

# StarCraft II as an Environment for Artificial Intelligence Research

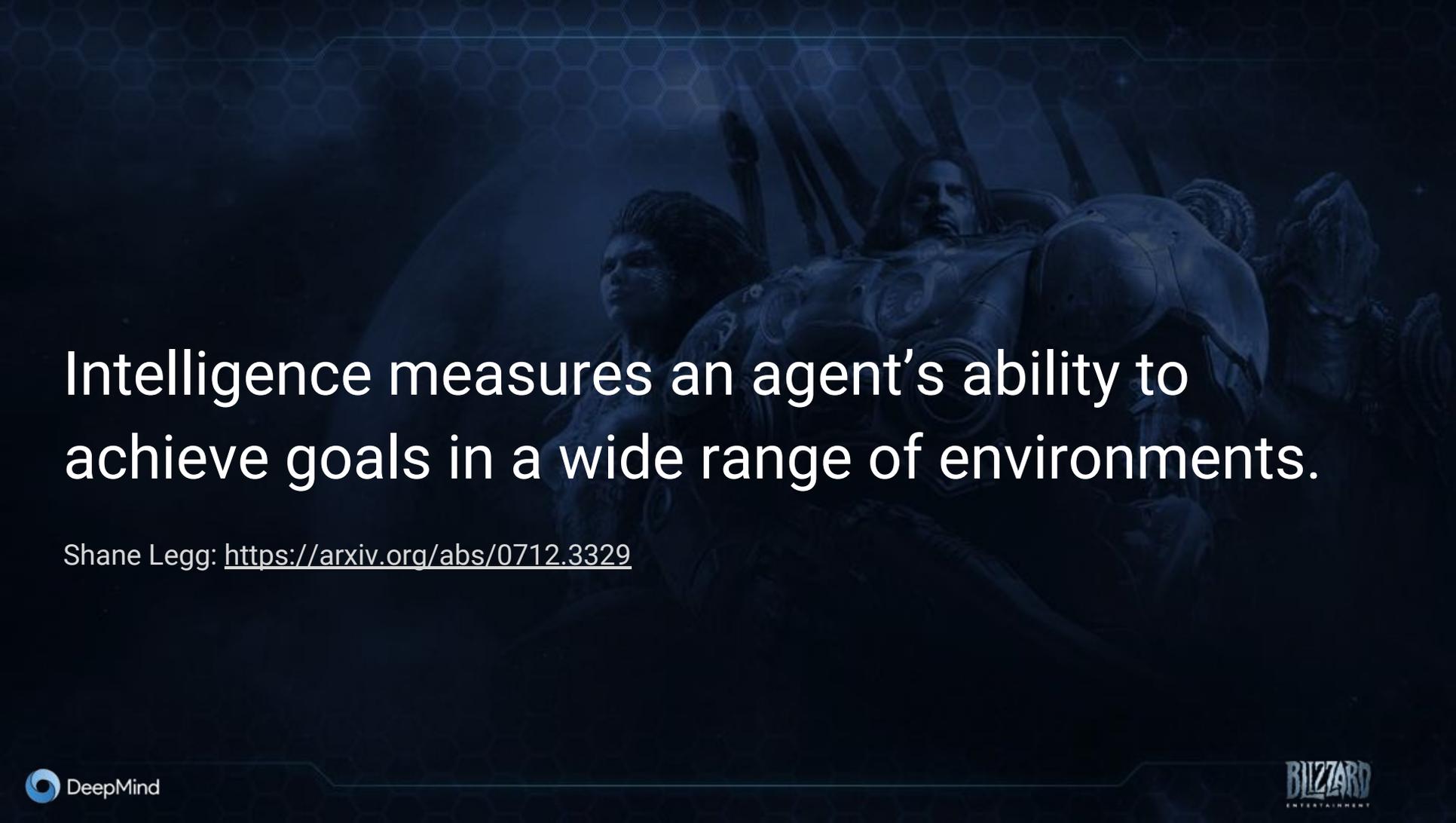
Timo Ewalds - DeepMind  
Chris Lee - Blizzard

The background of the slide features a dark blue, monochromatic image of several characters from the Warcraft universe. In the center, a character with long hair and a beard, likely a druid or shaman, is shown in profile. To the right, a large, armored figure, possibly a paladin or warrior, is visible. The background is overlaid with a faint, glowing hexagonal grid pattern, reminiscent of the Warcraft game's interface or environment.

DeepMind's Mission:

Solve intelligence.

Use it to make the world a better place.



Intelligence measures an agent's ability to  
achieve goals in a wide range of environments.

Shane Legg: <https://arxiv.org/abs/0712.3329>

# The Reinforcement Learning Paradigm

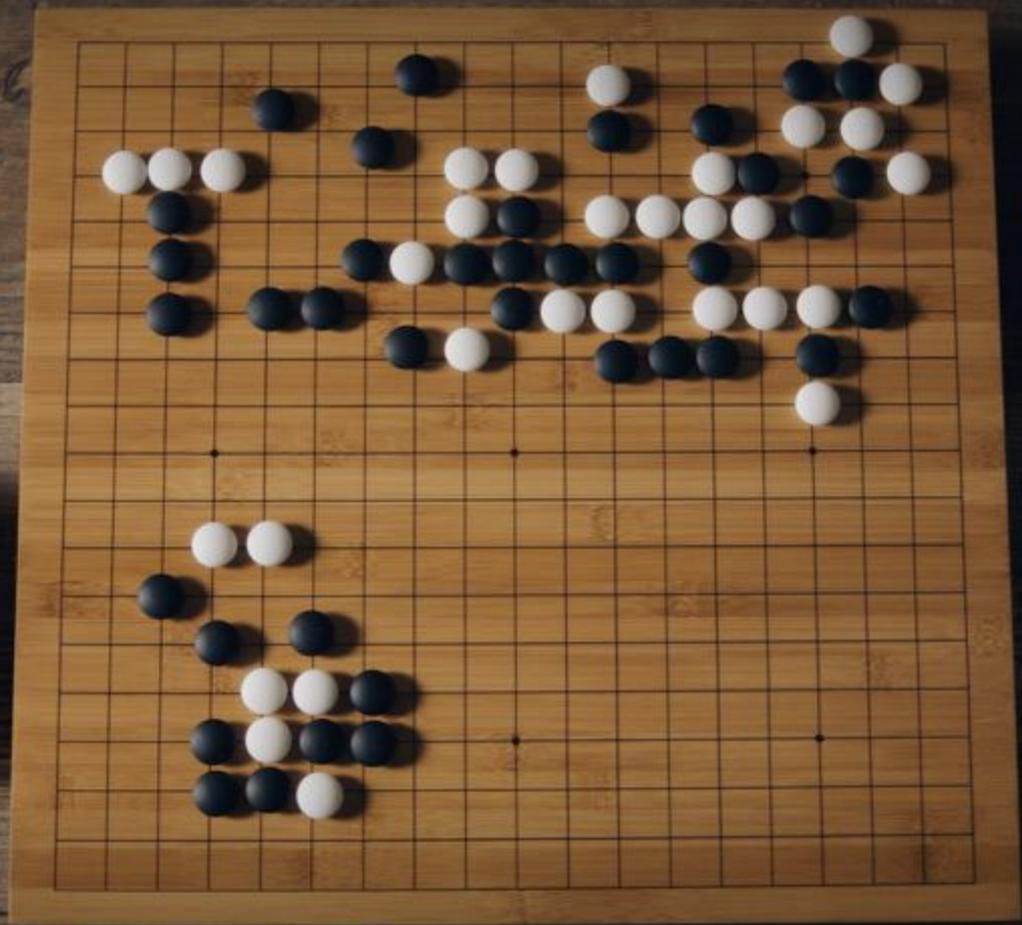


# DeepMind and Games

## 'End-to-end' agents: from pixels to actions

Games are the perfect platform for developing and testing AI algorithms





# Challenge Match – Seoul, March 2016

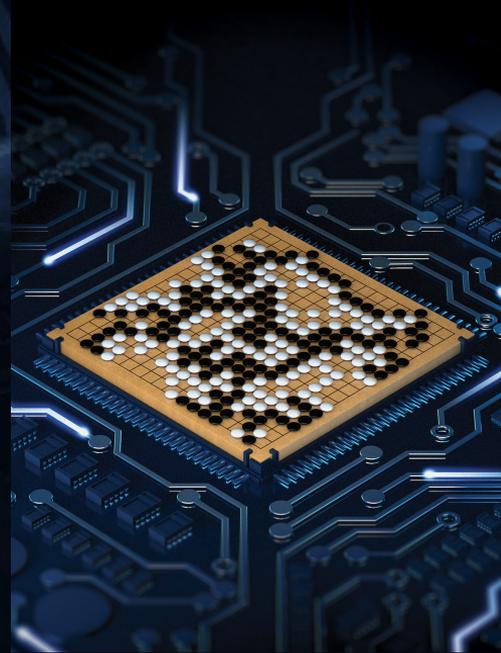
Legend of the game:  
18 world titles

Best player of  
the past decade



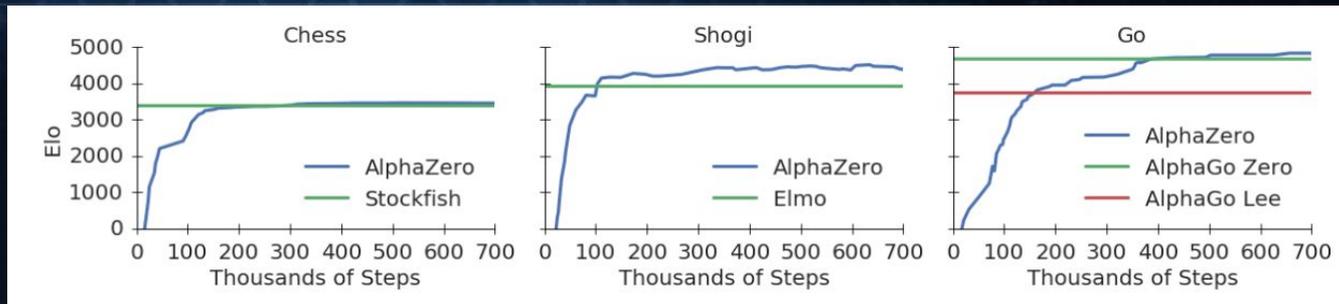
Lee Sedol

Vs

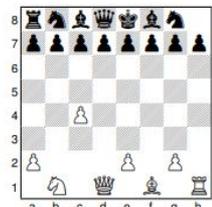


AlphaGo

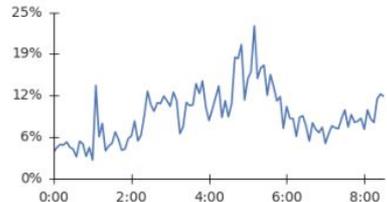
# AlphaZero



A10: English Opening

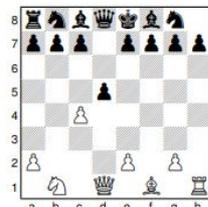


w 20/30/0, b 8/40/2

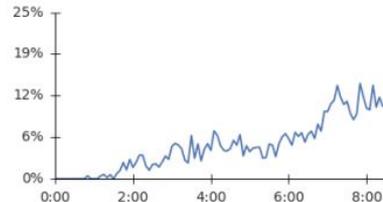


1...e5 g3 d5 cxd5 ♖f6 ♗g2 ♕xd5 ♖f3

D06: Queens Gambit

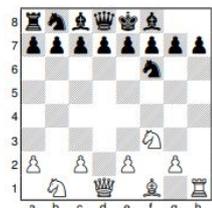


w 16/34/0, b 1/47/2

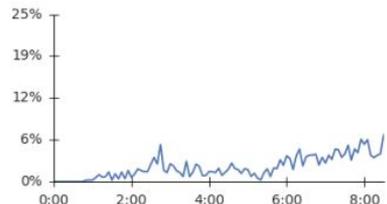


2...c6 ♖c3 ♖f6 ♖f3 a6 g3 c4 a4

A46: Queens Pawn Game

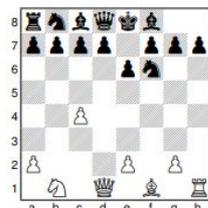


w 24/26/0, b 3/47/0

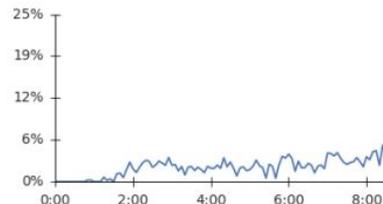


2...d5 c4 e6 ♖c3 ♗e7 ♗f4 O-O e3

E00: Queens Pawn Game



w 17/33/0, b 5/44/1



3.♖f3 d5 ♖c3 ♗b4 ♗g5 h6 ♖a4 ♖c6

# STAR CRAFT II



# StarCraft II 101

1 - Collect resources



2 - Build a base



3 - Build some units



4 - Engage the opponent





# TERRAN VS ZERG

# Decisions to make



# Why StarCraft II?

## **Imperfect Information**

guess what the opponent is doing

## **Huge Action Space**

need for hierarchical actions

## **Economy**

resource management  
expanding vs. defence

## **Real-Time**

simultaneous, fast paced decisions  
multi-tasking

## **Long Pay-Off**

strategy more important than micro

## **3 Asymmetric Races**

each with their own  
strengths and weaknesses

# Editor

The screenshot shows the StarCraft II Editor interface with a trigger configuration. The window title is "Triggers - [C:/.../StarCraft II/Maps/mini\_games/CollectMineralsAndGas.SC2Map] - StarCraft II Editor (rp, SC2.4.2.a)". The menu bar includes File, Edit, View, Data, Map, Modules, Window, Help, and Development. The toolbar contains various icons for editing triggers and maps. On the left, a tree view shows the map's structure, including "Init" and "Score Updates and Victory". The main editor area displays a trigger configuration under the "Init" tab. The trigger is a "General -if (Conditions) then do (Actions) else do (Actions)" trigger. The conditions include "Controller of player 1 != Computer". The actions include "Camera -Lock camera input for player 1", "Visibility -Reveal Playable Space for player 1 for 0.0 seconds and Do Not check cliff level", "Sound -Pause Music soundtrack for (All players) (Immediately)", "Game -Set the game speed to Faster", "Scenario setup", "Player -Modify player 1 Minerals: Set To 50", and "Unit -Create 1 Command Center for player 1 at Command Center using default facing (No Options)". The "Unit -Create 1 Command Center for player 1 at Command Center using default facing (No Options)" action is highlighted. Below the trigger configuration, a preview window shows the details of the selected action: "Create Units With Default Facing", "Count: 1", "Type: Command Center", "Player: 1", and "Point: Command Center". At the bottom of the editor, a text box displays the action's description: "Create 1 Command Center for player 1 at Command Center using default facing (No Options)".



# Built-In Scripted AI

Team 2

A.I.

Very Easy

Random

Any Build

Very Easy

Easy

Medium

Hard

Harder

Very Hard

Elite

Cheater 1 (Vision)

Cheater 2 (Resources)

Cheater 3 (Insane)

Any Build

Full Rush

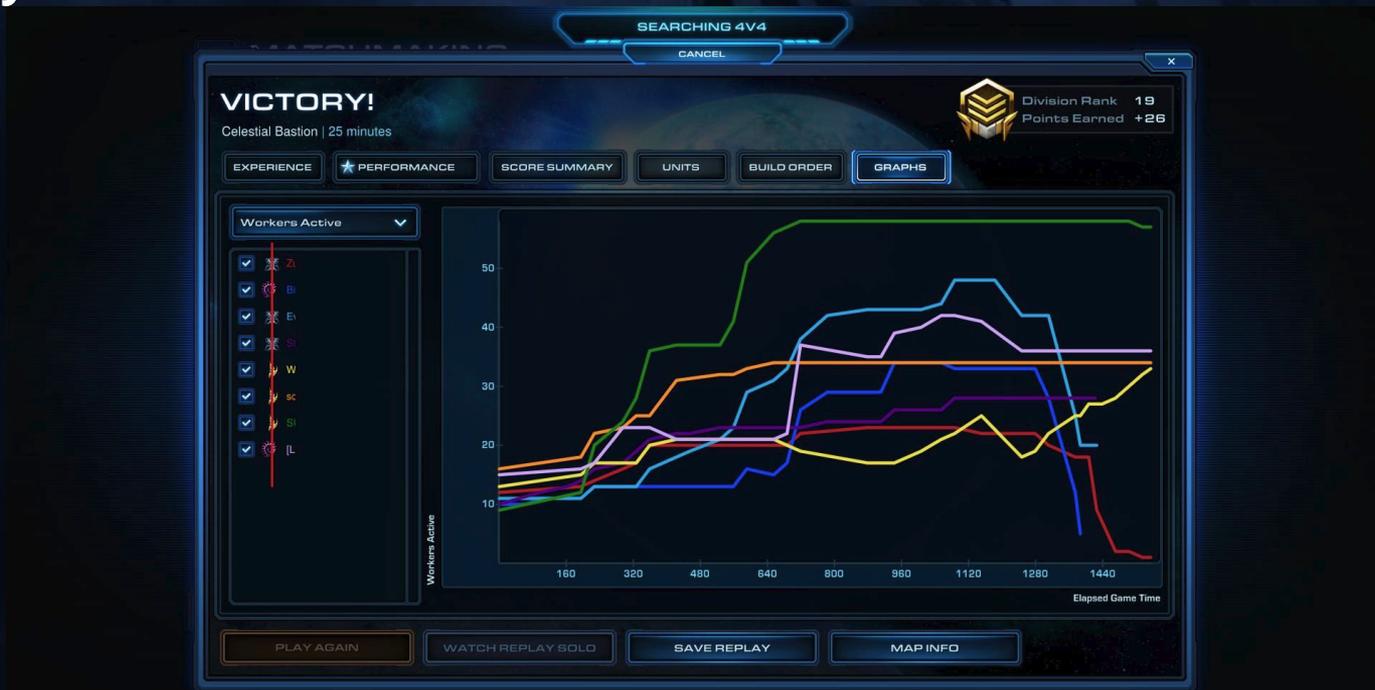
Timing Attack

Aggressive Push

Economic Focus

Straight to Air

# Replays

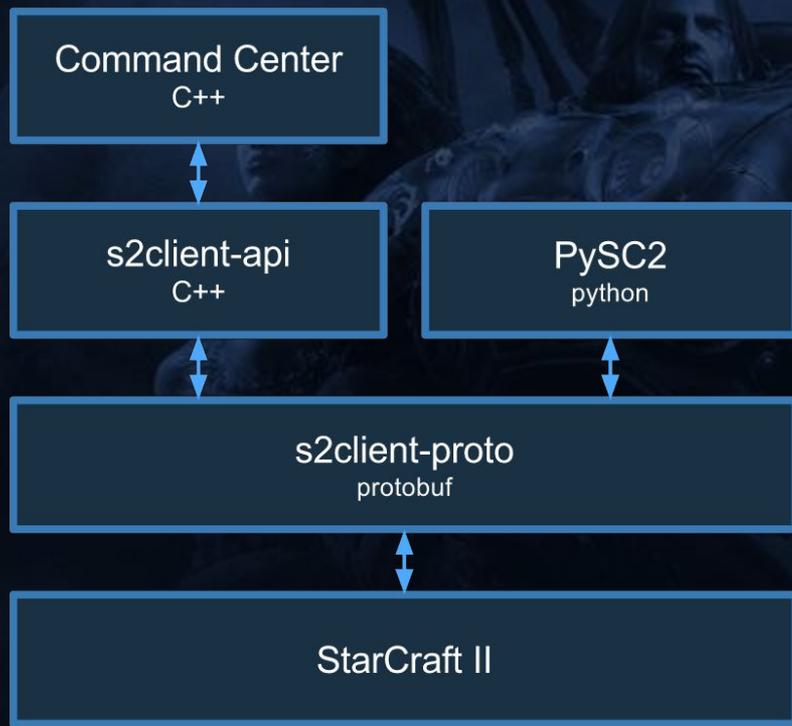


Millions of anonymized replays with their actions and observations

# Professional Scene



# SC2 API



# Language Agnostic Protobuf API

```
message Request {
  RequestCreateGame create_game;
  RequestJoinGame join_game;
  RequestStartReplay start_replay;
  RequestGameInfo game_info;
  RequestObservation observation;
  RequestAction action;
  RequestStep step;
}

message Action {
  ActionRaw action_raw;
  ActionSpatial action_feature_layer;
  ActionSpatial action_render;
  ActionUI action_ui;
}

message ResponseObservation {
  repeated Action actions;
  Observation observation;
  repeated PlayerResult player_result;
}

message Observation {
  uint32 game_loop;
  PlayerCommon player_common;
  repeated AvailableAbility abilities;
  Score score;

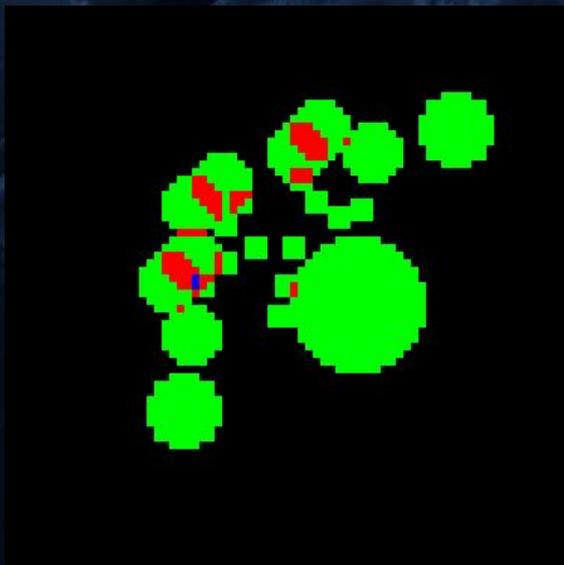
  ObservationRaw raw_data;
  ObservationFeatureLayer feature_layer_data;
  ObservationRender render_data;
  ObservationUI ui_data;
}
```

# Interfaces

Rendered



Feature Layer



Raw

```
units {  
  display_type: Visible  
  unit_type: 86  
  owner: 1  
  pos {  
    x: 135.5  
    y: 104.5  
    z: 11.980469  
  }  
  facing: 4.712389  
  is_selected: false  
  health: 1500  
}
```

# Interfaces - Spatial

## Rendered



Decomposed:

- Screen, minimap, resources, available actions

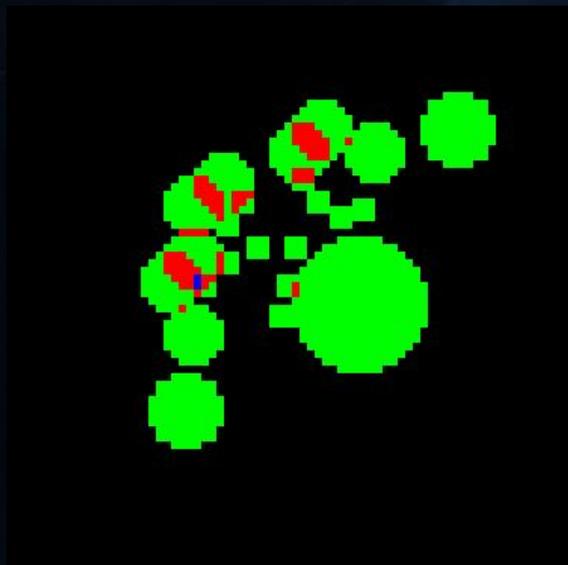
Same control as humans:

- Pixel coordinates
- Move camera
- Select unit/rectangle

Great for Deep Learning, but hard

# Interfaces - Spatial

## Feature Layer



Same actions: still in pixel space

Same decomposed observations, but more abstract

- Orthogonal camera

Layers:

- unit type
- unit owner
- selection
- health
- unit density
- etc

# Interfaces

## Raw

```
units {  
  display_type: Visible  
  unit_type: 86  
  owner: 1  
  pos {  
    x: 135.5  
    y: 104.5  
    z: 11.980469  
  }  
  facing: 4.712389  
  is_selected: false  
  health: 1500  
}
```

List of units and state

Control each unit individually in world coordinates

Gives all observable state (no camera)

Great for scripted agents and programmatic replay analysis

# C++ API

- Offers easier C++ object representations of raw protobuf data
- Includes example scripted agents using the raw observations
- Easy to get your own basic agent up and running
- [github.com/Blizzard/s2client-api](https://github.com/Blizzard/s2client-api)

# Command Center

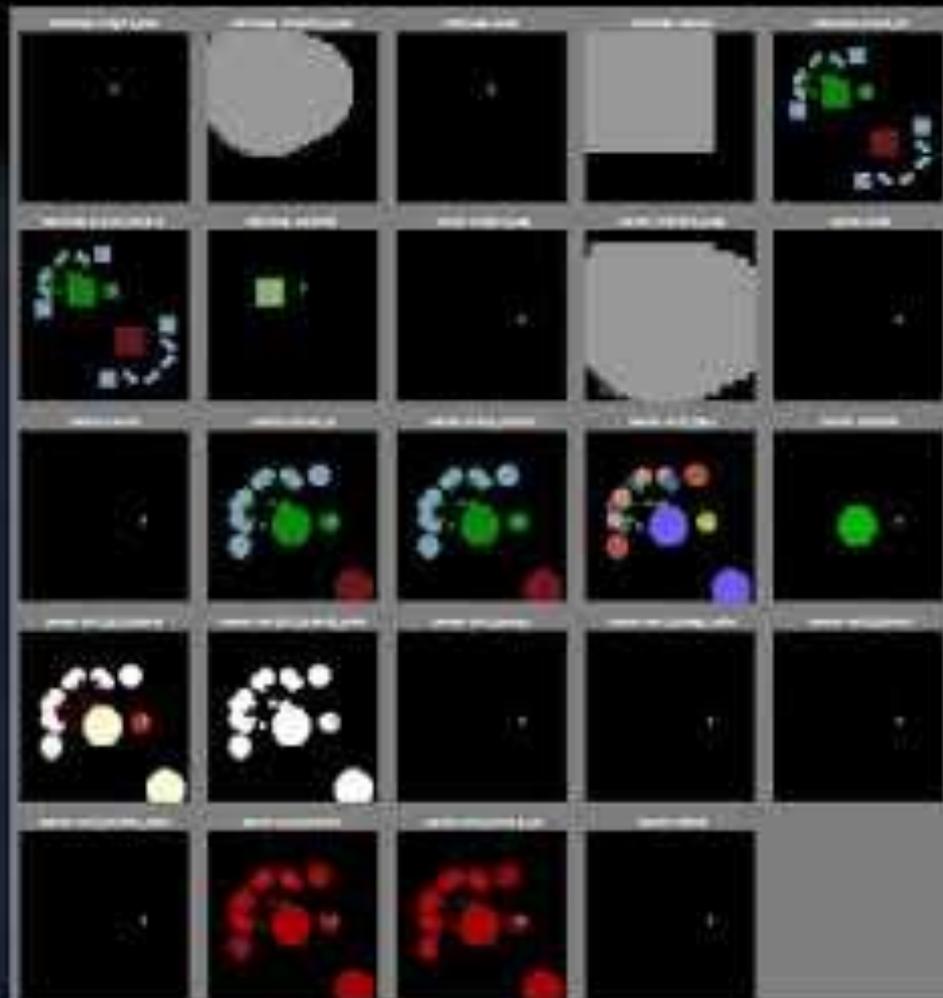
- Full featured agent that already knows basic strategies
- David Churchill's UAlbertaBot ported to StarCraft II
- Supports both BWAPI and SC2API
- Modular, easy to add your own strategies, build orders, etc.
- Basis of most bots on the community ladder: [sc2ai.net](http://sc2ai.net)
- [github.com/davechurchill/commandcenter](https://github.com/davechurchill/commandcenter)

# PySC2

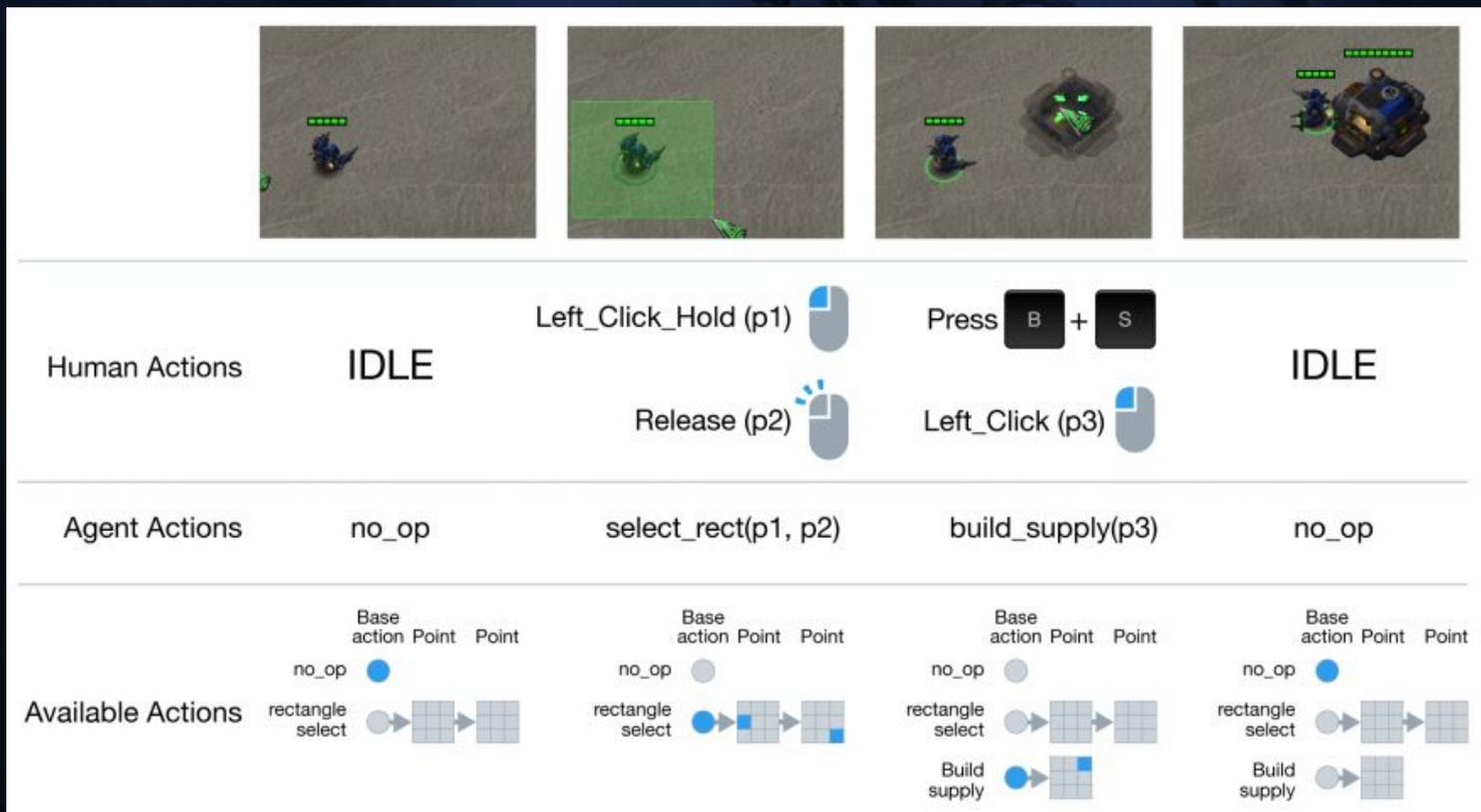
- Written in pure Python
- RL environment using the spatial actions and observations
- Includes a debug renderer for visualizing on Linux
- Works with self-play
- Can be installed from PyPI
- [github.com/deepmind/pysc2](https://github.com/deepmind/pysc2)

Blizzard: D:\S\_Visgenet\_0\_Pack: 18/23; Score: 444; Frame: 329; FPS: 022; R:40.2

Escape - Cancel  
o - ControlCenterLoad  
e - SCV  
y - Rally



# Action space



# StarCraft II: A New Challenge for RL, Vinyals et al, 17

- Mini-Games results
  - A3C
- Full game
  - A3C
  - Supervised

---

## StarCraft II: A New Challenge for Reinforcement Learning

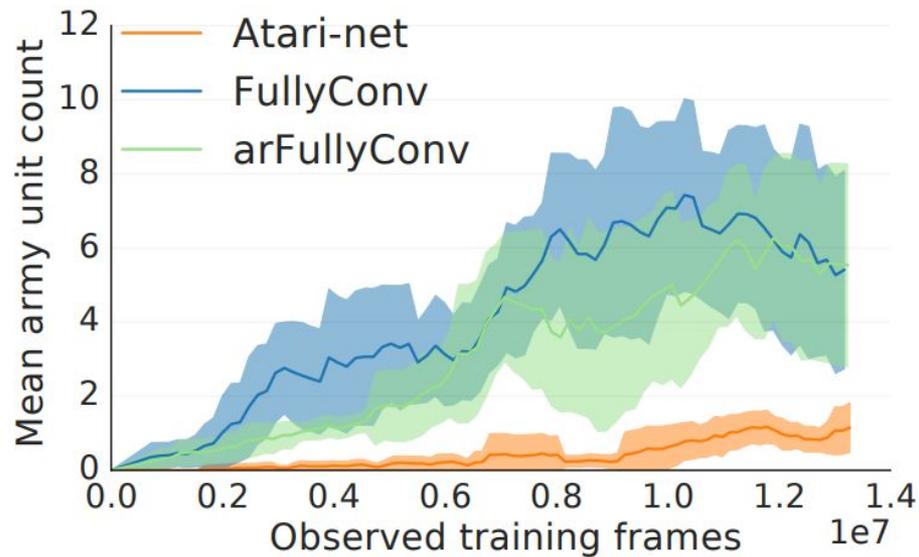
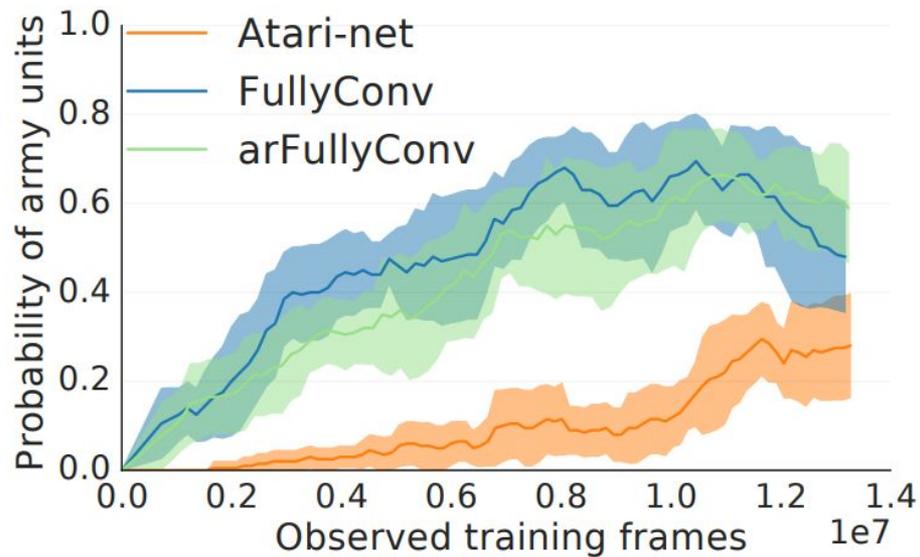
---

Oriol Vinyals Timo Ewalds Sergey Bartunov Petko Georgiev  
Alexander Sasha Vezhnevets Michelle Yeo Alireza Makhzani Heinrich Küttler  
John Agapiou Julian Schrittwieser John Quan Stephen Gaffney Stig Petersen  
Karen Simonyan Tom Schaul Hado van Hasselt David Silver Timothy Lillicrap  
*DeepMind*

Kevin Calderone Paul Keet Anthony Brunasso David Lawrence  
Anders Ekermo Jacob Repp Rodney Tsing  
*Blizzard*

<https://arxiv.org/abs/1708.04782>

# Full Game



# Minigames

Learning to play the full game is hard: start with minigames!

- Test a small aspect of the game:
  - Game concepts
  - Micro
  - Macro
- Easier reward: points for subgoals, instead of pure win/loss
- Fixed time limit (usually)
- We released 7 minigames, more made by the community

# Minigame Results

AGENT	METRIC	MOVE TO BEACON	COLLECT MINERAL SHARDS	FIND AND DEFEAT ZERG LINGS	DEFEAT ROACHES	DEFEAT ZERG LINGS AND BANELINGS	COLLECT MINERALS AND GAS	BUILD MARINES
		1	17	4	1	23	12	< 1
RANDOM POLICY	MEAN	1	17	4	1	23	12	< 1
	MAX	6	35	19	46	118	750	5
RANDOM SEARCH	MEAN	25	32	21	51	55	2318	8
	MAX	29	57	33	241	159	3940	46
DEEPMIND HUMAN PLAYER	MEAN	26	133	46	41	729	6880	138
	MAX	28	142	49	81	757	6952	142
STARCRAFT GRANDMASTER	MEAN	28	177	61	215	727	7566	133
	MAX	28	179	61	363	848	7566	133
ATARI-NET	BEST MEAN	25	96	49	101	81	3356	< 1
	MAX	33	131	59	351	352	3505	20
FULLY CONV	BEST MEAN	26	103	45	100	62	3978	3
	MAX	45	134	56	355	251	4130	42
FULLY CONV LSTM	BEST MEAN	26	104	44	98	96	3351	6
	MAX	35	137	57	373	444	3995	62



# Learning on CollectMineralShards from Raw Pixels



# Official resources

- SC2API: <https://github.com/Blizzard/s2client-proto>
  - Download links for Linux binaries, maps, replays
  - Main proto API, replay download api
- C++ API: <https://github.com/Blizzard/s2client-api>
- PySC2: <https://github.com/deepmind/pysc2>
- Paper: StarCraft II: A New Challenge for Reinforcement Learning
  - <https://arxiv.org/abs/1708.04782>
  - <https://deepmind.com/blog/deepmind-and-blizzard-open-starcraft-ii-ai-research-environment/>

# Community

- [wiki.sc2ai.net](http://wiki.sc2ai.net) - resources and links to get started
- [reddit.com/r/sc2ai](https://reddit.com/r/sc2ai) - discussion and links
- [Discord](#) - where the main discussion happens
- [sc2ai.net](http://sc2ai.net) - ladder for bots
- [Command Center](#) - basis of most scripted agents

# Contact

Chris Lee: [cmlee@blizzard.com](mailto:cmlee@blizzard.com)

Timo Ewalds: [tewalds@google.com](mailto:tewalds@google.com)