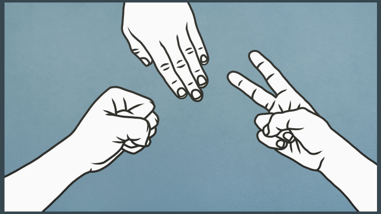
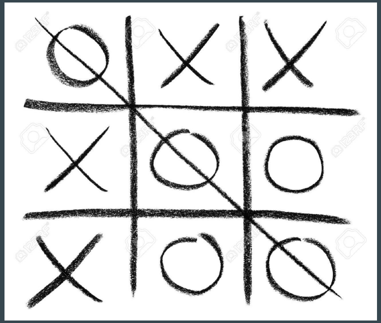
The notion of 'depth' in games: a case study with Quixo

Jau Tung Chan

(Advisor: Prof. James Glenn)

'Depth' in games: an intuition

?



HEARTHSTONE

Less 'depth'

More 'depth'

'Depth' in games: in literature

The AAAI-17 Workshop on
What's Next for AI in Games?
WS-17-15

Depth in Strategic Games

Frank Lantz,[*] Aaron Isaksen,[†]
Alexander Jaffe,[‡] Andy Nealen,[*†] Julian Togelius[†]

[*]NYU Game Center [†]NYU Game Innovation Lab [‡]Spry Fox

Abstract

This paper explores the question of whether it's possible to discover a well-defined property of game systems that corresponds to what game designers and players mean by the term "depth." We propose a measurable property of a game's formal system, which we call '*d*', that corresponds to the capacity of a game to absorb dedicated problem-solving attention and allow for sustained, long-term learning. To define this property we develop a formal model that measures how susceptible a game is to partial solutions under conditions of steadily increasing computational resources. We then sketch out several directions for using the model to investigate questions about the structural properties of games that produce these effects.

Introduction

Game designers and players often make reference to the concept of depth. This term has a broad, general meaning that expresses the idea that something is absorbing and profound. But there is a narrow application of the term which refers specifically to the formal system of strategic games. Our goal is to examine this particular meaning of the term as it is used to describe the abstract system of choices and outcomes within games of this type. In this context, depth refers to a game's capacity to provide a lifetime of study, learning, and improvement. A game with great depth is one that seems to unfold into an endless series of challenging problems and responds to serious thought by continually revealing surprising and interesting things to think about.

Games like Chess, Bridge, Go, StarCraft, Hearthstone, and League of Legends are able to absorb the dedicated efforts of a large community of expert players over many generations of serious competition and collaborative analysis while continually producing fascinating strategic problems. Does this capacity correspond to a property of game systems that can be objectively observed and measured? After all, there are many well-defined formal properties of game systems that can be analyzed and quantified, features such as state space and branching factor. We often talk about depth as if it were a property like this, but is it? Is there some precisely definable, objectively observable property of a game's underlying structure that allows it to exhibit this quality?

One challenge with the informal use of the word "depth" is that it is often used as a binary term, implying a quality that games either do or don't have. The real picture is much more complex; every game exists along a spectrum of depth. Moreover, the same game may exhibit different levels of depth in relation to different player communities, or in relation to the same community at different times. We are interested in examining depth as a quality that all games have to various degrees, understanding how this quality is related to a game's formal structure, and developing conceptual tools that allow us to explore this relationship with greater precision.

Depth is often referred to by game developers (Pulsipher and Others 2011; Kiley 2013; Ghostcrawler 2016) and in scholarly research (Browne 2008; Nielsen et al. 2015; Abbott 1975) but to our knowledge no attempts have been undertaken to make a thorough and rigorous investigation into the property to which it refers. The purpose of this paper is to lay the groundwork for such an investigation. We are attempting to establish a foundation, clarify the important questions, and suggest directions for further study. We are not at this time proposing final answers to the central question.

Goals and Clarifications

It is not our goal with this project to make normative claims about how games *should* be designed or what makes a *good* game. The term depth can be used casually as a general superlative but that's not the way we are using it here. We aren't claiming that this quality is the most important feature for judging a game's overall value; there are many ways for a game to be good that aren't related to the kind of depth this paper investigates. In addition, even though we are attempting to analyze precise and quantifiable properties of a game's formal system, we are not attempting to reduce aesthetic judgments to objective empirical claims. Instead, we wish to establish clarity regarding features of game systems which *can* be observed and measured in order to better inform aesthetic analysis and discussion. In this way, our project is analogous to research into color theory (Albers 1971), which doesn't claim to distinguish between good and bad paintings, but provides a powerful, consistent, fine-grained conceptual tool for artists and critics to use in analyzing painting's aesthetic qualities. We believe that this conceptual tool can be used by game designers to understand some of the effects that rule and parameter changes have on their games.

967

*Lantz et al., 2017: "Game designers and players often make reference to the concept of **depth**. This term has a broad, general meaning that expresses the idea that something is **absorbing and profound**."*

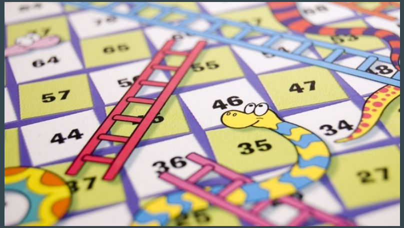
*"... depth refers to a game's **capacity to provide a lifetime of study, learning, and improvement**. A game with great depth is one that seems to unfold into an endless series of challenging problems and responds to serious thought by continually revealing surprising and interesting things to think about."*

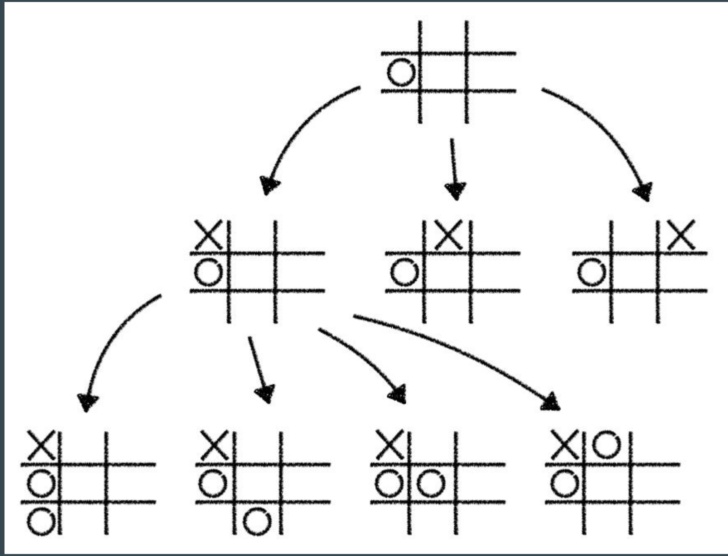
https://www.aaai.org/ocs/index.php/WS/AAAIW17/paper/download/15142/14780

'Depth' in games: a quantitative formalism?

State space?

... but like ...





Game (strategic) length?

... but like ...

Branching factor? ... but like ...

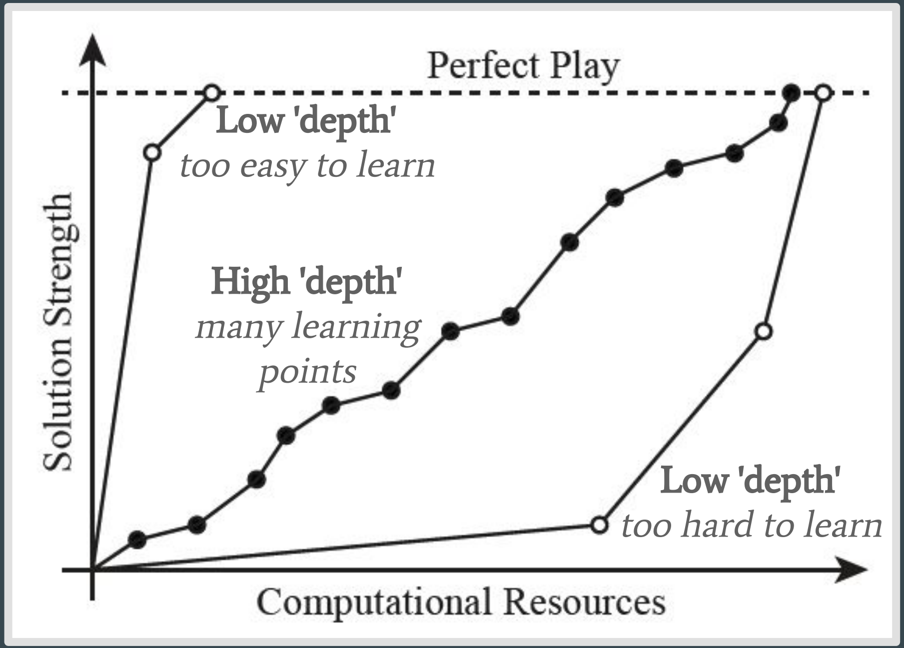
'Depth' in games: (back to) literature

Lantz et al., 2017: "What matters is ***meaningful state space****. The question of what makes state space meaningful in this way is exactly what our project seeks to understand."*

"A key concept in addressing this question is the idea of the ***skill chain*** *... the presence of a skill chain with a* ***large number of distinct steps*** *is evidence of a game in which a* ***player can improve through study****, a game in which the more you think about it the better you get. Players of a game like this are on a journey of gradual and continuous improvement, ascending ever-upwards towards better understanding and stronger play."*

'Depth' in games: (back to) literature

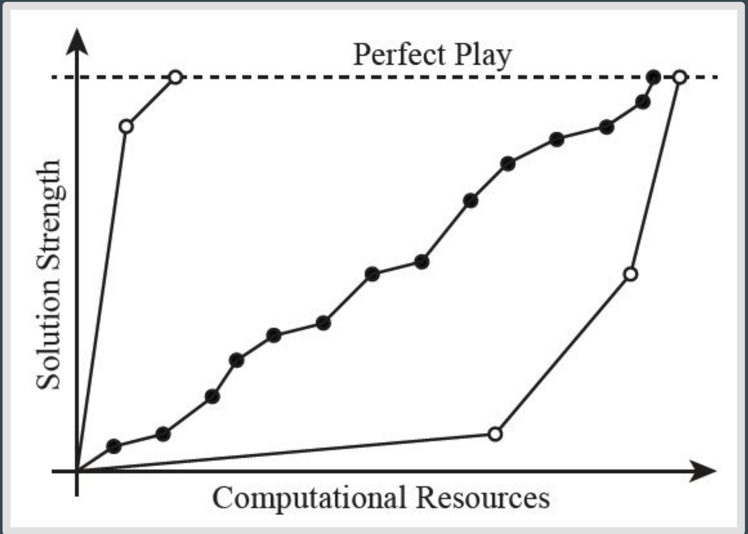
The 'strategy ladder' model



(Lantz et al., 2017)

*"Each dot represents a complete, **fully-defined algorithmic strategy** for playing a particular game represented by each path. Each dot is the **best strategy that can be achieved at that level of computational resources**."*

Objective of my senior project



So what is Quixo: the basics

Player X vs. Player O
(2-player turn-based game)

5X5 grid of tiles
(of Xs and Os)

Take turns to pick up a tile and
replace it back onto the board

First to make 5-in-a-row wins

So what is Quixo: a move

Player X vs. Player O
(2-player turn-based game)

5X5 grid of tiles
(of Xs and Os)

? Take turns to pick up a tile and **replace** it back onto the board

First to make 5-in-a-row wins

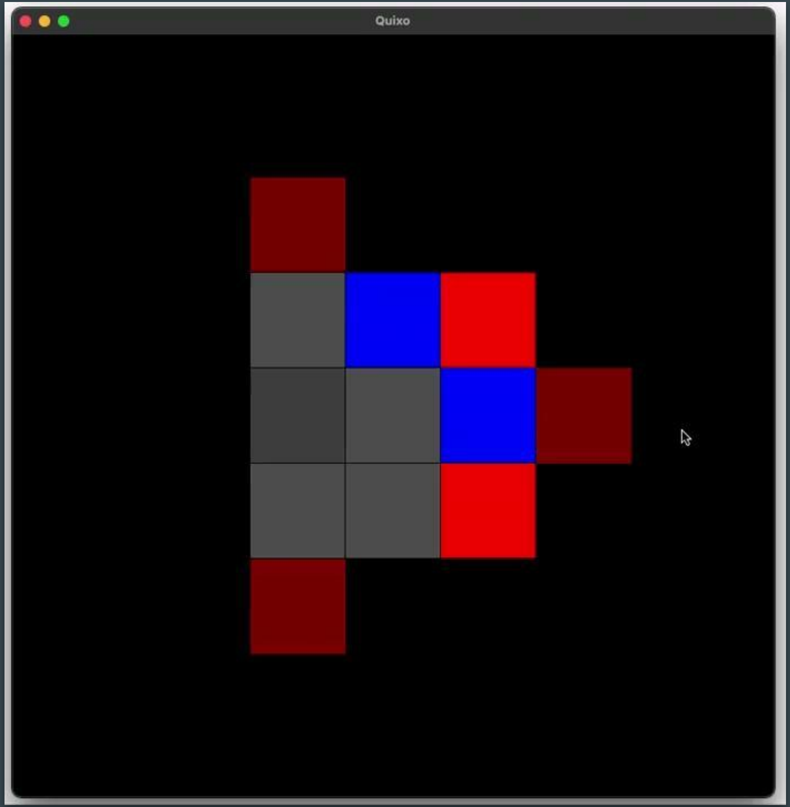
https://www.youtube.com/watch?v=-cG5eapomTE

Must pick up from boundary

Must pick up a tile that is either empty or already your shape

Always replace as your shape
(each cube has an X side and an O side for you to rotate it)

So what is Quixo: a (quick) game



An (analogous) 3X3 version

(i.e. 3-in-a-row wins)

Red = X, Blue = O

(I was too lazy to implement characters)

So why Quixo?

'Small' enough to compute a ***complete* solution**!

(this means we can compute the *exact* optimal move for any game state)

'Large' enough that the **complete solution is not trivial**

(definitely not trivial for a human, at least as far as I know...)

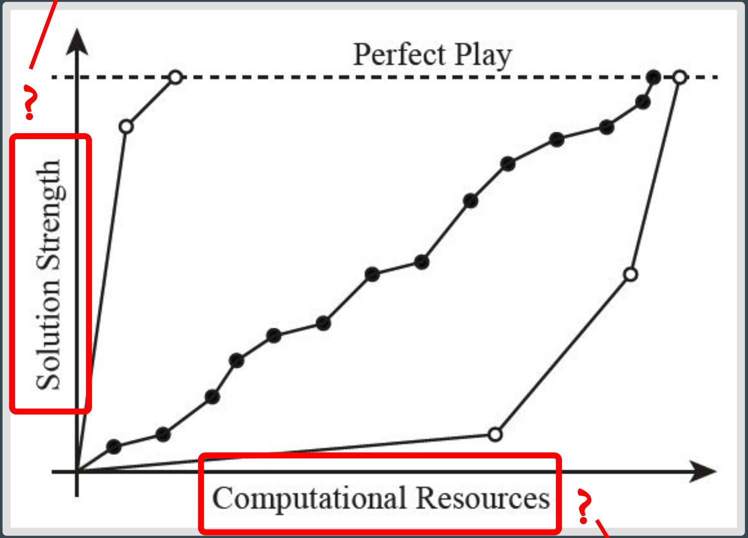
Possible to **vary sizes to vary 'depth'**

(i.e. 5X5 Quixo, 4X4 Quixo, 3X3 Quixo; presumably 5X5 is 'deeper' than 3X3)

I want to / because I can ☺️!

(Back to the) objective of my senior project

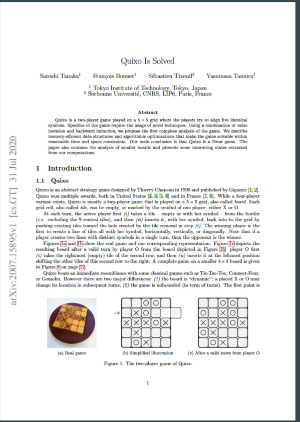
'Performance' against an optimal player



Time given for a computational agent to learn the game

3 main parts of my senior project

Compute Optimal Player



arXiv:2007.15895v1 [cs.GT] 31 Jul 2020

Quixo Is Solved

Satoshi Tanaka[1] François Bonnet[1] Sébastien Tixeuil[2] Yasumasa Tamura[1]

[1] Tokyo Institute of Technology, Tokyo, Japan
[2] Sorbonne Université, CNRS, LIP6, Paris, France

Abstract

Quixo is a two-player game played on a 5×5 grid where the players try to align five identical symbols. Specifics of the game require the usage of novel techniques. Using a combination of value iteration and backward induction, we propose the first complete analysis of the game. We describe memory-efficient data structures and algorithmic optimizations that make the game solvable within reasonable time and space constraints. Our main conclusion is that Quixo is a Draw game. The paper also contains the analysis of smaller boards and presents some interesting states extracted from our computations.

1 Introduction

1.1 Quixo

Quixo is an abstract strategy game designed by Thierry Chapeau in 1995 and published by Gigamic [1, 2]. Quixo won multiple awards, both in United States [3, 4, 5, 6] and in France [7, 8]. While a four-player variant exists, Quixo is mostly a two-player game that is played on a 5×5 grid, also called *board*. Each grid cell, also called *tile*, can be empty, or marked by the symbol of one player: either X or O.

At each turn, the active player first *(i)* takes a tile – empty or with her symbol – from the border (*i.e.* excluding the 9 central tiles), and then *(ii)* inserts it, with her symbol, back into to the grid by pushing existing tiles toward the hole created by the tile removal in step *(i)*. The winning player is the first to create a line of tiles all with her symbol, horizontally, vertically, or diagonally. Note that if a player creates two lines with distinct symbols in a single turn, then the opponent is the winner.

Figures 1a and 1b show the real game and our corresponding representation. Figure 1c depicts the resulting board after a valid turn by player O from the board depicted in Figure 1b: player O first *(i)* takes the rightmost (empty) tile of the second row, and then *(ii)* inserts it at the leftmost position shifting the other tiles of this second row to the right. A complete game on a smaller 4×4 board is given in Figure 8 on page 12.

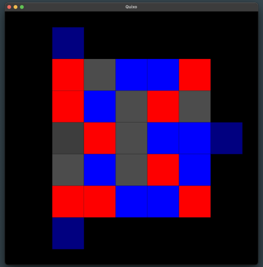
Quixo bears an immediate resemblance with some classical games such as Tic-Tac-Toe, Connect-Four, or Gomoku. However there are two major differences: *(1)* the board is "dynamic"; a placed X or O may change its location in subsequent turns, *(2)* the game is unbounded (in term of turns). The first point is

(a) Real game (b) Simplified illustration (c) After a valid move from player O

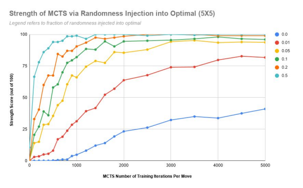
Figure 1: The two-player game of Quixo

1

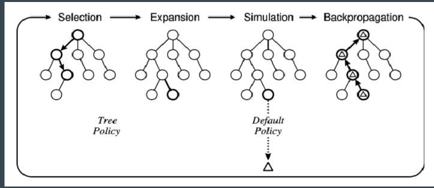
Implement Game Engine



Experiments & Results!

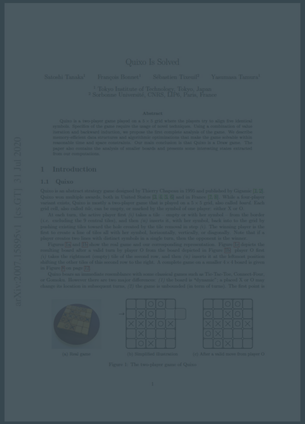


Implement Computational Learning Agent(s)

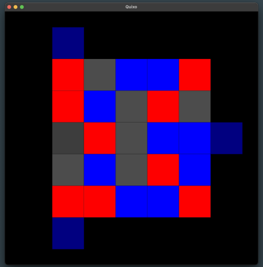


3 main parts of my senior project

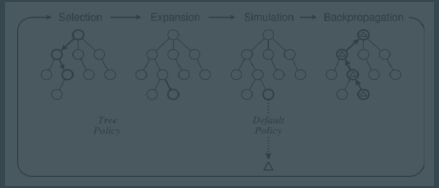
Compute Optimal Player



Implement Game Engine



Implement Computational Learning Agent(s)

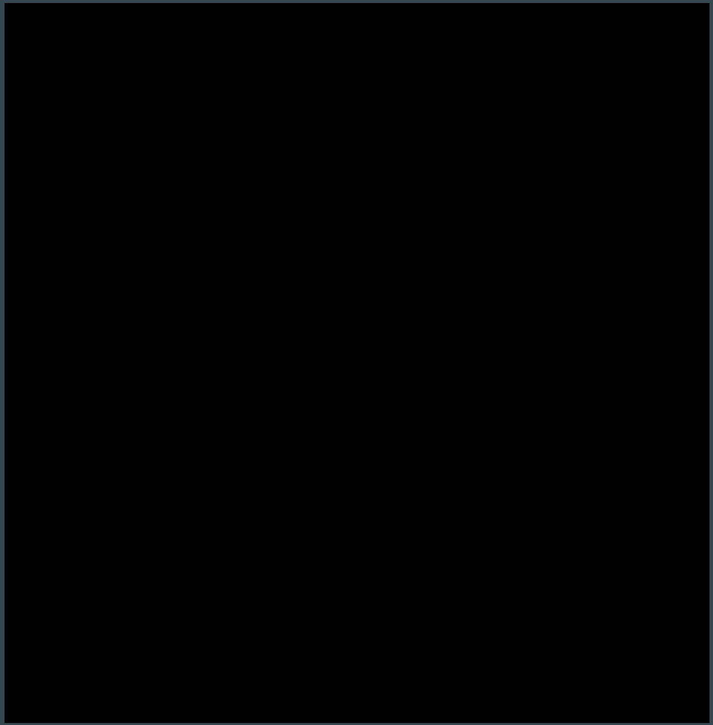


Experiments & Results!

Game engine

Booooring CS implementation details... 🥱

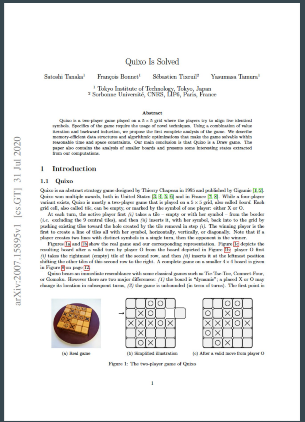
Implemented a **graphical interface** for interactive playing and for flashy graphics though ☺️!



2 random players battling it out!
(and frankly, not being very smart...)

3 main parts of my senior project

Compute Optimal Player



Quixo Is Solved

Satoshi Tanaka[1] François Bonnet[1] Sébastien Tixeuil[2] Yasumasa Tamura[1]

[1] Tokyo Institute of Technology, Tokyo, Japan
[2] Sorbonne Université, CNRS, LIP6, Paris, France

Abstract

Quixo is a two-player game played on a 5 × 5 grid where the players try to align five identical symbols. Specifics of the game require the usage of novel techniques. Using a combination of value iteration and backward induction, we propose the first complete analysis of the game. We describe memory-efficient data structures and algorithmic optimizations that make the game solvable within reasonable time and space constraints. Our main conclusion is that Quixo is a Draw game. The paper also contains the analysis of smaller boards and presents some interesting states extracted from our computations.

arXiv:2007.15895v1 [cs.GT] 31 Jul 2020

1 Introduction

1.1 Quixo

Quixo is an abstract strategy game designed by Thierry Chapeau in 1995 and published by Gigamic [1, 2]. Quixo won multiple awards, both in United States [3, 4, 5, 6] and in France [7, 8]. While a four-player variant exists, Quixo is mostly a two-player game that is played on a 5 × 5 grid, also called *board*. Each grid cell, also called *tile*, can be empty, or marked by the symbol of one player: either X or O.

At each turn, the active player first *(i)* takes a tile – empty or with her symbol – from the border (*i.e.* excluding the 9 central tiles), and then *(ii)* inserts it, with her symbol, back into to the grid by pushing existing tiles toward the hole created by the tile removal in step *(i)*. The winning player is the first to create a line of tiles all with her symbol, horizontally, vertically, or diagonally. Note that if a player creates two lines with distinct symbols in a single turn, then the opponent is the winner.

Figures 1a and 1b show the real game and our corresponding representation. Figure 1c depicts the resulting board after a valid turn by player O from the board depicted in Figure 1b: player O first *(i)* takes the rightmost (empty) tile of the second row, and then *(ii)* inserts it at the leftmost position shifting the other tiles of this second row to the right. A complete game on a smaller 4 × 4 board is given in Figure 8 on page 12.

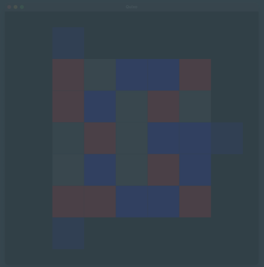
Quixo bears an immediate resemblance with some classical games such as Tic-Tac-Toe, Connect-Four, or Gomoku. However there are two major differences: *(1)* the board is "dynamic"; a placed X or O may change its location in subsequent turns, *(2)* the game is unbounded (in term of turns). The first point is

(a) Real game (b) Simplified illustration (c) After a valid move from player O

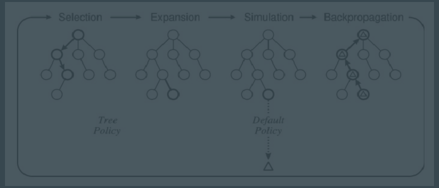
Figure 1: The two-player game of Quixo

1

Implement Game Engine

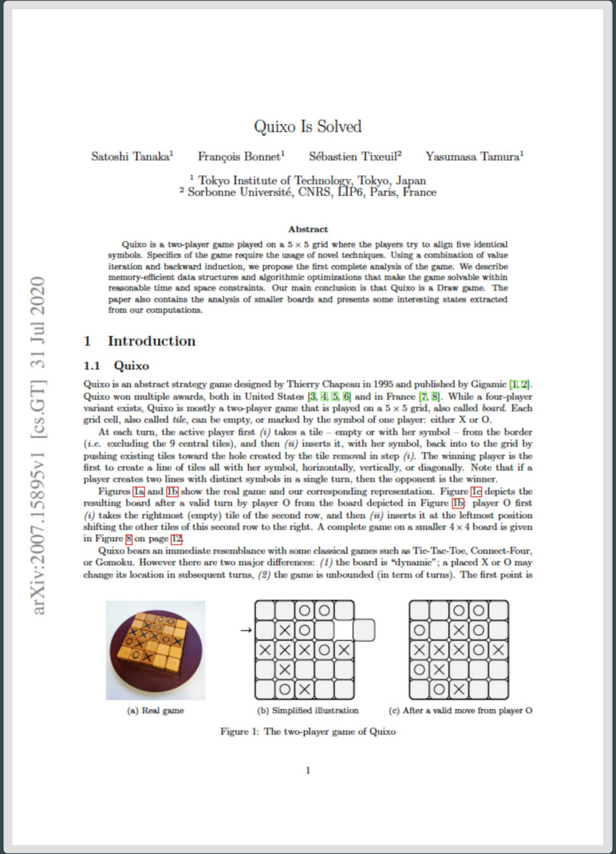


Implement Computational Learning Agent(s)



Experiments & Results!

Computing optimal player: brute-force backward induction



arXiv:2007.15895v1 [cs.GT] 31 Jul 2020

Quixo Is Solved

Satoshi Tanaka[1] François Bonnet[1] Sébastien Tixeuil[2] Yasumasa Tamura[1]

[1] Tokyo Institute of Technology, Tokyo, Japan
[2] Sorbonne Université, CNRS, LIP6, Paris, France

Abstract

Quixo is a two-player game played on a 5×5 grid where the players try to align five identical symbols. Specifics of the game require the usage of novel techniques. Using a combination of value iteration and backward induction, we propose the first complete analysis of the game. We describe memory-efficient data structures and algorithmic optimizations that make the game solvable within reasonable time and space constraints. Our main conclusion is that Quixo is a Draw game. The paper also contains the analysis of smaller boards and presents some interesting states extracted from our computations.

1 Introduction

1.1 Quixo

Quixo is an abstract strategy game designed by Thierry Chapeau in 1995 and published by Gigamic [1, 2]. Quixo won multiple awards, both in United States [3, 4, 5, 6] and in France [7, 8]. While a four-player variant exists, Quixo is mostly a two-player game that is played on a 5×5 grid, also called *board*. Each grid cell, also called *tile*, can be empty, or marked by the symbol of one player: either X or O.

At each turn, the active player first *(i)* takes a tile – empty or with her symbol – from the border (*i.e.* excluding the 9 central tiles), and then *(ii)* inserts it, with her symbol, back into to the grid by pushing existing tiles toward the hole created by the tile removal in step *(i)*. The winning player is the first to create a line of tiles all with her symbol, horizontally, vertically, or diagonally. Note that if a player creates two lines with distinct symbols in a single turn, then the opponent is the winner.

Figures 1a and 1b show the real game and our corresponding representation. Figure 1c depicts the resulting board after a valid turn by player O from the board depicted in Figure 1b: player O first *(i)* takes the rightmost (empty) tile of the second row, and then *(ii)* inserts it at the leftmost position shifting the other tiles of this second row to the right. A complete game on a smaller 4×4 board is given in Figure 8 on page 12.

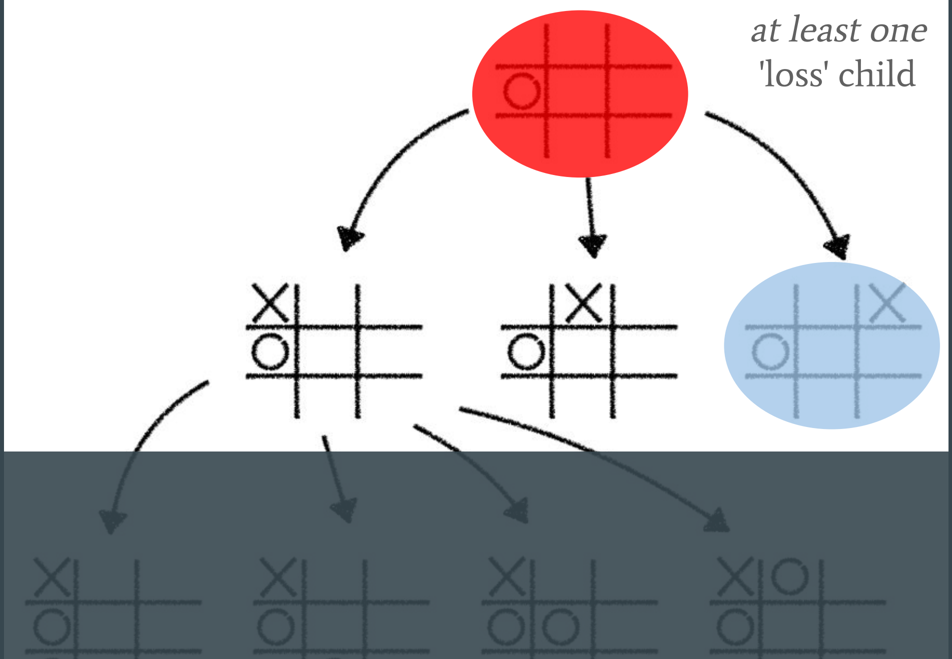
Quixo bears an immediate resemblance with some classical games such as Tic-Tac-Toe, Connect-Four, or Gomoku. However there are two major differences: *(1)* the board is "dynamic"; a placed X or O may change its location in subsequent turns, *(2)* the game is unbounded (in term of turns). The first point is

(a) Real game (b) Simplified illustration (c) After a valid move from player O

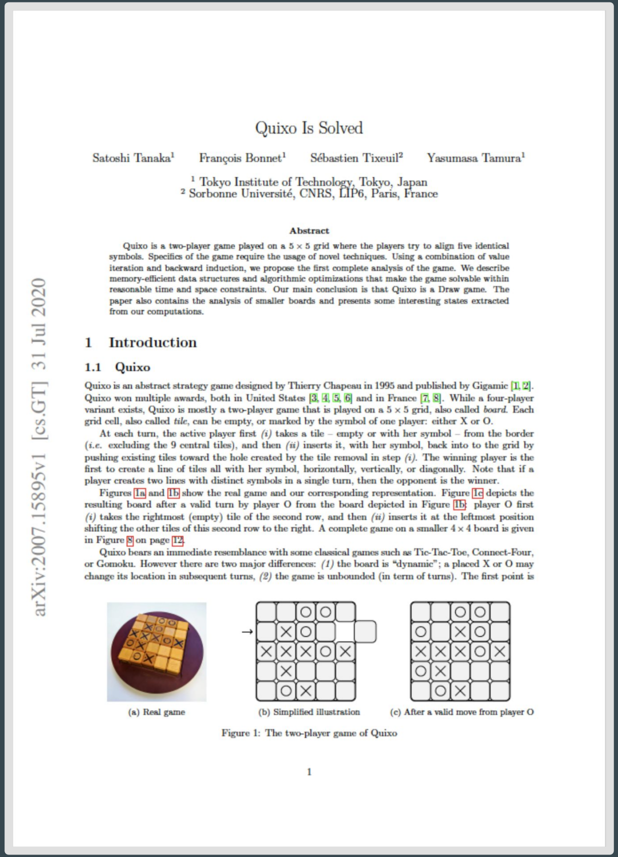
Figure 1: The two-player game of Quixo

1

Every state of the board is either first-player-win, first-player-loss, or draw



Computing optimal player: brute-force backward induction



arXiv:2007.15895v1 [cs.GT] 31 Jul 2020

Quixo Is Solved

Satoshi Tanaka[1] François Bonnet[1] Sébastien Tixeuil[2] Yasumasa Tamura[1]

[1] Tokyo Institute of Technology, Tokyo, Japan
[2] Sorbonne Université, CNRS, LIP6, Paris, France

Abstract

Quixo is a two-player game played on a 5×5 grid where the players try to align five identical symbols. Specifics of the game require the usage of novel techniques. Using a combination of value iteration and backward induction, we propose the first complete analysis of the game. We describe memory-efficient data structures and algorithmic optimizations that make the game solvable within reasonable time and space constraints. Our main conclusion is that Quixo is a Draw game. The paper also contains the analysis of smaller boards and presents some interesting states extracted from our computations.

1 Introduction

1.1 Quixo

Quixo is an abstract strategy game designed by Thierry Chapeau in 1995 and published by Gigamic [1, 2]. Quixo won multiple awards, both in United States [3, 4, 5, 6] and in France [7, 8]. While a four-player variant exists, Quixo is mostly a two-player game that is played on a 5×5 grid, also called *board*. Each grid cell, also called *tile*, can be empty, or marked by the symbol of one player: either X or O.

At each turn, the active player first *(i)* takes a tile – empty or with her symbol – from the border (*i.e.* excluding the 9 central tiles), and then *(ii)* inserts it, with her symbol, back into to the grid by pushing existing tiles toward the hole created by the tile removal in step *(i)*. The winning player is the first to create a line of tiles all with her symbol, horizontally, vertically, or diagonally. Note that if a player creates two lines with distinct symbols in a single turn, then the opponent is the winner.

Figures 1a and 1b show the real game and our corresponding representation. Figure 1c depicts the resulting board after a valid turn by player O from the board depicted in Figure 1b: player O first *(i)* takes the rightmost (empty) tile of the second row, and then *(ii)* inserts it at the leftmost position shifting the other tiles of this second row to the right. A complete game on a smaller 4×4 board is given in Figure 8 on page 12.

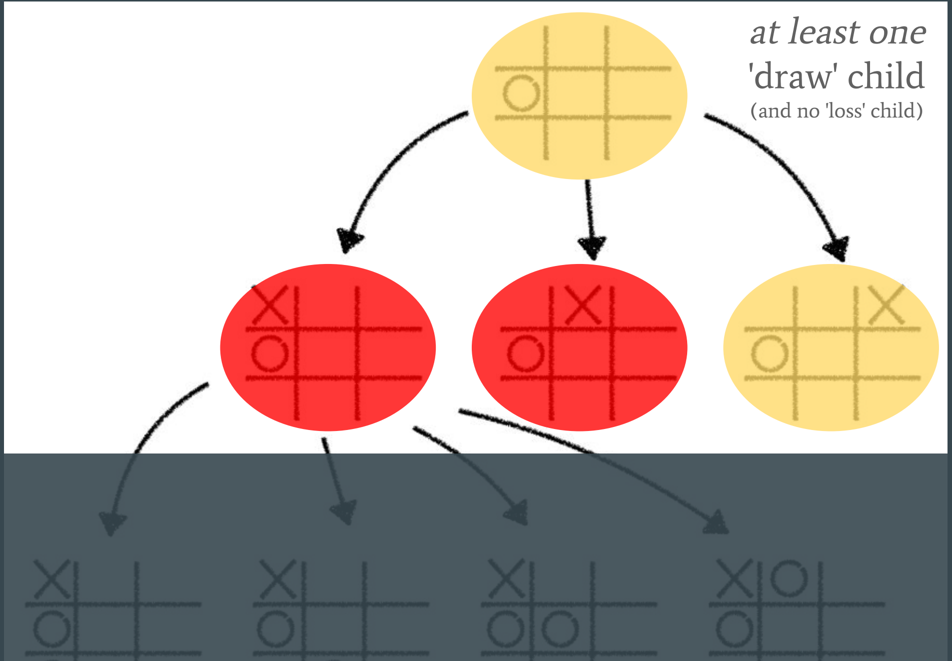
Quixo bears an immediate resemblance with some classical games such as Tic-Tac-Toe, Connect-Four, or Gomoku. However there are two major differences: *(1)* the board is "dynamic"; a placed X or O may change its location in subsequent turns, *(2)* the game is unbounded (in term of turns). The first point is

(a) Real game (b) Simplified illustration (c) After a valid move from player O

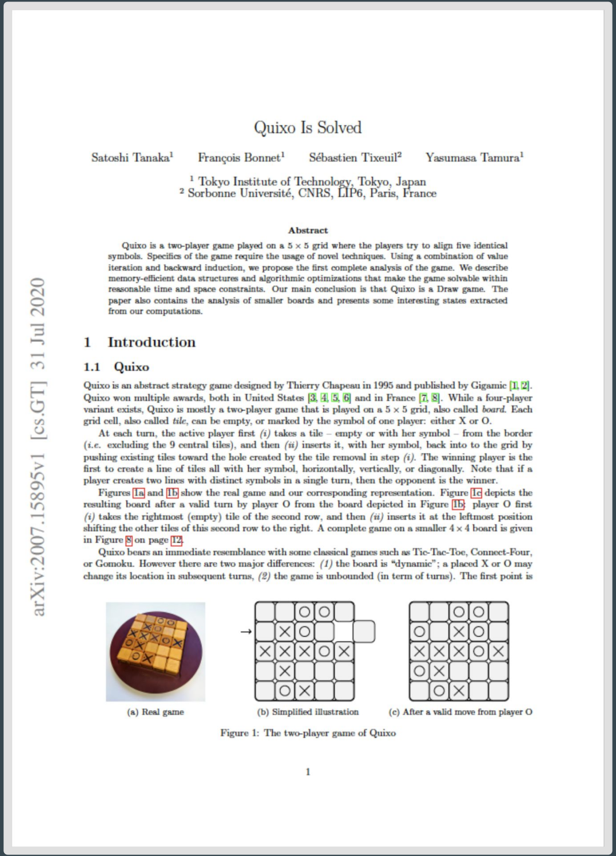
Figure 1: The two-player game of Quixo

1

Every state of the board is either
first-player-win, first-player-loss, or draw



Computing optimal player: brute-force backward induction


arXiv:2007.15895v1 [cs.GT] 31 Jul 2020

Quixo Is Solved

Satoshi Tanaka[1] François Bonnet[1] Sébastien Tixeuil[2] Yasumasa Tamura[1]

[1] Tokyo Institute of Technology, Tokyo, Japan
[2] Sorbonne Université, CNRS, LIP6, Paris, France

Abstract

Quixo is a two-player game played on a 5 × 5 grid where the players try to align five identical symbols. Specifics of the game require the usage of novel techniques. Using a combination of value iteration and backward induction, we propose the first complete analysis of the game. We describe memory-efficient data structures and algorithmic optimizations that make the game solvable within reasonable time and space constraints. Our main conclusion is that Quixo is a Draw game. The paper also contains the analysis of smaller boards and presents some interesting states extracted from our computations.

1 Introduction

1.1 Quixo

Quixo is an abstract strategy game designed by Thierry Chapeau in 1995 and published by Gigamic [1, 2]. Quixo won multiple awards, both in United States [3, 4, 5, 6] and in France [7, 8]. While a four-player variant exists, Quixo is mostly a two-player game that is played on a 5 × 5 grid, also called *board*. Each grid cell, also called *tile*, can be empty, or marked by the symbol of one player: either X or O.

At each turn, the active player first *(i)* takes a tile – empty or with her symbol – from the border (*i.e.* excluding the 9 central tiles), and then *(ii)* inserts it, with her symbol, back into to the grid by pushing existing tiles toward the hole created by the tile removal in step *(i)*. The winning player is the first to create a line of tiles all with her symbol, horizontally, vertically, or diagonally. Note that if a player creates two lines with distinct symbols in a single turn, then the opponent is the winner.

Figures 1a and 1b show the real game and our corresponding representation. Figure 1c depicts the resulting board after a valid turn by player O from the board depicted in Figure 1b: player O first *(i)* takes the rightmost (empty) tile of the second row, and then *(ii)* inserts it at the leftmost position shifting the other tiles of this second row to the right. A complete game on a smaller 4 × 4 board is given in Figure 8 on page 12.

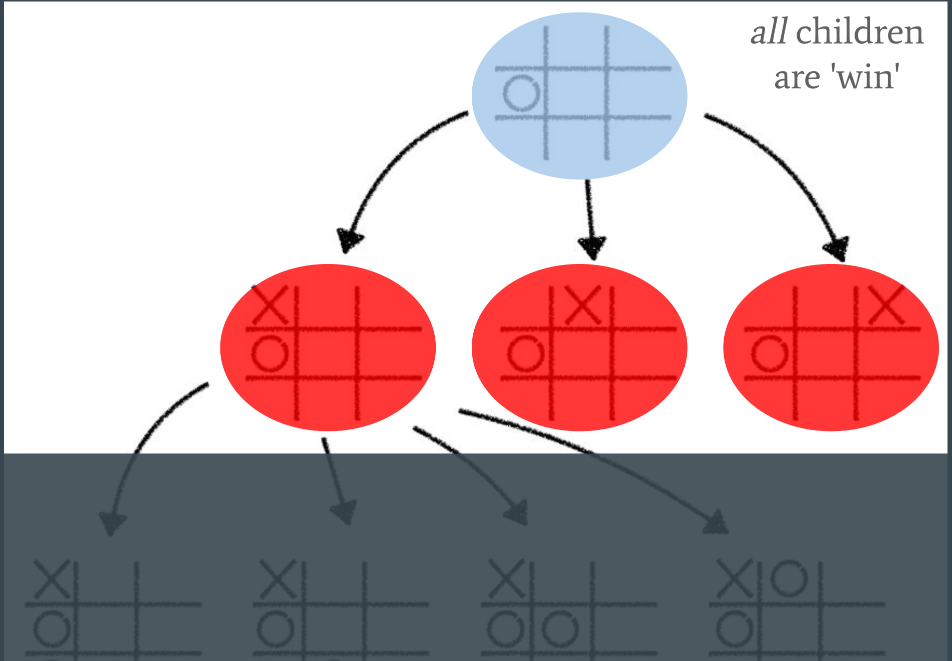
Quixo bears an immediate resemblance with some classical games such as Tic-Tac-Toe, Connect-Four, or Gomoku. However there are two major differences: *(1)* the board is "dynamic"; a placed X or O may change its location in subsequent turns, *(2)* the game is unbounded (in term of turns). The first point is

(a) Real game (b) Simplified illustration (c) After a valid move from player O

Figure 1: The two-player game of Quixo

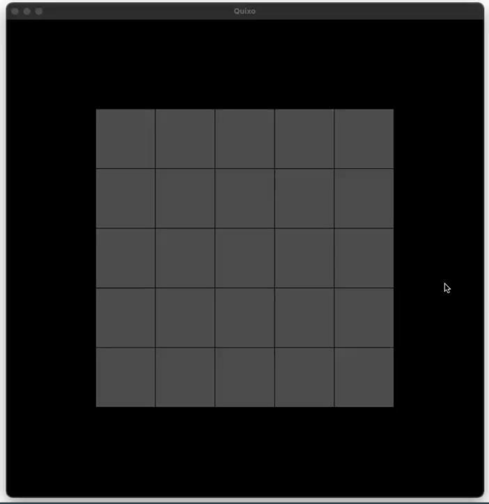
1

Every state of the board is either
first-player-win, first-player-loss, or draw



Computing optimal player: implementation

Some booooring CS implementation details, some optimization, and ***many many many many many many many many*** hours of debugging later…



Random player X against optimal player O

(X does not stand a chance, btw…)

Computing optimal player: some sense of scale

Full solution took ***~2 weeks*** to compute with 16 threads, even on the Zoo...

(for comparison, Tanaka et al. took ~2 weeks with 1 thread, and ~32 hours with 32 threads)

(... i.e. yes I know my implementation sucks)

Full solution occupies ***~200GB*** of hard drive space

(for the nerds here: 3^{25} total possible states × each state needs ≥2 bits to store win/lose/draw)

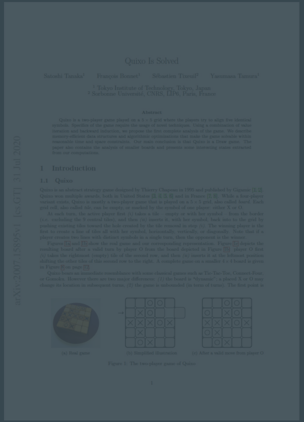
(... yes, this is still sitting in my laptop at this very moment, can't wait to delete it ...)

This problem grows (probably) ***super-exponentially***

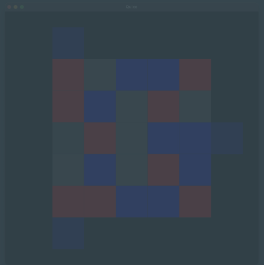
3X3 Quixo takes <1s and 4X4 Quixo takes ~20s to compute

3 main parts of my senior project

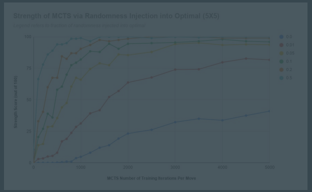
Compute Optimal Player


Quixo Is Solved

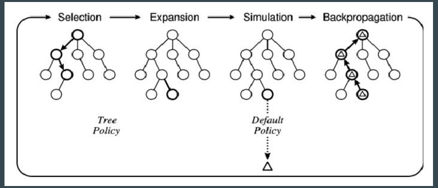
Implement Game Engine



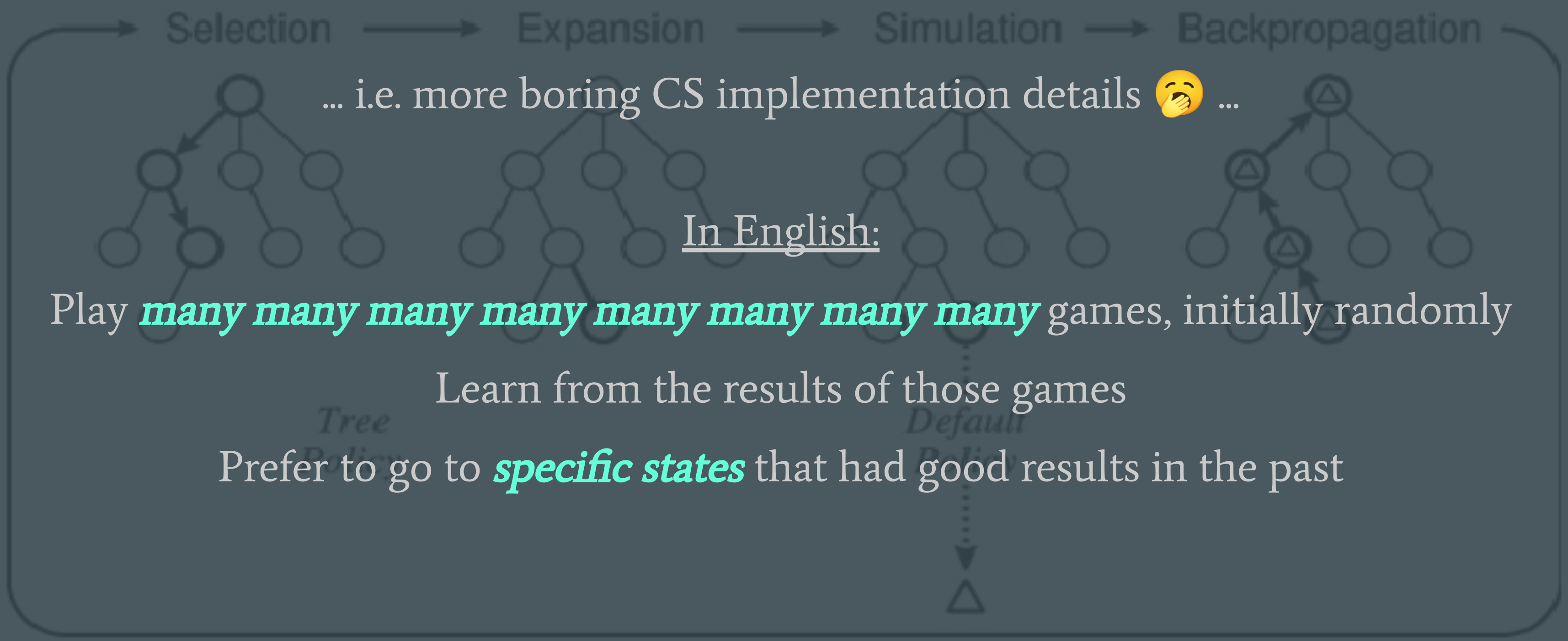
Experiments & Results!



Implement Computational Learning Agent(s)



Computational learning agent: Monte Carlo Tree Search (MCTS)



… i.e. more boring CS implementation details 🥱 …

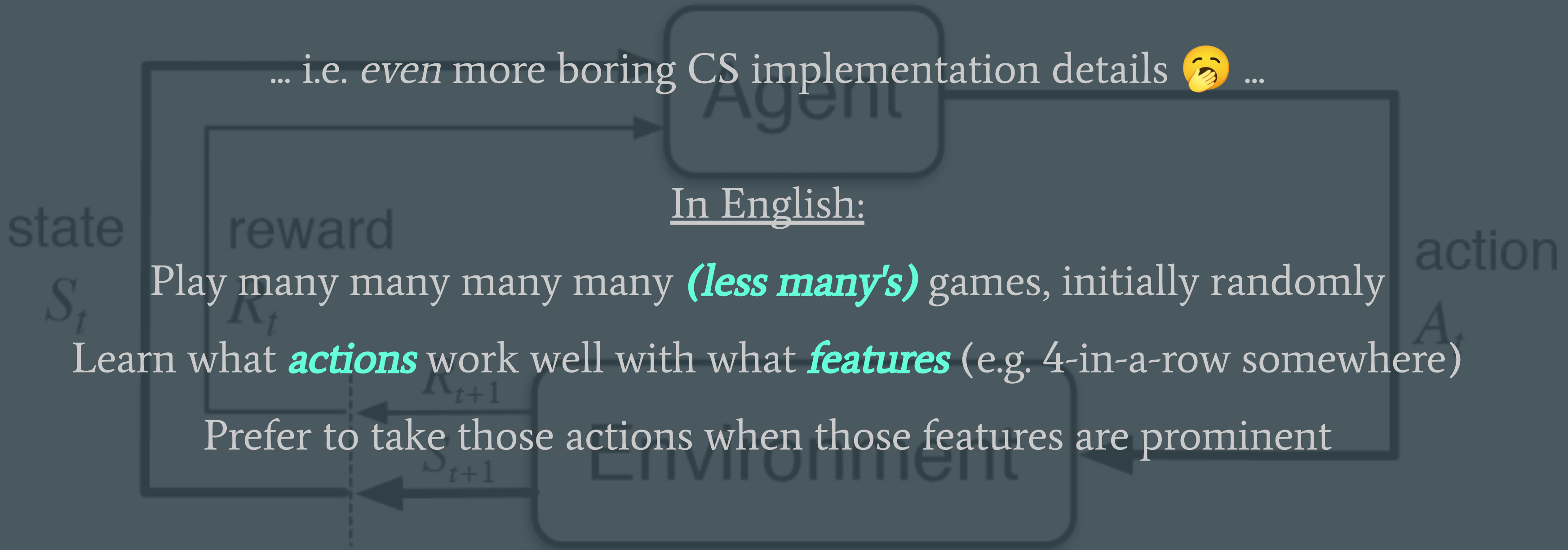
<u>In English:</u>

Play ***many many many many many many many many*** games, initially randomly

Learn from the results of those games

Prefer to go to ***specific states*** that had good results in the past

Computational learning agent: Q-learning (with feature approximators)



... i.e. *even* more boring CS implementation details 🥱 ...

<u>In English:</u>

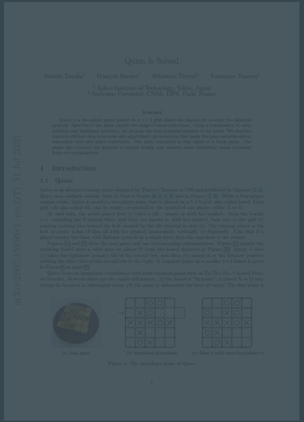
Play many many many many ***(less many's)*** games, initially randomly

Learn what ***actions*** work well with what ***features*** (e.g. 4-in-a-row somewhere)

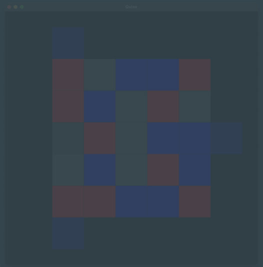
Prefer to take those actions when those features are prominent

3 main parts of my senior project

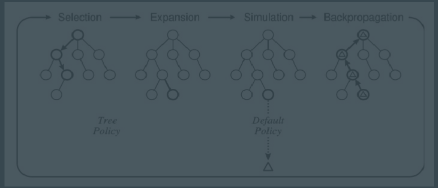
Compute Optimal Player



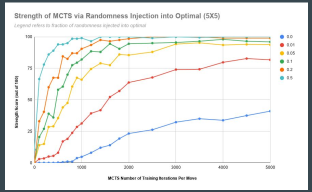
Implement Game Engine



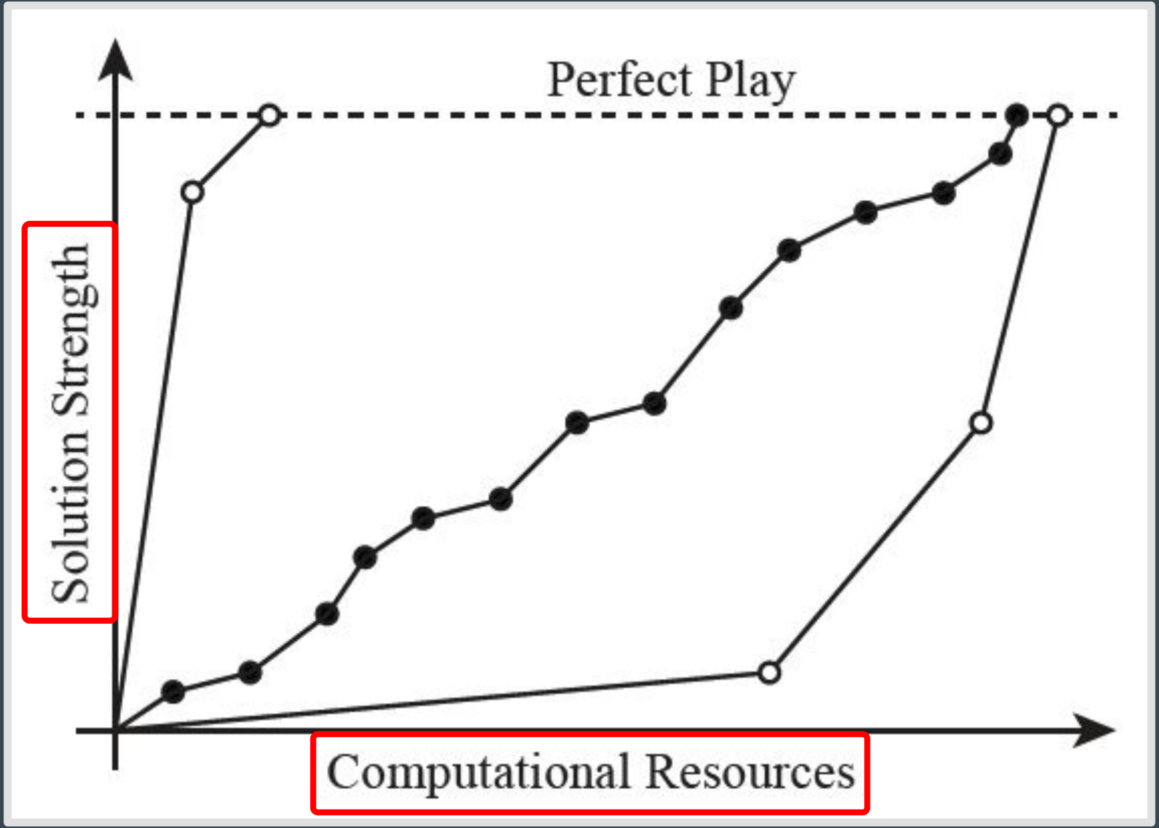
Implement Computational Learning Agent(s)



Experiments & Results!

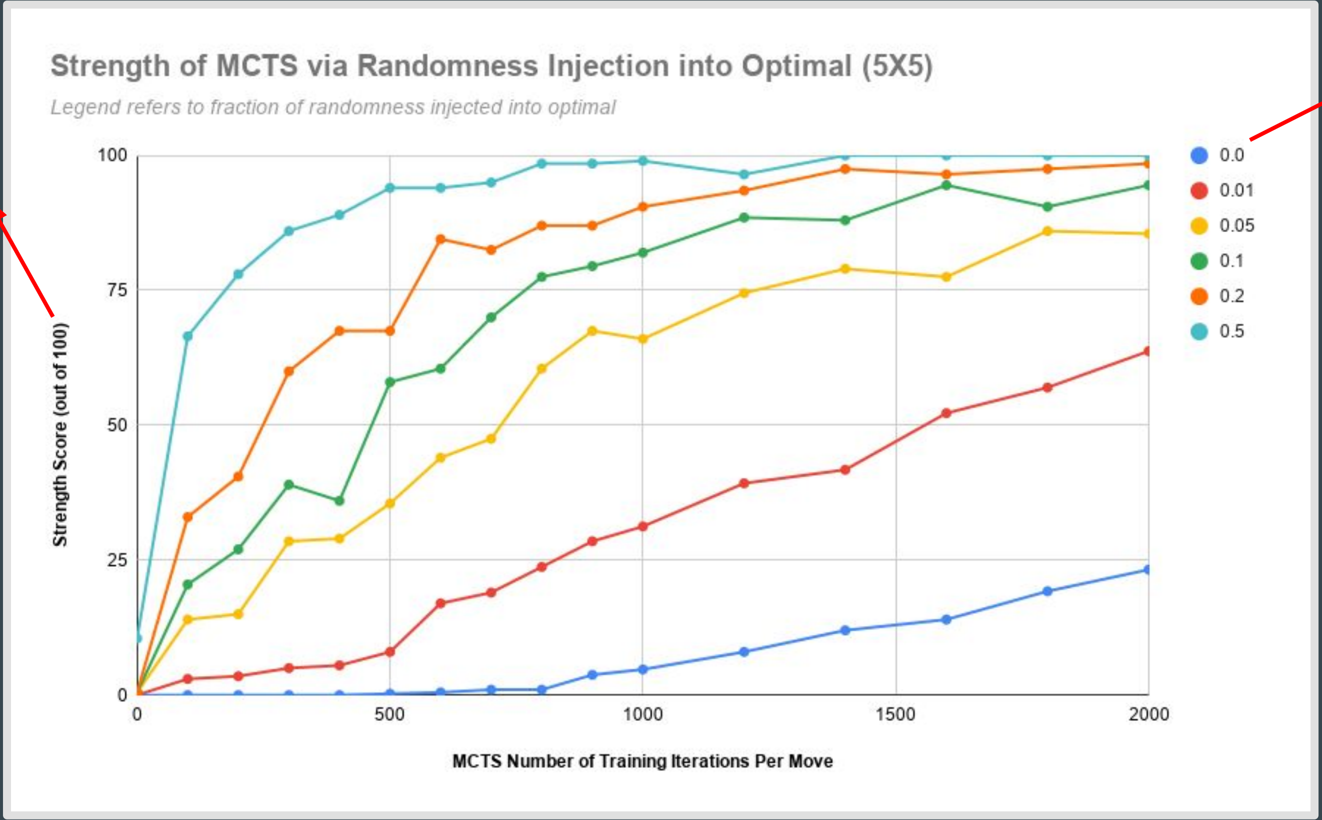


Results: … recall that we want to make this graph …



Results: solution strength via (1) game outcomes

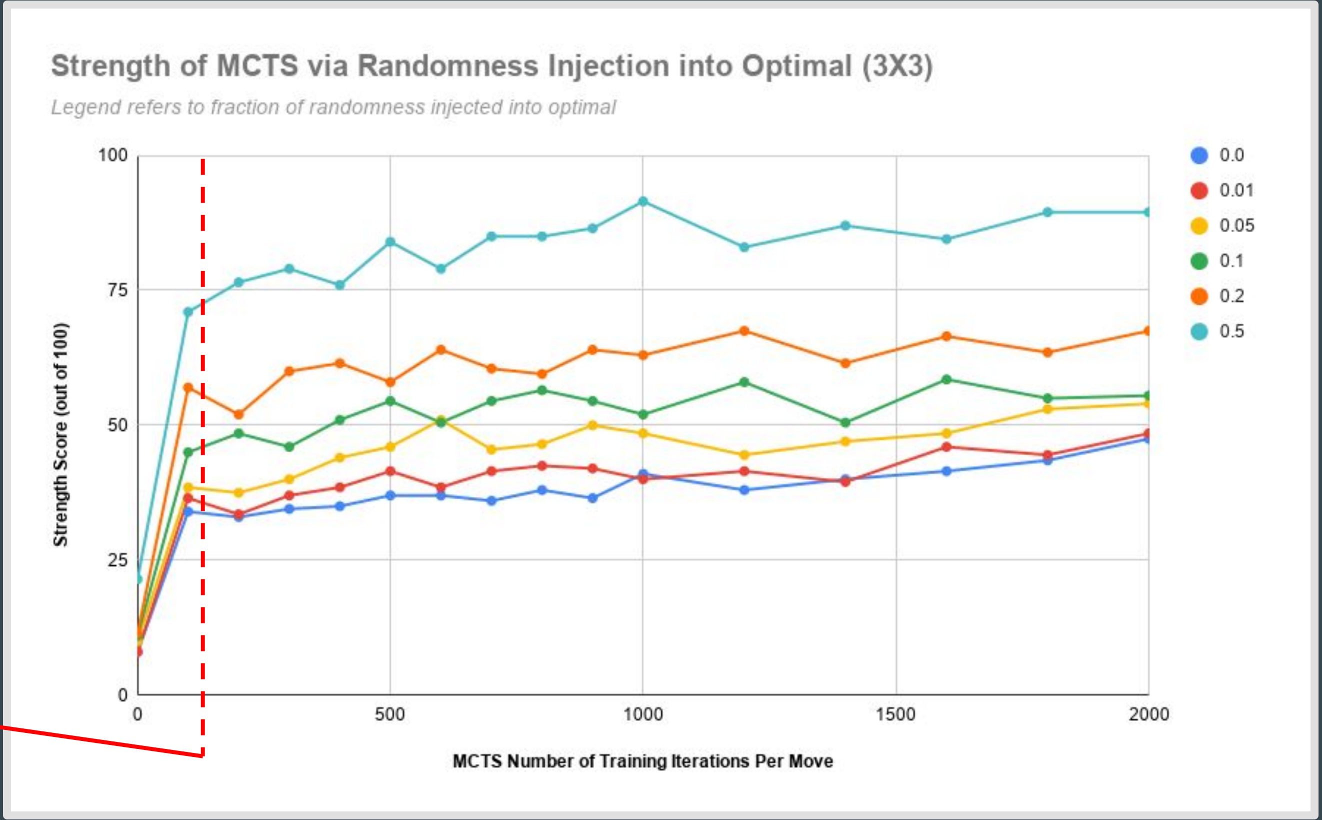
#wins + #draws/2
(over 100 games against optimal player)



Optimal player 'blunders' and makes a random move with some probability
*(otherwise, MCTS will **never** win)*

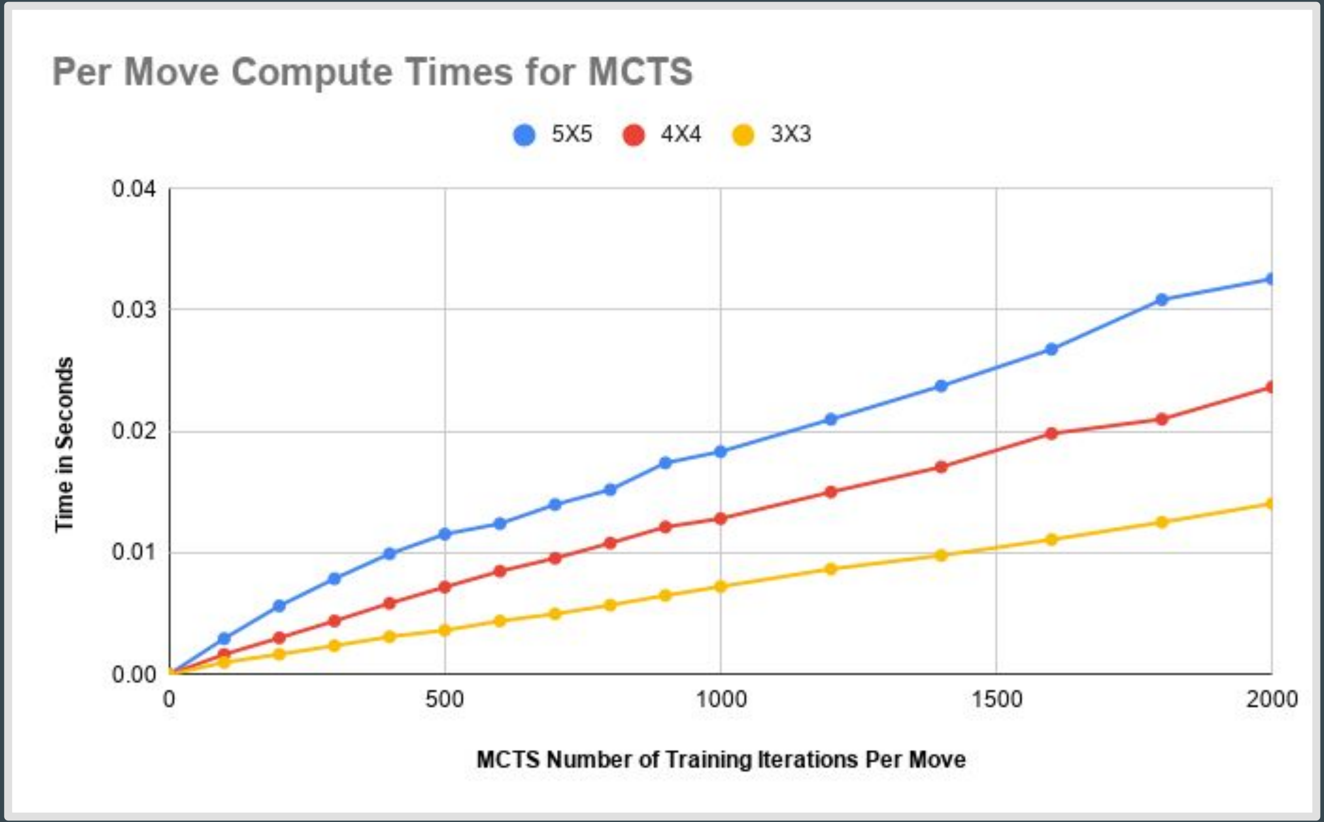
Note: even just this graph took ~7 hours to run

Results: solution strength via (1) game outcomes



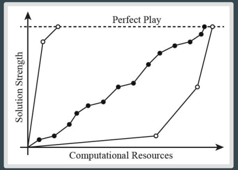
Much lower 'depth' for 3X3?

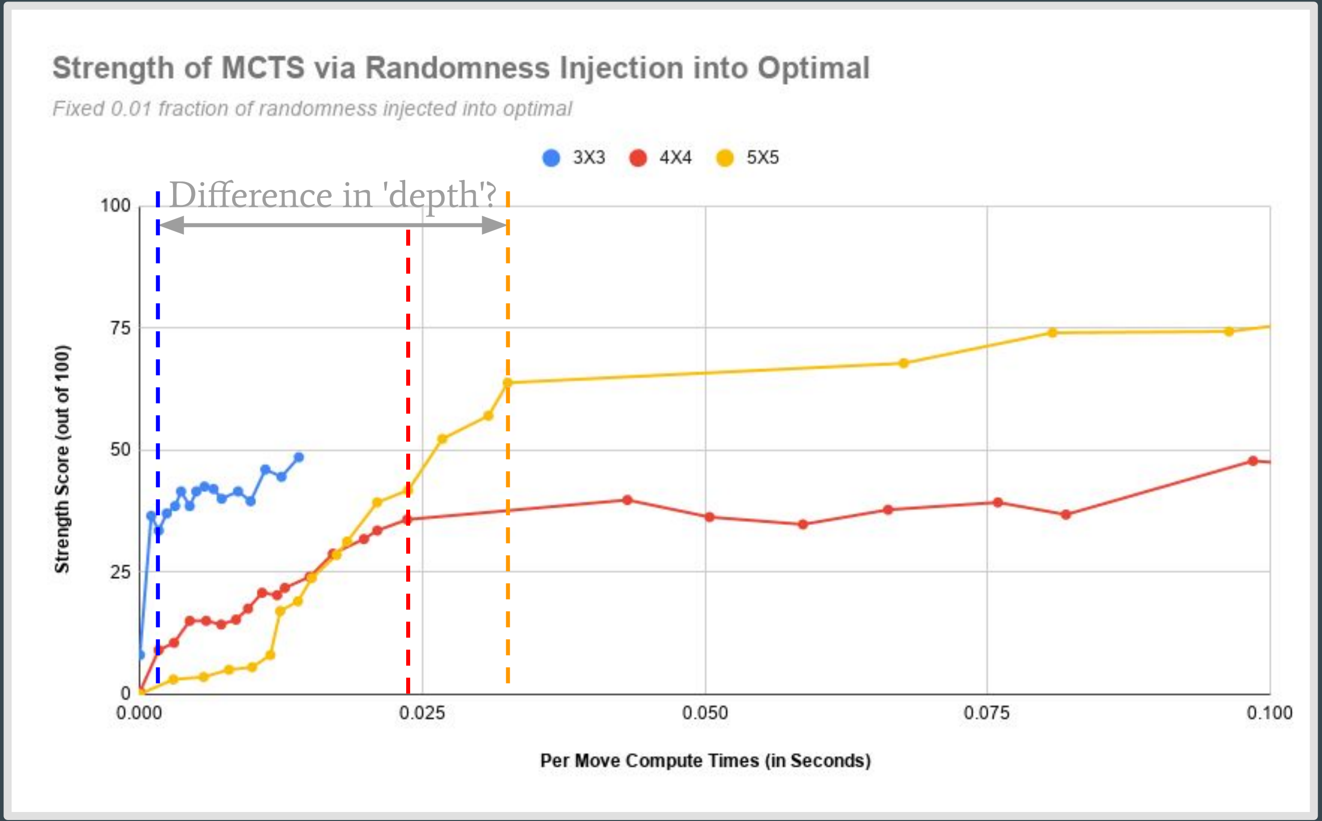
Results: side note: normalizing MCTS for different board sizes



Results: solution strength via (1) game outcomes

Aiming for...

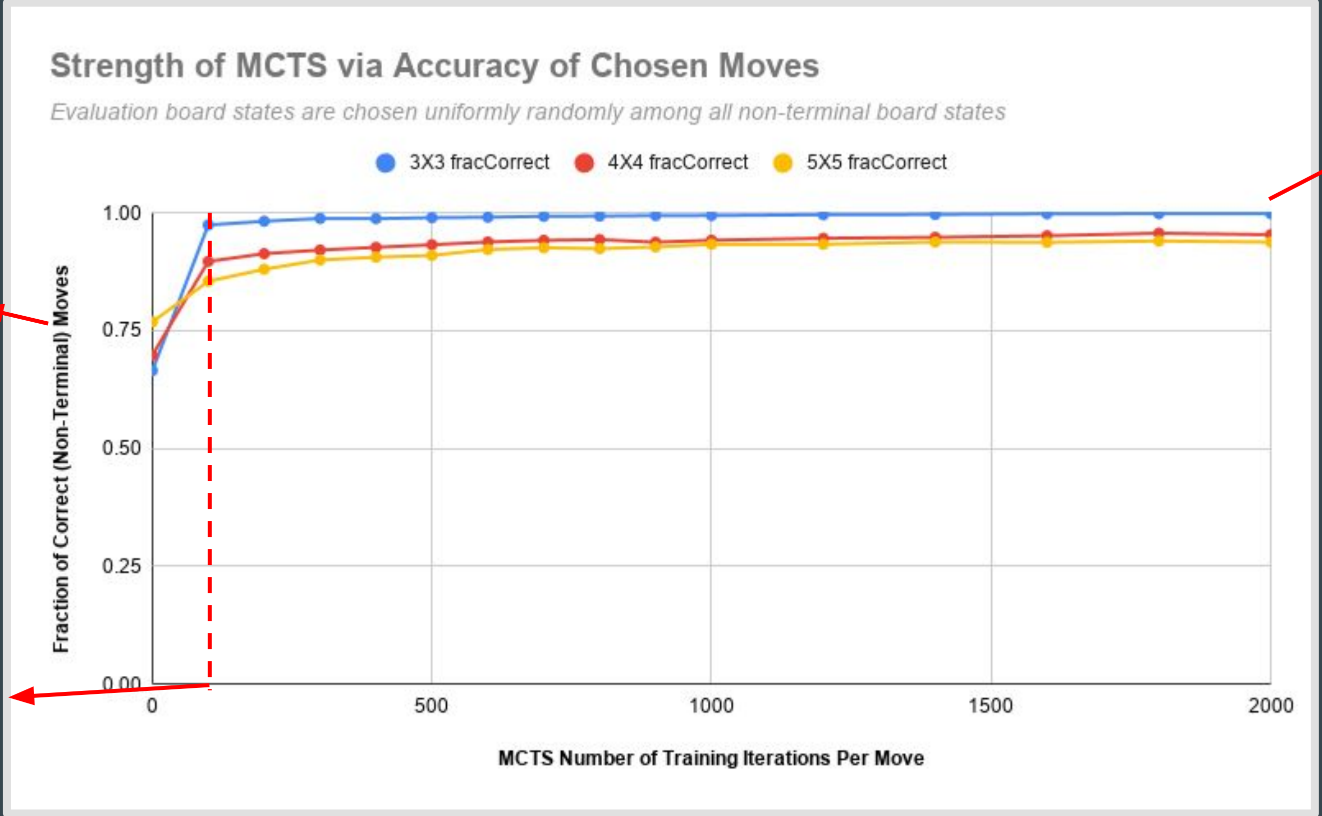




Results: solution strength via (2) accuracy of chosen moves

Since optimal player knows the *correct* moves

Not much difference in 'depth'?



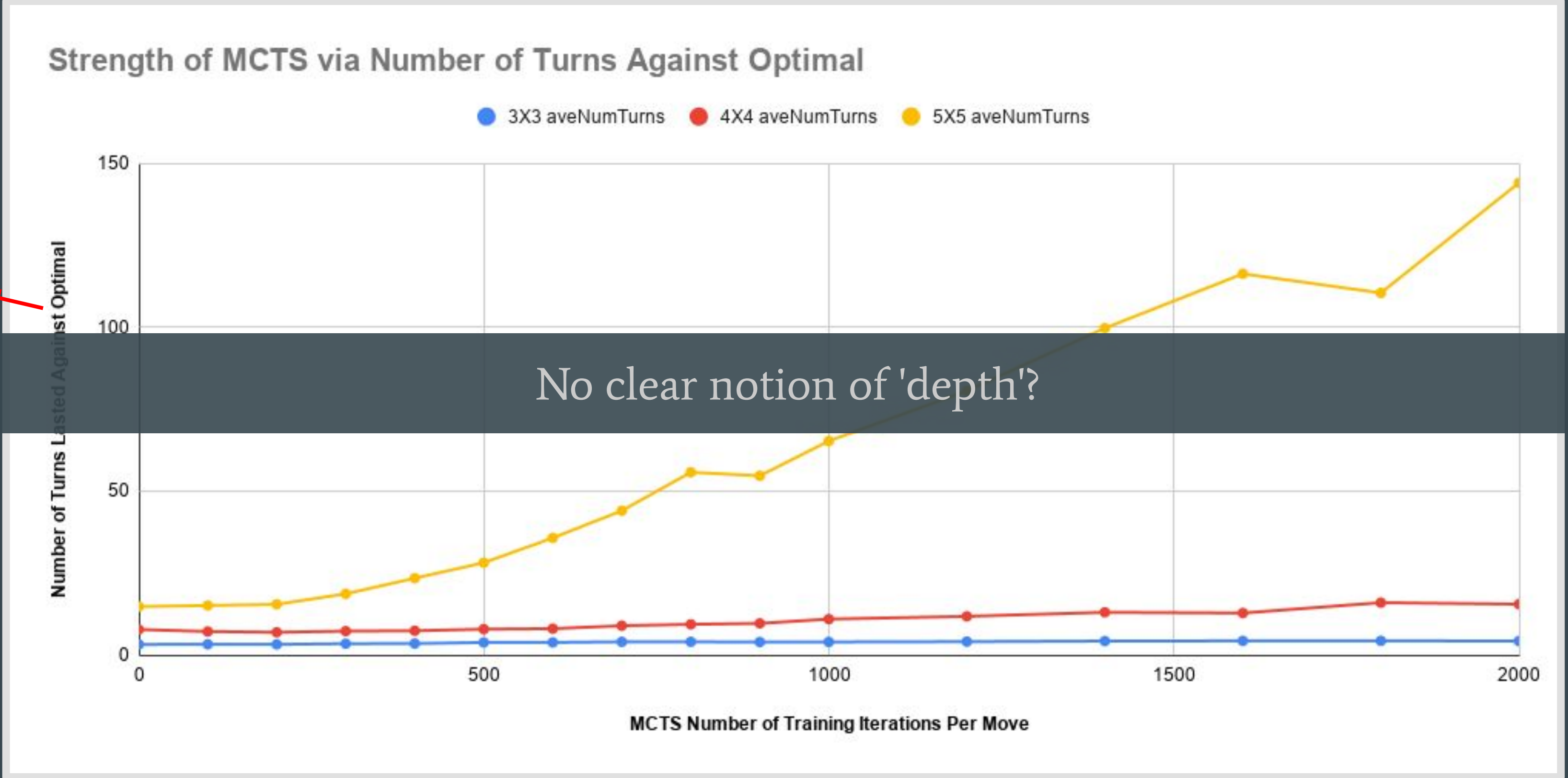
Note: each point is a fraction over 10,000 board states

At just 2000 training iterations, **99.8%, 95.4%, & 93.8%** accuracy for 3X3, 4X4, 5X5 respectively!

Together with game outcomes earlier, indicates that Quixo is a very 'mistake-punishing' game?

Results: solution strength via (3) number of turns against optimal

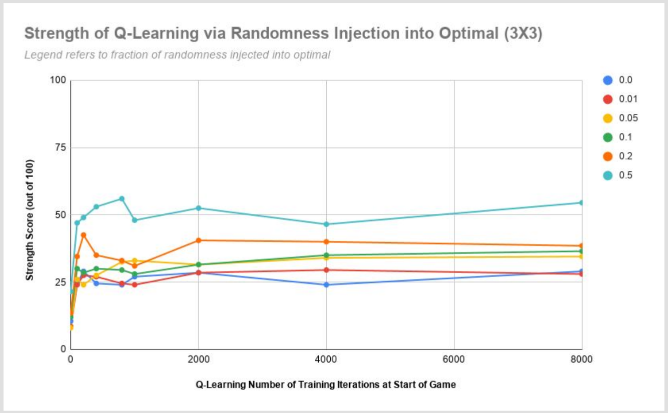
Recall that optimal *always* wins

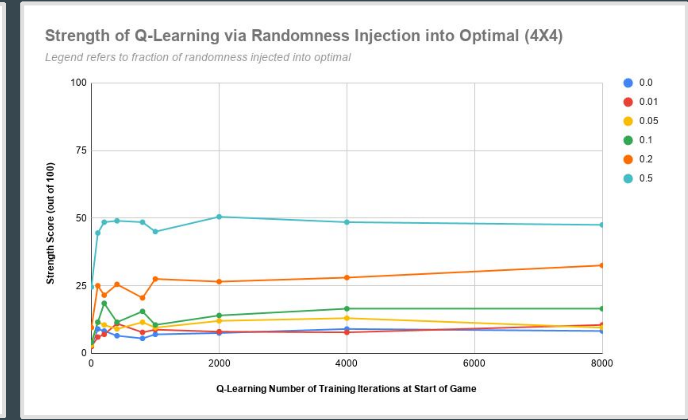


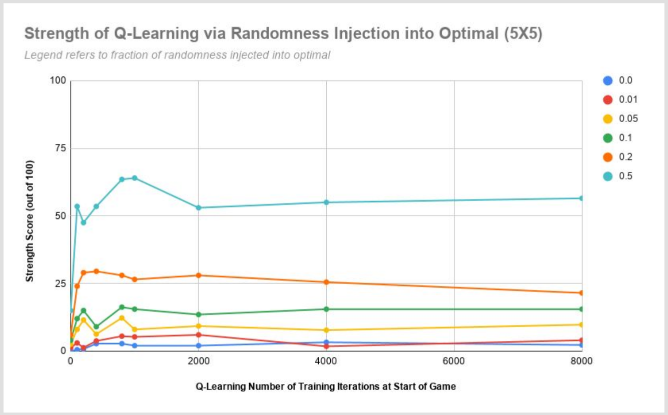
No clear notion of 'depth'?

Note: each point is an average over 200 games

Results: a final note about Q-learning







Q-learning does ***very poorly*** compared to MCTS

Indicates that Quixo is ***hard to learn via easily-defined features*** (hence is hard for humans)?

Thank You! Questions?