General Game Learning

Matthew Fisher

What is general game learning?

Learning to play games you have never seen

- Reading or listening to the rules
- Asking questions about the rules
- Learning the rules by playing (ex. Tetris)
- Figuring out the objective
- Being reprimanded for making illegal moves
- Learning from mistakes
- Deciding on good strategies
  - The focus of 99% of game AI research

Goal of this talk

Convince you that general game learning is:

- Important
- Interesting
- Possible

Why is GGL specifically interesting?

Task Automation

- Automating real tasks is a big AI goal (folding clothes, driving cars, SQL-injection attacks)
  - manually code specialized controllers for each task
  - have controllers that can learn a task from examples
- Real tasks are hard to setup and evaluate
- Digital games are plentiful, complex, accessible to a computer, and easily evaluated
- To a computer there is no difference between the real world and a good simulation

Games

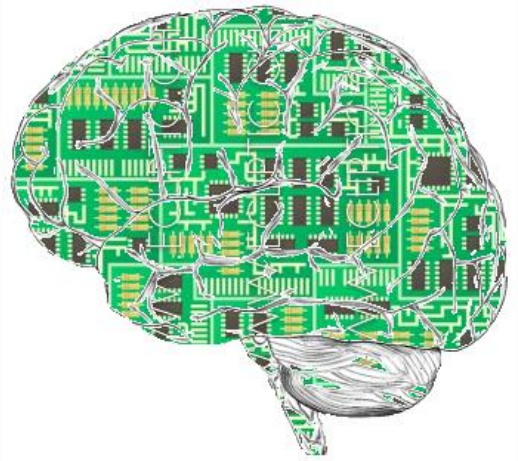
Why are games a great task?

- Games have a clear goal that can be quantified
- Games are challenging to understand
  - Games require complex reasoning
  - Games require often incomplete rule-learning
- **Digital games are fully accessible to computers**
- Games are one way humans and computers interact
- Games are fun and popular
  - Deep Blue, Watson
- **The dataset of available games is huge and untapped**

We will use digital games to study…

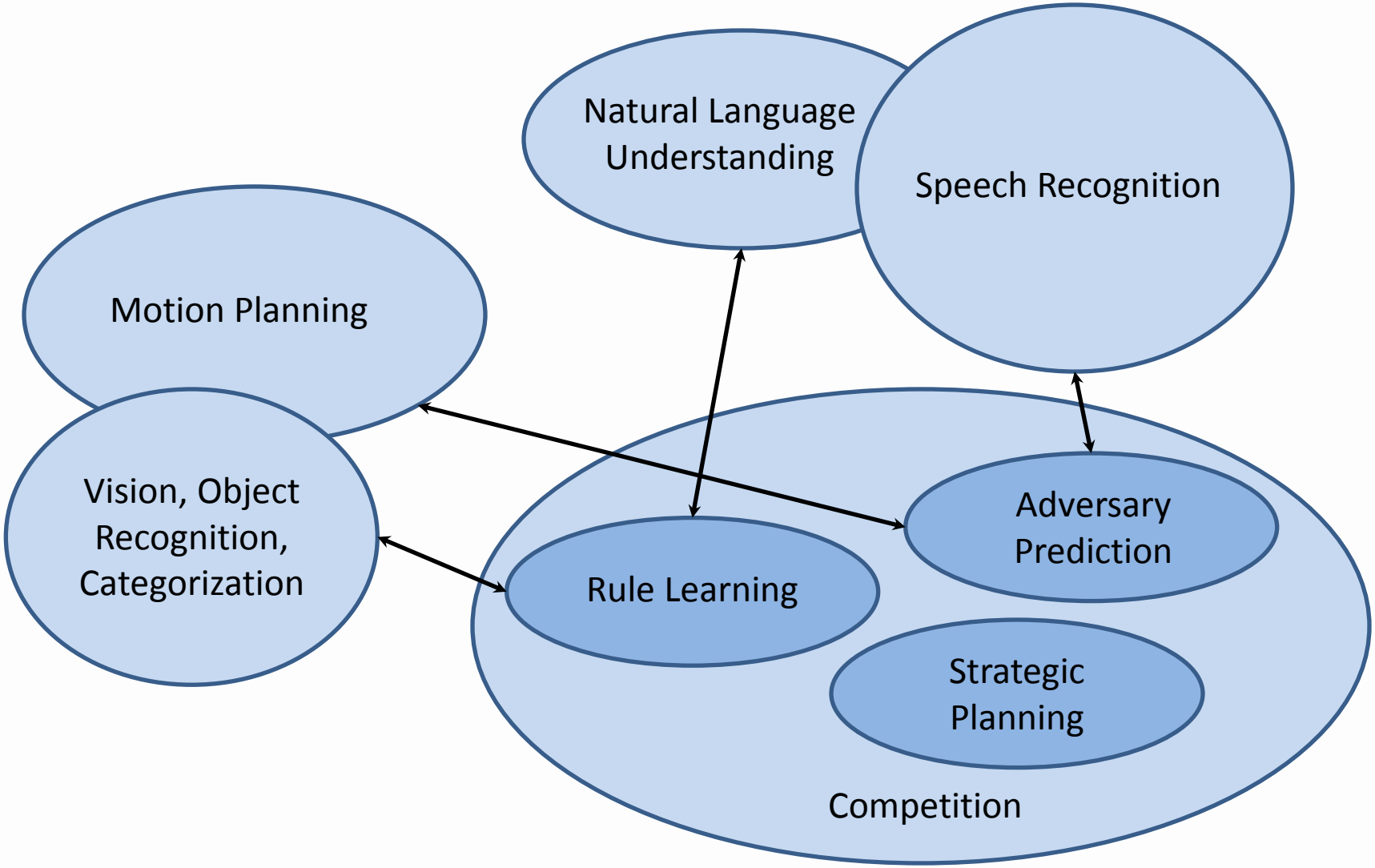
- **Unsupervised** object segmentation and categorization
- Rule learning
- Knowledge transfer between categories
  - Between Mario & Luigi
- Goal inference
- Game state value estimation
- Action planning
- Knowledge transfer between games
  - Between Super Mario 1 and Super Mario 3
  - Between Shogi and Chess

Artificial intelligence

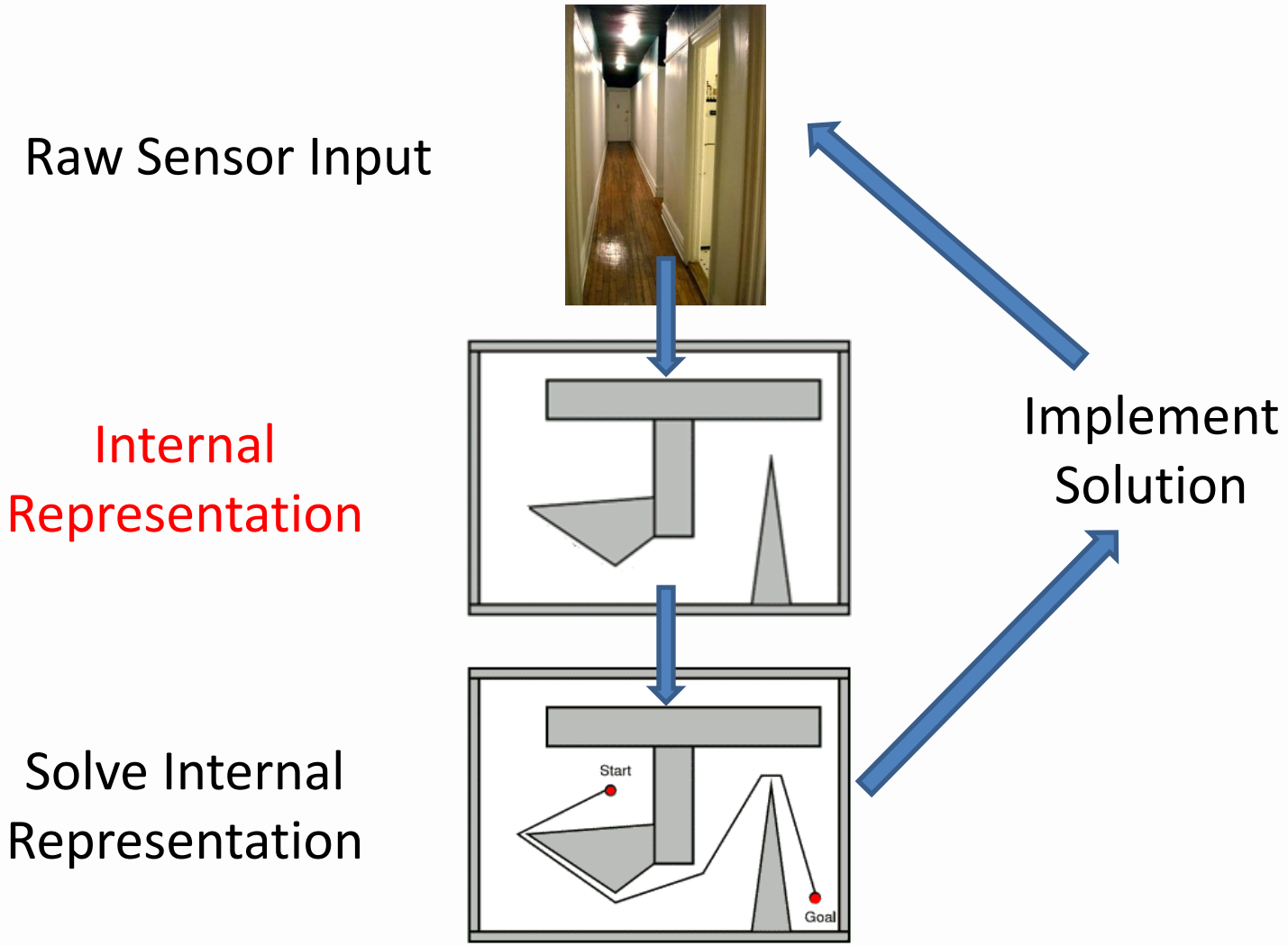


Goal is to solve problems

Problem Space (Humans)



Solution Space (Humans)



(Some) Tasks in General Learning

- Define meaningful internal representation
  - Categorize, classify, and learn from raw data
- Process raw signal
- Map raw signal onto internal representation
- Solve problem using the internal representation
- Map solution to an action and execute it

Representation is everything

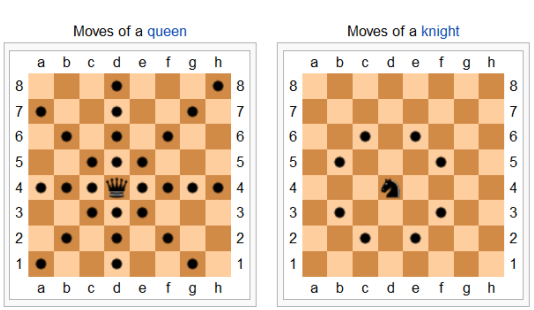
Trying to learn rules without a good representation is impossible

- Imagine trying to learn the rules of Chess at the level of pixels or 2x2 blocks
- We can guess that one representation is better than another if it is easier to learn rules about how the environment behaves using the representation

Real-world
General Game Learning

Human Game Learning Process

Natural Language Understanding



Rule Learning

Strategic Planning

Learning from Mistakes

Object Categorization and Recognition (Vision)

Conversation

Real-world Game Learning

Research is bottlenecked by

- 3D object segmentation and categorization
- 3D environment reconstruction
- Natural language processing

Human game learning in the physical world still provides us with useful insights

- Constructing a good representation for rule learning and action planning is rarely challenging to humans

Existing Game Research

Game AI Research

An AI that plays Chess well is only the summit of Mount Everest

- It assumes that you already have a perfect game representation and understanding of the rules that define this representation

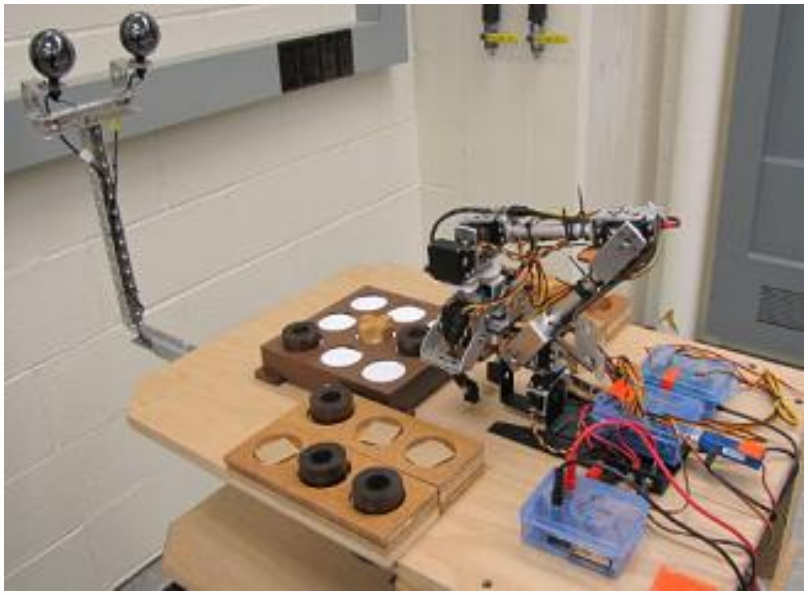
General Game Playing (Michael Genesereth) generalizes learning strategy across games but ignores learning the game representation

Forcing General Game Learning

A few people have tried general game learning

- Dramatically reduce space of games under consideration (ex. RPS, TTT variants)
- Manually teach system all relevant categories and train classifiers for them
  - the largest number of categories that has been successfully used is 3
- Focus is on robots interacting with humans, not so much on game learning

State of the art (2011)



- People have taught robots to observe two entities playing tic-tac-toe or rock-papers-scissors in contrived settings and then attempt to play these games

Digital-world
General Game Learning

What games

Many different types of games

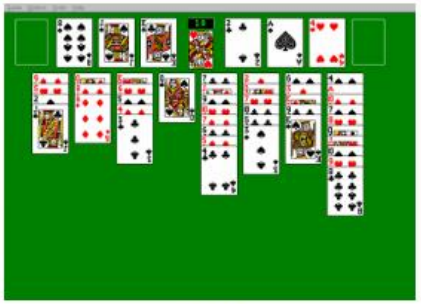
- Real-time (Mario) vs. turn-based (Reversi)
- Discrete (Tetris) vs. continuous (Space Invaders)
- Complete information (FreeCell) vs. incomplete information (Poker)
- Single player (Zelda) vs. competitive (Chess)

For now, focus is on 2D, sprite-based games

Bejeweled

Mahjong Solitaire

Peg Solitaire



FreeCell

Settlers of Catan

Pandemic

Braid

VVVVVV

Poker

Warcraft II: Tides of Darkness

SimCity 2000

The Sims

Eternal Champions

Final Fantasy 3

Megaman X

Metal Slug 3

(One Possible) General Game Learning Pipeline

Inputs

AI gets a video feed of the game

AI can take actions as if it were a human (ex. with the keyboard, mouse, or game controller)

AI may have access to: (semi-supervised)

- Human playthroughs of the game
- Partial annotations of playthroughs denoting specific objectives (ex. “reach the final boss”, “don’t die”)

Pipeline Strategy

Construct many possible representations

- Each segmentation of the input video feed leads to different symbolic representations
- Attempt to learn rules on each symbolic representation independently
- The representation which leads to the simplest rules and best predicts future states of the game is the best representation

Game Learning Pipeline

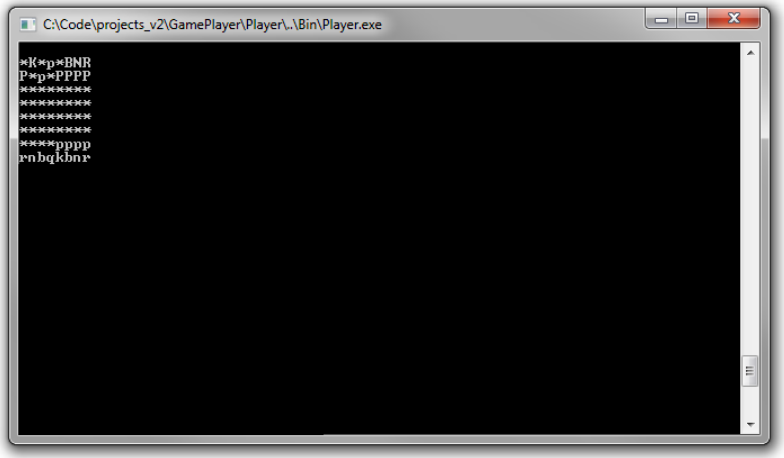
- Map video feed of the game onto a candidate symbolic representation
- Decide what kind of actions are possible
- Learn a model for how symbolic representation evolves as a function of the action taken
- Determine the goal of the game
- Plan actions that accomplish the goal

Specializing for a given game is “easy” – our focus is on generalizing learning across games

Pipeline Example: Chess Titans

1. Map to symbolic representation

Observable State



+

Hidden Variables

WhitePlayer has control

The black rooks and black king have not moved

No En passant is possible

…

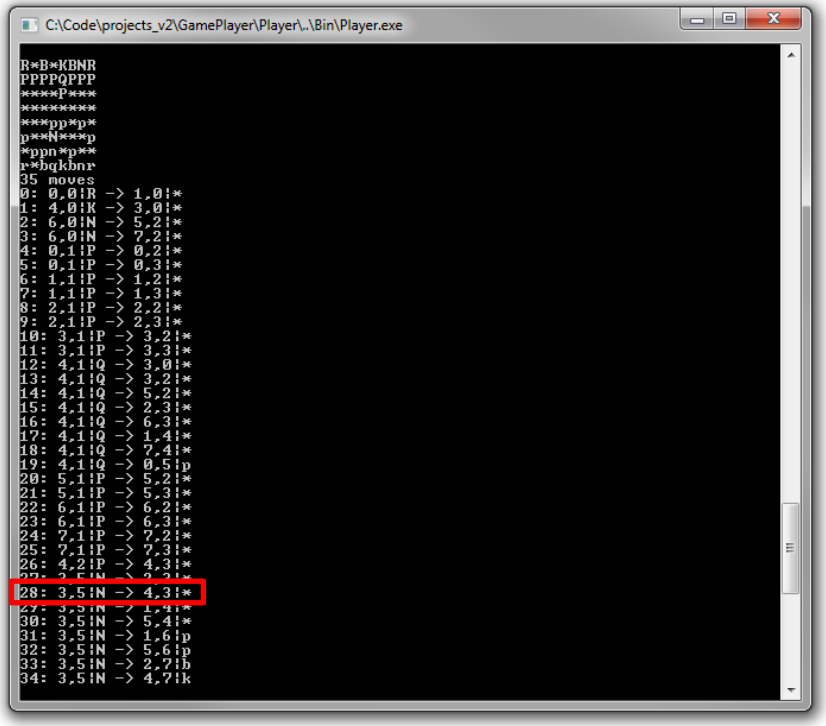
2. What actions are possible

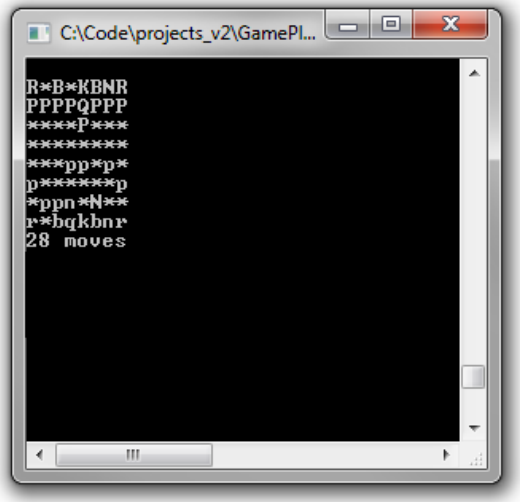
Physical Actions

Symbolic Actions

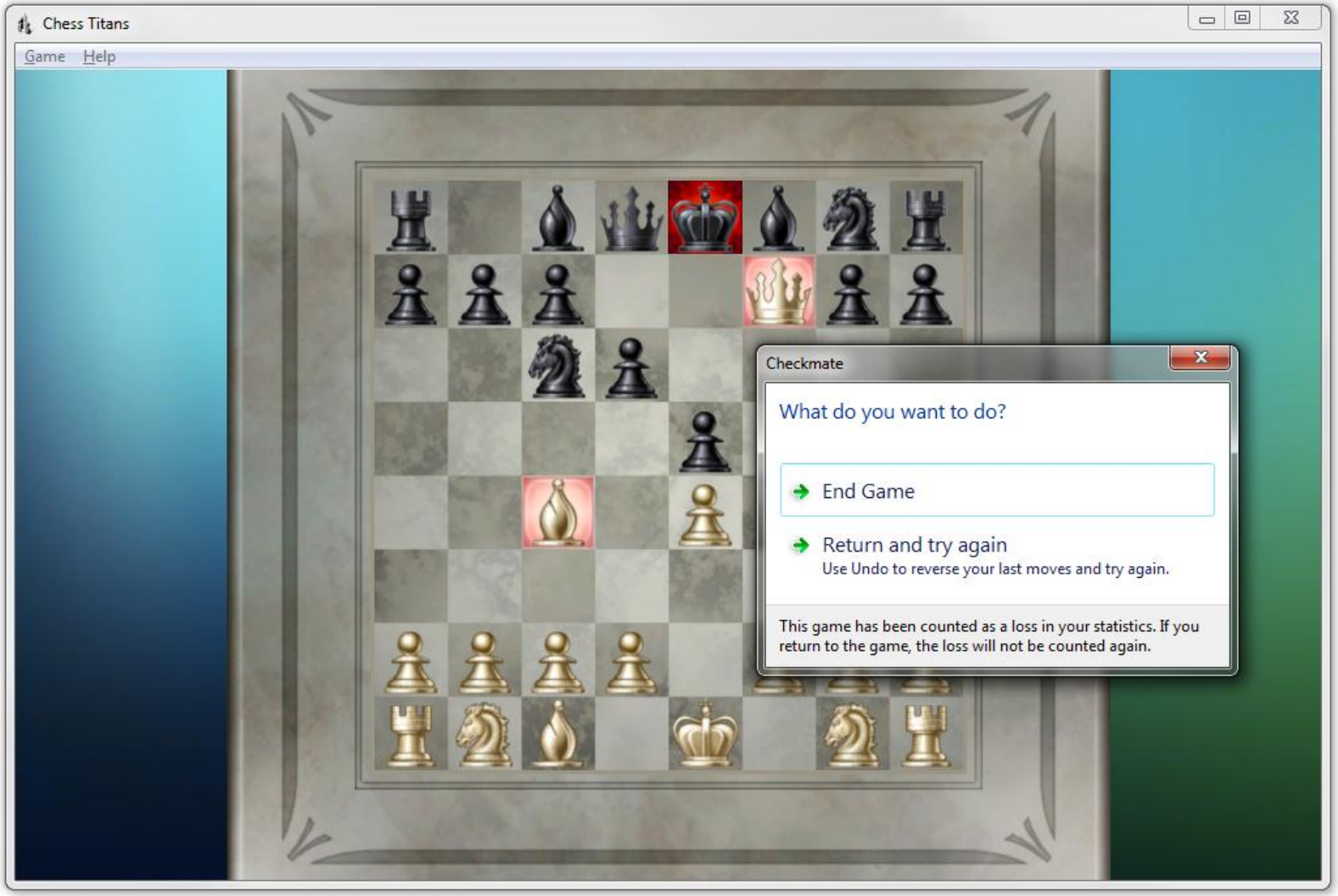
3. What rules govern the game

How does the game state evolve in response to actions and time?





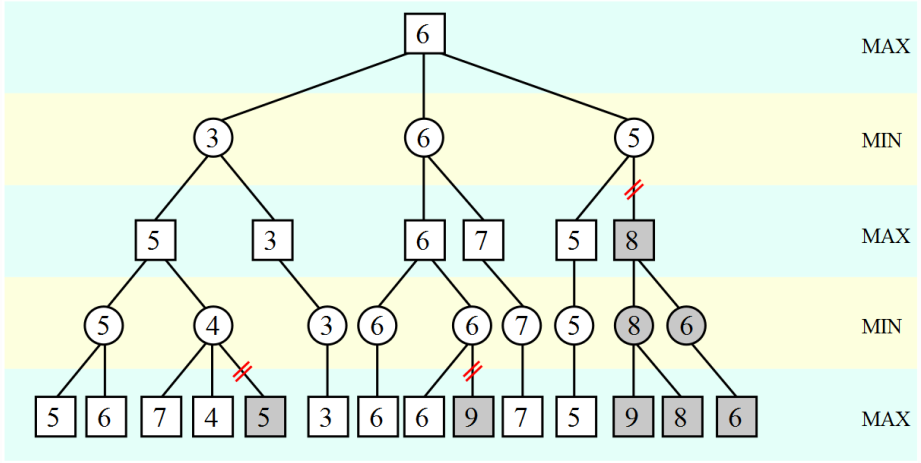
4. Determine the goal

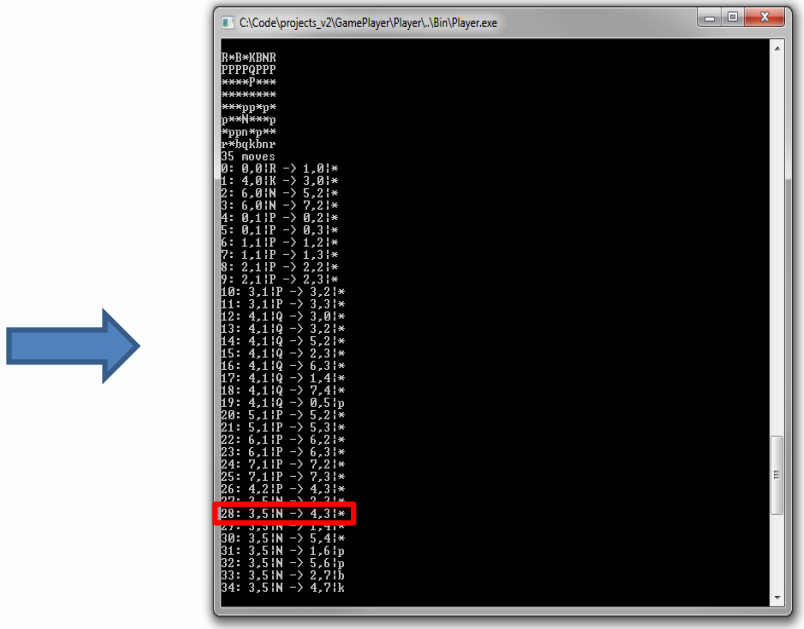


5. Execute actions leading to goal

Game State Value Estimator

+





Imperfect game models

Learning at each stage in this pipeline is going to be hard, but it doesn't need to be perfect

- You can play a fine game of Chess if you don't know the en passant or castling rules
- The rules that occur often are typically the ones that are most important

Pipeline Stages

Stage 1: Extract Symbolic Representation

Extract Symbolic Representation

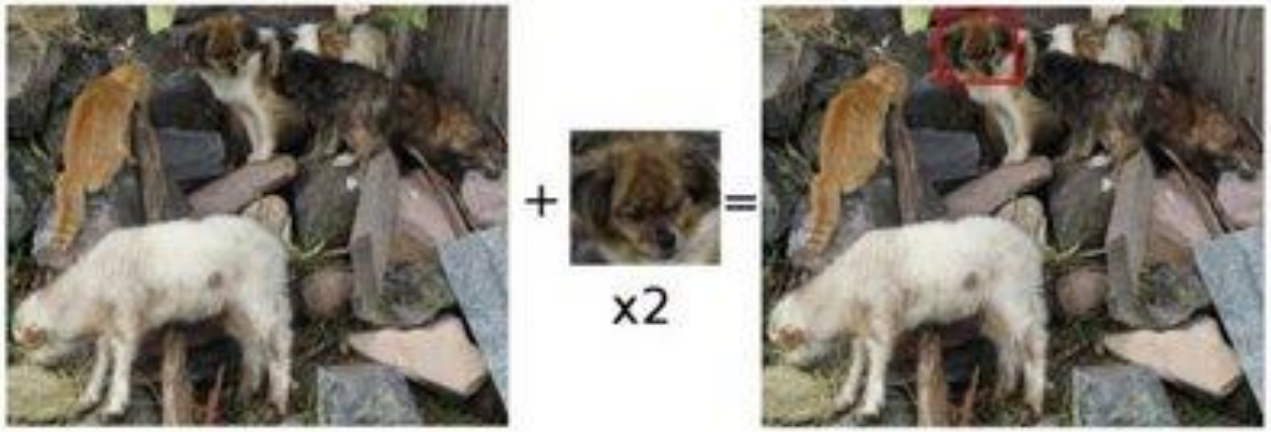
Co-segmentation

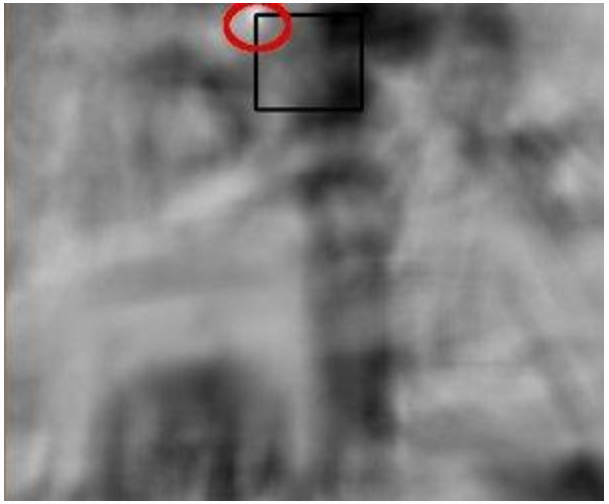
- Assume the set of input images contain many instances of similar objects
- Simultaneously segment the images and extract a set of commonly-occurring templates

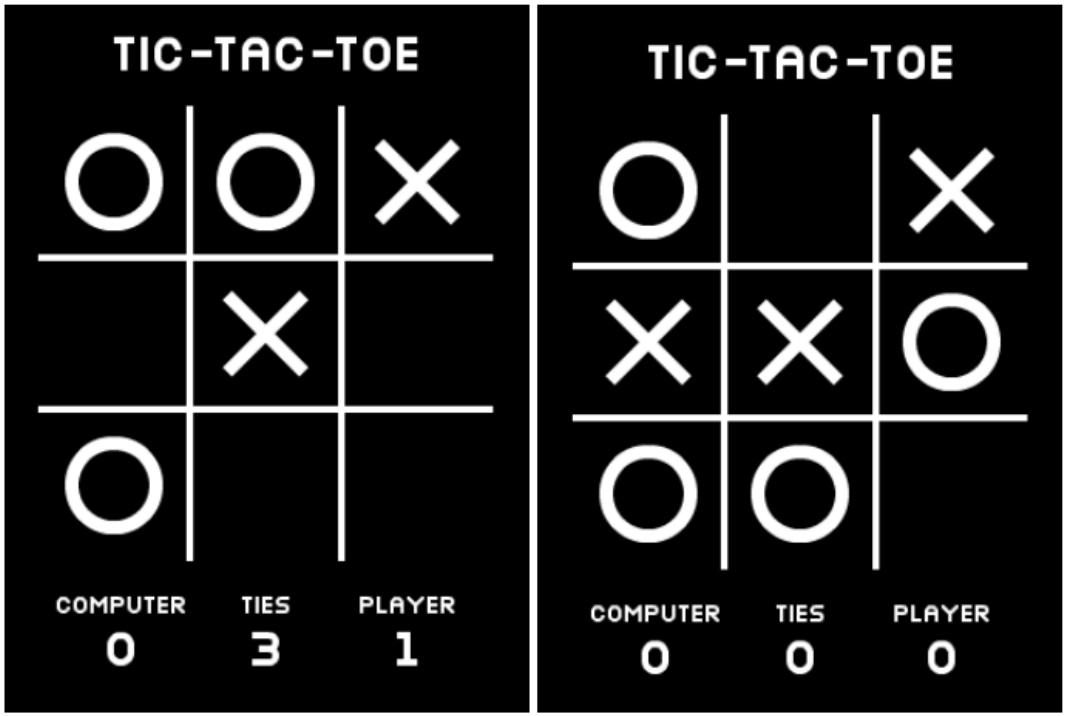
Template matching

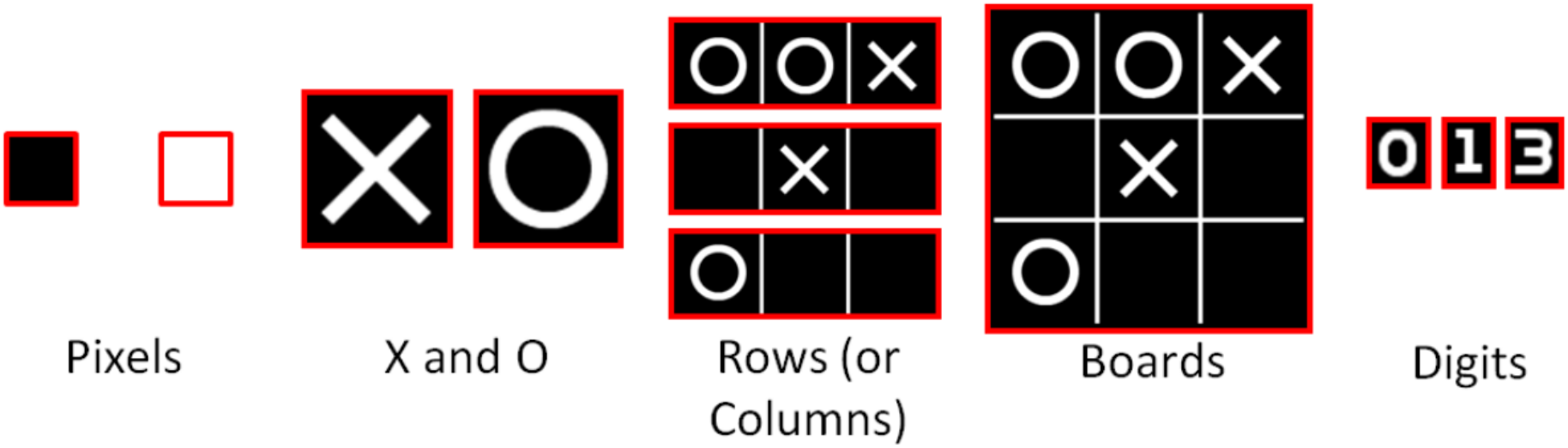
- Use template matching to isolate the instances and extract a set of representations

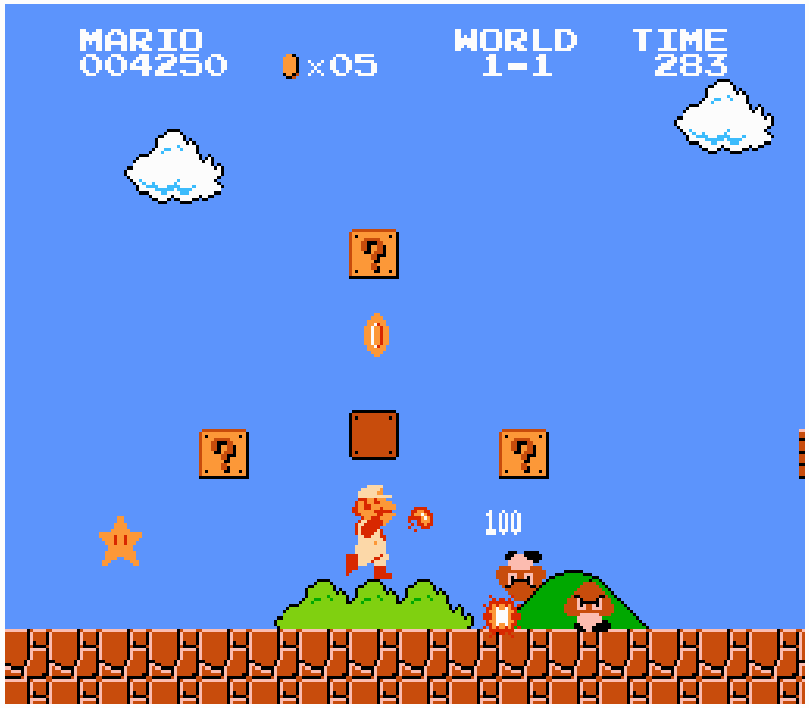
Template Matching

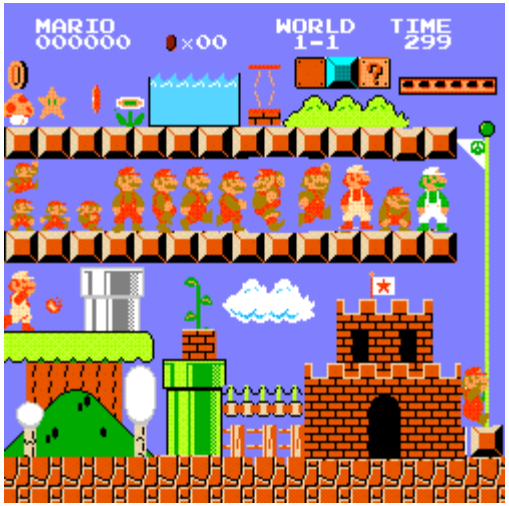


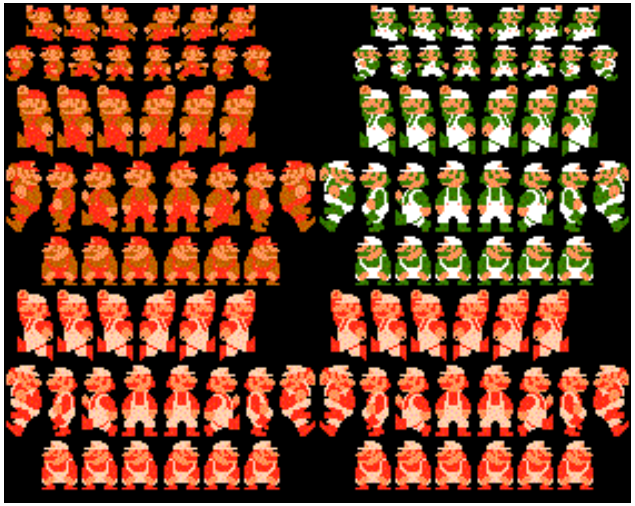



TIC-TAC-TOE
COMPUTER
0
TIES
3
PLAYER
1
TIC-TAC-TOE
COMPUTER
0
TIES
0
PLAYER
0


Pixels
X and O
Rows (or Columns)
Boards
Digits
0 1 3


MARIO
004250
×05
WORLD
1-1
TIME
283
100


MARIO
000000
×00
WORLD
1-1
TIME
299



Human Knowledge

The AI cannot be expected to re-derive the meaning of human language and numerals

- Search for dominant patterns among the symbol sets (grids, graphs, text)
- When certain patterns are detected (ex. a string of glyphs) use character recognition to replace them with a related symbolic form (ex. “Score” or “9000”)
- One goal of GGL is to discover what patterns are necessary and which can be learned

Stage 2: Determine Valid Actions

Discrete input

Near-continuous input

Use symbolic representation

Physical Actions

Symbolic Actions

Use symbolic representation

Stage 3: Learn a Game Model

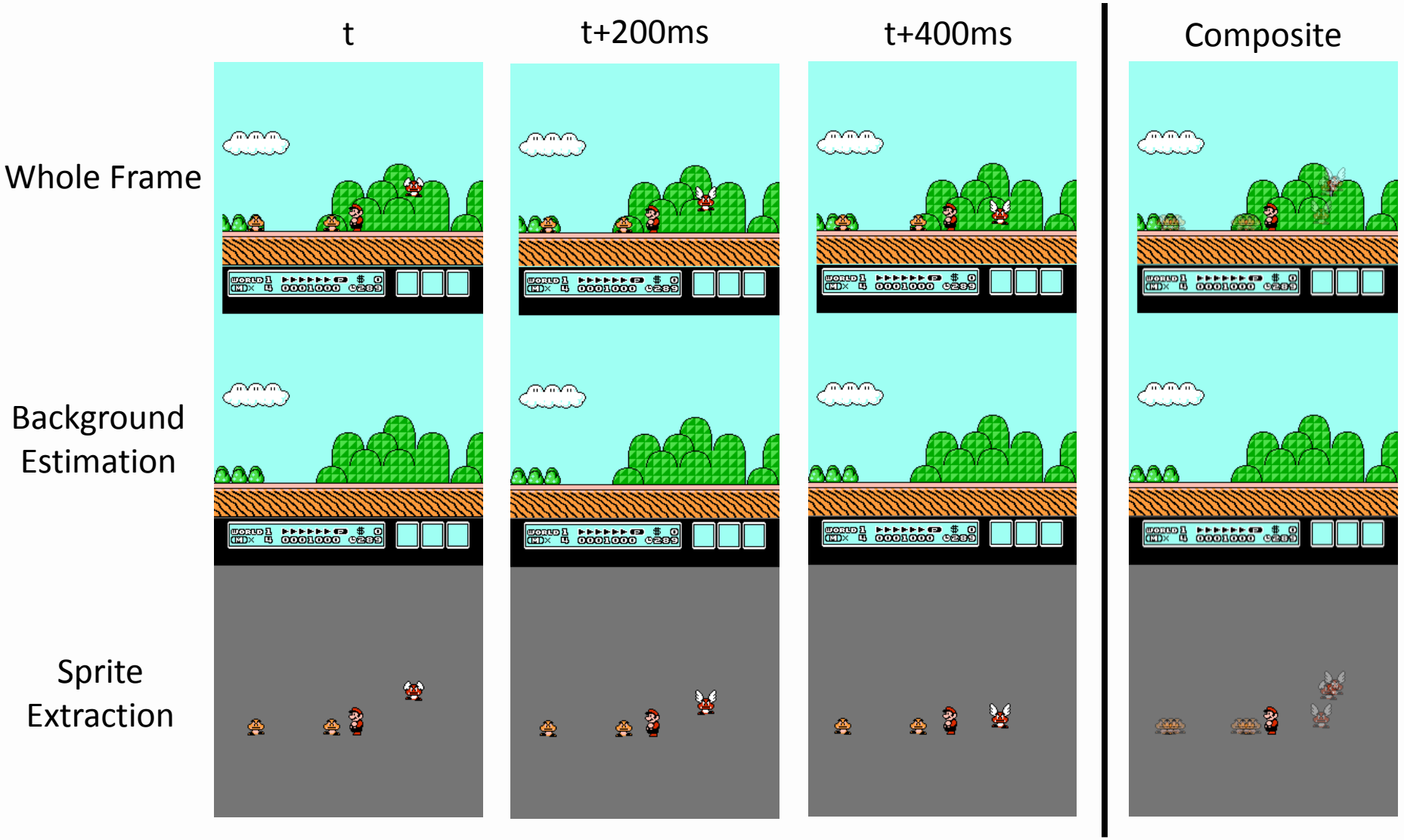
Game Model

A game model encodes all the relevant rules of the game

- Defines which actions are legal (Chess, Reversi)
- Takes the current state and a user action and returns the next state
- Need to learn from observations and experiments
- Markov decision process

Games are complicated but the behaviors of subcomponents is often simple

Simple Entity Behavior



Natural Language Rules

Chess

- A player can move a rook or queen along any horizontal or vertical straight line so long as the intervening pieces are all empty
- Players can only move pieces of their color
- It is legal for white player to move a white pawn forward two squares if it is in its home row and the two squares in front of the pawn are empty

Natural Language Rules

Super Mario Bros.

- Pressing right moves Mario to the right
- If Mario falls into a pit or fire he will die
- Goombas move slowly in one direction and can fall off ledges
- If Goombas hit an obstacle they will flip directions
- Jumping on Goombas kills them but colliding with them from the side will hurt Mario
- When Mario reaches a flag at the end of a stage he advances to the next stage in that world

Models encode rules

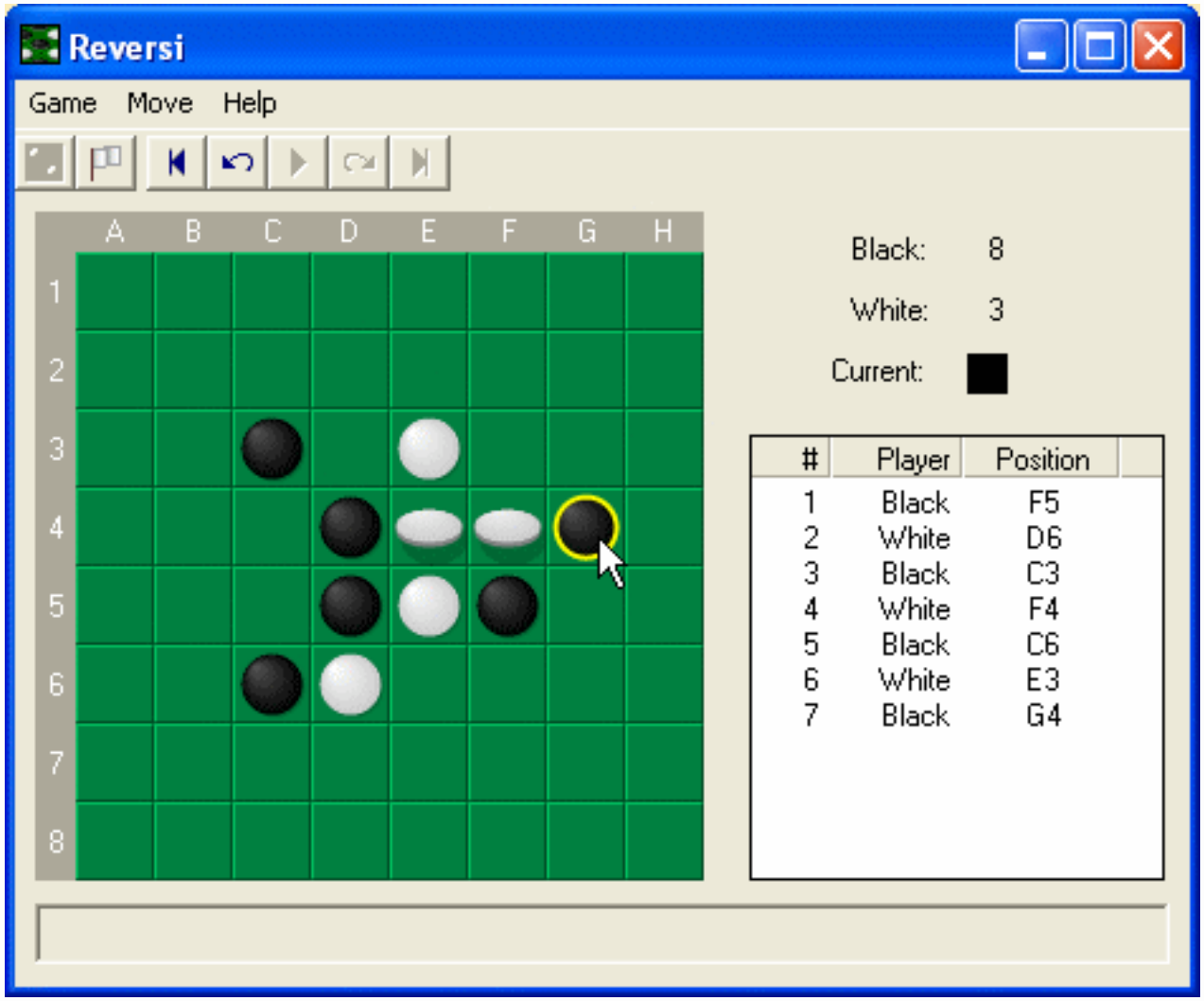
Need to specify a language that can encode game rules

- Complex enough to model important game behavior
- Simple enough to be learned from observations

Genre-specific vocabulary

- Piece, grid, board location, move
- Collision, velocity, parabola, atlas/map, portal

Reversi


Reversi
Game
Move
Help
A B C D E F G H
1 2 3 4 5 6 7 8
Black: 8
White: 3
Current:
Player Position
1 Black F5
2 White D6
3 Black C3
4 White F4
5 Black C6
6 White E3
7 Black G4

Game Model Languages

$$\text{Legal}(\text{whitePlayer}, \text{place}(x),$$
$$\quad \text{line}(x, n, m) \in \text{blackPiece} \;\&\&$$
$$\quad \text{lineEndPt}(x, n, m) \in \text{whitePiece})$$

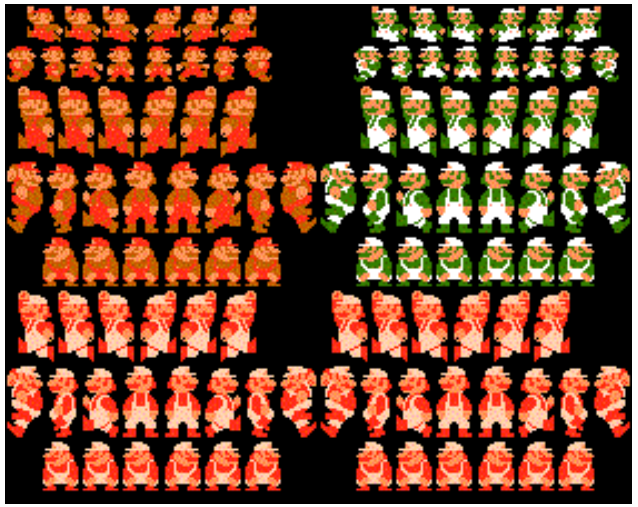
"It is legal for white to place at a coordinate x if the line starting at x parameterized by n and m contains only black pieces and the end point of this line is a white piece for all board locations x and certain values of n and m"

$$\text{Terminal}(\text{countPieces}(\text{whitePiece}) +$$
$$\quad \text{countPieces}(\text{blackPiece}) == 64)$$

Learning Rules and Representation

Rules that apply generally are more likely than a bunch of special cases

- Information gain criteria
- Occam's razor



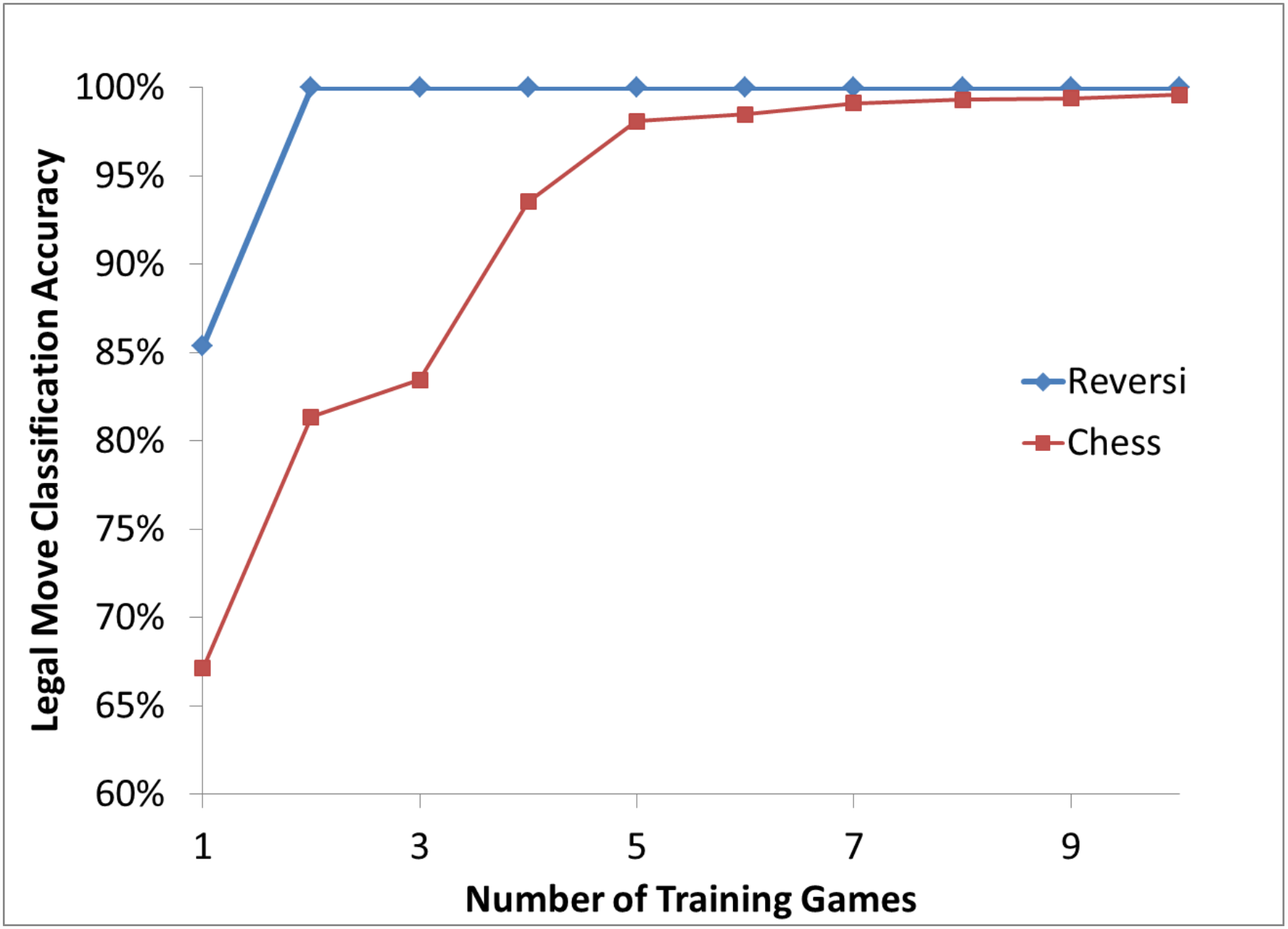
Categories are important

- Two entities probably belong to the same category if most of the rules that apply to entity A apply to entity B

Learning Rules and Representation

A representation is good if it is easy to build rules that describe it

- We can use this to decide between many possible candidate representations


100%
95%
90%
85%
80%
75%
70%
65%
60%
Legal Move Classification Accuracy
1
3
5
7
9
Number of Training Games
Reversi
Chess

Stage 4: Determine the Goal

Giveaway Checkers

Giveaway Checkers

How can we tell if we are playing regular Checkers or Giveaway Checkers?

- If we have observations of humans playing the game, what value are they placing on their pieces?
- If we have an AI (or human) opponent, are they trying to capture our pieces or not?

State Value Inference

True Goal of Chess

- Construct a situation in which your opponent is unable to prevent you from capturing their king

Inferred Value Function

- Capture enemy pieces, keep your own pieces from being captured
- King=20, Queen=10, Rook=6, Bishop=4, Knight=3, Pawn=1
- We can play the game just fine using this value function, in fact, it is ultimately more useful

State Value Inference

GAME OVER
DEAD
DIED

THANK YOU
CONGLATURATION !!!
WINNER IS
HAPPY END!

General Values

Novelty

- Seeing the death screen 10,000 times is not as interesting as exploring the world. If you explore a game and manage to see the ending, you've won.

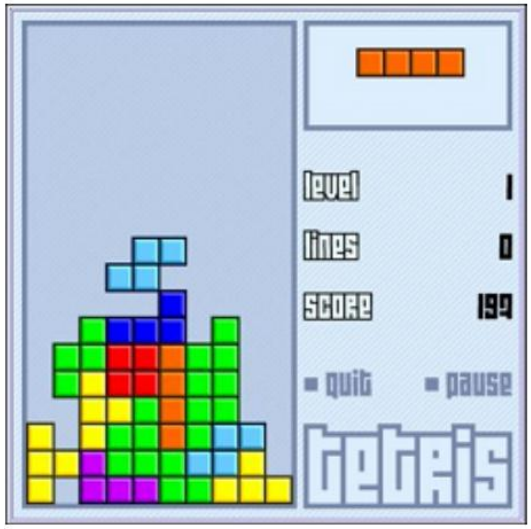
Canonical directions

- In many platformer games "proceed to the right, and don't die" is a good approximation of the goal

Number of actions

- Maximizing the number of actions available is a good idea in many piece-type games

State Value Inference



Tetris

The Legend of Zelda

Desktop Tower Defense

Level, lines, score, life

- See what humans appear to be optimizing
- Try to optimize each possible number on many different passes, see which results in the best behavior (ex. exploration criteria)

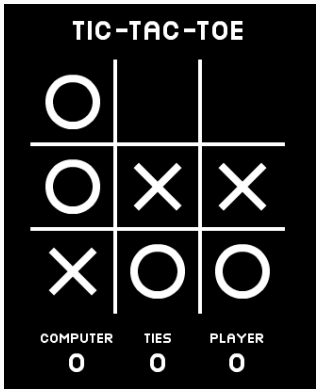
Prior work

Value function inference for specific games is often used to derive information from experts

- Value function for Chess was parameterized with approximately 8,000 parameters
- Parameter values were learned by analyzing thousands of grandmaster Chess games

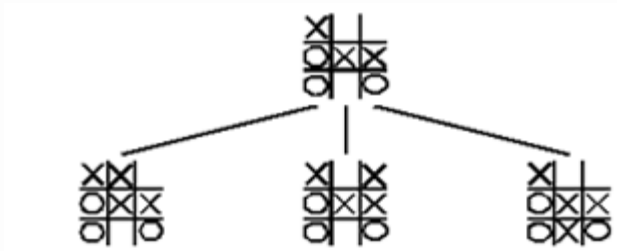
Goal is to use this idea on learned game models and construct “general parameterizations” of the value function

Stage 1 – 4 Summary

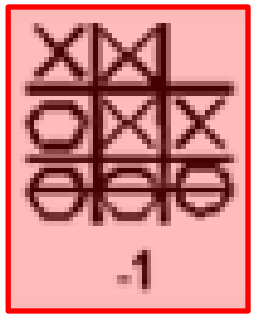


Pixels

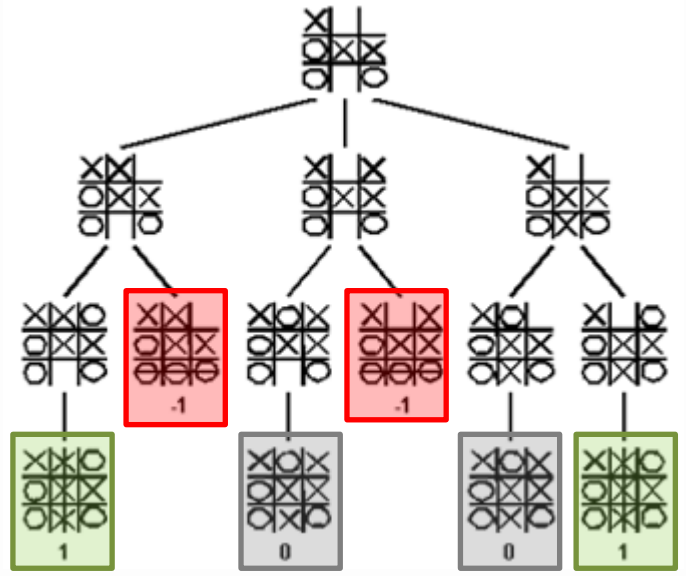
→ Symbolic Game State



Symbolic Game State → List of Actions and Resulting Game States



Goal Evaluator



Complete Game Description

Stage 5: Plan Actions

Execute actions leading to goal

Different games are solved in different ways

- Minimax tree search (Chess, Chinese Checkers)
  - Requires value function:
    - Use one modeled from examples in Stage 4
    - Use one derived from General Game Playing
- Motion planning (Mario, Tetris)
  - Try many different state value functions
  - Reinforcement learning techniques
    - Markov decision process formulation

Execute actions leading to goal

This is what General Game Playing is for

- Active area of research
- CS227B, Michael Genesereth
- Not designed for incomplete game models

Uses Game Description Language

- Mapping from a symbolic representation of the game state to a list of actions and the resulting states
- Many general game players have been developed
  - Bandit-based Monte Carlo
- Can easily convert game model to GGP

Execute actions leading to goal

```
legal(Y,mark(M,N)) <=
  true(cell(M,N,b)) &
  true(control(Y))

legal(white,noop) <=
  true(cell(M,N,b)) &
  true(control(black))

legal(black,noop) <=
  true(cell(X,Y,b)) &
  true(control(white))
```

```
next(cell(M,N,x)) <=
  does(white,mark(M,N)) &
  true(cell(M,N,b))

next(cell(M,N,o)) <=
  does(black,mark(M,N)) &
  true(cell(M,N,b))

next(cell(M,N,W)) <=
  true(cell(M,N,W)) &
  distinct(W,b)
```

```
goal(white,100) <= line(x)
goal(white,50) <=
  ~line(x) &
  ~line(o) &
  ~open
goal(white,0) <= line(o)
goal(black,100) <= line(o)
goal(black,50) <=
  ~line(x) &
  ~line(o) &
  ~open
goal(black,0) <= line(x)
```

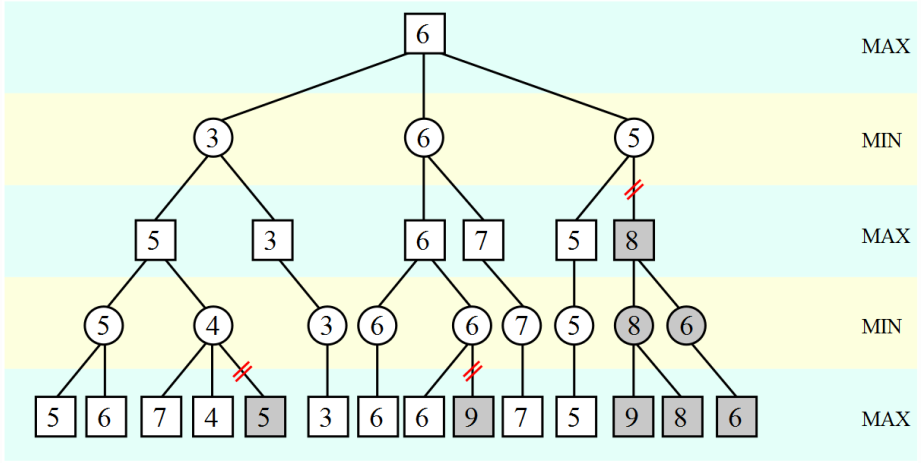
```
init(cell(1,1,b))
init(cell(1,2,b))
init(cell(1,3,b))
init(cell(2,1,b))
init(cell(2,2,b))
init(cell(2,3,b))
init(cell(3,1,b))
init(cell(3,2,b))
init(cell(3,3,b))
init(control(white))
```

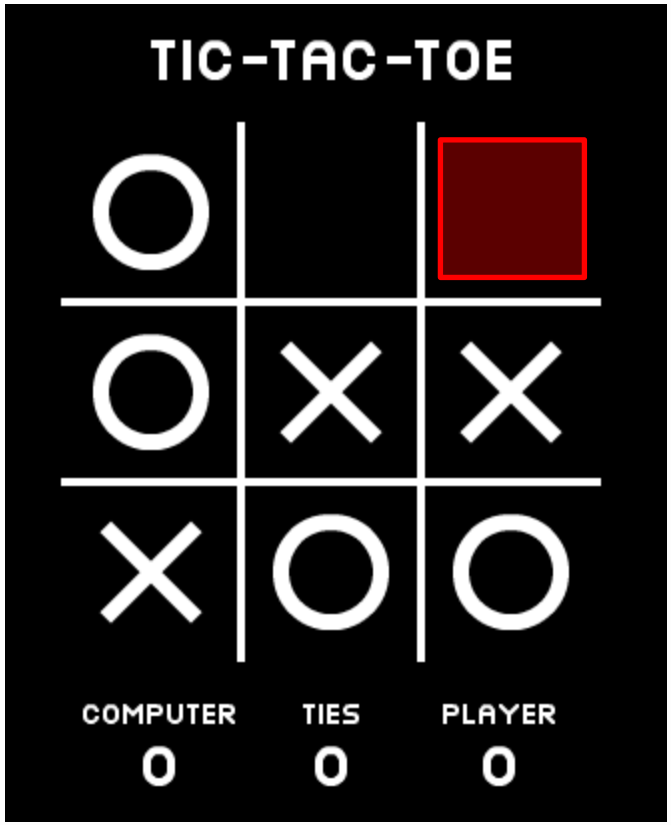
```
terminal <= line(x)
terminal <= line(o)
terminal <= ~open
```

Typical general game player

"Game state value estimator" trained from many game simulations

+





5. Execute actions leading to goal

General Game Playing is not going to help much for Mario or Zelda

Can use basic search and control algorithms

- Relies on a reasonably good model of the state space transitions for the game

Try and try again

- Millions of deaths are fine as long as it eventually learns; once it does we can look at how to make it learn faster or better

Mario AI Competition

Conclusions

Results: Deliverables are easy

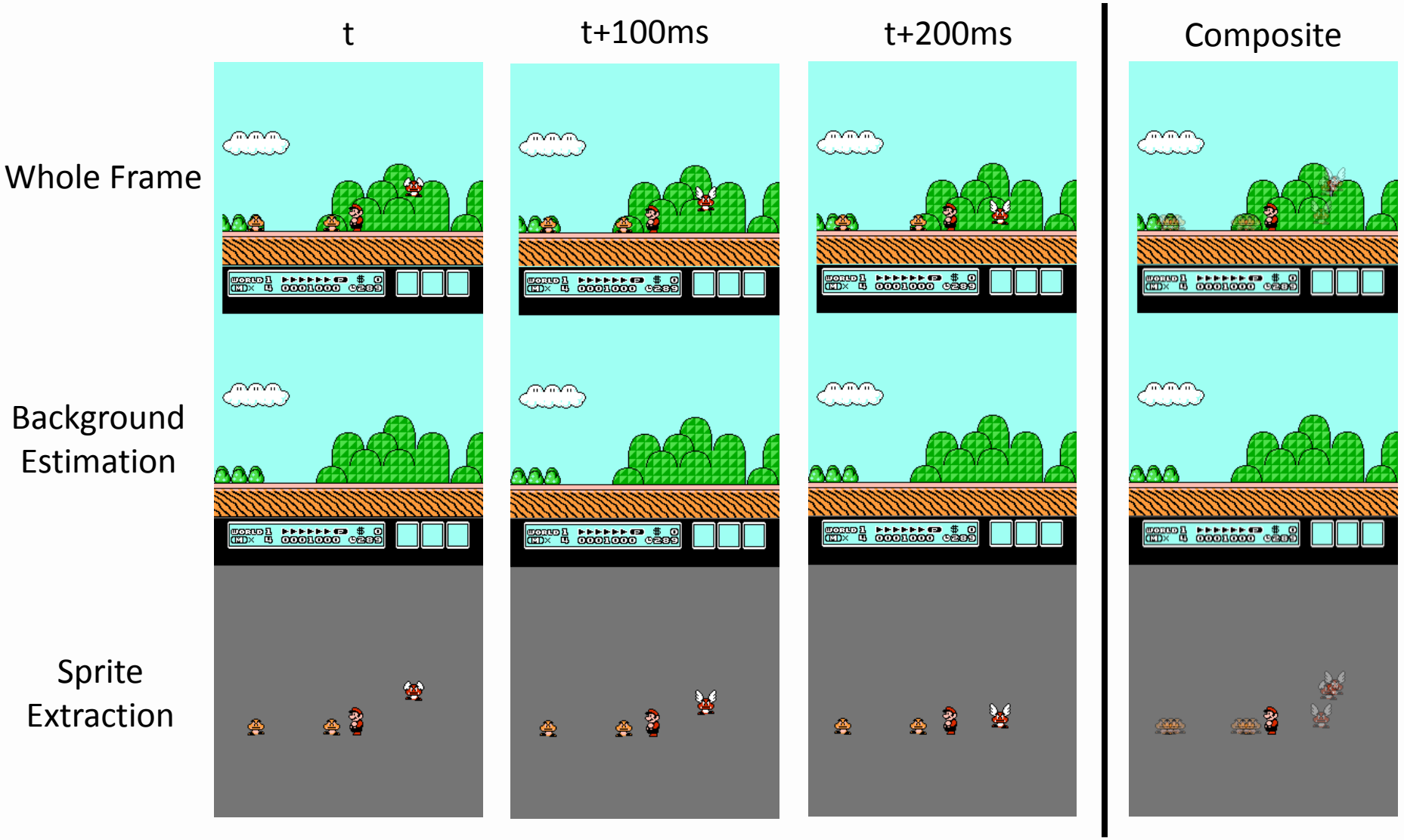
Viable game learners answer a lot of questions

- Can we determine what components are meaningful?
- What games are hard to learn?
- How many games do you need to learn rules?
- Can we construct a useful set of rules that generalize across many games?
- How well can we play even if we only know some of the rules?
- Can we compute a useful reward function from watching human players?

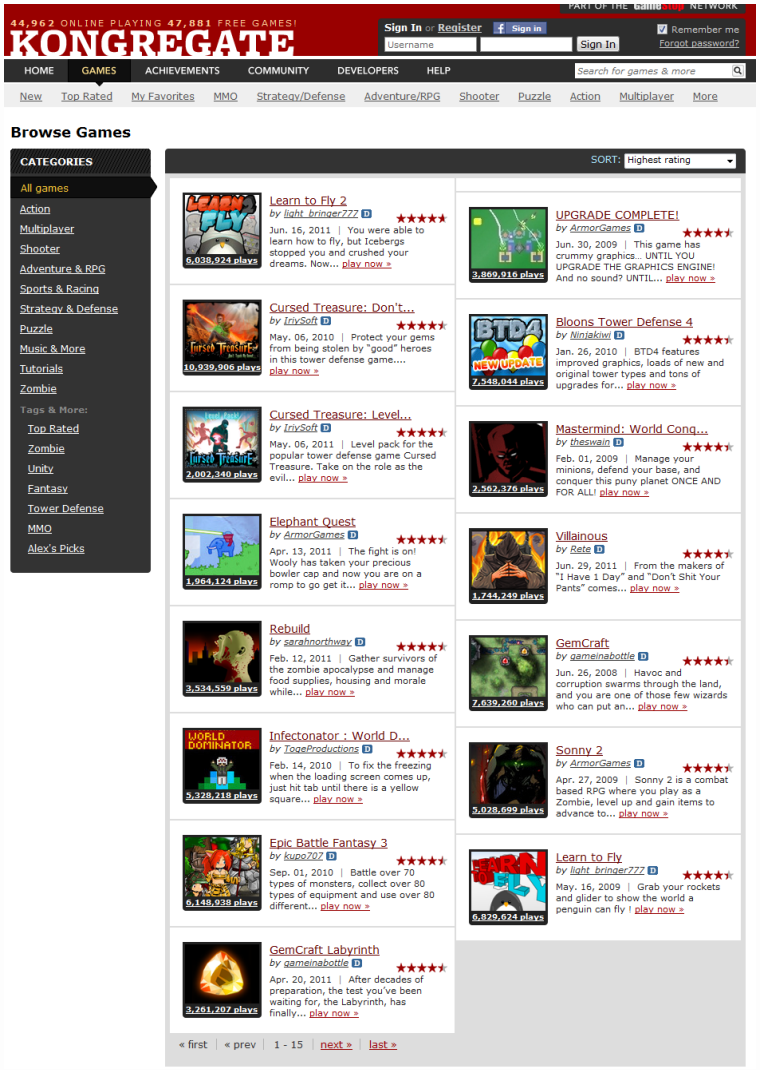
We will use digital games to study…

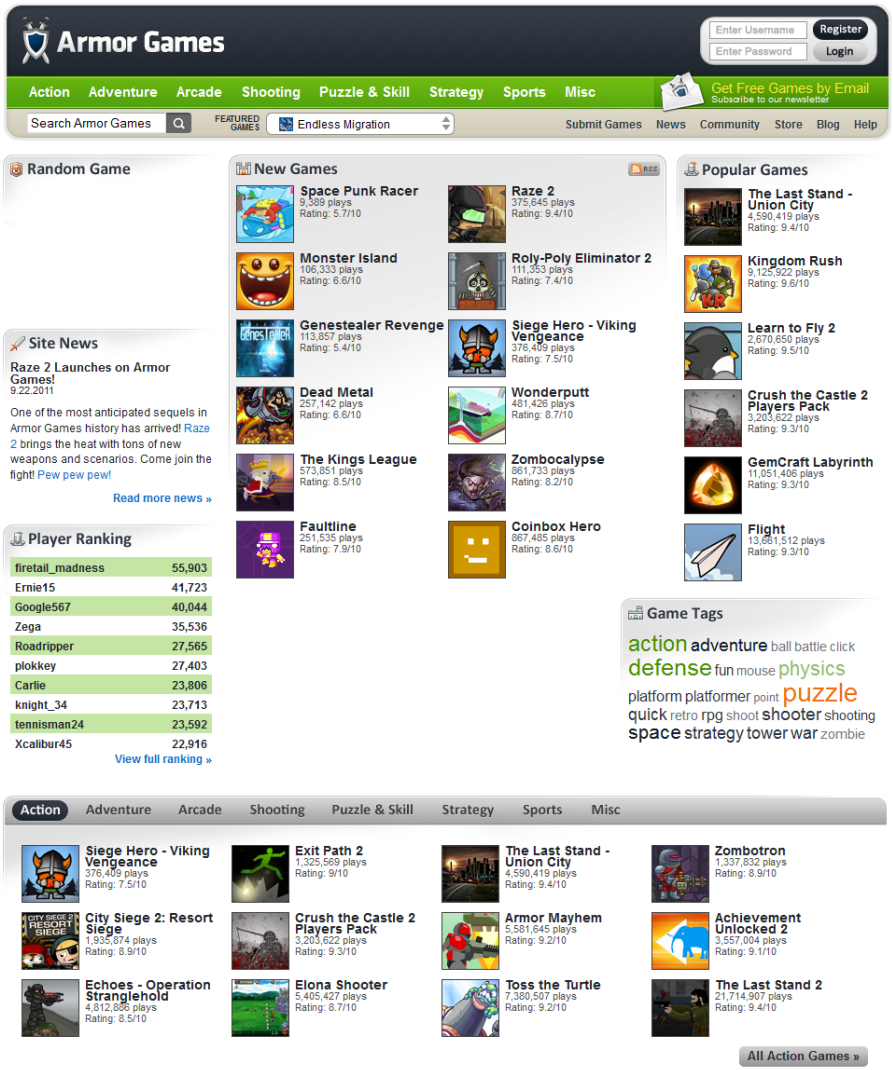
- **Unsupervised** object segmentation and categorization
- Rule learning
- Knowledge transfer between categories
  - Between Mario & Luigi
- Goal inference
- Game state value estimation
- Action planning
- Knowledge transfer between games
  - Between Super Mario 1 and Super Mario 3
  - Between Shogi and Chess

First steps: NES games



Lots and lots of 2D games (>100,000)





2D vs. 3D games