Dungeon Crawl Stone Soup as a Goal Reasoning Challenge Problem

Dustin Dannenhauer

GRW @ ACS - 11/15/2021

Outline

Dungeon Crawl Stone Soup (~17 min)

DCSS as an AI Challenge Problem and GR Problem

Thoughts on Building an AI to Win DCSS

The API: *DCSS-AI-Wrapper*

What is Dungeon Crawl Stone Soup?

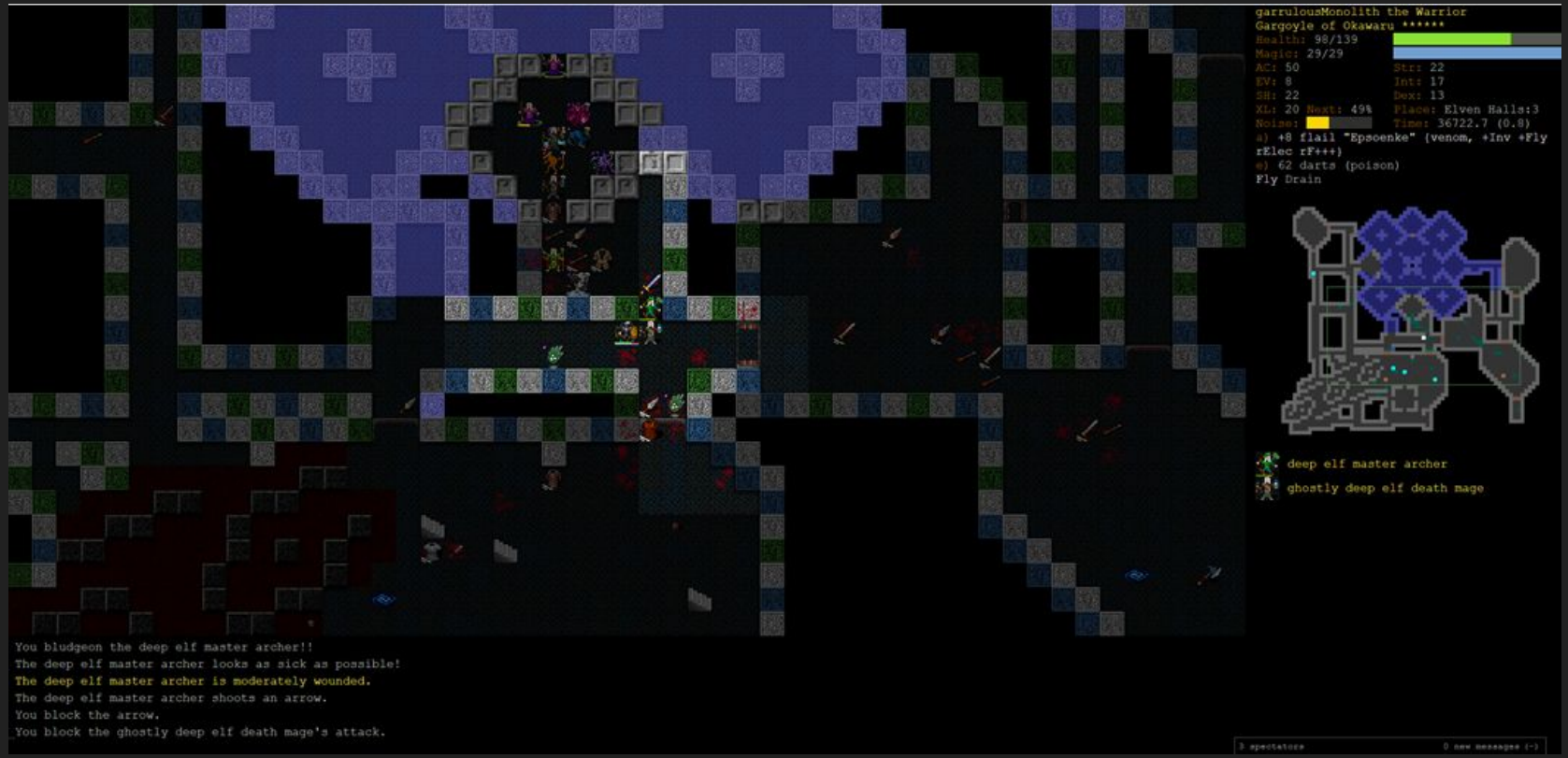
Rogue-like video game under active development since 2006

Game Objective: Navigate your character through a series of 2D grid-based levels to retrieve the Orb of Zot, and return to the entrance.

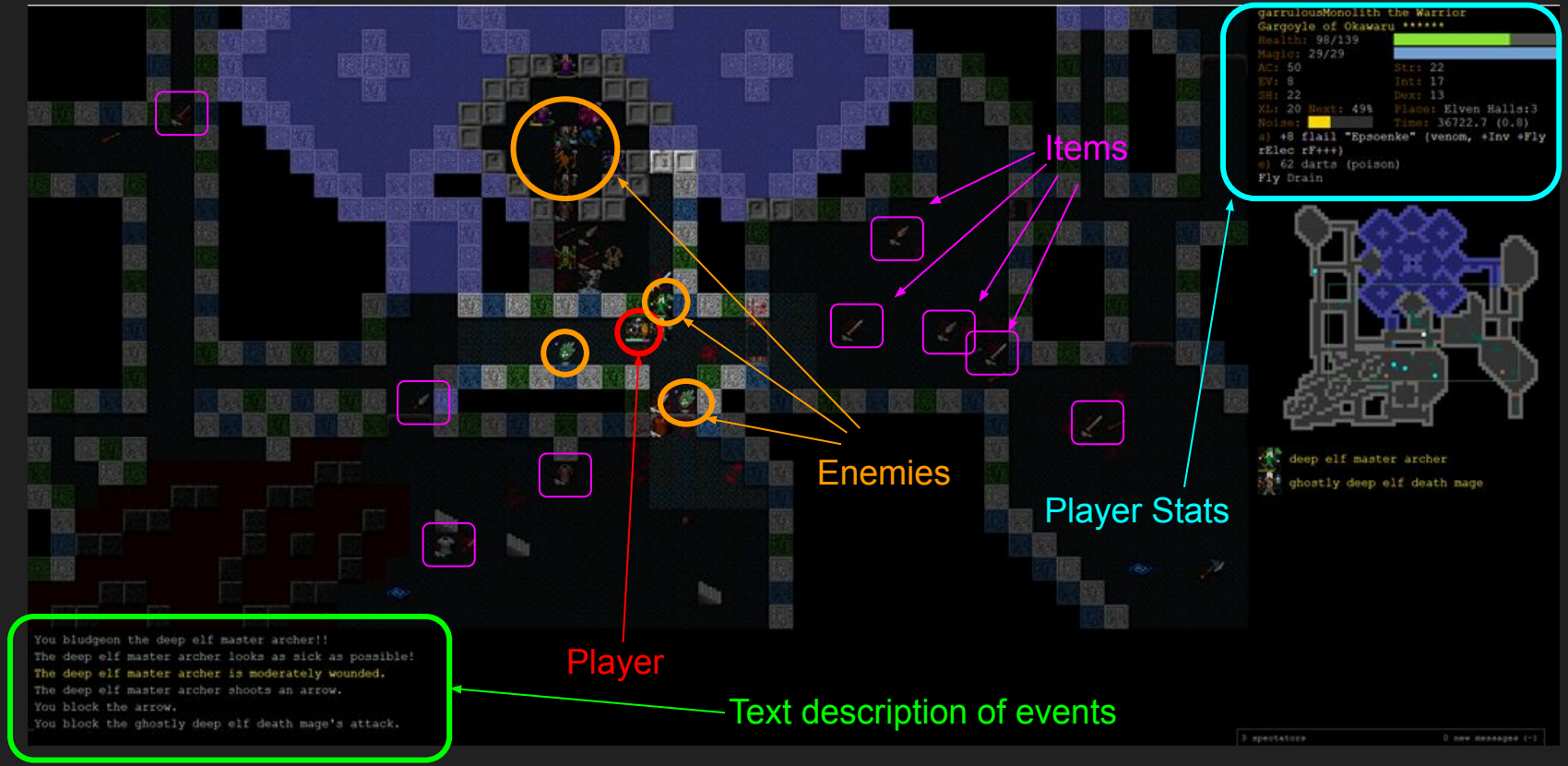
Open-source, available on Github: https://github.com/crawl/crawl

Often referred to as Crawl or DCSS

Screenshot of DCSS



Screenshot of DCSS (annotated)



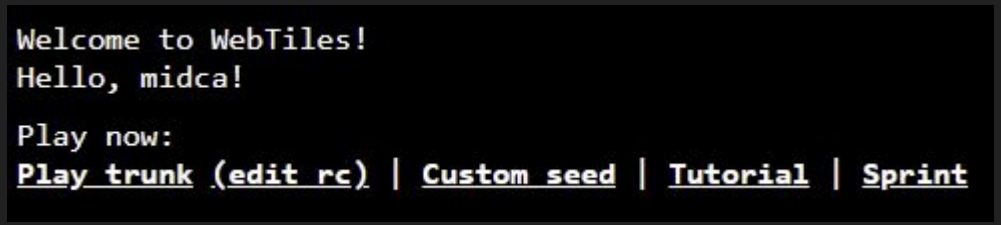
Game Modes

4 primary modes:

- Trunk (*main game mode)*
- Trunk w/ Seed
- Tutorial
- Sprint

Game Mode Selection in Desktop Client:

Browser:



Picking your starting character

Starting the Game (GIF)

Actions

Movement: Melee

To move in a direction or to attack, use the numpad (try Numlock off and on) or vi keys:

```
 7 8 9      y k u
  \|/        \|/
 4-5-6      h-.-l
  /|\        /|\
 1 2 3      b j n
```

Other Gameplay Actions: Magic

a : use special Ability (a! for help)
z : cast spell, abort without targets
Z : cast spell, no matter what
I : list all memorised spells
M : Memorise a spell from your library
t : tell allies (tt to shout)

Rest: Do-nothing

. : wait a turn (also s, Del)
5 : rest and long wait; stops when Health or Magic become full or something is detected. If Health and Magic are already full, stops when 100 turns over (numpad-5)

Dungeon Interaction and Information:

O/C : Open/Close door
</> : use staircase
; : examine occupied tile and pickup part of a single stack
x : eXamine surroundings/targets
X : eXamine level map (X? for help)

Item Interaction:

{ : inscribe item
c : Chop up a corpse on floor
e : Eat food (tries floor first)
f : Fire next appropriate item
F : select an item and Fire it
Q : select item slot to be Quivered
q : Quaff a potion
r : Read a scroll (or book on floor)
w : Wield an item (- for none)
' : wield item a, or switch to b (use = to assign slots)
v : eVoke power of wielded item
V : eVoke wand and miscellaneous item
W/T : Wear or Take off armour
P/R : Put on or Remove jewellery

Sensing Actions

Everything in the game has a text description.

Sometimes additional information too.

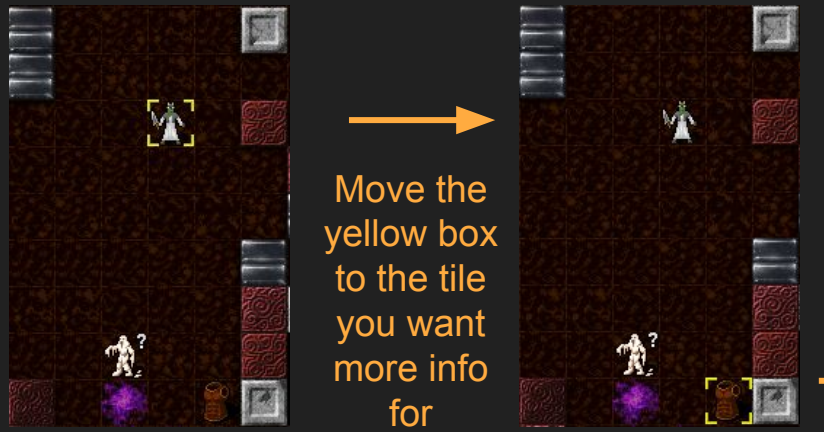
Pressing x switches to examine mode

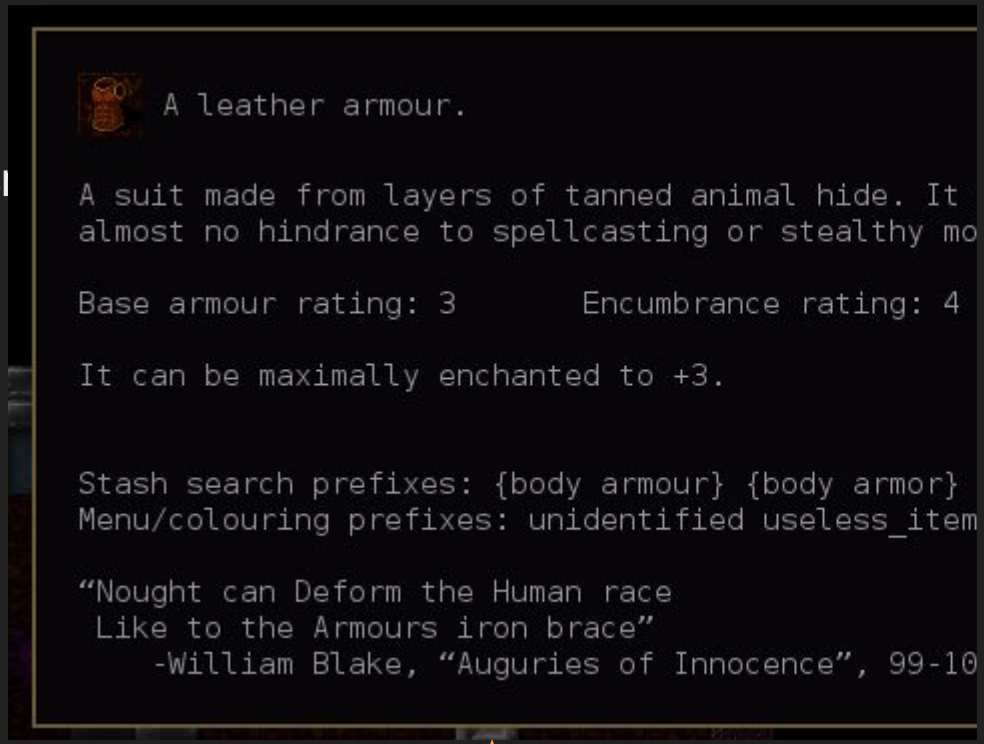
Sensing Actions

Everything in the game has a text desc

Sometimes additional information too.

Pressing x switches to examine mode

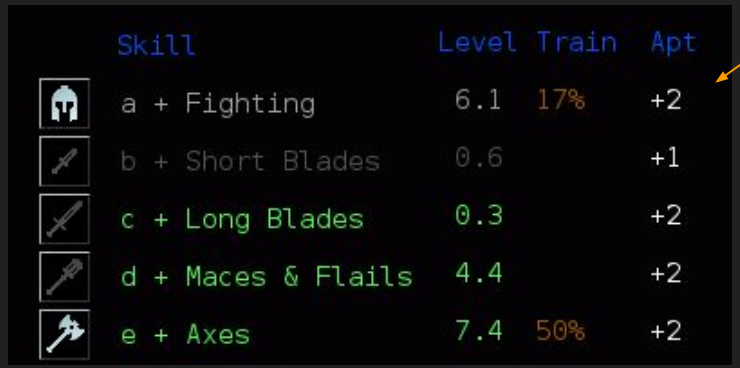




Leveling up in DCSS - Skill Points

Defeating an enemy provides experience

Wiki: “Allocating your XP to align with your aptitudes, available equipment, and playstyle will make or break your character.”



Characters have 33 Skills

Inventory Management

52 Slots in Inventory; Permanent vs. consumable items

Combat

Primary activity of the game

Many different play styles

- Melee
  - Stealthy backstabber
  - Antimagic berserker worshipping Trog
  - Ranged character
- Spells
  - Situational magic: traps and clouds
  - Transformations - become a dragon, etc.
  - Summoning - control a horde of zombies, orcs
- Hybrid
  - Battlemage

A Difficult Combat Situation

Situation

Character stats and inventory

Solutions recommended by other players:

- Use a Phantom's Mirror
  - Must have in inventory
  - Requires sufficient evocations skill
- Use a Scroll of Summoning
  - Must have in inventory
- Use a Wand of Scattershot
  - Must have in inventory
- Fight directly if AC > 30 and mid-tier weapon
- Summoning Scroll + Immolation Scroll
- Invis potion + run north + wall jump boost to slow "Lernie"
- Run away?
  - Are you fast enough?
  - How close are the nearest stairs?

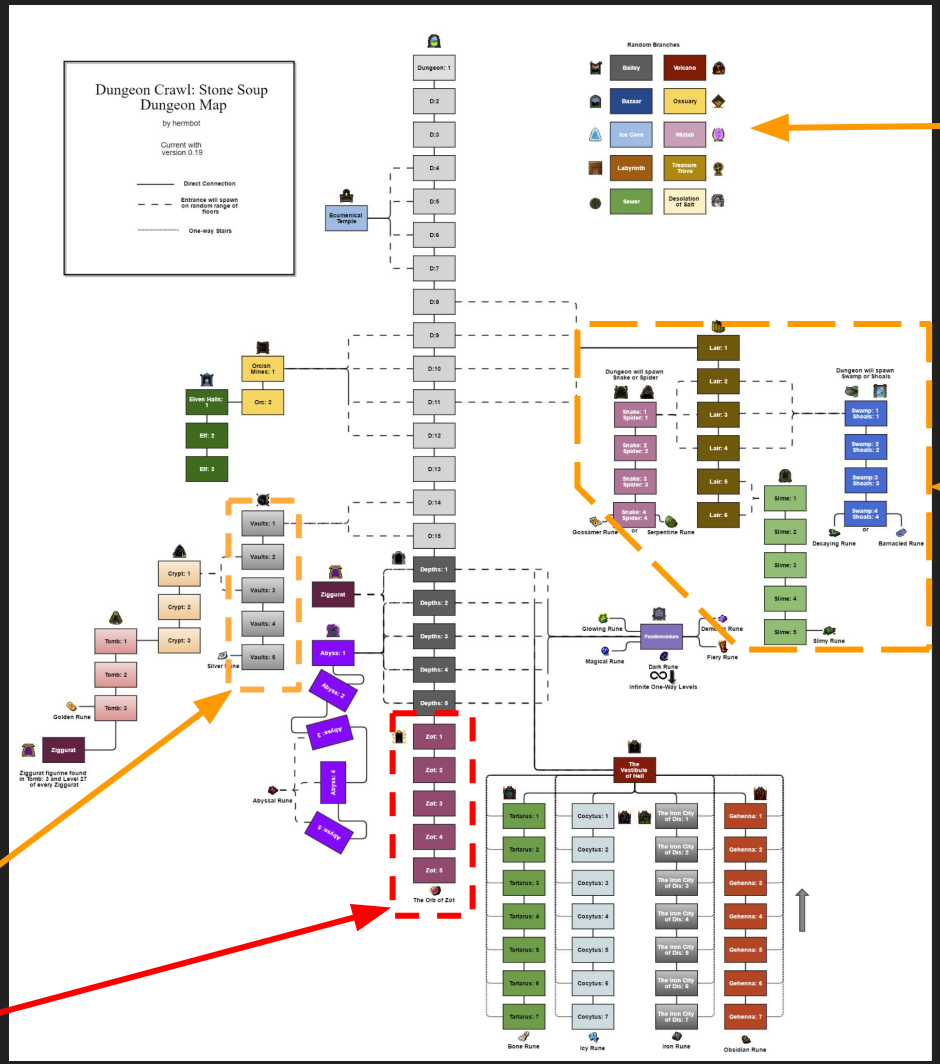
https://www.reddit.com/r/dcss/comments/o79cfm/need_help_fofi_in_swamp_with_lernie/

Dungeon Map

- More than 100 Levels
- 17 possible runes
- 3 runes needed to enter Zot

2 infinite realms:

- Pandemonium
- Abyss



8 Random Branches

Easiest runes

Easier rune

Orb of Zot

DCSS as an AI Challenge Problem

Features of Dungeon Crawl Stone Soup

- Stochastic Actions and Events
  - Ex: melee attacks may miss, hit for partial health, or kill a monster
- Non State Changing Sensing Actions
- Partially observable, unknown until explored
  - Line of sight is 7 tiles in all directions
- Procedurally Generated World
  - Unique every playthrough except if seed value specified
- Complex State Space
  - Minimum requirements to win include visiting ~70k tiles, thousands of monsters and items
  - 2 levels are infinite, change with every turn (abyss, labyrinth)
  - 650+ types of monsters
- Complex Action Space
  - More than 100 spell and melee action types, many actions requiring choosing targets
- Permanent Death

State Space Complexity - Lower Bound, 3-rune Game

Assumptions:

- 70,000 Tiles
- 900 Items
- 2000 Monsters

$$|S| = 70000^{2900} \approx 10^{14000}$$

DCSS has an infinite number of tiles, although visiting the infinite realms is not required to complete the game

Action Space Complexity

~50 action types

- 24 directional move / attack actions
- 2 macros for moving and fighting (o, tab)
- 2 resting actions
- 18 item actions (firing a missile, equip armour, evoke)
- 4 dungeon interaction actions (doors, stairs)

Players can learn up to 21 spells; carry multiple ranged weapons

Most spells and ranged weapons can target any creature in Line of Sight (7 tiles)

- Therefore up to 225 targets for a spell, including player tile

Result - upper bound could be ~30 types of ranged attacks… yielding 6750 possible instantiated actions

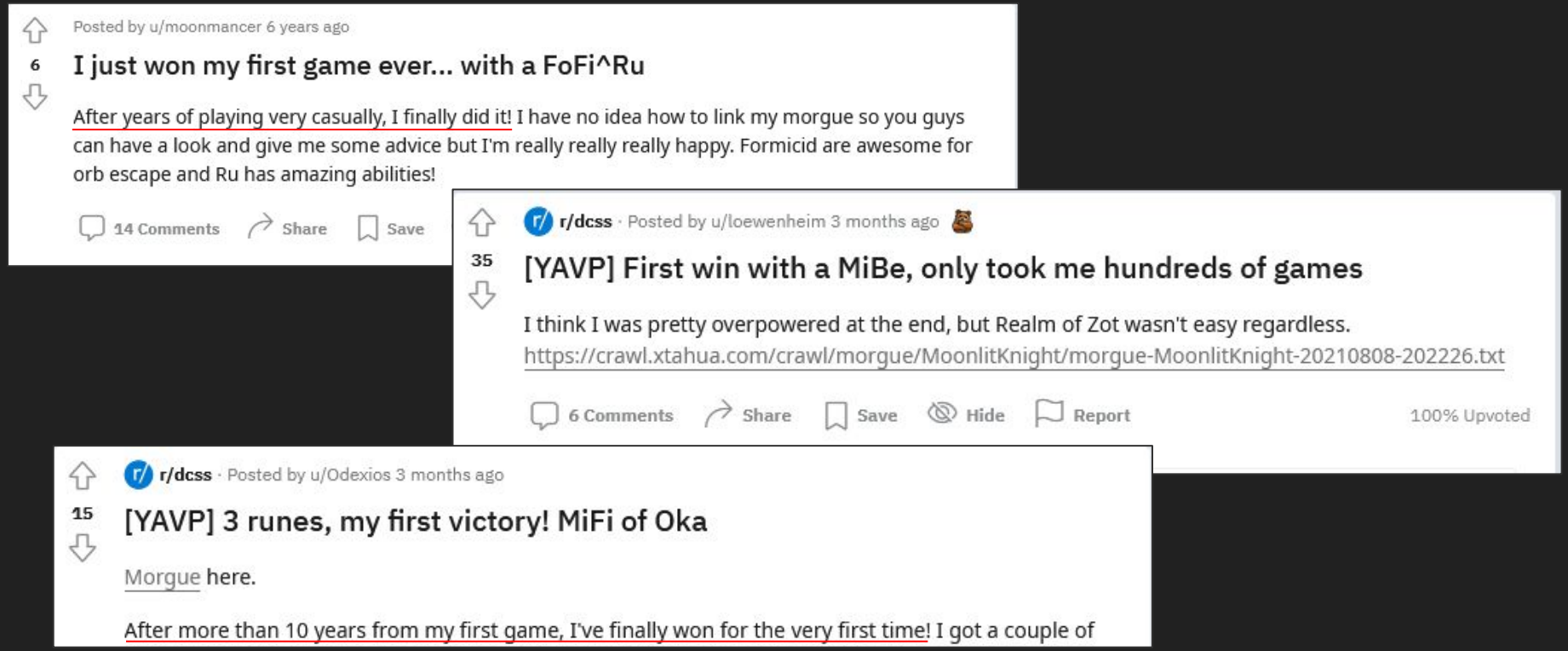
Features of a Good AI Challenge Problem

- Challenging
- Popular
- Available
- Easy to start
- Supported and maintained
- Publishable

Rogue-like Games as an AI Challenge Problem

Property	Dungeon Crawl Stone Soup	Nethack
Challenging	Currently Humans > AI	Currently Humans > AI
Popular	Active subreddit, IRC, wiki, discord	Active community
Available	AI API: ***dcss-ai-wrapper***	AI API: ***Nethack Learning Env.***
Easy to Start	Yes - tutorials available	Yes - tutorials available
Supported and maintained	Yes - under development (me) ***Open invitation to contribute!***	Yes - supported by Facebook AI Research
Publishable	Yes	Yes

Additional Benefit: It's hard for most humans



Features of a Good **Goal Reasoning** Challenge Problem

- Multiple Types of Goals
  - Achievement goals
  - Maintenance goals
  - Learning goals
- Goals over varying time horizons
  - Near term and long term goals
- Rich goal space
  - Many specific state features are desirable as goals
    - Not just a short list of goal types that a human provides (such as navigation)
- Pursuing more than one goal at a time improves performance

Achievement Goals in DCSS

A non-exhaustive list:

- Exit the dungeon with the orb of Zot (Highest goal)
- Obtain a rune (direct pre-req. to get orb)
- Acquire strength in:
  - Player stats / skills / weapons & armour
- Defeat a monster
- Acquire items
- Navigate to a tile

Maintenance Goals in DCSS

A non-exhaustive list:

- Keep health points high
- Keep magic points high
- Keep other player stats high
- Piety
- Stealth

Knowledge to Learn in DCSS

State Transitions:

- Player’s actions
  - Only living monsters will follow you up stairs
- Events
  - Fire spells over water may produce steam
- Processes
  - The rate at which health regenerates
  - Damage from actions change as you get stronger

Strategy:

- Bees travel in groups, most characters cannot outrun bees
- Bladed weapons against hydras will make them stronger unless they are branded with fire
- It is often easier to fight many enemies in a hallway 1v1 than in large open areas

Fighting in a hallway (GIF)

Winning DCSS

Some speculative thoughts

Learning as a Requirement

Intractable to human-author a complete, accurate model

- Hint: It's always changing

Any AI that's going to solve DCSS must be a learning agent

Which approach is best to solve DCSS:

Cognitive System approach? Or Deep RL End to End system?

Measuring Performance in DCSS

Well-known Metrics:

- # of wins / attempts
- Efficiency of wins
- Resources required for training (simulations, etc.)

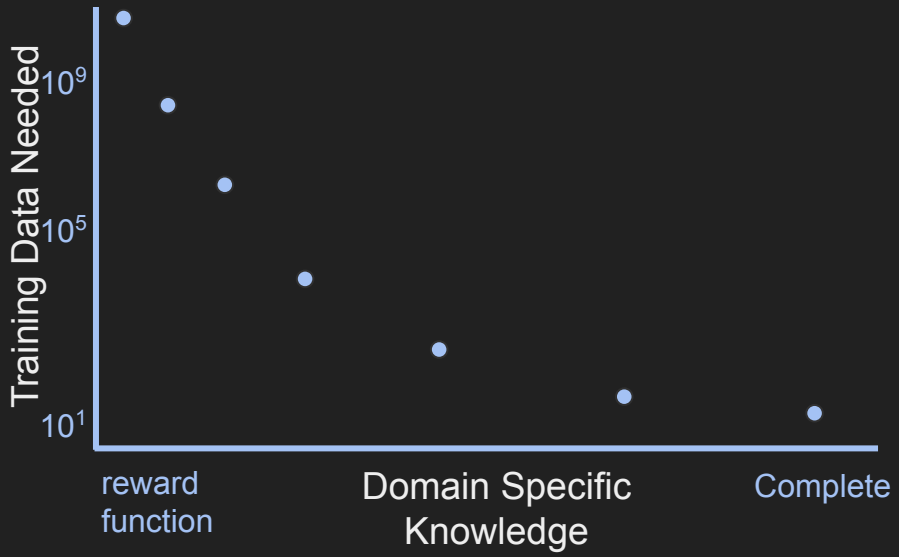
Less-obvious, critical metrics should address:

- How much knowledge is given to the agent?
- How much human time is spent encoding the knowledge to the agent?
  - *Wishful thinking?*

Measuring Human-Authored Knowledge

How much knowledge has been given to the agent?

Speculative Hypothesis:



Common phrase heard from cog. sys. folks:

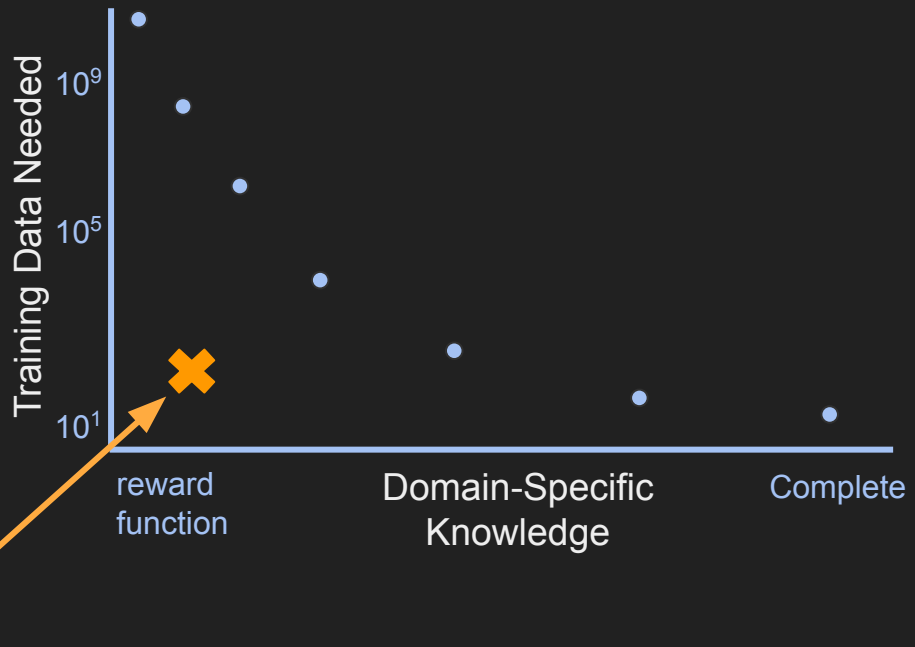
"With just a little bit of knowledge, it could learn way faster. After all, humans don't need so many examples"

DCSS-AI-Wrapper seeks to measure this precisely!

<u>Types of Knowledge to Measure:</u>

- Reward functions (incl. curriculums)
- Action / Event / Process Models
  - Varied by accuracy and completeness
- Strategy Knowledge
  - May take many forms
    - Schedules
    - Sub-goal Trees
    - Facts (ex: monster X weakness)

Bigger Goal: Can we do better?

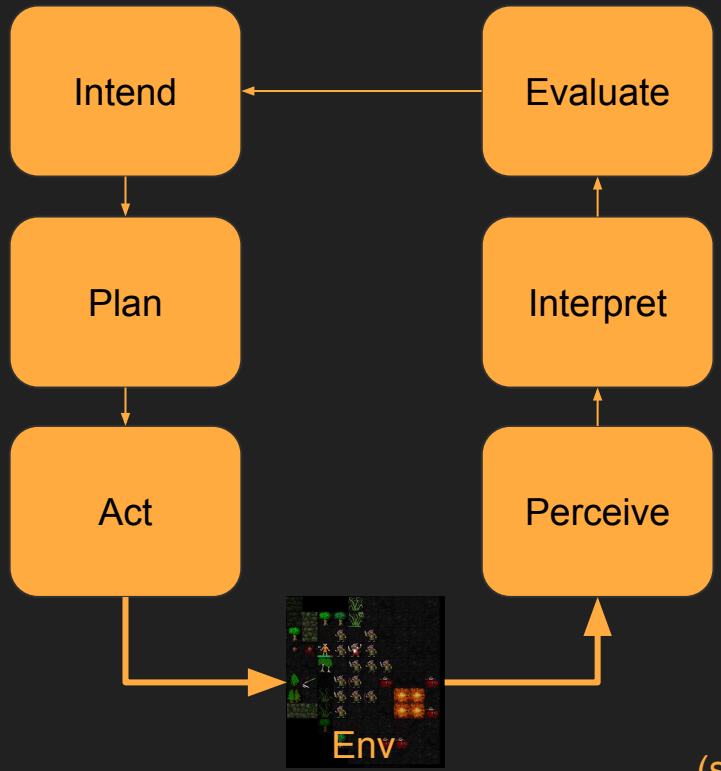


Learning as a Metacognitive Process

Cognition is Reasoning About the World

Examples:

- Beliefs and Goals are World States
- Actions change world states
- Interpret detects anomalous world states

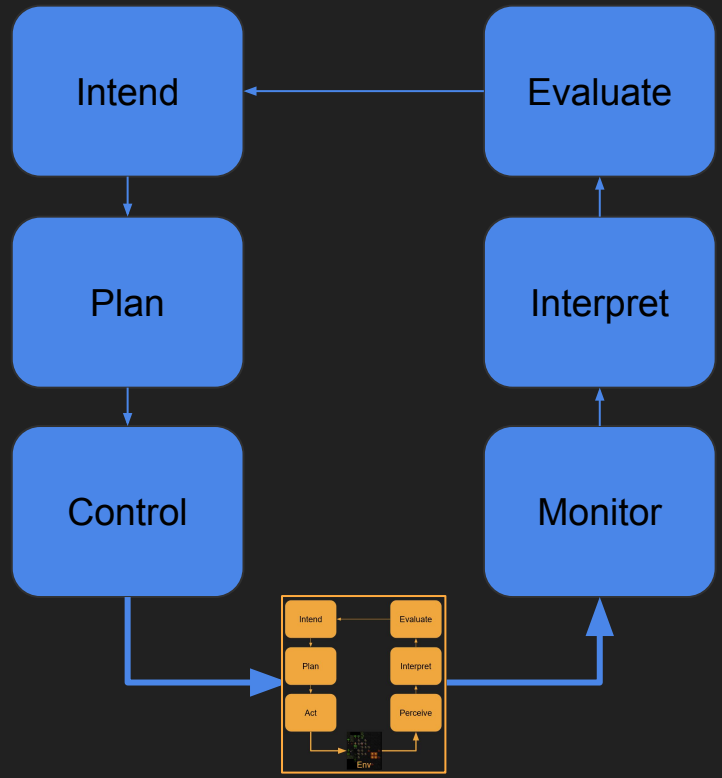


MIDCA
(see Cox, et al. 2016)

Metacognition is reasoning about Cognition

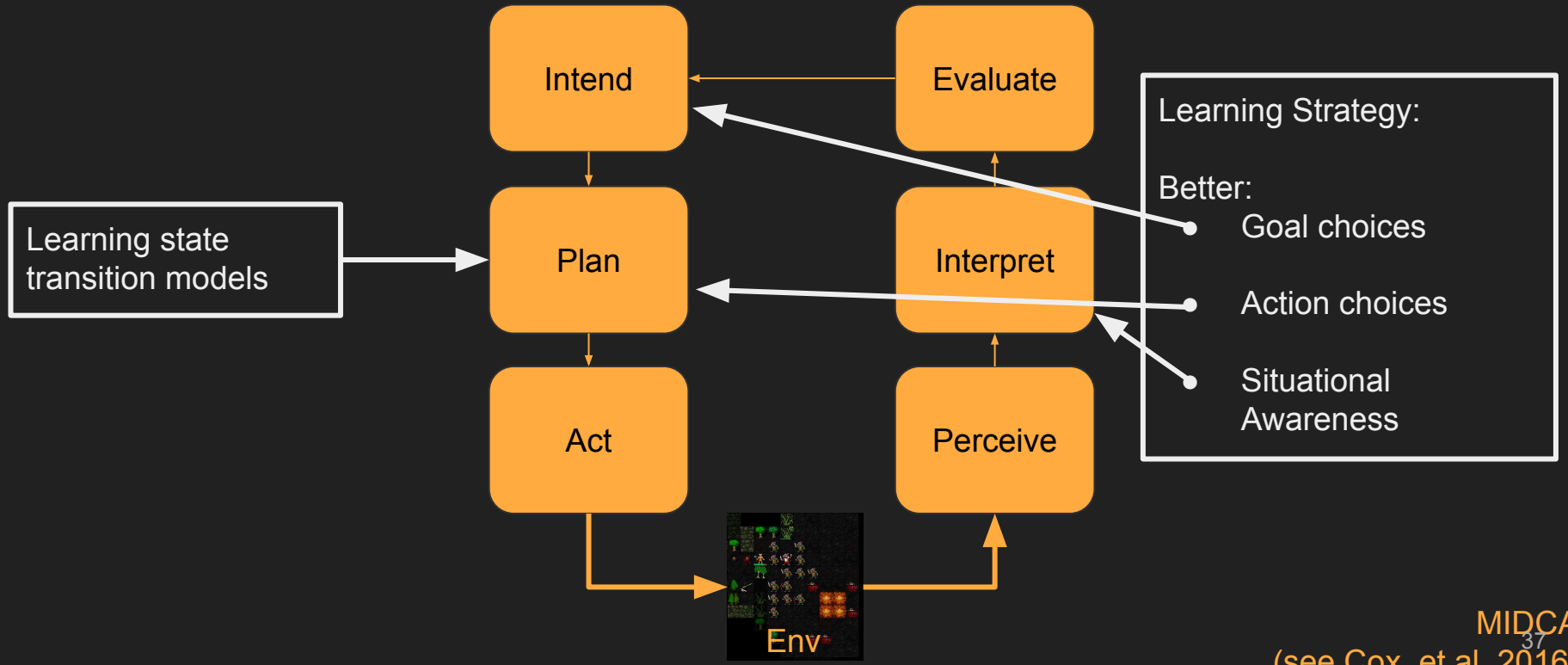
Examples:

- **Monitor** perceives the state of cognition
- **Interpret** detects anomalies in cognition
  - Failure to explain
  - Failure to plan
  - Failure of cyclic behaviors
- **Control** modifies cognition
  - Change parameter of an algorithm
  - Change algorithm
  - Change knowledge



MIDCA
(see Cox, et al. 2016)

Changing Knowledge in MIDCA



MIDCA
(see Cox, et al. 2016)

DCSS-AI-Wrapper

DCSS-AI-Wrapper

An AI-friendly API to DCSS

Goals:

- Make it easy for AI researchers to use DCSS as a test environment
- Preserve the DCSS game engine

Motivation:

- Push the field to use more complex domains as testbeds in every-day research
  - Only as an addition, not to replace other domains!

Features of the DCSS-AI-Wrapper

- Multiple state representations
  - Currently vector-based representations and PDDL 2.2
  - Low-effort to add new representations (such as other PDDL variants)
- An approximate, high-level PDDL domain file
  - ~2100 lines, includes all types of objects, monsters, spells, items, etc. & basic actions
  - Currently works with the FastDownward planner
- Cross-platform - works on Windows, Mac, Linux
- Support for running many experiments
  - Configuration options to easily set parameters
    - Auto-restart a game
    - End a game after X actions

Code to create an agent

```python
from dcss.agent.base import BaseAgent
from dcss.state.game import GameState
from dcss.actions.action import Action

class MyAgent(BaseAgent):

    def __init__(self):
        super().__init__()
        self.gamestate = None

    def get_action(self, gamestate: GameState):
        self.gamestate = gamestate
        # get all possible actions
        actions = Action.get_all_move_commands()
        # call your planner or policy instead of random:
        return random.choice(actions)
```

Only Requirements:

1. Subclass BaseAgent
2. Override get_action()

Code to run an agent

```
1   from dcss.websockgame import WebSockGame
2   from dcss.connection.config import WebserverConfig
3
4   def main():
5       my_config = WebserverConfig
6
7       # set game mode to Tutorial #1
8       my_config.game_id = 'tut-web-trunk'
9       my_config.tutorial_number = 1
10
11      # create game
12      game = WebSockGame(config=my_config,
                           agent_class=MyAgent)
13      game.run()
```

Option to set configuration values when running

Set your agent here

game.run() is a blocking call

Planning Agent Demo

Hallmark of DCSS-AI-Wrapper: Both Vectors & PDDL

Vector Based Representation

- Player stats (vector)
- Player inventory (vector)
- Player spells (vector)
- Player abilities (vector)
- Player skills (vector)
- Map data Line-of-Sight (LOS) (vector)
- Map data current level (vector)
- Map data all (vector)

PDDL State Representation

- Player stats (PDDL)
- Player inventory (PDDL)
- Player skills (PDDL)
- Map data Line-of-Sight (LOS) (PDDL)
- Map data current level (PDDL)
- Map data all (PDDL)
- Static Background Knowledge (PDDL)

PDDL Model of the game (~2k lines):

- Works with the FastDownward planner
- In the repo: **models/fastdownward_simple.pddl**

https://dcss-ai-wrapper.readthedocs.io/en/latest/usage/state_representation.html

Hallmark of DCSS-AI-Wrapper: Both Vectors & PDDL

Vector Based Representation

- Player stats (vector)
- Player inventory (vector)
- Player spells (vector)
- Player abilities (vector)
- Player skills (vector)
- Map data Line-of-Sight (LOS) (vector)
- Map data current level (vector)
- Map data all (vector)

The map data LOS vector is comprised of 225 tile vectors, with each tile vector length 30

(total vector size being 6750)

PDDL State Representation

- Player stats (PDDL)
- Player inventory (PDDL)
- Player skills (PDDL)
- Map data Line-of-Sight (LOS) (PDDL)
- Map data current level (PDDL)
- Map data all (PDDL)
- Static Background Knowledge (PDDL)

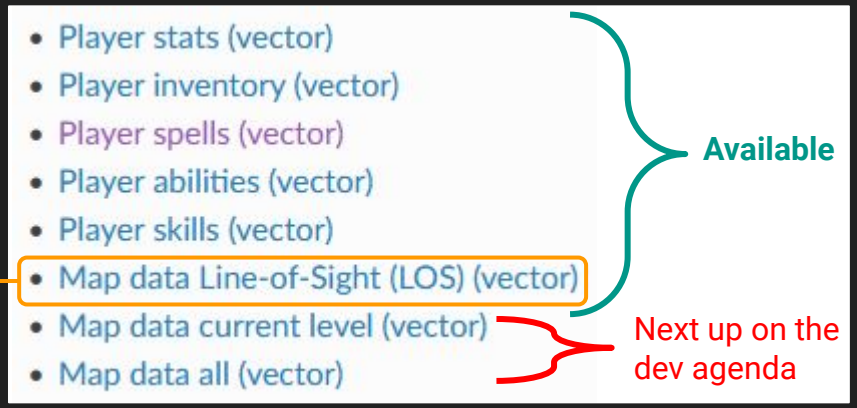
PDDL Model of the game (~2k lines):

- Works with the FastDownward planner
- In the repo: **models/fastdownward_simple.pddl**

https://dcss-ai-wrapper.readthedocs.io/en/latest/usage/state_representation.html

Hallmark of DCSS-AI-Wrapper: Both Vectors & PDDL

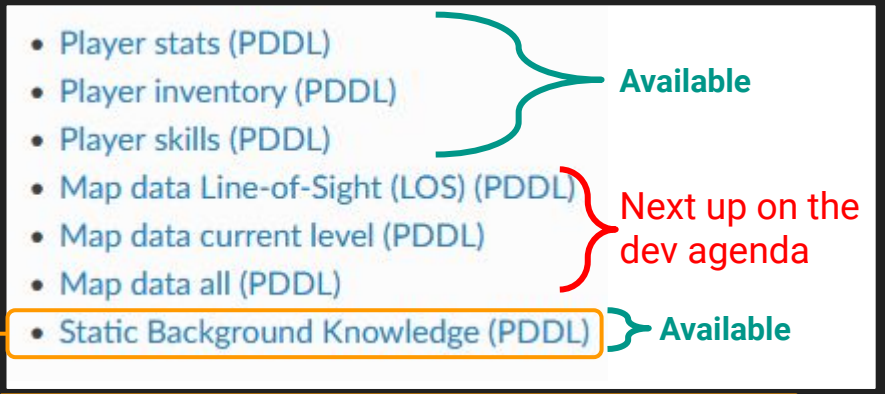
Vector Based Representation



The map data LOS vector is comprised of 225 tile vectors, with each tile vector length 30

(total vector size being 6750)

PDDL State Representation

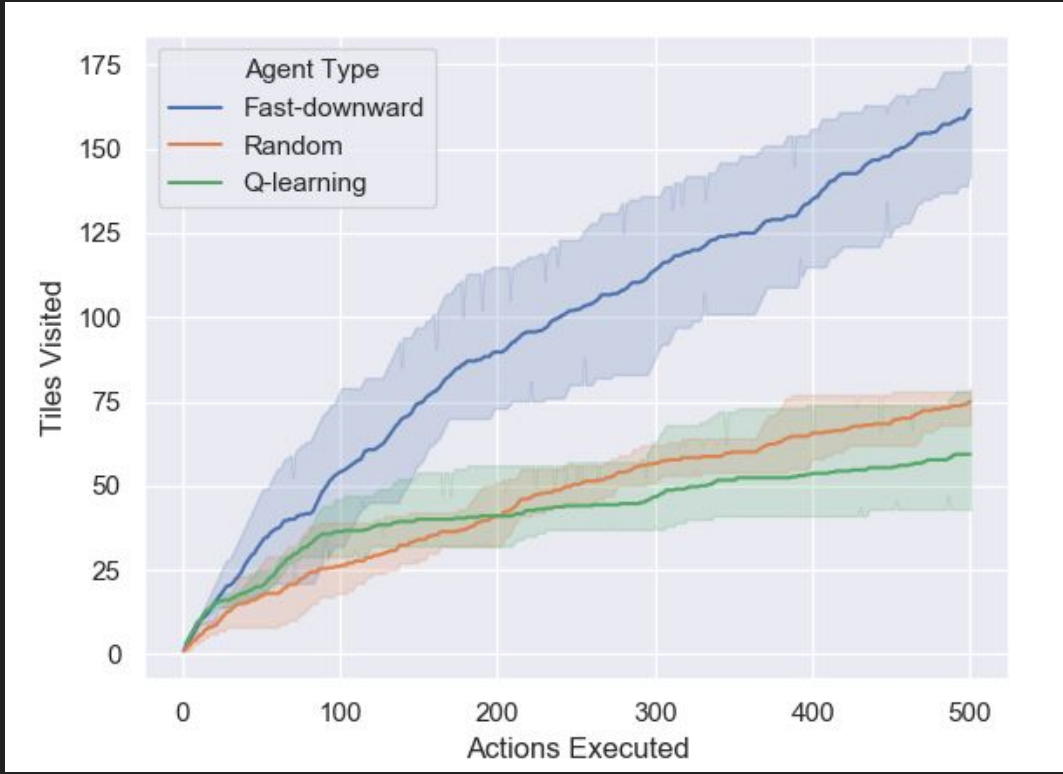


PDDL Model of the game (~2k lines):

- Works with the FastDownward planner
- In the repo: **models/fastdownward_simple.pddl**

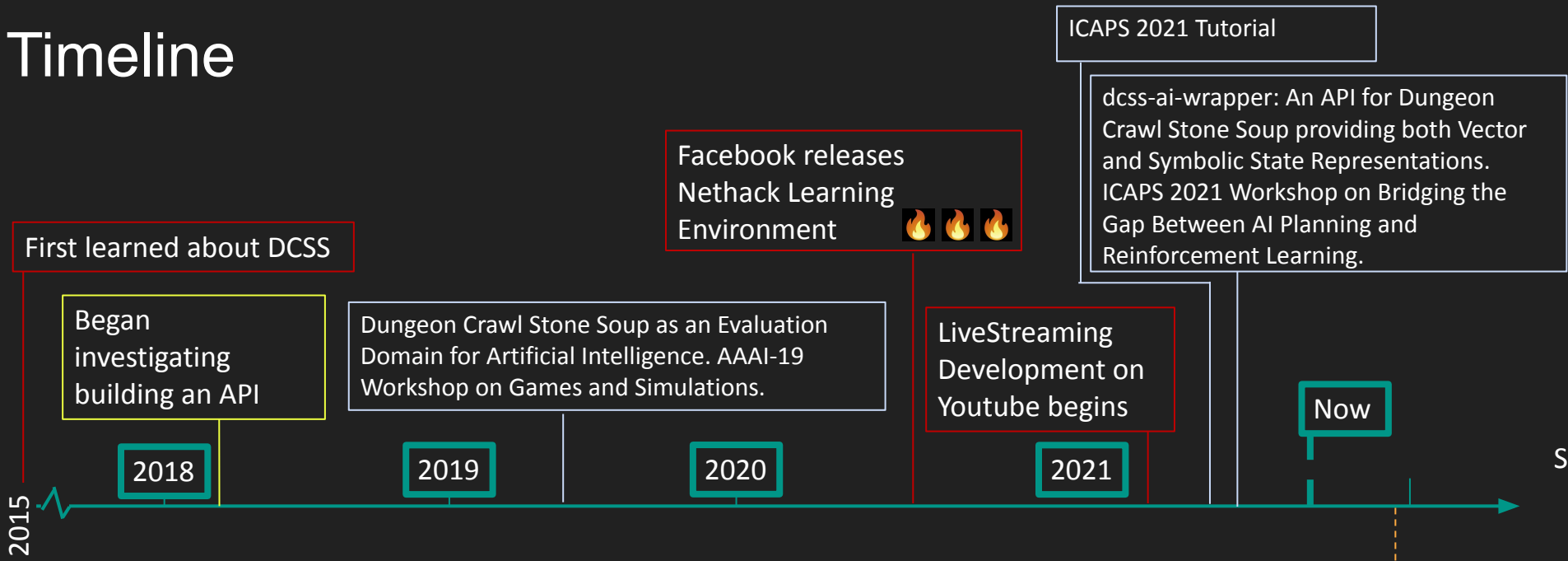
https://dcss-ai-wrapper.readthedocs.io/en/latest/usage/state_representation.html

Comparing Diverse Agents



Data collected from agents using the dcss-ai-wrapper API

Timeline



Near-term Goals

- **Add monster details to API**
- **Finish PDDL API**
- **Optimize terminal mode**
- **Refactoring & polishing**
- **… agent development??**

Getting Started

Code:

Github: dcss-ai-wrapper & Documentation

Resources:

Recent ICAPS Tutorial Documentation

ICAPS Tutorial Youtube Video

Join the conversation and ask questions at the Gitter

Thank you to my collaborators

Michael W. Floyd @ *Knexus Research Corporation*

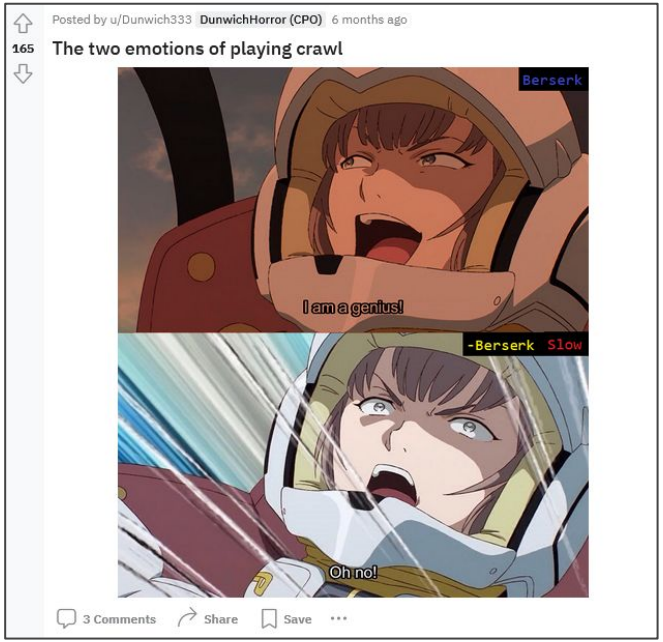
Zohreh A. Dannenhauer @ *Knexus Research Corporation*

Adam Amos-Binks @ *Applied Research Associates*

Jonathan Decker @ *Naval Research Laboratory*

David Aha @ *Naval Research Laboratory*

Noah Reifsnyder @ *Lehigh University*



End