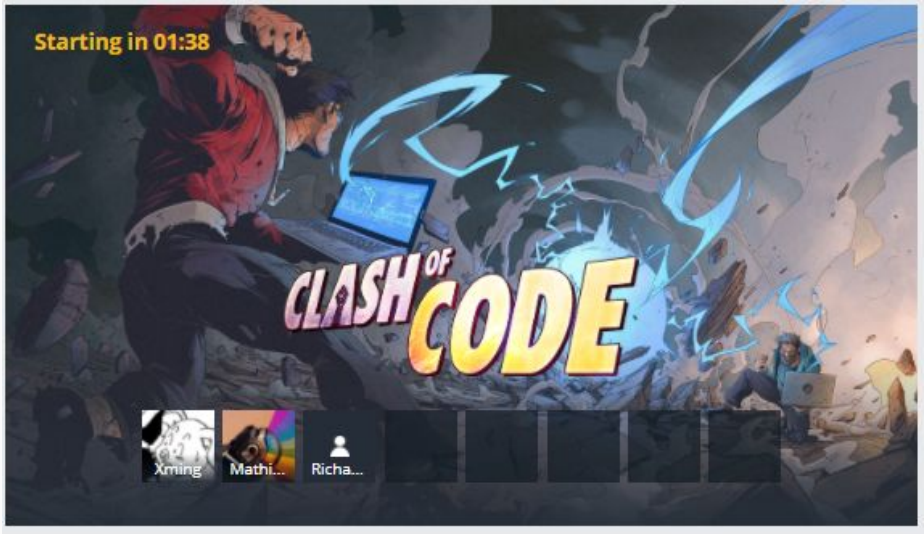
Online competition in which you have to solve small programming problems in a short time “better” than other participants.

There are three types of tasks:

- ➔ Fastest
- ➔ Shortest source code (and fastest)
- ➔ Reverse (no statement, fastest)

Some given problems are interesting, some are weird, but still it’s a great activity for learning useful language structures.

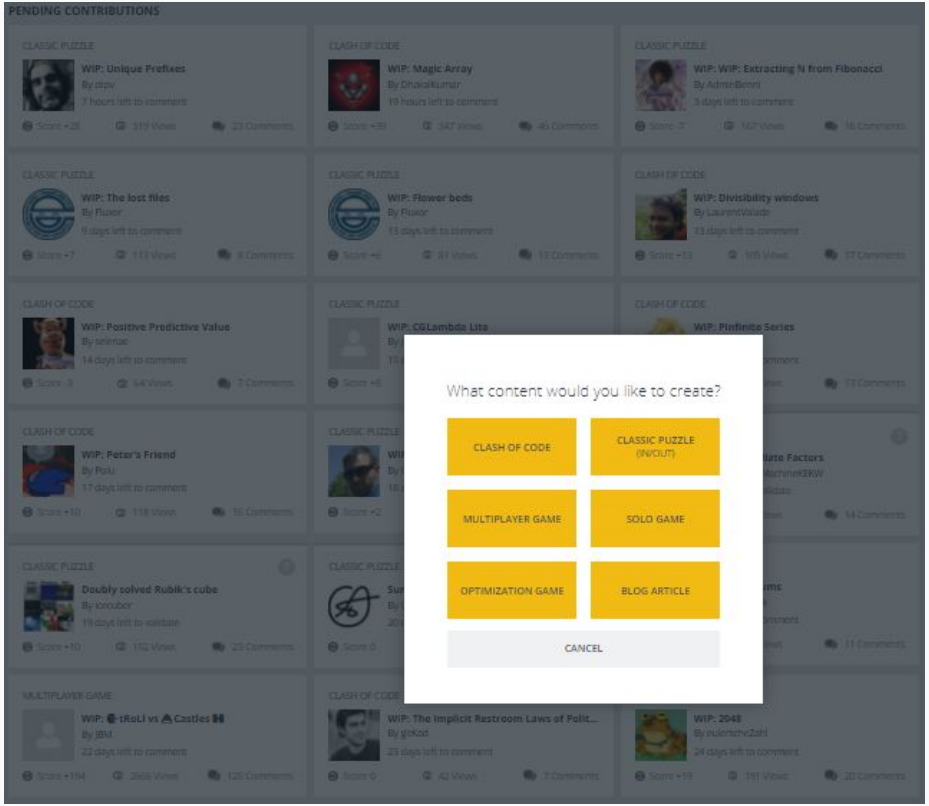
Also, this is fast and small activity so it’s definitely the least “demanding” one.



User Contributions

Last but not least, it is possible to make your own tasks. Not only simple in-out puzzle, but also full-graphic interactive puzzles/optims/multis with CodinGame SDK.

Some of this lecture tasks will be based on such contributions. Also, creating such games devoted for AI-playing is a task of the planned for the next semester project *Programming Programming Games*.



Basic Info

- ★ Community
- ★ Coding IDE
- ★ User profile
- ★ Expert Mode
- ★ CG Sync

Community

CG has very friendly and helpful community, including a group of very devoted top-tier players that are around nearly any-time and you can ask them questions.

Also, what's very important, the main point of CG is to learn. So although there are competitions, and no one will send you his source code, people share their knowledge, approaches, exchange ideas, etc.

Communication channels:

Official CG Discord

Also, CG-related info can be found (and shared) on #codingame channel on my teaching discord.

Last but not least, I definitely encourage you to join the Polish CG community, run by Zylo and me.

Here is the invite link

User Profile and Gamification

XP

Given for almost every activity: solving puzzles (more for original CG ones), getting achievements, advancing to next leagues, contributions, various upvotes, etc. You can never lose xp, even when downvoted.

Used to gain **CG levels**, which gives additional privileges, e.g., ability to review contributions, modifying them.

Achievements

Yay, achievements, tons of achievements...

CP

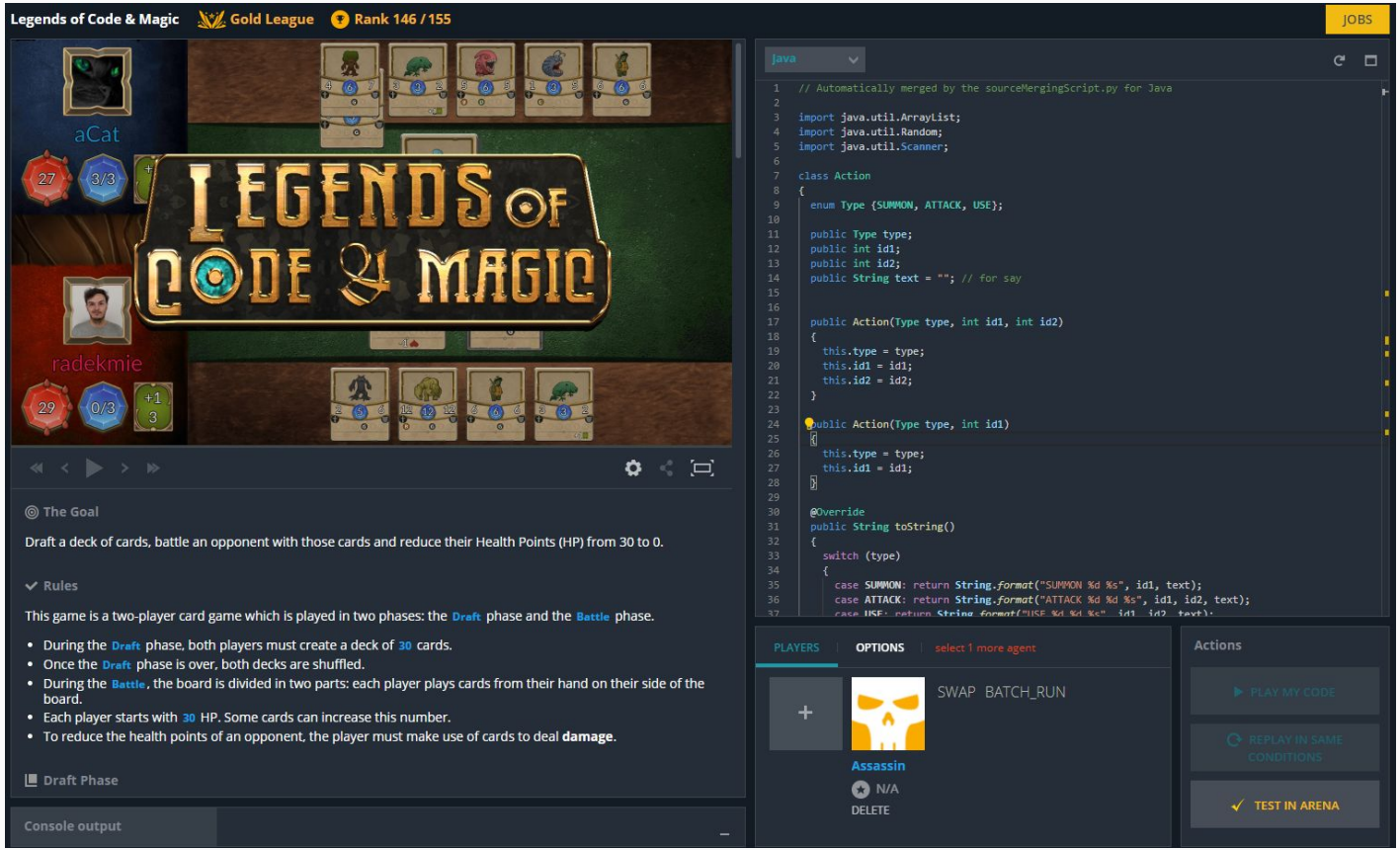
Given for competitive activities, summary of the scores/ranks given in individual games comparing to other players. Changes dynamically, updated once per day.

Used to compute **CG rank**.

Quest Map

New feature, guiding the process of learning the platform. Contains suggestions “what to do next”.

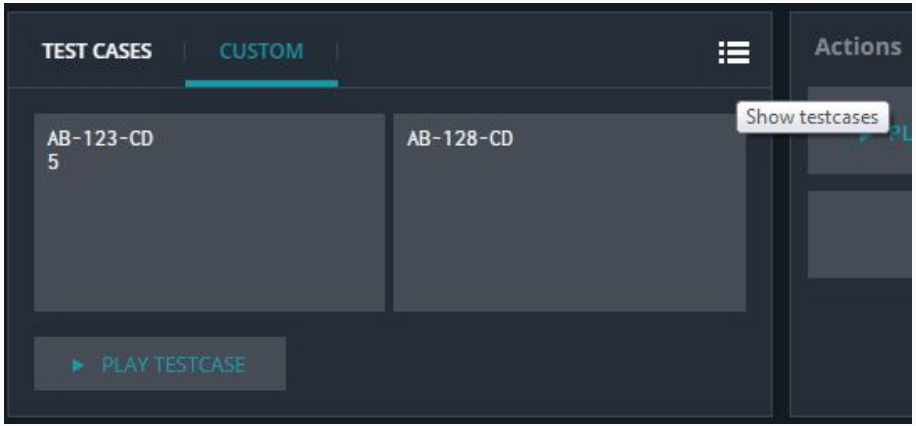
Coding IDE

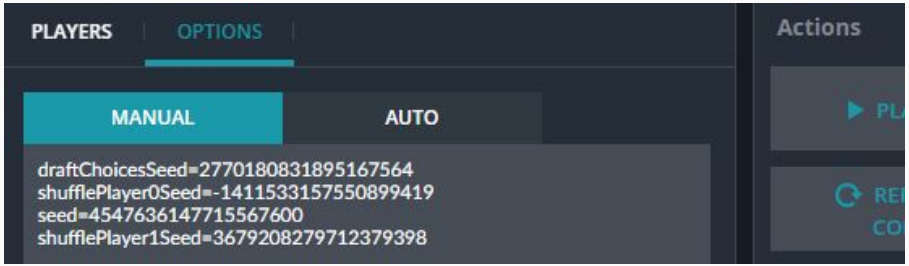

Legends of Code & Magic
Gold League
Rank 146 / 155
JOBS
aCat
27
3/3
LEGENDS OF CODE & MAGIC
radekmie
29
0/3
+1
3
The Goal
Draft a deck of cards, battle an opponent with those cards and reduce their Health Points (HP) from 30 to 0.
Rules
This game is a two-player card game which is played in two phases: the Draft phase and the Battle phase.
During the Draft phase, both players must create a deck of 30 cards.
Once the Draft phase is over, both decks are shuffled.
During the Battle, the board is divided in two parts: each player plays cards from their hand on their side of the board.
Each player starts with 30 HP. Some cards can increase this number.
To reduce the health points of an opponent, the player must make use of cards to deal damage.
Draft Phase
Console output
Java
// Automatically merged by the sourceMergingScript.py for Java
import java.util.ArrayList;
import java.util.Random;
import java.util.Scanner;
class Action
{
enum Type {SUMMON, ATTACK, USE};
public Type type;
public int id1;
public int id2;
public String text = ""; // for say
public Action(Type type, int id1, int id2)
{
this.type = type;
this.id1 = id1;
this.id2 = id2;
}
public Action(Type type, int id1)
{
this.type = type;
this.id1 = id1;
}
@Override
public String toString()
{
switch (type)
{
case SUMMON: return String.format("SUMMON %d %s", id1, text);
case ATTACK: return String.format("ATTACK %d %d %s", id1, id2, text);
PLAYERS
OPTIONS
select 1 more agent
SWAP BATCH_RUN
Assassin
N/A
DELETE
Actions
PLAY MY CODE
REPLAY IN SAME CONDITIONS
TEST IN ARENA

Expert Mode

Turning this option is very handy for solving some puzzles (where you can see inputs and outputs of all test cases, and try the program on your own input), and multiplayer games (where you are given control of the game parameters, random seeds, etc.).

It does not apply to all puzzles and games, so the option is **not always** available from the settings menu (it should be accessible for e.g. community puzzles).





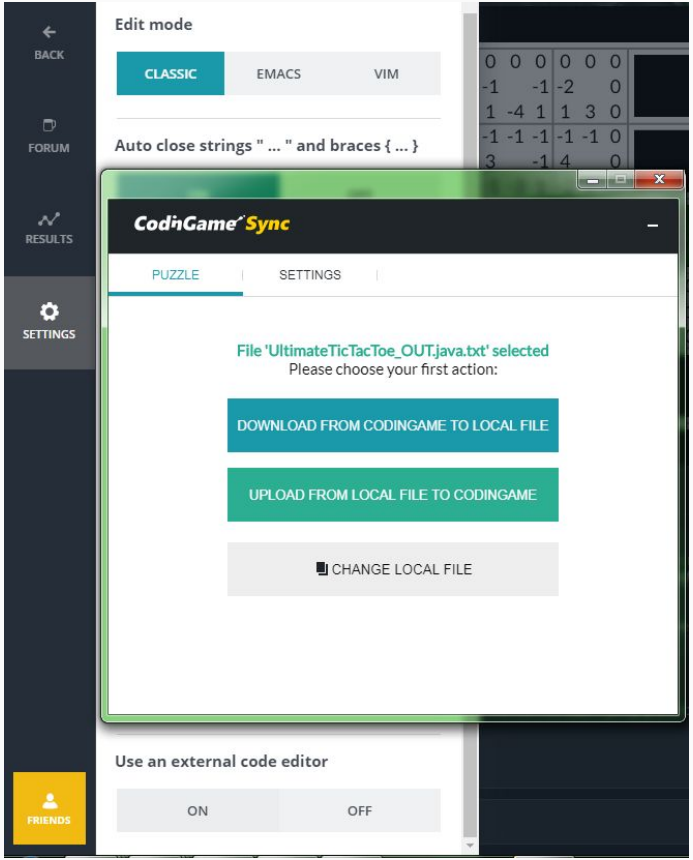
CG Sync

This is a great tool that allows you to synchronize a file on a hard drive with the content of the CG IDE window.

It comes handy if you develop your programs offline and want to test them online without too much of bothersome copypasting.

In my opinion, a must have addon.

Link to description and download



Advanced Tips

- ★ Mergers
- ★ CG Benchmark
- ★ Brutaltester
- ★ Command Line Referees
- ★ CG Tools

Mergers

CG requires you to create an entire program in one file, which, of course, is very problematic in case of large files with complicated code.

The solution then is to use some merging scripts that join multiple source files. So that you can normally code offline in your favourite IDE and keep common-sense file structure and the resulting file will be CG-ready (and can be automatically synchronized with CG sync).

Usually everyone writes such script for himself, as they are often tailored for a specific use, but if anyone's interested I can share the scripts I made for:

- ★ C#
- ★ Java
- ★ Lua

CG Benchmark

CGBenchmark is a tool that allows to run your local code against the other users' submitted agents in a batch and returns links to the game replays.

Thus, it allows you to extensively test your agent strength on a selected set of opponents and compare different versions of your program.

The latest build is available here.

If you are loging to your CG account with other means than email/password, you may retrieve your password simply by using *forgot your password* option.

To work properly, CGBenchmark requires properly configured config file. In particular, for UTTT you have to set

```
multiName: tic-tac-toe
isContest: false
```

To get the opponent's program id, you have to find it on the CG Stats page. For example:

```
- agentId: '2995053'
  name: aCat
```

A reason for `internal_error` might be incorrect code language name. I advise to experiment with proper spelling/spacing then.

Brutaltester

Brutaltester is a program that allows to run matches of multiplayer games using offline agents. It may be useful for offline debugging as well as improving the strength of your program:

- ➔ the program is then free to create any artifacts as additional logfiles, unlimited stderr output, etc.)
- ➔ each simulated game creates a reply you can analyze
- ➔ you can test multiple versions of your agent against each other and manually/automatically tune their parameters

Brutaltester works off-the-shelf given the current build. It requires properly formatted game referee (engine) as one of the parameters. A compatible referee for UTTT can be found here.

This is the command template to run brutaltester on UTTT for 100 games with a player swap:

```
java -jar cg-brutaltester-1.0.0.jar
   -r "java -Dleague.level=2 -jar tictactoe.jar"
   -p1 "./myCode.exe" -p2 "./myOtherCode.exe"
   -t 2 -n 100 -l "./logs/" -s
```

If the output informs about *Negative score during game*, it means that for an enlisted player number (minus one) some error occured.

Command Line Referees

Before running brutaltester it might be worth to try if the standalone referee works properly.

```
java "-Dleague.level=2"
   -jar tictactoe.jar
   -p1 "./myCode.exe" -p2 "./myOtherCode.exe"
   -l tmp.log -s
```

You may need to experiment with proper parameter quoting to make the thing work.

For newer java versions you may need to add `--add-opens java.base/java.lang=ALL-UNNAMED` to java call.

The `-s` flag runs a localhost server where you can see the game. It may not work properly if the default path to the web server dir is too long. You may need to add another system property, e.g.: `"-Djava.io.tmpdir=D:\cg_tmp01"`

Also, it may happen that the brutaltester shows you multiple errors due to the timeouts, although on the CG site everything works properly. In such case you may tweak your program to send answers earlier for the test purposes.

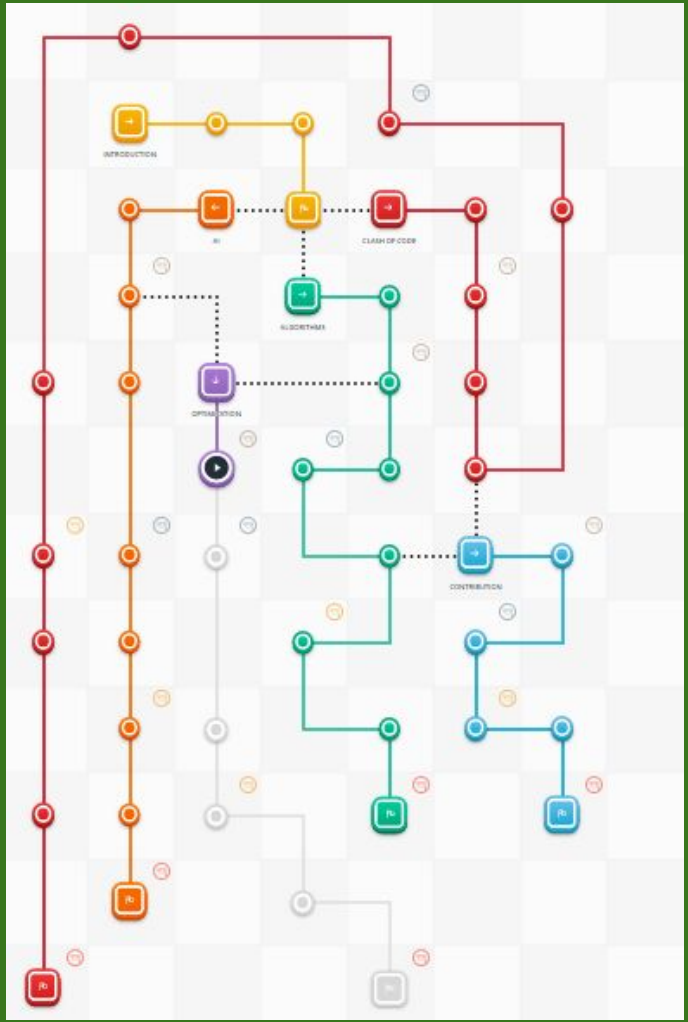
CG Tools

CG Tools is a collection of community-made websites that usually contain some useful information and statistics.

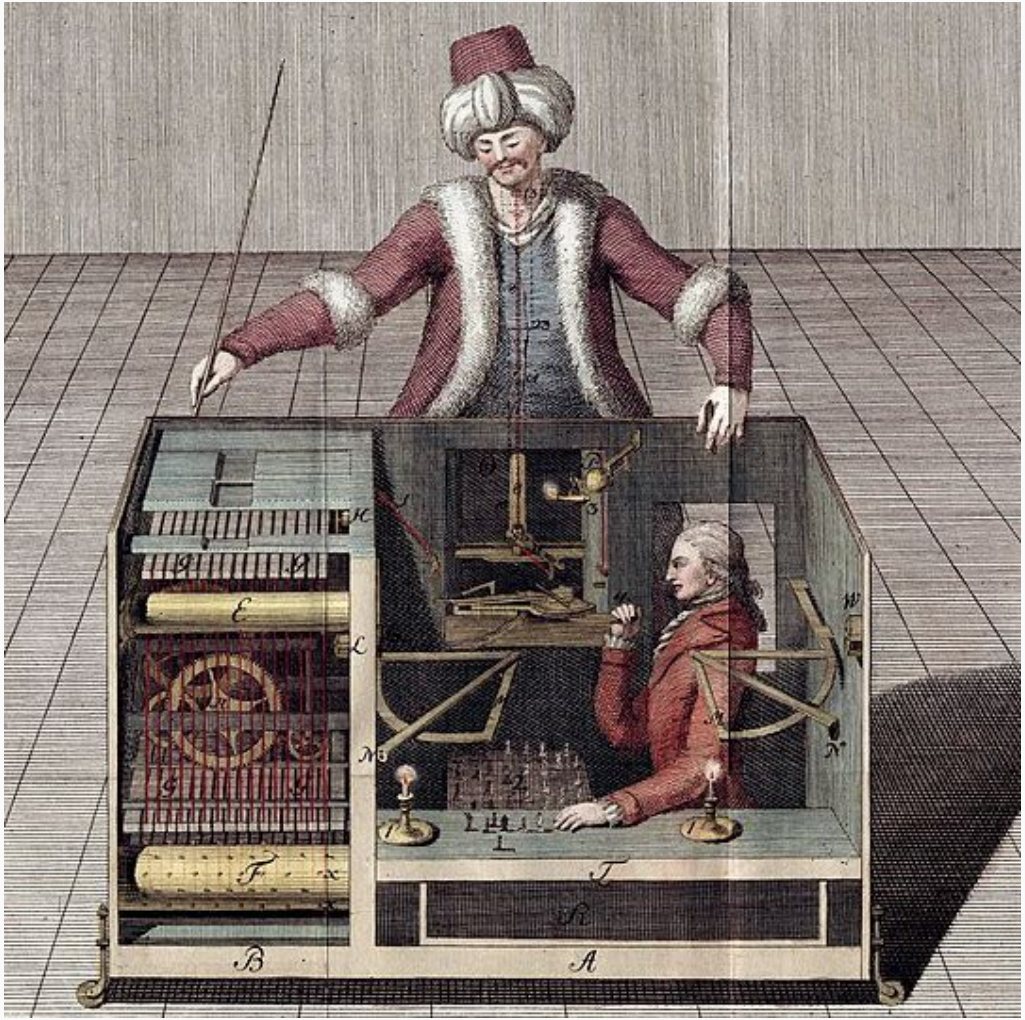
- ★ Puzzles list
  Gives nice overview of all CG puzzles
- ★ CG multi
  Handy list of multiplayer puzzles with your rank on it.
- ★ XP Leaderboard
  Contains detailed leaderboard with useful sorting options.
- ★ CG Points Details
  Shows missing CPs for each competitive task. Profile ID is the last part of the user's profile page address.
- ★ CG Stats
  Shows detailed winrate of your bot against concrete opponents it was playing when submitted.Very useful especially during the competitions.

Summary

- ★ Course’s labs will be based on CodinGame
- ★ Have fun discovering the platform on your own
- ★ And use it as a tool for gaining experience



Thanks!



Bonus reference quiz