

# **Mastering the game of Abalone using deep reinforcement-learning and self-play**

Ture Claußen

Bachelor thesis in "Applied computer science"

November 17, 2021



**Author** Ture Claußen  
Matriculation number: 1531067  
tu.cl@pm.me

**First examiner:** Prof. Dr. Adrian Pigors  
Abteilung Informatik, Fakultät IV  
Hochschule Hannover  
adrian.pigors@hs-hannover.de

**Second examiner:** Prof. Dr. Vorname Name  
Abteilung Informatik, Fakultät IV  
Hochschule Hannover  
email-Adresse

### **Declaration of authorship**

I hereby declare that I have written this thesis independently without any help from others and without the use of documents or aids other than those stated. I have mentioned all used sources and cited them correctly according to established academic citation rules.

Hannover, November 17, 2021

Signature

# Contents

- 1 Introduction 6**
- 2 System architecture 7**
  - 2.1 Software . . . . . 7
    - 2.1.1 Training framework . . . . . 7
- 3 Analysis 8**
  - 3.1 Environment . . . . . 8
    - 3.1.1 Abalone rules . . . . . 8
    - 3.1.2 Abalone complexity . . . . . 9

# List of Figures

3.1	Basic moves [S.A] . . . . .	8
3.2	Sumito positions allow pushing the opponent's marbles [S.A] . . . . .	8
3.3	Counts of moves available for random for random player in 5 games . . .	9

# List of Tables

3.1 Abalone in comparison with other games [Cho09] . . . . . 10

# 1 Introduction

Programming is severely limited by our ability to properly model the problem in advance. Problems with large complexity would require large efforts, that at a certain point just become unfeasable. Instead, learning the task from a blank slate, *tablula rasa*, would allow to imitate human adaptability.

Instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulates the child's? If this were then subjected to an appropriate course of education one would obtain the adult brain. Presumably the child-brain is something like a note-book as one buys it from the stationers. Rather little mechanism, and lots of blank sheets. [...] Our hope is that there is so little mechanism in the child-brain that something like it can be easily programmed. [TUR50]

In the field of artificial intelligence the construction of game-playing

## **2 System architecture**

### **2.1 Software**

#### **2.1.1 Training framework**

As there are existing frameworks that have implemented the system described in the AlphaZero paper in a more general and adaptable fashion, it has to be considered building on their foundation.

# 3 Analysis

## 3.1 Environment

### 3.1.1 Abalone rules

The goal of the game is to push six of the opponent's marbles off the playing field. The game's starting position is depicted in figure 3.1 (a). One, two, or three adjacent marbles (of the player's own color) may be moved in any of the six possible directions during a player's turn. We differentiate between broadside or "side-step" moves and "in-line" moves, depending on how the chain of marbles moves relative to its direction, which is shown in figure 3.1 (b) and (c).

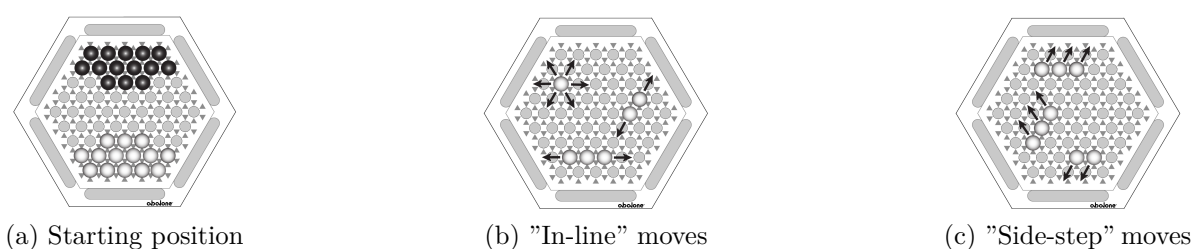


Figure 3.1: Basic moves [S.A]

A move pushing the opponent's marbles is called "sumito" and comes in three variations, as shown by figure 3.2. Essentially, the player has to push with superior numbers and the opponent's marbles can not be blocked. This is the game mechanic that allows for pushing the marbles out of the game and winning.

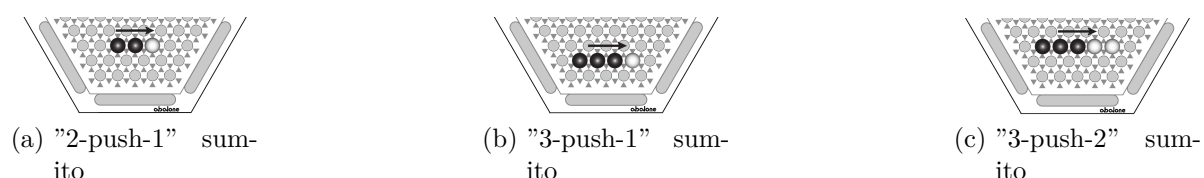


Figure 3.2: Sumito positions allow pushing the opponent's marbles [S.A]



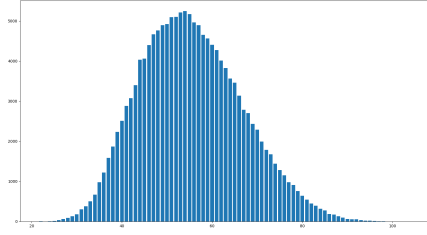


Figure 3.3: Counts of moves available for random for random player in 5 games

### 3.1.2 Abalone complexity

An important characteristic of a game environment is its complexity, which can be described in two relevant dimensions.

**State space complexity** The state space is the collection of all possible states the agent can be in. [RN21, p. 150] For Abalone this means we have to consider all possible board configurations with different numbers of marbles present. Additionally, we would have to correct duplicates that arise from the symmetries of the board. Ignoring this fact the following gives a good upper bound:

$$\sum_{k=8}^{14} \sum_{m=9}^{14} \frac{61!}{k!(61-k)!} \times \frac{(61-k)!}{m!((61-k)-m)!}$$

**Game tree complexity** The game tree defines the dependencies between board positions (nodes) and moves (edges). First we consider the branching factor (how many moves are possible in one position) of the game tree, which is on average 60. We combine that number with the height of the tree to get the total number of leaves. As the length of a game varies greatly, we use the average length of a game which is 87:  $60^{87}$  [Lem05]

Putting Abalone's complexity in relation with other popular games, its state space complexity is on the same level as Reversi, whilst its game tree surpasses chess in complexity (c.f. table 3.1)

Game	state-space complexity (log)	game-tree complexity (log)
Tic-tac-toe	3	5
Reversi	28	58
Chess	46	123
Abalone	24	154
Go	172	360

Table 3.1: Abalone in comparison with other games [Cho09]

# Bibliography

- [Cho09] Pascal Chorus. Implementing a computer player for abalone using alpha-beta and monte-carlo search. Master’s thesis, Citeseer, 2009.
- [Lem05] NPPM Lemmens. Constructing an abalone game-playing agent. In *Bachelor Conference Knowledge Engineering, Universiteit Maastricht*. Citeseer, 2005.
- [RN21] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, Inc, fourth edition, 2021.
- [S.A] Abalone S.A. Abalone rulebook. <https://cdn.1j1ju.com/medias/c2/b0/3a-abalone-rulebook.pdf>.
- [TUR50] A. M. TURING. I.—COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, LIX(236):433–460, October 1950.