Claiming the Throne: A Digital Chess Game

Zongming Han
Game Science and Design
Northeastern University
han.zo@husky.neu.edu

ABSTRACT
“Claiming the Throne” is an interesting 2-player strategy board game that incorporates a famous ancient Chinese lore during the “Warring Age” into the Western feudal wars of chivalry, in the form of an ingenious combination of the Chess, Checkers and Chinese Checkers. The players will be able to use a myriad of possible combination of the strategies to reach a same goal: Race to the palace and become the King! This paper illustrates why and how the author made this game, and the insights the author gained from designing and implementing this game.

Keywords
digital game; chinese-checkers; chess; checkers; board game; digital board game; digital chess game

INTRODUCTION
Once upon a time during the famous “Spring-Autumn Warring Countries” Age (“Warring Age”) in the Chinese history, the ruler of a declining dukedom, Qi, passed away. The two sons of the duke, Xiaobai and Jiu, raised their army and raced to the capital to inherit the power. On their way to the capital their armies clashed, and a civil war broke out. During the time, Xiaobai’s advisor Bao Shuya, and Jiu’s advisor Guan Zhong were both brilliant strategies and tried their best to serve their lords in order for them to be the duke. In the end, Xiaobai reached the capital first and became the duke, and Jiu fled abroad to Lu. Jiu’s strategist Guan Zhong was imprisoned, and Bao Shuya was invited to be the prime minister of the country. But Shuya refused and recommended Zhong instead, who was once his enemy, saying Zhong was more capable than him in governance in order to make Qi great again. Xiaobai took his advice and years later, he with the great service of Zhong brought Qi one of the 5 most strong dukedoms across the land.

This game was based on this fascinating ancient lore from the 9th year of “Zhuang gong” of Qi from the history tome “Zuo Zhuan”(The Chunqiu; with the Zuo Zhuan, English Translation). Basically, two princes race to the palace to inherit the throne. They need to command their armies wisely, to both escort them safely, and try to capture the enemy prince at the same time. The game was put into a Western setting, under the consideration of localization, as well as a reference to a famous ancient Western World war-themed game, Chess. In this game, there are footmen(pawns), horsemen (knights), and pikemen (rooks), and similar to chess, each has their special abilities. Besides since fighting for power over the local throne was present throughout the human history anywhere in the world, this game naturally came from the ancient Chinese history and adopted a Westernized look and theme.

Then, why did I adopt a form of “Chess game”? The reasons are as below:

1. **Learn from these time-tested games** such as Chess, Checkers and Chinese Checkers about their rules and strategy depth, and try to **design and develop a game of a similar strategy depth**.
2. **Chess games are popular.** Chinese Chess (also called “Xiangqi”) is a strategy board game that is in the same family as Chess. According to the data from SteamSpy, a
game “Chinese Chess” (2017) has sold 20k copies since its first release in Oct 2017, which features an unchanged Chinese Chess rule and with an Artificial Intelligence opponent (AI). Also from SteamSpy, another game “Simply Chess” (2015) has sold more than 20k copies, which features an unchanged Chess rule. For this “Chess-like” game “Claiming the Throne”, there is a similar game depth, and has a wider search branch than in Chess.

3. Improve my game design and programming skills. Throughout the 6-month development process, a board game version was made after 5 iterations; then an arcade game was made to test the core-mechanic in preparation of the AI; and finally a 3D demo version was finished before a PAX East showcase. The development process technically involved data structures, fast and clean coding practice, advanced C#.NET/Unity programming skill, etc.

BACKGROUND
There were many war-themed turn-based strategy “chess-like” games (“chess-likes” in short) throughout human history, such as Chess, Checkers and Chinese Checkers. This section will go over the attractive features of them and compare the pros and cons. In the Method section, we will look at how this game “Claiming The Throne” inherits these two features from them.

Feature 1: Strategic depth
This paper assumes the reader has an understanding of the rules of the Chess and Checkers. So this paper will not cover the rules of them. Whereas, the rules of the traditional Chinese Checkers will be briefly introduced.

Chess is a strategy board game well-known for their depth. As per the “depth” of a strategic game (Lantz, Isaksen, Jaffe, Nealen, & Togelius, 2017), the deeper the “depth” is, the more players will generally enjoy it as a strategy game. This definition uses the following 4 rubrics to measure the “depth”. In general, the 4 rubrics measure the available choices and states in a strategy game, and the more, the “deeper”. The following is the analysis for Chess:

1. State Space. The more the possible states, the “deeper” the depth. Chess has 16 pieces, 6 piece types for each side and 64 squares. Chess has 3.8e+45 different states (How many legal states of chess exists. 2015)
2. Branching Factor. It measures the effective number of branches actually explored. Using an alpha-beta pruning technique (Pearl, 1982), one can use this formula to calculate the branching factor of the game, where n is the degree of a finite tree (The maximum number of child nodes a node can contain), and d is the depth of a finite tree (different from the “strategy depth”, this “depth”refers to the distance between the root node and the deepest child node):

   \[ R_{ab} = \lim_{d \to \infty} (N_{nd})^{1-d} \]

   **Equation 1.** The formula for calculating the branching factor of Chess.

   The \( N_{nd} \) is the average number of leaf nodes explored, given the different possible shapes of the tree.

   Using an alpha-beta pruning minimax search, Knuth and Moore (1975) claimed the branching factor was \( O(n/\log n) \). Pearl proved that it could be improved into \((0.925)n^{0.747}\). The “n” in both formula refers to the degree of a tree. In the case of Chess, n for the nodes of depth 1 ranges from 0 (no pieces can move. Draw) to 16 (all the 16 pieces can move). And the n for the nodes of deeper depth ranges from 0 (no available moves for a piece) to 27 (the maximum number of squares a queen can control). So using either approach, the branch factor will still be quite large.
3. Computational Complexity: The computational complexity refers to the efficiency of the algorithm for solving a certain problem (Papadimitriou, 2003). As stated above, the computational complexity for an alpha-beta pruning minimax search, which is widely used for a Chess AI algorithm, is $O(n/\log n)$, in term of the time efficiency (Knuth et al. 1975). Generally, it takes “EXPTIME”, or $O(2^n)$ in terms of time, where $n$ is the depth of the tree (Fraenkel & Lichtenstein. 1981).

4. Strategy Ladder Model: Strategy Ladder is a model for measuring the number of algorithm strategies and human strategies a game can have. Each strategy has a distinct difficulty factor $d$ across the ladder. The more $d$ a games has, the “deeper” a game from the lens of the Strategy Ladder Model (Lantz et al., 2017). In chess, the mechanics can be used across the rounds in different way, hence there can be different strategies, good or bad. The more types of mechanics exist, the more strategies, or “dynamics” these mechanics will form (Schell, 2015), hence the more $d$ there will be.

Whereas Checkers isn’t as “deep” as Chess as per these definitions:

1. State Space: Checkers has 12 pieces and 2 piece types for each side and 32 active squares. It has $5.0e+20$ possible states, which is less than the state space of Chess (Schaeffer et al., 2007)
2. Branching Factor. Each piece has at most 4 choices to pick each turn. In the equation, the degree $n$ for the nodes of depth 1 ranges from 1 to 12 (at least one piece and at least twelve pieces one player can have), and $n$ for the nodes of deeper depth ranges from 0 to 4 (0, 1, 2 for a regular piece or a partially blocked king, 3 or 4 for a king). It is obvious to see the ranges are smaller than that in Chess.
3. Computational Complexity: The best case is at the start, where there are 24 pieces on 32 squares, so for each side there are 6 choices to make. The worst case is each side only has kings that each king can make 4 choices. There’s also not many choices to make.
4. Strategy Ladder Model: There is a strategic ladder for Checkers, too. There are some strategies/dynamics for Checkers (Brown 2019) that can be distributed on a strategy ladder. But since the number of mechanics in Checkers is less than the number in Chess (Chess have more piece types, more pieces and more available squares hence more mechanics to use), the $d$ will be larger in Chess than in Checkers.

Before we dive into the depth of Chinese Checkers, let us take a look at its rules (Chinese Checkers on Wikipedia):

1. The goal is to move all of the pieces in one triangle to the triangle on the other side.
2. Players take turns moving a single piece, either by moving one step in any direction to an adjacent empty space, or by jumping in one or any number of available consecutive hops over other single pieces.
3. Can be played by 2 players, 4 players or 6 players at the same time.

It has a similar depth (in terms of a 2-player game) as Chess because:

1. State Space. Though Checkers only has 10 pieces for each side and every piece is same to each other, it has 73 cells. But notice that if it’s a 2-player game, the 4 side triangles are meaningless since they have nothing to do with the goal (moving all the pieces to the opposite triangle), hence there are only 73 - 6 - 6 = 61 useful cells. Since a Checkers player has 12 pieces and 32 squares, and a Chess player has 64 pieces and 64 squares, One can deduce that the Chinese Checkers have a large State Space than Checkers, but a smaller one than Chess.
2. Branching Factor. Unlike Chess or Checkers, each piece can continue to jump if applicable. This means one piece in one turn is able to end up in every cell, based on the formation of other pieces. What’s more, one piece can have multiple ending positions for each turn, based on the formation of other pieces. So the degree n for the nodes of depth 1 is 10 (Each player has 10 pieces throughout the game), and the degree of the nodes of deeper depth ranges from 0 to 36 (there are 73 cells, need to jump over 1 piece to jump father so the piece takes over 1 space. So in order to be able to cover the board, which is possible, one needs 36 pieces so 73 -36 -1 itself = 36 possible cells). This seems large, but remember that the scenario of a piece having a lot of available cells to land on is not very often. So it’s hard to say if Chinese Checker has a higher branching factor than Chess or Checkers, though it should be big.

3. Computational Complexity: Fonseca (2015) used a “depth-first” search using an approach similar to alpha-beta tuning, skipping the values lower than a heuristic value. But it’s still hard to say if Chinese Checker has a higher computational complexity than Chess or Checkers.

4. Strategy Ladder Model: As discussed above, the number of d, the number of possible strategies a player can take for the game closely depends on the branching factor. Since it’s hard to compare the branching factor of Chinese Checkers to Chess/Checkers, it’s hard to compute the d for them as well. That said, Chinese Checker can have a large d value as well, based on the big branching factor it has.

So we can safely assume that Chess, Checkers and Chinese Checkers all have ample game “depth”. Then the next question is, will a game sharing a combined quality of these four games have a deep game “depth”? The answer is affirmative and will be explained in the “Results” section.

**Feature 2: Single-Player and Online Multiplayer**

In the modern days, Chess and Checkers no longer just exist in their analog form, where two human players play face-to-face. There are digital games that support playing with a computer opponent, or match two players online to play. Some other digital variants infuse new stories into the old games to bring new gaming experience.

For example, Simply Chess (BlueLine Games, 2015) is a popular chess game on Steam\(^1\) that features a smart and efficient AI (though many other Chess programs do). Battle Chess (Interplay Entertainment Corp., 1988) is another popular chess game that features a humanoid avatar for each piece and animations for each piece on capturing. The modern version of this game, Battle Chess: Game of Kings (Olde Sküül &amp; Sculptured Software, 2015), though known for serious quality problems (a lot of crash and bug reports), features an online multiplayer match as well. As for Chinese Chess (Wang Wenxi, 2017), this game “Chinese Chess” is gains its popularity by featuring a single player mode against AI, a skirmish mode where you solve Chinese Chess puzzles, and a multiplayer mode where you play with other online players.

---

\(^1\) Steam is a digital distribution platform developed by Valve for purchasing and playing video games. More information on Steam: [https://en.wikipedia.org/wiki/Steam_(software)](https://en.wikipedia.org/wiki/Steam_(software))
Nevertheless, all of the games above inherit the exact same rules of these classic games. There are games that are different. Chessaria: The Tactical Adventure (Chessaria) (Pixel Wizards, 2018) added a fairy story to the game and used the Chess as the game mechanic. On top of that, Chessaria adds new rules. The players can arrange their formations before the battle starts, and they can have different numbers of pieces than the classic, for example: they can start with more than one queens. Similarly, another game Really Bad Chess (Zach Gage, 2016) randomizes the startout of the formation for both sides, for example: one side can have 4 queens, 8 bishops and 4 pawns. Breaking the rules of the existing games makes the games more interesting (Kaitlyn Burnell, 2012).

In conclusion, a good modern “Chess-like” game should have a good game depth, and support digital version that has single-player with AI and multi-player. The following section introduces the implementation of these game design criteria.

**METHOD**

This section introduces how the author designed and implemented this game “Claiming The Throne” to give it a game “depth” similar to Chess and Chinese Checkers, and improved the idea from the board game version to the digital version through multiple iterations.

**Strategic game depth:**

This game has the following rules that increase the number of game states and possible move branches.

Rules similar to Chinese Checkers: (difference are marked **bold**)

1. Uses a hexagon board **different from a traditional Chinese Checkers board.**
2. Players take turns to move a piece by either moving one step, or jumping over another piece/other pieces consecutively until can’t jump.
3. Two players try to move **only the prince piece** to **a common spot - the “palace”**.

Rules similar to Checkers:

1. A piece “captures” another piece by jumping over it. **A piece can move in any of the 6 directions.**

Rules similar to Chess:

1. There are 4 types of pieces: prince, footman(pawn), horseman (knight), and pikeman (rook). A prince and a soldier moves exactly the same, despite the importance. But a horseman and pikeman has their unique moves. That said, **every piece moves either one step or jumps over other pieces to advance.**
2. A piece only captures an enemy piece by **jumping over it**, not **landing on it.** You can’t land on a piece.

The author also added unique rules to make this game more interesting:

1. Conscription.
2. Conquer/Liberate Fief.
3. Pikeman’s “Leap” move and Horseman’s “Charge” move.
4. “Sacrifice” move.
5. “Loyal” move.
The details of the final rules are listed after the illustration of the rules changes through iterations at the end of the Method section.

*The rules listed above are the final rules after all the iterations.*

We will compare its strategic depth with Chess and Chinese Checkers in the Results section. For now, we can clearly see that this game has more rules and possible moves to explore, thus a deeper tree to search for available moves.

Nevertheless, there’s another aspect this game should achieve, as the following shows.

**Going digital (Single player and multiplayer)**

The author planned to make this game digital, which has a single player mode with an intelligent computer opponent, and a multiplayer mode that features online games.

To achieve this goal, the author followed these steps:

1. **Make a board game prototype to establish the core mechanics.**

   The purpose is to reduce the risk in the later development of the digital version. According to the Agile Development guideline (Greer & Hamon, 2011), the cost of making a change will skyrocket. That is to say, the cost of making a change will be much larger at the later development stage than at the earlier stage. Hence the need of making cheap paper prototypes before starting the digital implementation (Marco Mignano, 2016).

The author then developed a rule and playtested the game iteratively. For the first iteration, the author positioned the two sides on the left and right and made their prince race to the opposite side and then capital. The goal was to encourage the players to move, as Shell (2015) pointed out in the Lens of Indirect Control that a game needs to encourage players do what the designer wants them to do:

**Iteration #1**

Overall rules:

1. Each player brings the prince to the other side, then to the capital (the blue area in Figure 1).
2. Starts with 2 horsemen on the sides.

Move rules:

- **Step:** Same as in the Chinese Checkers, where a piece moves one step to an empty cell in any direction.
- **Jump:** Same as in the Chinese Checkers, where a piece jumps over one adjacent piece, and can jump continuously until there are no other adjacent positions in the final position.
- **Leap:** Like jump, but jumps over two adjacent pieces.
- **Charge:** After “jump”, can move one more step onto an empty position, in the same jump direction, and can “jump” again in any direction if “jumpable”.

---

The rules listed above are the final rules after all the iterations.
Piece rules:
- Footman/Prince: can step/jump.
- Horseman: can step/jump/charge.
- Pikeman: can step/jump/leap.

While I encountered some problems during the brief playtest with 3 players:

![Figure 1](image.png)

**Figure 1.** Two players start with 10 pieces, and race to the opposite side before going to the capital.

Problems:
1. Once a player has a major victory, it’s very hard for the other player to win.
2. There are few ways for the defeated side to recover quickly.

According to the Lens of Balance (Schell, 2015), the players need to feel right about the game, otherwise they will lose interest. Having a major defeat and not being able to recover soon is definitely a destroyer of the Lens of Balance. So I started to add more balance in order to preserve the “game flow” (Elias, 2012).
Iteration #2

Overall rules change (solution to the last iteration):

1. Players’ princes directly head to the capital (the blue area), but they have troops on the other side.

![Figure 2](image-url)

Figure 2. Each player starts with 12 pieces on both sides, and directly races to the capital.

The balance problem was rectified shown in the playtest of another 3 players. But new problems arose, which were about the motivation and the board.

Problems:

1. There’s no drive to use the “hidden” most powerful unit - pikeman.
2. Board is exactly the same as Chinese Checkers. Not creative/original.
3. The “white” area is never used.
4. Board is so small that it is not friendly to elders to clearly see the path, or select a piece.

Iteration #3

In the last iteration, the players start with horsemen and footmen only, though they can recruit a pikeman at any time. The playtests showed that there were no motivations to recruit a pikeman, but I wanted the players to use this elite troop type since it’s supposed to be an important mechanic in the game. Hence according to the Lens of Motivation (Schell, 2015), I made each player start with one pikeman so they can use it right away.
Also, I remade the board to make it bigger and more unique, and enlarged the capital area from a 1-4 triangle (10 cells) to a 1-5 triangle (15 cells), in order for a larger “contention space” during battle (Totten, 2014):

Overall rules change (solution to the last iteration):
1. Make a larger board with clear margins.
2. Make a board different from the default.
3. Able to arrange the places of 1 Footman, 2 Horsemen and 3 pikemen at the front lines at the beginning of the game.

![Figure 3. A larger board and a richer start setting.](image)

This round of playtest involved another 4 players, and they gave positive feedback on the board size and recognized the power of the pikeman through using them at the beginning. One of the players got interested in this game and played with me 4 rounds, and discovered a “dominant strategy” in the last round, which is a strategy that will almost ensure that you will win, hence damages the balance (Schell, 2015).
Problem:
  - A dominant strategy caused by the horsemen and footmen, because of the “charge” rule. Currently, a horseman can jump into any direction after charging. This means a horseman surrounded by footmen will be able to hit more than cells out of cells, as shown below, which severely breaks the balance. It’s almost impossible to defeat this strategy.

Figure 4. The dominant strategy arises when a horseman is guarded by footmen.
Iteration #4

Since a dominant strategy is a destroyer of game experience (Totten, 2014), I started to think of a way to eliminate this dominant strategy, right after we found this problem. The solution to the problem to answer the question of why this strategy will happen: the knight has powerful control because of the number of direction, plus the number of steps he can move, compared to a footman and pikeman. So I disabled the freedom of choice of the moving direction, hence the “charge” move:

Overall rules change (solution to the last iteration):
- For a “charge” move, the piece can only jump in the same direction as before, after charging. This reduces the power of the horseman yet remains more powerful than a footman.

This time, I conducted the playtest publicly on Winter Play event, to evaluate the rules after all these iterations. It was well-received and attracted many strategy game fans. In majority the players enjoyed this strategy journey through the game, but also one concern were raised:

Problem:
- The bottom two rows are never used:

Figure 5. The bottom two rows are never used.
One of the players made a suggestion of deleting the two rows, whereas another player suggested that I add different tile types to make use of these two rows. But Lens of Complexity (Schell, 2015) became my concern, since too much complexity may cause a loss of flow channel. So I didn’t add new tile types.

The suggestion of deletion of the two rows made a good point because as per the Lens of Control (Schell, 2015), the vast majority of the information should be “clear to the players”. That said, the problem of not using a space was not severe because I didn’t expect players to go to these, unlike I wanted the players to use the pikemen. So the author decided to keep these two rows for now.

Since the public test didn’t reveal any severe problems that needed a timely change, I made the final rules as shown below, considering the balance, flow channel, motivation and player effort (Elias, 2012), as shown below.

**Final Game Rules:**

1. Bring the “Prince” to the “Palace” in the “Capital”.
2. Place 1 “Pikeman”, 2 “Horsemen” and 3 “Footmen” at the designated 6 spots at the beginning of the game.
3. Bring a “Prince” to the “Palace” to win.

**Pieces:**

A Footman is the basic piece for this game. A Horseman or a Pikeman can move in the same way as a Footman, but both of them have their unique moves. A Prince moves the same as Footmen, and the only difference is their importance.

1. **Footmen:** can move, jump, and loyal.
2. **Horsemen:** can move, jump, charge, sacrifice and loyal.
3. **Pikemen:** can move, leap, sacrifice and loyal.
4. **Prince:** The most important. Once jumped over, game is over. Can move and jump.

**Places:**

1. **Palace:** The top corner of the board. Its coordinate is (-6, 12) (Figure 10)
2. **Capital:** The top orange 5-row triangle area (Figure 9)
3. **Fief:** The two black and red triangle areas on the edges of the board (Figure 9)
4. **Border:** Any position that is on the edge of the board. For example, from (-6, 7) to (-6, 12) they are all considered “Border”. Notice that (-6, 6) is not “Border”, while (-7, 7) is. (Figure 10)

**Moves:**

1. **Move:** move 1 step.
2. **Jump:** jump over 1 piece until can’t jump. If jumping over an enemy piece, remove the enemy piece from the board. Can’t jump over a piece on the border.
Special moves:

1. **Charge**: Move 1 extra step after jumping over a piece. Can only charge once during a turn. After charge, can only move or jump in the same direction. Available with Horsemen.

   ![Figure 6](image.png)

   Figure 6. “Charge” of a horseman.

   When a Horseman “charges”, he goes an extra step after jumping, in the same direction of his jump. After he charges, if there’s another piece adjacent to his new position, in the same direction of his jump, he may jump in that direction until he can’t jump any more.

2. **Leap**: jump over 2 pieces until can’t jump. If jumping over an enemy piece, remove the enemy piece from the board. Available with Pikemen.

   ![Figure 7](image.png)

   Figure 7. “Leap” of a pikeman.
3. **Sacrifice**: jump over an enemy piece on border (Figure 8). Available with Horsemen and Pikemen.

4. **Loyal**: jump over the enemy Prince on border. Available with Footmen, Horsemen, and Pikemen.

A horseman or a pikeman can perform a move called “loyal”/”sacrifice”, jumping off the board and sacrificing themselves to capture a piece on border. They cannot jump over a piece of the same faction that is on border. Besides, they can jump back if there’s a way.
5. **Conscription:** Spend one turn to recruit a Footmen/spend two turns to recruit a Horsemen or Pikemen in any of the 9 positions of the two fiefs, as shown below:

![Diagram of recruitment areas](image)

**Figure 9.** The positions a player can recruit their pieces.

6. **Conquer/Liberate Fief:** If one fief (either a 6-position triangle area or a 3-position triangle area) has one or more enemy pieces, the fief is called “Conquered” by the enemy, and cannot have any new recruits until there is no enemy in the fief, which is called “Liberated” by the player.
2. **Make a “logic layer” prototype**

Now that we had a working game mechanic, the next thing is to test whether making the digital board game version is feasible. Also, since the single-player mode of the digital version needs artificial intelligence (AI), the digital implementation should go in a similar way to the digital implementation of Chess: a “logic layer” for quick searches for available moves (for the AI algorithm), and the “representation layer” of the game, including the User interface, and the visuals of the game world.

Thus the author started to make the “logic layer” prototype for the game. For the convenience of representation, the author chose the .NET C# console application, and implemented the core classes of the game in an efficient way:

i. Board representation.

   Similar to Fonseca’s approach (2015) for implementing an AI for a Chinese Checkers game, I applied the Cartesian coordinate system on to the board:

![Diagram](image.png)

Figure 10. Design of the coordinate system for this game.
The reason to use this system is that it’s simple to cover every position on the board. As shown in the picture, the position in the center is (0, 0) and the position of the palace (at the top, which both princes are racing to) is (-6, 12) Thus the hexagon board (left) can be “distorted” into a rectangle board (right).

![Figure 11. Comparison of the board and the distorted board. (Left: the board; Right: the distorted board)](image)

ii. Object Oriented Design

After having a board representation, the author designed the classes using the Object Oriented Programming practices:

- Game Manager has 2 Players and a board.
- Game Manager has the following stages:
  - StartGame(): Initialize everything.
  - do{
    - WaitingForInput Stage: Get input of unit selection from the player. The player can be either human or AI.
    - UnitSelected Stage: Since the move can be continuous, the game gets further inputs from the player. Again, the player can be either human or AI.
    - TurnChanging Stage: Display the moves a player made in an animation after a player makes a move.
    - while(!GameOver); // Repeat the loop while the game is not over.
    - GameOver Stage: Display the results of the game after it is over.
- Board has a hashtable, mapping coordinates (two ints x and y) to a piece struct which stores info as bytes.
- Each player stores a Move struct, which has a list of AvailableMove structs, which contains byte info as well.
- The classes are designed to use as less memory (byte type fields) and computation (byte comparisons are fast) as possible, in order for the implementation of tree-search for the AI.
Here’s a brief illustration of the structure of the game.

The GameManager is a controller, who has two Player instances and a Board instance. A player records a Move each turn, which records AvailableMoves and MoveInfo. Static class such as Type, Faction and MoveType has functions returning single bytes representing each case for that type.
This version\(^2\) was to test the core mechanic in the preparation of the future implementation of the AI. Thus the author rolled out a human versus human console-application game as shown below.

![Figure 13. The logic layer prototype.](image)

The author conducted a brief user test on this prototype to test the digital representation of the core mechanics of the board game. There’s no AI - two human players take turns to make a move. An “P” represents a footman, “C” a horseman, “R” a pikeman, and “A” the prince.

\(^2\) The trailer of this arcade game prototype can be found at: [http://splashingspray.com/claiming-the-throne/](http://splashingspray.com/claiming-the-throne/)
3. 3D game implementation

Since the core mechanic of this digital board game is working, the author moved onto the making of the 3D game. The author used Unity\(^3\) and Visual Studio\(^4\) for designing the User Interface (UI) and coding. Basically, the author added the UI representation classes to the game. To differentiate the UI classes and the core classes, the following naming convention was used:

- CTTS prefix means it’s a class in the simulator core code.
- CTT prefix means it’s a class in the 3D game, usually for UI representation.

For example:

- CTTBoard is the UI representation of CTTSBoard;
- CTTPlayer is the UI representation of CTTSPlayer;
- CTTCCell is the cell object in the game used for UI representation;
- CTTUnit is the UI representation of CTTSPiece;

![Figure 14. Object-Oriented Design for the 3D game prototype.](image)

The GameManager keeps track of the players and the board, so CTTPlayer and CTTBoard don’t have a reference to CTTSPlayer and CTTSBoard. Whereas, in order to play animations of unit moving and capturing, the CTTUnit has a reference to CTTSPiece, and stores both the mock move position (mr_mockCell) and the final position (mr_cell).

---

\(^3\) More information about Unity: https://en.wikipedia.org/wiki/Unity_(game_engine)

The piece models were made by the author using MagicaVoxel⁵, which is a modelling tool that helps make 3D models.

![Image of 3D model](image)

Figure 15. The horseman model I made in MagicaVoxel.

The game/UI animations were made using both Unity coroutines and the free version of DOTween plugin.

⁵ More information about MagicaVoxel: [https://ephracy.github.io/](https://ephracy.github.io/)
The iterations of the making of the board:

At first, the author made a linked-node board, where pieces stand on nodes, which are connected by lines:

![Figure 16. The linked-node board.](image)

According to the second rule of Nielsen's Usability Heuristics (Jakob Nielsen, 1994), the design of the human-computer interaction interface should “speak the users' language”, in other words, cater to the user experience of the similar games. Though a Chinese Chess (Xiangqi on wikipedia.) or Chinese Checkers board (Chinese checkers on wikipedia.) is a linked-node board, A Chess or Checkers board is a solid board. Since this game for now targets at the Western market, a solid board fits better in terms of the current user experience for a digital chess game.

Thus the board becomes a solid board, where each node is a hexagon next to each other:

![Figure 17. The solid hexagon board for Claiming the Throne.](image)
Notice that in the linked-node board, there is a background projecting a battleground (an open field with trees), but in the solid board, there is no background. That’s because making the background is optional for the current development, which focuses more on the implementation of the gameplay, than the polishing of the user experience.

4. **(Future work) AI implementation and online-multiplayer feature.**
The core mechanic has been made to cater to the needs of a fast AI algorithm that uses little memory. But I didn’t implement AI algorithm and the multiplayer feature in this iteration.

The final 3D version prototype of the game is available on this website: splashingspray.com
It’s a human versus human game.

Figure 18. The 3D game prototype.
RESULTS

This section discusses about the strategic depth of the game, and the result of the playtests for the three prototypes.

Strategy depth of this game:

In the “Background” section, we asked a question: “will a game sharing a combined quality of Checkers, Chinese Checkers, Chess and Chinese Chess have a deep game “depth”? The answer is yes, with the following reason and results:

1. State Space. Each player starts with 12 pieces (Chinese Checkers has 10, Chess has 16, and Checkers has 12), on a board consists of 208 cells (Chinese Checkers has 121, Chess has 64 and Checkers has 32). While unlike any of the four games, in this game each player can spend one or two turns to recruit a footman/horseman/pikeman at any of the 9 positions within areas called “fief”. This opens up the state space of this game.

2. Branching Factor. Each piece can either move one step, or jump over another single piece consecutively. Besides, a footman/pikeman piece can charge/leap, and both can “loyal” (a move to jump off the board). Finally, a player can choose to spend a turn or two to recruit a piece. Thus this game has a larger branching factor than Chinese Checkers.

3. Computational Complexity: Though the data structure for this game is consist of bytes, each piece has multiple possible positions to go to, and can go to several positions in a row, with a combination of jump,charge, and leap. In Chinese Checkers, a piece can also moves to different positions consecutively in a turn, but there’s only one type of pieces. In Chess, once a piece moves to a position, the turn is done.

4. Strategy Ladder Model: There are different strategy levels for this game. An easier strategy, which has been noticed often times during the Winter Play and PAX EAST playtests, is to use two pieces jump over each other to advance. A stronger strategy is to set up a formation of pieces and then quickly move them to the front. Furthermore, since you have two winning conditions in this game (bring the prince to the palace/capture the other prince), you can develop either an offensive and defensive strategy, each of which has multiple levels of difficulty, such as forming a wall near the capital, or assault the fief to disable recruit. Whereas in Chinese Checkers, since there’s no capture mechanic, the main strategy would be only about building smart “bridges” to get your pieces into the positions faster. Chess, on this other hand, also has different levels of strategies, such as the “control the center” opening moves and avoid doubling pawns.

Hence we can see this game has a depth deeper than Chinese Checkers and similar to Chess (The author didn’t do an accurate calculation comparing the depth of Chess and this game). Now that we know this, let’s look at how the playtesters think of this game.

Board game public playtest on Winter Play event:

Winter Play is a game showcase event in Boston, MA, U.S. hosted by PlayCrafting. I showed this game with its board game version. This game became one of the most popular board games on the event. There were 13 rounds of play tests conducted during the 3 hours of the event, and 9 of them were a player against the author, and the rest 4 of them were a player against another player. There 2 pairs of players played 2 rounds, 2 players played 2 rounds against me, and 1 player played 3 rounds against me. Each player was asked a brief question when they were going to leave: “How do you think of this game”. Most of the answers were compliments, plus some comments such as “I like Chinese Checkers in this way”.
There was one player who has beaten the author at his third round, and commented that he likes “the strategy depth and the number of possible choices you can make”.

During this playtest, most of the players were tenages from 8 to 24. There were two players above this range and nobody below this range.

![Figure 19. Players enjoying this game on the Winter Play event.](image)

**Arcade game private playtest:**

This brief playtest involved 3 players who were fellow students. They confirmed that the prototype contained the core mechanic of this game, thus the author can move on to add UI representation on it.

**3D digital board game public playtest on PAX EAST:**

PAX EAST is a game festival happening annually in Boston, MA, U.S. The author presented the 3D digital prototype as an outstanding student game at Northeastern University’s booth during the event. The game was on show for one day and got 7 rounds of playtests, 2 out of which were a player against another player other than the author. Like the playtest of the board game, each player was asked a question about their feeling about this game, and the feedbacks were positive. But unlike the board game playtest, Most players were highschool students, parent and children, and college students. In other words, the player
demographic was heavily influenced by the location I was presenting my game, which was the Northeastern University booth.

Figure 20. Players enjoying this game on the PAX EAST.
CONCLUSION
This digital board game “Claiming The Throne” is a 2-player turn-based strategy game inspired from and ancient Chinese lore, and some famous classic board games such as Chess, Checkers and Chinese Checkers. During the game, two players command their troops to bring their prince to the palace, while protecting the prince from being captured.
The game is interesting because of the depth of its strategy. The development of this game so far has been through 3 stages: board game, a digital version without UI representation, and a digital 3D version. All of them are still prototypes. The author is continuing making the game to add AI opponent, improve user interface, and add online play mode.

ACKNOWLEDGMENTS
Thanks to all the playtesters:
Ryan Nuz, Carter Seggev, Uttkarsh Narayan, Ruiming Wang, Josiah Velosso, John Hart, Christoffer Holmgård, Yoshi Lu, All the players at the Winter Play, Jiawei Tang, Stan Fan, Ricardo Chiesa, Carlos Corona, Varun Sriram, All the players at PAX EAST.

Thanks to Stan Fan for the help in presenting my game at the Winter Play, and the lovely poster he made for Claiming The Throne.

Thanks to the author’s little cousin Zonglin Han, without playing with whom I would never come up with this idea of “Claiming The Throne”.

If I missed anyone who did playtest please do let me know and pardon me.

Last but not least, thanks, Christoffer, for advising!
ANNOTATED REFERENCES

Chinese Checkers on wikipedia. Retrieved March 27, 2019, from


https://www.nngroup.com/articles/ten-usability-heuristics/


https://www.thesprucecrafts.com/how-to-win-at-checkers-411170


https://steamspy.com/app/722670


Zuo Qiuming. (late 4th century BC). *The chunqiu; with the zuo zhuan (english translation).* Retrieved April 19, 2019, from

http://www2.iath.virginia.edu/saxon/servlet/SaxonServlet?source=xwomen/texts/chunqiu.xml&style=xwomen/xsl/dynaxml.xsl&chunk.id=d2.9&toc.id=d2.9&doc.lang=bilingual

**APPENDIX**

Claiming the Throne page: splashingspray.com/claiming-the-throne

PAX Demo version downloadable: https://splashingspray.itch.io/ett