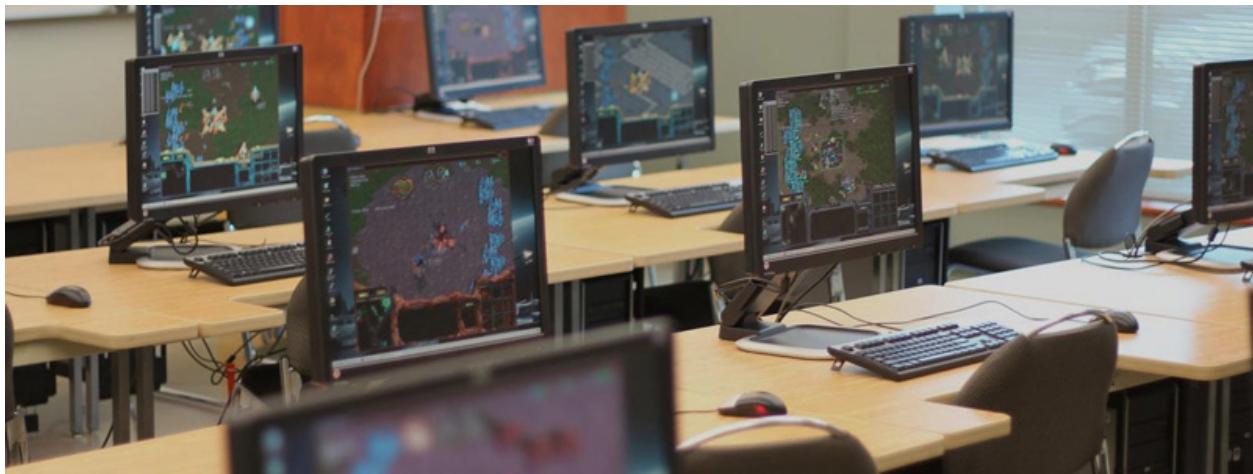


[Home](#)[Compete](#)[Results](#)[Rules](#)[Resources](#)[Contact](#)

## 2015 AIIDE StarCraft AI Competition Report

[Introduction](#) - [Entrants](#) - [Results & Analysis](#) - [Human vs. Machine](#) - [Further Discussion](#)

[Post](#)[Follow @davechurchill](#)

Written By: [David Churchill](#) dave.churchill@gmail.com

### Introduction

The 6th annual AIIDE StarCraft AI Competition was held in October/November 2015, hosted by the [University of Alberta](#) Department of [Computing Science](#), and organized by [David Churchill](#) and [Michael Buro](#). This year the competition ran for two full weeks - from October 21st to November 4th, the state of the art in StarCraft AI bots competed against each other 24 hours a day in the [Computing Science Dapartment](#) at the U of A.

### Competition Format

The tournament format for 2015 remains unchanged from previous years, with bots playing one on one in the full game of Starcraft: BroodWar using [BWAPI](#). Bots play against each other in round-robin format with each bot playing against each other bot on each of the 10 maps. After a round robin has been played on one map, another round robin begins on the next map, and this process is repeated until the tournament time limit is reached. At the end of the tournament the results are trimmed to the end of the last completed round robin (for fairness) and bots are ranked based on their overall win percentage. No playoffs are played.

Bots are given access to read and write directories throughout the tournament in order to implement AI techniques such as learning about opponents for strategy selection. Bots can continuously write to a specific 'write' folder, and read from a specific 'read' folder. The contents of the 'write' folder are copied to the 'read' folder after each full round robin has been played - this ensures that no bot has any advantage over another bot due to individual game scheduling. The result of this process is that bots have access to information from every

previously completed round. Full details of the file IO system can be seen on the [rules page](#).

For full details on tournament format and rules, check the [official competition rules section](#).

## Competition Environment

In order to play as many games as possible between the bots, custom Starcraft AI tournament managing [software](#) was written to allow the automatic scheduling and running of games on a local network. With many bots incorporating AI techniques which learn about opponents over time, it was very important to play as many games as possible to ensure that bots have enough time to learn as possible, and to get statistically significant results. Starcraft AI matches have a notoriously high amount of results variance over just a few games per pairing, often taking thousands of games for results to converge.

Since 2011, competitions at the UofA were held in an undergraduate [computer lab](#) utilizing 20 Windows XP machines. Since the computer lab was actively used by students, this meant that games could only be played between the end of one school term and the beginning of the next (usually at the end of August). This year the competition was run on Virtual Machines so that games could be run for much longer - thanks to a significant effort by Nicolas Barriga (another PhD student in RTS AI at the UofA). In total, 4 Linux servers each hosted 3 Windows XP virtual machines for a total of 12 virtual machines for the tournament to run on. This, combined with the fact that AIIDE was scheduled later than normal (in November) meant that we could let the competition run for a full two weeks - more than double the running time from last year. An additional advantage to virtual machines was that the competition could be monitored and controlled remotely with remote desktop software. By using [KRDC](#) through an ssh [tunnel](#), all 12 machines were able to be simultaneously controlled and the tournament could be stopped and restarted in just a matter of minutes from home. In previous years, debugging or restarting the tournament involved driving to campus, physically walking around a computer lab and restarting the software manually.

## "Why not StarCraft 2?"

This is the question we always get asked when we tell people we are doing a BroodWar AI competition. This competition relies completely on [BWAPI](#) as a programming interface to BroodWar. BWAPI was created by reverse engineering BroodWar and relies on reading and writing to the program memory space of BroodWar in order to read data and issue commands to the game. Since any program that does this can essentially be seen as a map hack or cheat engine, Blizzard has told us that they don't want us to do anything similar for StarCraft 2. In fact, most of the [StarCraft 2 EULA](#) specifically deals with not modifying the program in any way. We are happy that Blizzard have allowed us to continue holding tournaments using BWAPI, and they have also helped out by providing prizes to the AIIDE tournament, however until their policy changes we will not be able to do the same for StarCraft 2.

There are other RTS game engines available for competitions as well. One such engine is [ORTS](#) - a free software RTS game engine which ran several [competitions](#) until 2010 when BWAPI was released and the [first](#) AIIDE Starcraft AI Competition was held. Another engine is [microRTS](#), a Java RTS engine which plays a simplified grid-based RTS game and is designed specifically for testing AI techniques.

## RTS AI Techniques

For an excellent up to date overview on the state of the art in StarCraft AI techniques and bot architecture descriptions, I highly recommend reading the following publications:

- [RTS AI Problems and Techniques](#) [2015]  
S. Ontanon, G. Synnaeve, A. Uriarte, F. Richoux, D. Churchill, and M. Preuss  
Springer Encyclopedia of Computer Graphics and Games
- [A Survey of Real-Time Strategy Game AI Research and Competition in StarCraft](#) [2013]  
S. Ontanon, G. Synnaeve, A. Uriarte, F. Richoux, D. Churchill, and M. Preuss  
Accepted to TCIAIG (August 2013)

## Entrants

This year, 25 teams registered to compete in the competition, however 3 bots were withdrawn after problems arose during the testing phase of the tournament. One registered bot was unable to submit the source code for

the proper version of the bot (a requirement for the open source AIIDE competition) and two other bots crashed a majority of their games, and decided to withdraw rather than compete this year. With 22 entries, this was the largest competition ever for AIIDE - a huge success!

The following table lists the entrants for this year's competition. On the right-hand side of the table you will see the 'Download' column which has 3 links: 'bot' (download the bot's submitted source code), 'rep' (download the bot's replays from this tournament), and 'faq' (survey responses from each bot author giving details about each bot and describing the techniques they used). You can **click on a bot name** to view details about the bot and its results in past competitions - it will be displayed in place of the table below, try it out! This functionality is provided by our new [Starcraft AI Competition Data Archive](#)! You can click on one of the following links to see details about past AIIDE competitions. Note:

AIIDE Competition Entrants: [2015](#) [2014](#) [2013](#) [2012](#) [2011](#) [2010](#)

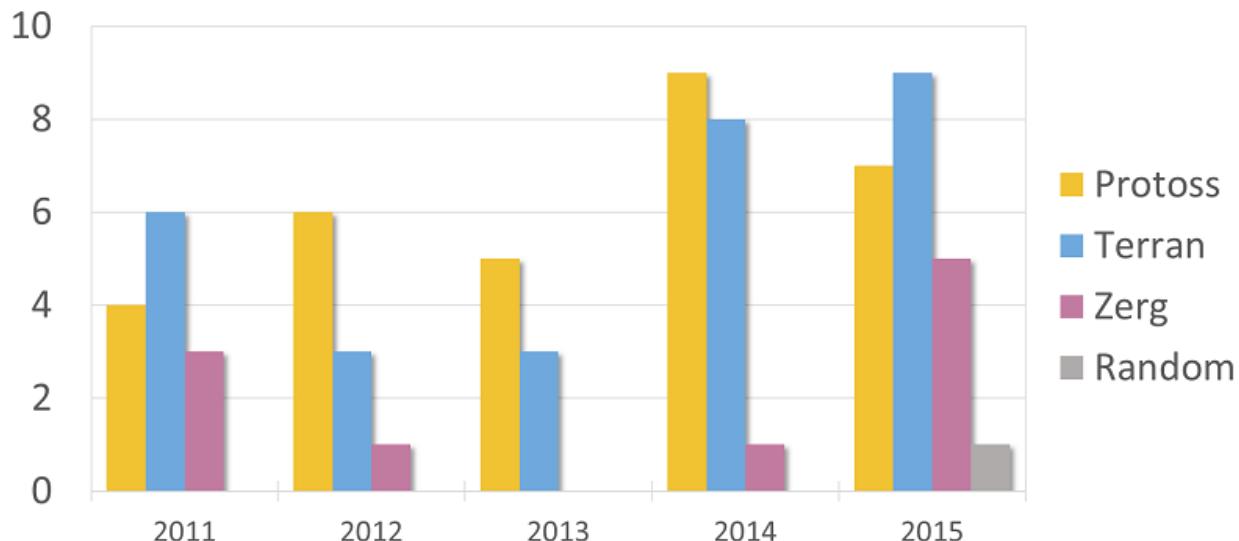
2015 AIIDE Starcraft AI Competition Competition Website - Official Results - Map Pack							
Rank	Bot Name	Author	Win %	Games	Wins	Race	Download
1	<a href="#">Tscmoo</a>	Vegard Mella	88.52	1890	1673	Zerg	<a href="#">bot</a> <a href="#">rep</a> <a href="#">faq</a>
2	<a href="#">ZZZKBot</a>	<a href="#">Chris Coxe</a>	87.83	1890	1660	Zerg	<a href="#">bot</a> <a href="#">rep</a> <a href="#">faq</a>
3	<a href="#">Overkill</a>	<a href="#">Sijia Xu</a>	80.69	1890	1525	Zerg	<a href="#">bot</a> <a href="#">rep</a> <a href="#">faq</a>
4	<a href="#">UAlbertaBot</a>	<a href="#">David Churchill</a>	80.2	1889	1515	Random	<a href="#">bot</a> <a href="#">rep</a> <a href="#">faq</a>
5	<a href="#">Aiur</a>	<a href="#">Florian Richoux</a>	73.02	1890	1380	Protoss	<a href="#">bot</a> <a href="#">rep</a> <a href="#">faq</a>
6	<a href="#">Ximp</a>	Tomas Vajda	67.78	1890	1281	Protoss	<a href="#">bot</a> <a href="#">rep</a> <a href="#">faq</a>
7	<a href="#">IceBot</a>	Kien Nguyen Quang	64.21	1889	1213	Terran	<a href="#">bot</a> <a href="#">rep</a> <a href="#">faq</a>
8	<a href="#">Skynet</a>	<a href="#">Andrew Smith</a>	64.13	1890	1212	Protoss	<a href="#">bot</a> <a href="#">rep</a> <a href="#">faq</a>
9	<a href="#">Xelnaga</a>	Ho-Chul Cho	62.7	1890	1185	Protoss	<a href="#">bot</a> <a href="#">rep</a> <a href="#">faq</a>
10	<a href="#">LetaBot</a>	Martin Rooijackers	60.95	1890	1152	Terran	<a href="#">bot</a> <a href="#">rep</a> <a href="#">faq</a>
11	<a href="#">Tyr</a>	Simon Prins	53.79	1889	1016	Terran	<a href="#">bot</a> <a href="#">rep</a> <a href="#">faq</a>
12	<a href="#">GarmBot</a>	Aurelien Lermant	52.58	1890	988	Zerg	<a href="#">bot</a> <a href="#">rep</a> <a href="#">faq</a>
13	<a href="#">NUSBot</a>	<a href="#">Authors</a>	39.37	1890	744	Protoss	<a href="#">bot</a> <a href="#">rep</a> <a href="#">faq</a>
14	<a href="#">TerranUAB</a>	Filip Bober	38.15	1890	721	Terran	<a href="#">bot</a> <a href="#">rep</a> <a href="#">faq</a>
15	<a href="#">Cimex</a>	Hiroto Takino	35.63	1889	673	Zerg	<a href="#">bot</a> <a href="#">rep</a> <a href="#">faq</a>
16	<a href="#">CruzBot</a>	Daniel Montalvo	32.38	1890	612	Protoss	<a href="#">bot</a> <a href="#">rep</a> <a href="#">faq</a>
17	<a href="#">OpprimoBot</a>	<a href="#">Johan Hagelback</a>	27.88	1890	527	Terran	<a href="#">bot</a> <a href="#">rep</a> <a href="#">faq</a>
18	<a href="#">Oritaka</a>	Yoshitaka Hirai	26.03	1890	492	Terran	<a href="#">bot</a> <a href="#">rep</a> <a href="#">faq</a>
19	<a href="#">Stone</a>	Igor Dimitrijevic	24.66	1890	466	Terran	<a href="#">bot</a> <a href="#">rep</a> <a href="#">faq</a>
20	<a href="#">Bonjwa</a>	Dustin Dannenhauer	22.91	1890	433	Terran	<a href="#">bot</a> <a href="#">rep</a> <a href="#">faq</a>
21	<a href="#">Yarmouk</a>	Abdelrahman Elogeel	9.1	1890	172	Terran	<a href="#">bot</a> <a href="#">rep</a> <a href="#">faq</a>
22	<a href="#">SusanooTricks</a>	<a href="#">Tetsuro Kamada</a>	7.83	1890	148	Protoss	<a href="#">bot</a> <a href="#">rep</a> <a href="#">faq</a>

2015 was the most international competition ever with 12 countries represented:



This year we also had the most even race distribution of any year at AIIDE. In previous years, Zerg had been poorly represented, but this year we had 5 new Zerg submissions! We also had the first ever Random race submission this year with UAlbertaBot. By choosing Random, you get randomly assigned one of the three races by Starcraft after the game has started. This means that it is harder to program a bot to play Random since you need to have strategies for all three races, however it does provide an advantage since your opponent does not know what race you are until they actually scout you in the game.

### AIIDE Starcraft AI Competition Race Distribution



## Results & Analysis

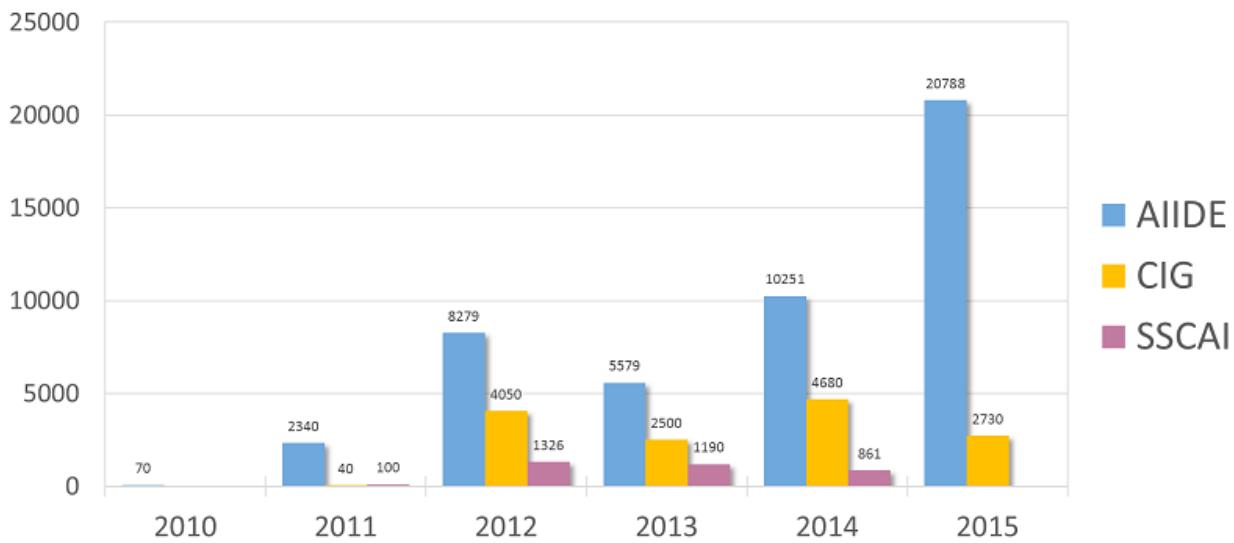
### Overview

All of the 2015 detailed results can be seen here: [2015 AIIDE StarCraft Competition - Final Results](#).

This year the tournament ran for two full weeks on 12 machines, resulting in 20,788 total games played - more than twice the previous record of 10,251 the year before. In the graph below you can see the number of games played in every major Starcraft AI competition so far. The Computational Intelligence in Games (CIG) competition uses the same tournament managing software as AIIDE, but does not run for quite as long. And while the Student Starcraft AI tournament (SSCAI) usually has more entrants than AIIDE it uses a different tournament structure -

like playing 2 or 3 rounds of round robin before having a playoff for the top few bots. This makes the results of this year's AIIDE competition the most accurate representation of bot skill so far.

## Starcraft AI Competitions - Total Games Played



### Final Results

The following table shows the final results for this year's tournament. This year's competition was extremely close with 1st/2nd place and 3rd/4th place being statistical ties with less than a 1% difference in win percentage between them. The three new Zerg entrants did extremely well - with tscmoo coming 1st, ZZZKBot coming 2nd, and Overkill coming 3rd. UAlbertaBot came in fourth place while playing Random which is also a great accomplishment. If you check out the [detailed results](#) you will also see that UAlbertaBot was the only bot to finish with a > 50% win rate against every other bot - even though it did not have the overall highest winning percentage. You can click on the column headers in the table below to sort the rows based on the results.

Bot	Games	Win	Loss	Win %	AvgTime	Hour	Crash	Timeout
tscmoo	1890	1673	217	88.52	12:06	3	0	0
ZZZKBot	1890	1660	230	87.83	6:35	3	0	0
Overkill	1890	1525	365	80.69	11:15	17	21	7
UAlbertaBot	1889	1515	374	80.2	10:47	52	0	10
Aiur	1890	1380	510	73.02	14:11	54	0	0
Ximp	1890	1281	609	67.78	15:56	9	48	50
IceBot	1889	1213	676	64.21	14:28	17	13	0
Skynet	1890	1212	678	64.13	11:01	12	0	3
Xelnaga	1890	1185	705	62.7	15:51	91	111	0
LetaBot	1890	1152	738	60.95	10:45	24	58	0
Tyr	1889	1016	873	53.79	19:07	125	23	0
GarmBot	1890	988	902	52.28	16:06	15	0	0
NUSBot	1890	744	1146	39.37	12:23	37	254	103
TerranUAB	1890	721	1169	38.15	16:01	67	69	2
Cimex	1889	673	1216	35.63	16:12	87	435	11
CruzBot	1890	612	1278	32.38	19:03	173	2	41
OpprimoBot	1890	527	1363	27.88	19:42	0	116	0
Oritaka	1890	492	1398	26.03	15:56	131	1	0
Stone	1890	466	1424	24.66	14:51	123	1	6
<b>Total</b>	20788	20788	20788	N/A	14:38	732	1351	318

Bot	Games	Win	Loss	Win %	AvgTime	Hour	Crash	Timeout
Bonjwa	1890	433	1457	22.91	17:57	176	10	1
Yarmouk	1890	172	1718	9.1	14:00	102	86	0
SusanooTricks	1890	148	1742	7.83	17:49	146	103	84
<b>Total</b>	20788	20788	20788	N/A	14:38	732	1351	318

The **AvgTime** column indicates the average game length of a bot in minutes:seconds. You can see from these results that ZZZKBot's zergling rush strategy resulted in far quicker games than any other bot, with its average game less than half the time of the tournament average.

The **Hour** column shows how many games a bot took to the hour time limit, for which the game was then decided by in-game score. Games typically only go to the full hour when one bot fails to find the last remaining buildings of a vanquished opponent. This year 3.5% of all games went to the hour time limit, but the majority of these hour long games are played by the worst performing bots. This is probably because of logic bugs which the these bots that make them unable to find and/or finish off their enemies.

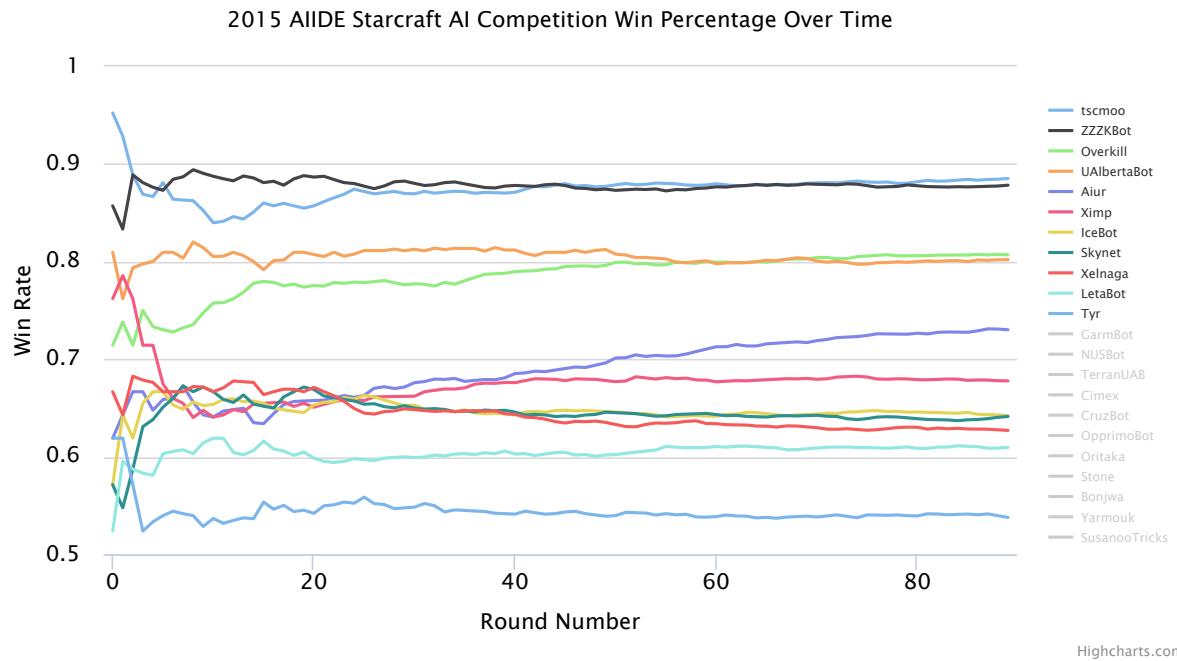
The **Crash** column indicates how many times the bot crashed during the competition. A crash in this sense is a programming error that caused the bot's program to terminate, resulting in a windows error shutting down StarCraft. The tournament manager software monitors the Starcraft.exe process and can detect when it has shut down, recording it as a crash.

The **Timeout** column indicates how many times a bot 'timed out', resulting in a game loss. The exact conditions for a time out are specified on our rules page, but the main two reasons are when a bot takes more than one minute on a single frame (usually caused by infinite loops), or when a bot has more than 320 frames which take more than 55ms to compute. Since this is a real-time strategy game, we want to enforce strict rules on bots taking too long to decide on what to do.

### Win Percentage Over Time

The following graph shows each bot's win percentage over time throughout the course of the competition. You can zoom in on the graph by dragging an area with the left mouse button, or you can view a larger version in a new window by [clicking here](#). Only those bots with a > 50% win rate are shown by default, but you can show or hide any of the results by clicking a bot's name in the graph legend to the right.

This graph shows just how close the competition was, with 1st/2nd place and 3rd/4th place alternating positions several times about 2/3 of the way through the tournament. In fact, if the tournament had ended with the same number of games played as in 2014, the results for 1st/2nd and 3rd/4th would have been reversed. We can also clearly see the effects of bots that learned throughout the competition with the most exceptional example being Aiur who finished in 5th place. During the course of the competition Aiur went from 63% wins to 73% wins between rounds 20 and 90 - a huge gain of 10%. Overkill's learning also helped it gain the few percentage points necessary to overtake UAlbertaBot (which did not do any learning this year) during the last half of the competition. A graph showing each bot's wins per round can be seen by [clicking here](#).



## Human vs. Machine

Each year we want to answer the question "How do the best bots perform vs. the best human players?", so we hold a human vs. machine match between the best ranking bots in the competition and an expert human player. This year our human vs. machine match's human player was [Djem5](#), a Russian Broodwar player who is widely regarded as one of the best non-Korean Protoss players in the world. Djem5 has been playing Starcraft since 2003 and has recently won several top foreign Broodwar tournaments such as the Team Liquid Legacy Cup. This year 3 matches were played: A best-of-5 match vss tscmoo, and two best-of-3 matches vs. ZZZKBot and UAAlbertaBot. It is important to note that these human vs. machine matches are a little bit unfair for the bots themselves, as they were designed to play in a tournament consisting of thousands of games, and as such the bots may be trying out risky strategies or tactics that they would not have chosen if designed to play in a shorter best-of-n type matchup where they may have implemented safer strategies.

Each match was commentated by Paul Paradies and can be watched on [Youtube](#), or you can download the Starcraft replay files and watch in-game. Click the Show Spoiler button for a quick analysis of what happened in each game.

### Djem5 vs. Tscmoo - [Youtube](#) - [Replays](#)

Show Spoiler / Analysis

### Djem5 vs. ZZZKBot - [Youtube](#) - [Replays](#)

Show Spoiler / Analysis

### Djem5 vs. UAAlbertaBot - [Youtube](#) - [Replays](#)

Show Spoiler / Analysis

## Further Discussion

It is obvious from the results of the human vs. machine match that Starcraft AI bots still have a long ways to go before beating the best expert human players. However, despite Djem5 making the bots look silly this year, several bots such as Tscmoo kras10 have done additional testing vs. human players on [ICCup](#) in which they were able to defeat D-ranked, and even some C-ranked players. After the human players have played one or two games against the bots they are then easily able to detect and exploit small mistakes that the bots make in order to easily win the majority of games. So while bots are able to win some games against decent amateur human

players, they do not do well in best-of-n game scenarios where the humans get to learn their behaviours.

Overall I believe the skill level of the bots has gone up quick quickly in the past two years. The 1st, 2nd, and 3rd place finishers in the 2014 competition placed 7th, 6th, and 10th place respectively in the 2015 competition. With the skill level bots continuing to rise, and by keeping the competition open source and fixing the blunders that the bots made this year I believe that in a year or two we will have several bots which will be able to easily defeat amateur human players. We are still a long way from beating the professionals though! For additional discussion feel free to read the concluding remarks from the [2013 report](#) which I believe remain true today.

Thanks to everyone for competing, and I hope you enjoyed the 2015 AIIDE Starcraft AI Competition! gg