AI for Strategy Games (SG) and Esports Analytics (EA) Workshop Report

Derek Martin, Michael Buro, Ben Watson, Erica Kleinman

https://skatgame.net/mburo/aiide22ws

Workshop Goals

AI for Strategic Games (SG) [organized by Derek Martin and Michael Buro]

- Bring together AI researchers and game AI programmers from industry, who are interested in strategic game AI
- Present and exchange ideas on the subject
- Discuss how academia and game companies can work together to improve the state-of-the-art in AI for games

Esports Analytics (EA) [organized by Ben Watson and Erica Kleinman]

- Begin to address the gap between esports' popularity and esports' research scarcity
- Attracting research surrounding the use of esports data

Hybrid Format - Oh My!

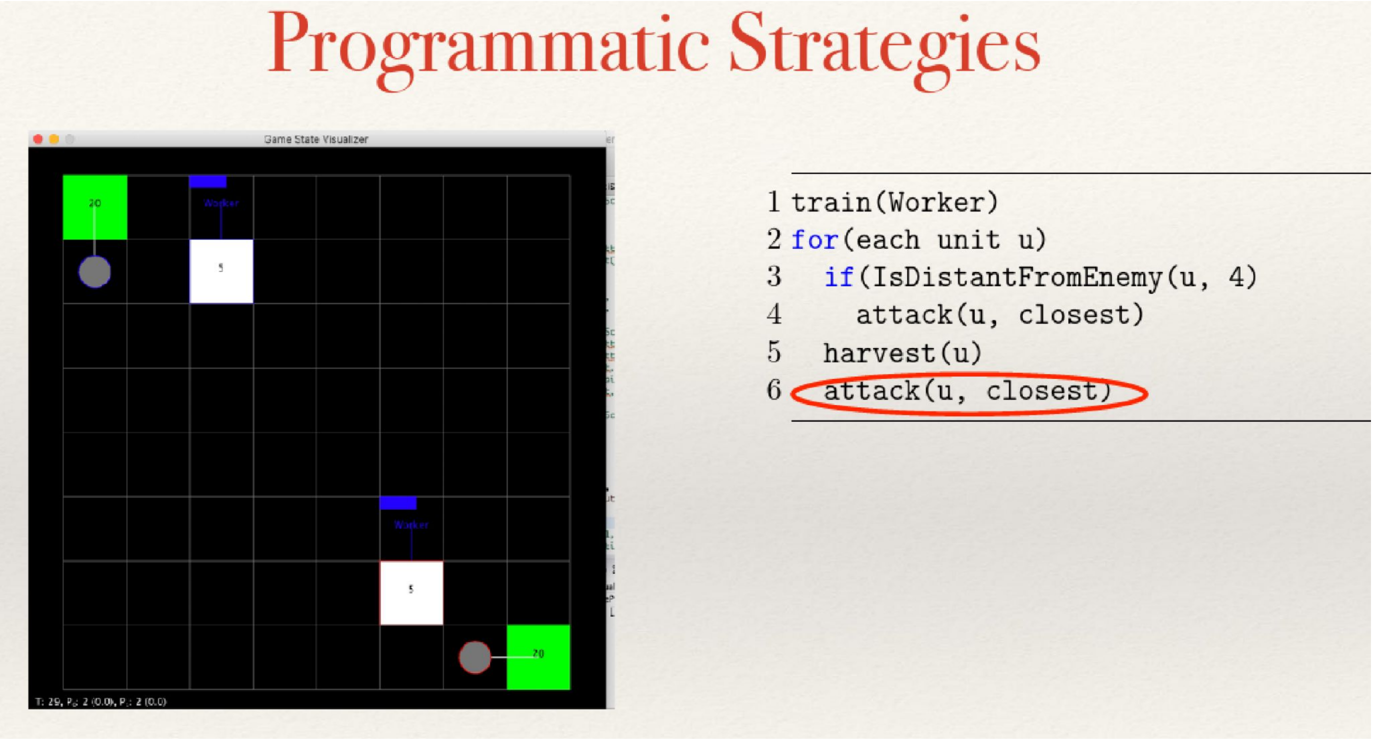
- 26 workshop attendees (combined)
- Used google meet room and projection in conference room
- The original schedule contained sufficient breaks to compensate for A/V setup and LA traffic delays
- All worked out fine after shuffling presentations and shortening breaks

Combined SG+EA Program (Full-Day)

- Accepted Paper Presentations (3 x SG)
- Invited Industry Talk (1 x EA)
- Show-and-Tell Sessions (5 x SG, 1 x EA)
- StarCraft AI Competition Report
- Group Discussion

Theme 1: Programmatic Strategies

How to search in program space to improve playing strength?



Programmatic Strategies

```
train(Worker)
for(each unit u)
  if(IsDistantFromEnemy(u, 4)
    attack(u, closest)
  harvest(u)
  attack(u, closest)
```

“Choosing Your Opponents Well: How to Guide Synthesis of Programmatic Strategies”

Rubens Moares [Show-and-Tell (SG)]

- How can we synthesize strategies to find a dominant strategy against an opponent ?
- Use domain specific programming language and Iterated-Best-Response algorithms
- This is difficult if the search space is large
- Compared three algorithms for learning how to navigate through the strategy search space
    - Fictitious Play (FP)
    - Double Oracle (DO)
    - Proposed Neighborhood Curriculum (NC) greedily selects from groups of strategies to guide the search
- NC is better in finding counter strategies in micro-RTS than FP and DO

“Show Me the Way! Hierarchical Search for Synthesizing Programmatic Strategies”

Levi Lelis [Show-and-Tell (SG)]

- Presented a two-level approach to synthesize programmatic strategies
- The top-level search tries to optimize a feature vector (e.g., unit counts to achieve) which is passed on to the second level - strategy search - for breaking ties
- Also discussed how to apply transfer learning for improving the feature learning
- A micro-RTS tournament with previous competition winners indicated that combining both methods works well
- Also discussed surprising program instances that perform well in unexpected ways

“Synthesizing Chess Tactics from Player Games”

Abhijeet Krishnan [Accepted Late Paper (SG)]

- Motivation: Humans can learn better strategies by studying experts (e.g., super human AI)
- Methodology
    - Trained an agent to learn chess tactics using Popper, an inductive logic programming (ILP) system
    - Encoded tactics (like pins and forks) in first-order-logic (PROLOG like)
- Evaluation
    - The agents were tested on a collection of human chess games
    - Performance metrics: coverage, accuracy, divergence
    - Baselines: random, ground, Maia-1600, stockfish 14
- Results
    - Agents trained using Maia-1600 (average) performed worse than the Random agent
    - Agents trained using Stockfish 14 (very strong) performed better, but still a lot of had divergence
- Conclusion
    - Formulated tactic synthesis the problem
    - Showed promising initial results
- Future work
    - Learn tactics based on player skill and multi-step tactics
    - Check interpretability with user study

Theme 2:
Esports Analytics

How can we improve human game playing experiences by analysing game data?

“The Consumer Side of Esports Analytics”

Amine Issa (Mobalytics) [Invited Presentation (EA)]

- Company motivated by what makes people good at what they do
- Mobalytics provides personalized recommendations to players based on their previous performances or the current game meta
- Uses 500 different metrics (some combined to create other metrics) to create a GPI (game performance index)
- Worked with multiple professional teams to help design the metrics/performance evaluation model
- Ran a study with ~2M players who used Mobalytics and ~2M who did not
  - Found that people who used their app climbed ranks faster than those who did not
- Mobalytics also provides means for players to just play using the best/current meta strategies or best builds for winning without personalized instructions

- Conclusion
  - Understanding how people play games helps us to learn more about humans
  - Players learning about themselves and understanding how they play games can help them recognize their strengths and weaknesses, and improve faster

“Learning to Spectate Games for Esports using Object Detection Mechanism”

H.-T. Joo, S.-H. Lee, C.-M. Bae, K.-J. Kim [Show-and-Tell (EA)]

Problem

- Automatic observer acts as camera man for the crowd and tries to show the most exciting scenes
- Existing methods for automatic observers are not effective because they are event-based or rule-based and require expert domain knowledge

Proposed Approach

- Collect and preprocess replays of human observational data
- Train a Mask-R-CNN for their automatic observer, using the replays
- Evaluate the models by the amount of overlap between target data (human data) and the model’s predictions

Conclusions and Limitations

- Their approach and other Mask-R-CNN algorithms best predicted interesting locations in the game world
- Behavior cloning and rule-based algorithms performed the worst
- It may be difficult for their approach to directly replace human observers, but it can help as an aid to commentators

Theme 3:
AI for Strategic Games

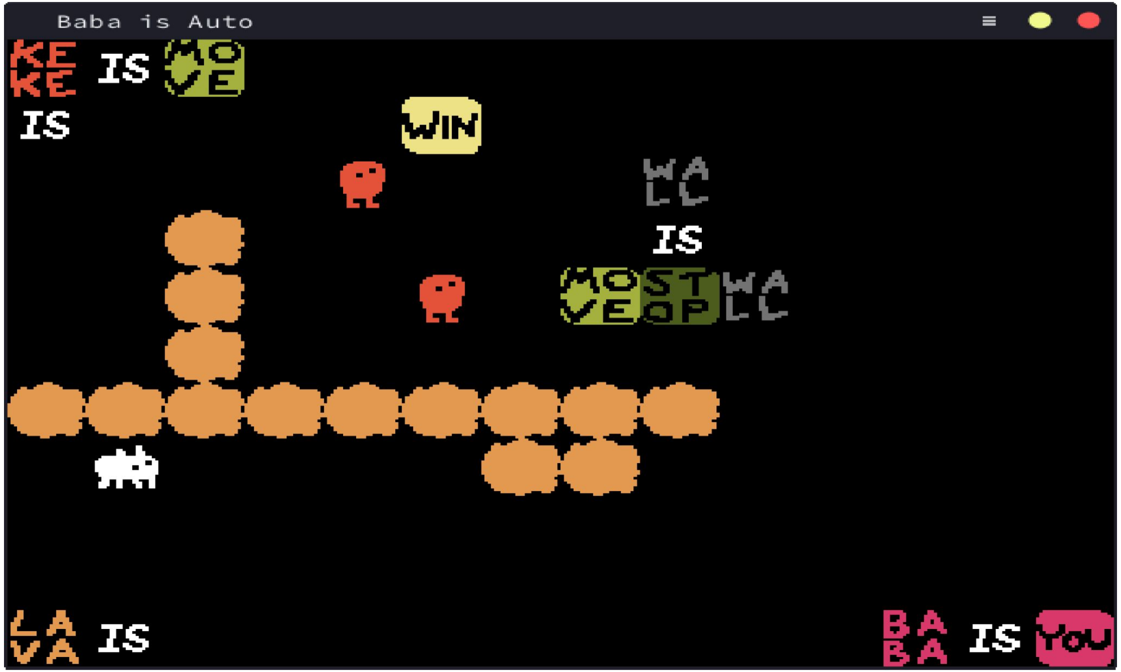
How to make programs stronger game players?

“Baba is Smart”

J. Ahn, **M. Korolkov, S. Madineni,** J. Waidhofer, **R. Canaan** [Accepted Paper (SG)]

- “Baba Is You” is a Sokoban-style game and it's an interesting environment because game rules can be changed
- Contributions
  - Map Conversion for the “Baba is Auto” RL environment
  - Three agents for solving puzzles (IDA* [Search], SAC [RL], A2C [RL)])
  - Level Difficulty Calculation (# of rules needed to solve the level and openness of the map)
  - Modified the state observation to make it more descriptive for RL agents to learn
- Evaluated the agents by testing them on levels from the Keke Competition
  - IDA* and Random were tested on 225 levels from the Keke AI Competition
  - RL agents were trained on 70% of the levels and tested on the other 30%
- Results and Future Work
  - IDA* was the most successful agent and was competitive against agents in the Keke Competition
  - RL agents need to be goal-driven to perform better
  - Add custom rules to IDA* to prompt the agent to form new rules to solve harder levels

Baba is You


Baba is Auto
KEKE
IS
MOVE
IS
WIN
WALL
IS
MOVE
STOP
WALL
LAVA
IS
BABA
IS
YOU

“Entity-Oriented Reinforcement Learning for RTS Games”

Costa Huang [Show-and-Tell (SG)]

- Objective: Find a way we can conveniently formulate RTS games as RL environment
- Problem Statement
  - Applying Deep-RL to JSON-like game observations and actions is hard because it expects fixed-size input
  - Cannot reuse the same model for maps of different sizes
- Method
  - Use entities/objects as observations, and each entity can perform actions
  - Treat entities like NLP tokens and utilize NLP techniques (e.g., transformers)
  - Propose using ragged buffers/jagged arrays that allow the agent to run multiple simulation at once to improve speed
- Contribution
  - Apply entity-oriented RL to Gym-Micro-RTS
  - Simplifies game description for ML purposes
  - Can reuse models for maps for different sizes

“Transformers as Policies for Variable Action Environments”

Niklas Zwingenberger [Accepted Paper (SG)]

- Using standard neural networks in variable action environments is challenging
- Using transformers allows to input a feature map and output logits of the same size
- Designed a weight embedding for condensing features into a more dense representation
- Resulting in better reward maximization than Unit-Action-Simulation (UAS) and faster training than GridNet in micro-RTS
- Limitation: the self-attention head of the transformer is computationally expensive
- Future work
  - Exploring sparse win/loss rewards
  - Adding self-play to architecture
  - Improving the efficiency of the self-attention head
  - Working on agent for partial observation domains

“Strategic Local Navigation using RL in an Adversarial RTS Games Environment”

Debraj Ray [Show-and-Tell (SG)]

Motivation: How can we navigate an adversarial environment quickly and safely?

Methodology

- Combines global guidance (A*) and local planning (RL) to find shortest paths and avoid enemies
- Learning objectives: survive dynamic enemy attacks while making progress along a global/optimal path

Results

- Approach significantly lowers the number of collisions with enemies compared to baseline agent
- lower amount of memory used / nodes expanded during search
- lower runtime

Conclusions

- Approach can help players with micro-management of units for navigation
- Design changes in the game won’t affect the agents overall performance
- Can effectively scale to larger maps

Other Topics

“Towards Semi-Exhaustive Procedural Content Generation for Safari Rush Hour Puzzles”

J.-A.-Solar Zavala, N. Barriga [Show-and-Tell (SG)]

Goal: Eventually generate interesting Safari-Rush-Hour puzzle instances

- Safari-Rush-Hour is a sliding piece puzzle; the goal is to get the safari-rover off the board
- Not much work done on the game yet, but some PCG work done on Fling!

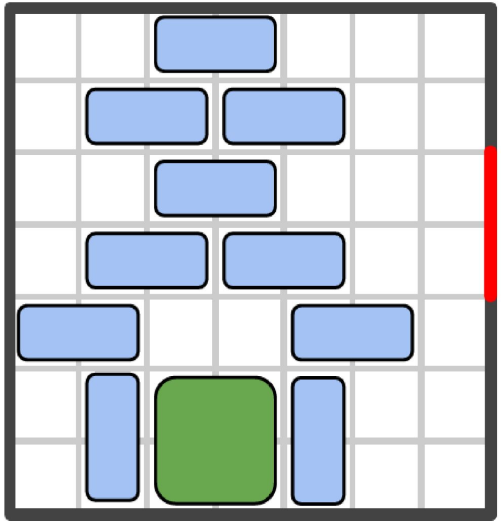
Methodology

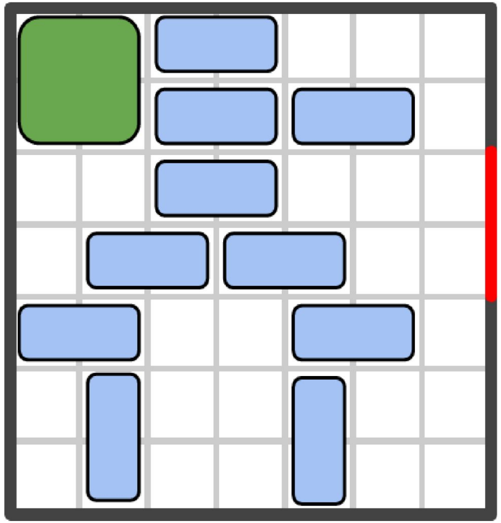
- Use search, combinatorics, and a super-computer
- Devised effective board cluster encoding scheme
- Went through some clusters and ran A* on each instance; redundant computation (repetition) => slow
- Ran BFS with shared cache to compute connected components within clusters
- Used retrograde analysis to speed up solving clusters and combine it with BFS to compute connected components

Results: Solved all instances in a few clusters, showed a sample position, and a harder variant

Future Work: Solve all instances in all clusters on a super-computer and identify interesting instances

Safari Rush Hour





“StarCraft BroodWar Competition Report”

Dave Churchill

- Details in Dave’s conference presentation

Last Session: Discussions and More Show-and-Tell

More discussions on programmatic strategies

- what the current objectives of programmatic AI are (improving AI strength, build better tutoring systems,...?)
- how it could help humans learn to play because agents will be programs that are ideally readable / interpretable
- how to make programmatic strategies more interpretable to humans
- how the language used to represent programs can help make programs more interpretable to humans

Costa Huang described a new RL library

- one file
- efficient PPO implementation - performs better than other popular implementations

Lastly, we discussed the future of the EA workshop at AIIDE, seeing potential in helping human players and AI systems to play better by analyzing large amounts of data

FIN

Looking forward to next year’s AIIDE SG and EA workshops!

Workshop URL: https://skatgame.net/mburo/aiide22ws

“StarCraft BroodWar Competition Report”

Dave Churchill

- 13th year of this competition !
- Playing the full StarCraft Broodwar Game
  - Fog of War enabled
  - Round-Robin tournament with more than 20k games played on a virtual machine cluster
- 9 Submissions
  - 2 new bots
  - 2 resubmissions (one good bot from last year, and UAlbertaBot which hasn’t changed in years)
  - No Terran bot among new submissions
- Results
  - BananaBrain won with a 85.5% win-rate, placed 2nd last year
    - Author worked on it for 5 years
    - 38k loc
    - Strategy: ranges from aggressive over defensive to greedy
    - AI: uses jump-point-search for path-finding for retreating, A*/Potential Fields for combat positioning, learning from previous encounters
  - The top four bots were repeats from last year