Title	コンピューターゲームにおける人間らしさの分析と再現
Author(s)	テンシリリックン，シラ
Citation	
Issue Date	2020-03-25
Type	Thesis or Dissertation
Text version	ETD
URL	http://hdl.handle.net/10119/16653
Rights	
Description	Supervisor:池田　心，情報科学研究科，博士

Contents

Chapter 1

Introduction

During a long period of research, Artificial Intelligence (AI) has progressed, and in recent years, has had an impact on our lives. Many innovations, such as self-driving cars and medical analysis systems, have introduced AI methods to improve the quality of life in various ways. AI has become a part of the modern lifestyle now.

Research on AI has been conducted broadly, for example, natural language processing (NLP), computer vision, or speech recognition. Recently, these studies have shown dramatic results, due to the availability of large-scale computing with GPU. In computer vision, AI can perform separation and classification tasks accurately. Currently, research has expanded to include automatic image and video captioning. In NLP and speech recognition, in 2016, Google announced that 20% of mobile users performed searches with the voice function. This shows the improvement and reliability of the current system. The focus of research in this field is now shifting to interaction with humans [1].

Games are also an important target of AI research because many games are well known and the rules can be simply defined, then it is relatively easy to reproduce or compare different methods. Although games are simple, creating smart computer players is not since it requires various intelligent abilities such as search, optimization, or reasoning.

The first and major purpose of game AI research is to create strong computer players. During 60 years of research on game AI, one of the most notable progress was shown in 1997, when IBM's computer chess player, called Deep Blue, beat Garry Kasparov, the world champion [2]. In the 2000s, the performance of computer shogi players (mainly oriented from Bonanza) reached the professional level. Moreover, in 2016, AlphaGo beat Lee Sedol, who is one of the strongest professional players and won 18 times world champion [3]. The triumph of computer players, such as Deep Blue and AlphaGo, over human professional players proved their ability. In addition, in modern computer games, such as Super Mario Bros. [4] and Starcraft [5], computer players have become stronger than intermediate human players. A famous example can be found on YouTube, where the A* player for Infinite Mario Bros. (the public domain clone of Super Mario Bros. of Nintendo)[1] has been shown with accurate and fast movement. These successes show that it is perhaps time to focus on other issues of game AI than making strong computer players, such as the ability to entertain humans, which is, in fact, the primary purpose of computer games. If we want to consider a method to entertain human players, it will be necessary to understand human players.

Strong computer players are likely to play optimal actions for given states. In contrast, human players are often affected by not only given states but also external factors such as their physical or mental statuses. So, optimal actions may not be selected, intentionally or unconsciously. Typical characteristics of such behaviors of human players are often called "human-likeness." In the case of the A* Super Mario Bros. AI's video where the behavior looks remarkably mechanical, humans understand that a machine controls the player. In the case of two or multi-player games, such mechanical behaviors may greatly harm human players' entertainment. Usually, human players prefer to play against/with human players or human-like players and do not want to play against too strong/accurate/fast players like a "cheater." Also, human-likeness is almost necessary for automatic test players. Let's consider that a computer player evaluates a stage of a shooting game. If the player is very strong/accurate/fast that human players cannot reach the level, even a very difficult stage may be evaluated as "can be cleared easily." Thus, human-likeness is important for the entertainment of games and should be analyzed, reproduced, and well-considered.

Many published papers dealt with and discussed human-likeness on various topics, purposes, methods, and aspects. For example, Fujii et al. [6] focused on physical constraints and employed reinforcement learning to create human-like players. Different from their approach, we focused on mental aspects and employed a modified A* algorithm [7], for making human-like Super Mario Bros. players. In this field, there are no common definitions of human-likeness or human-like behaviors; therefore the evaluation methods vary from different research groups.

In this dissertation, we classified the current literature to understand the current situation of human-likeness in academic research and look for whether there is any essential but less studied aspect. As further studies of human-likeness, we worked on three different topics which we believe will be useful for advanced research in this field.

The structure of this dissertation is as follows: In Chapter 2, the previous literature is classified by the purpose, the method, and the perspective. In Chapter 3, the approaches of our research are briefly explained and the details are introduced in Chapters 4 to 6. In Chapter 4, the emotional aspect of human-likeness is discussed. Particularly, using Super Mario Bros., some behavior models and transition rules are proposed and implemented so that the player seems to have emotions. In Chapter 5, the emergence of human-like behaviors based on sub-goals is discussed. In Chapter 6, we show the generation of biased random sequences to make human players believe the random numbers are nat-

[1] https://www.youtube.com/watch?v=DlkMs4ZHHr8, Accessed May 13th, 2019

ural without any bias. Finally, we explain the contribution of the research in Chapter 7.

This research contains various approaches. However, they have a common primary purpose, which is to analyze and reproduce human-likeness for the entertainment of games.

Chapter 2

Literature Review

Much of the research on computer game players (called game AI) thus far has been conducted for its "strength," and classic board games (e.g., chess, Go, and shogi) have been targeted. A certain strength is necessary as the basis for any other purpose; however, the performance of game AI had been insufficient for a long time, even for such simple games.

Through breakthroughs by the Bonanza method [8] and Monte-Carlo tree search [9] at the end of the 2000s and deep learning in the mid-2010s, the strength of game AI has exceeded that of human players, or at least average players in many games, not limited to classic board games [3]. Recently, other directions than strength are attracting attention.

As one of the next research targets of game AI, "human-like behavior" is a topic that has become popular very recently. To enhance entertainment, game AI may be necessary to behave like humans. We can consider the effects of human-likeness on two levels. In the case where humans do not participate in the games, e.g., the gameplay movie of A* player in Infinite Mario Bros. [4], the unnaturalness of the game AI might be harmless. In contrast, if a promotion movie of a game is generated by employing two AIs, it may be better to employ human-like AI for making exciting movies. In two-player or multi-player games, where a human player simultaneously plays with one or more game AI (AI is assigned as partners, opponents, or neutrals), the unnaturalness of the AI's behavior may harm the entertainment of the games directly.

The topic of human-likeness of humans/machines is popular in the field of philosophy of the mind and cognitive science. The research in this field has been discussed widely since Alan Turing proposed the Turing test in his article "Computing Machinery and Intelligence" [10]. In the area of computer games, the study of human-like behavior is broad and arouses interest from many researchers. However, as far as we know, the definition of human-like behavior is still ambiguous, even in the specific case of computer games.

There have been several attempts to define the measurement of parts of human-likeness. Togelius et al. [11] defined **player believability** as "whether someone believes that the player, who controls the character/bot, is a human, i.e., a human is playing." This measurement is mainly used for video games, such as Starcraft and Super Mario Bros., where either a computer player or a human player can control a character in the games. Human subjects observe the behaviors and judges whether a human player is playing. Based on this definition, various human-like game AI approaches were proposed. However, believability is only a part of human-likeness. There are many other aspects and ways of measurement of human-likeness. In the remaining of this chapter, we collected and classified the current literature to understand the situation of human-likeness in academic research.

2.1 Various definitions of human-likeness and various approaches

Much research has been conducted on games, and many studies focused on the strength of game AI, but there are also many studies on human-likeness. There are a variety of uses, methods, and perspectives of the research in human-likeness (various categories are briefly summarized in Table 2.1). Thus, we surveyed many papers related to human-likeness and investigated them on the following points: 1) for which purpose human-likeness is needed/used, 2) by which method the purpose is realized, and 3) on which perspective of human-likeness is focused. We classified and grouped the papers to understand the situation of recent research on human-likeness and the result is summarized in Table 2.2. This survey was published in a domestic conference [12].

Method	supervised learning, reinforcement learning, evolutionary computation, heuristics and search, etc.
Purpose	entertainment, procedural content generation, education, etc.
Perspective	abstract observation, overlook in observation and search, physical limitation, fun and curiousity, emotion, etc.

Table 2.1: Various methods/purposes/perspectives of human-likeness

In Section 2.2, we introduce some of the research according to the purposes of human-likeness. The most obvious purpose of the research on human-like behavior is to improve human players' satisfaction by making computer players, which play against/with human players, act like human beings. It may be an opponent (enemy), a friend (alliance), or a neutral player such as a villager or a crowd. In some cases, not only human-like but also specific-player-like may be preferred. Further, other purposes, such as automatic content generation (procedural content generation, PCG), or education, have also been broadly studied recently.

In Section 2.3, we discuss the methods employed for realizing human-likeness. To make game AI behave like humans, some simple and straightforward methods are to reproduce similar actions to those made by human players in the same situations. Reproduction is usually done by case-based reasoning or supervised learning. These methods have been commonly used for “strengthening” purposes. Besides, there are many other methods for achieving human-likeness or for analyzing human-likeness of human players.

Humans have many unique characteristics that computer players usually do not have. Humans cannot recognize the game screen or information precisely all the time. Blurring often occurs in observation, omitting often occurs in thinking, or fatigue of hands usually occurs in control. Also, sometimes humans are influenced by emotions, and sometimes, they may play games for purposes other than the purpose defined by the game rules just for fun. Thus, we divided these characteristics into several groups, and their perspectives are introduced in Section 2.4.

2.2 Purpose

Many researchers have analyzed or reproduced human-likeness for many purposes. In this section, we introduce and discuss such purposes with typical examples.

2.2.1 Agent players for entertaining human players

Among the studies on human-like behaviors, the most direct usage is to introduce game AI as agent players (opponent, friend, or neutral character) to replace human players. Most of these kinds of research aim to satisfy human players by using human-like computer agent players.

For example, Mandziuk et al. [13] claimed that, if the behavior of a game AI is constant or poorly changed in a certain scene of a game, human players can predict the behavior of the game AI by repeatedly learning and the entertainment of the game might be impaired. Thus, he proposed a method to produce agents’ individuality to solve the problem.

Bailey et al. [14] considered that the limited behavior of the game AI affects the entertainment of a game, and various elements, such as individuality, emotion, motivation, and social relationships, should be introduced to give a sense of immersion in the in-game content.

Fujii et al. [6] claimed that the optimal actions to win games may look machine-like to humans. For game AI to play with a human player, it is necessary to give the player the feeling that the AI is also playing with him or her. Thus, biological constraints (tremor, delayed response, and fatigue) are introduced into the game AI to create human-like behaviors.

Ito et al. [15] claimed that a drama or happening should occur in a game match to increase its entertainment. They also claimed that such drama or happening among human players are often made by some errors. The lack of human-like errors caused players to lose interest in the game. For this, they classified the errors in shogi and proposed a human error model.

For the game of Go, Ikeda et al. [16] also provided an overview of various techniques used for entertainment. They claimed that computer players should select natural moves to entertain human players and to encourage them to continue playing Go.

Wada et al. [17] claimed that computer players' human-likeness as teammates is more important than that as enemies. They also claimed that teammate AI should estimate human players' intentions or utility and respect it. They proposed a method to estimate a human player's utility from the human player's history of play and try to optimize the utility.

2.2.2 Procedural content generation

In a game, not only basic rules and the system, but also game data, also called "contents", are required. The scope of the word is wide and depends on the context. Contents can refer to the part of games that is strongly related to gameplay, such as the construction of the map (in games), the additional rules, or the statuses of characters. In addition, content can refer to small details, such as the appearance of the character, the appearance of the map (small articles, texture, etc.).

Contents are usually created by human designers. In some games, automatic content generation has been used, such as automatic map generation in Rogue-like games[1]. Recently, maintaining the excitement of games by updating the contents which is called "download content", such as game rules, additional maps or characters, new stories, or events, becomes important tasks. And to reduce costs, assisted or automated content generation is needed. Automatic content generation is also called procedural content generation, or PCG.

PCG requires technologies in various fields, such as natural language processing, image processing, or speech recognition, depending on what the content refers to. In the study of game informatics, gameplay-related contents, such as map generation, are considered. There are many ways to generate such contents, e.g., supervised learning and generative adversarial network [18] using existing maps. One of the most major ways is the generate-and-test approach, where contents (such as a map) are randomly generated and then evaluated/filtered by computer players. In other words, playability ("can human players clear this map?") or entertainment of the contents ("do human players feel fun by playing this map?") are estimated by AI agents. If the AI behaves too differently from human players, the evaluation may not be able to point out anything (only the AI can play or use the created contents).

In some PCG works, human-likeness of test agents was explicitly considered. Togelius et al. [19] modeled players with focuses on the driving style, ability, etc. for race games, and used them to generate race tracks that human players enjoy playing.

[1] https://www.greenmangaming.com/blog/what-is-a-roguelike/

Sometimes, human-likeness is considered, but not as the property of test agents. Shaker et al. [20] proposed a model of the impression of a player (fun, challenging, or dissatisfaction) by using machine learning. This research attempted to generate a game stage which has a high possibility for entertaining human players, by using the impression model as the state evaluation function.

Chanel et al. [21] estimated emotions based on physiological signals of human players during gameplay and adjusted the difficulty of the game to suit the players. The approach was not based on questionnaires which are usually done in human-likeness research.

2.2.3 Education

One of the purposes of the works on human-likeness is education/coaching. Computer game players are strong, then it is natural that they are employed to coach human players. Although they may be easy to tell the best move for each state, such ability is not sufficient as a good coach. Good human coaches respect the characteristics and intentions of their students, consider why students played bad actions, and explain so that students can understand. For such activity, human-likeness should be well considered.

For example, Omori et al. [22] presented a method to reproduce many different playing styles of professional players in shogi for coaching purposes. Professional players are divided into two groups, offensive and defensive, and supervised learning was conducted to imitate their playing styles.

Ikeda et al. [23] aimed at education for the game Go. Instead of simply giving guidance for bad moves with poor evaluation values, the researchers tried to find bad moves that "human instructors are likely to point out." The work tried to reproduce a human-like instructor, not simply human-likeness.

Takahashi et al. [24] aimed to educate players to master Puyo Puyo, one of the Japanese popular puzzle games. They proposed to generate game subproblems to improve players' skills. Thus, the work can also be classified as PCG. To generate interesting and useful training problems, estimation of interestingness by supervised learning was used.

2.2.4 Other purposes

There are some other purposes of works on human-likeness which are difficult to be classified into the previously shown groups. For example, some of them discussed essential questions such as what human-likeness is, how it can be evaluated, and what kind of evaluation method is undesirable [11] [25] [26] [27]. Also, there is another study that focused on commercial and advertising perspectives. The research focused on the positioning of the augmented reality (AR) icon that is easy to be recognized but not visually disruptive [28].

Due to the popularity of e-sports in recent years, Lu [29] tried to create an observation video of a fighting game with game AI for e-sports training.

Thus far, we showed the various purposes of human-likeness in this section.

However, what we have presented here are only parts of the purposes of human-likeness. There are also other purposes that have not been introduced.

2.3 Method

Various artificial intelligence techniques can be used to reproduce human likeness. In this section, we provide an overview of current techniques that have been used. Also, the following table (Table 2.2) lists our classification of papers by purposes and methods.

Purpose	Method				
	SL	RL or EC	HS	Phys, Rb, Grp	Other
Make game players	[30] [31] [32] [33] [34] [35] [36]	[6] [37] [27] [38] [39] [40] [41]	[13] [28] [42] [43] [44]	[45] [14] [46]	[47]
PCG	[19] [20] [48] [49] [50]	[48]	-	[21]	-
Education or training	[22] [51]	-	[24]	-	[52]
Etc.	[53] [54]	-	[55] [56]	[57]	[11] [58]

Table 2.2: The literature classified by the purpose and the method (SL: Supervised learning, RL: Reinforcement learning, EC: Evolutionary computation, HS: Heuristic method and search, Phys: Physical testing, Rb: Robotic, Grp: Graphic, Other: Discussion or proposing assessment method)

2.3.1 Supervised learning

To create human-like game AI, learning from human play data is considered the simplest method. Many approaches have been tried and employed in the context of reproducing human professionals' actions as strong computer players. For example, the realization probability search [59] and Bonanza method [8] originated in Japan. These methods attempted to imitate strong human players' thinking and/or evaluation function by using supervised-learning methods and strong human players' game records.

Shaker et al. [20] used a neural network to express human-likeness in a car racing game. The authors combined two types of networks, one determining the trajectory of a car and the other determining the speed, to mimic the behavior of human players.

Munoz et al. [31] tried to not directly reproduce the behavior (playing) of human players, but to estimate the course preference of human players in a car racing game. They constructed an evaluation function, from a course to a value, and applied it to create non-player characters (NPCs).

Miyashita et al. [37] proposed to combine supervised learning and reinforcement learning. Their goal was to create strong while human-like computer players.

In Omori et al.'s research [22], they separated playing styles in shogi into offensive and defensive, and trained an evaluation function by machine learning. Research targeted on specific playing styles or specific players is often performed.

Moreover, many other studies used supervised learning to reproduce human-likeness, and we discussed only some of them [17] [19] [24] [32] [60].

2.3.2 Reinforcement learning

Reinforcement learning is a learning process that imitates humans interacting with environments [61]. Reinforcement learning is notable in terms of creating strong game AI without giving any learning data, for example, AlphaGo [3] and DQN [62]. However, there are also no guarantees whether human-like behaviors can be obtained, which highly depends on given rewards. For example, DQN for Atari games does not look human-like, because in some cases, the responses are too quick.

Fujii et al. [6] tried to introduce human-like delays and sensory errors to conventional Q-learning for producing human-like behaviors. To express a delay, data obtained from the game were fed into Q-learning several frames later, and noises were added to the input data as sensory errors. The authors expressed fatigue by giving negative rewards when the operation (button) was changed rapidly. These modifications greatly benefited the acquisition of human-like behaviors.

Lu et al. [29] aimed to combine reinforcement learning with neural networks to create game AI that can play several games of the same type. The general game playing (GGP) and general video game playing (GVGP) frameworks pursue human-likeness in the sense of a human-like quick adaptation from current knowledge.

Rule-based modeling has been widely used for making computer players, and often the rules were defined manually. Ortega et al. [33] employed the "dynamic scripting" method to adapt the rules by reinforcement interaction with environments or opponents. Such adaptation ability can be also considered as a part of human-likeness.

Other attempts introduced external rewards that were not obtained from the games. For example, when humans are playing an unfamiliar game but keep failing, they may try to reach some new states first. The attempt to reproduce this eagerness of exploration is called curiosity-based reinforcement learning and many researchers focused on this area. In addition, Phuc et al. [27] pointed out that it would be better to give "fun" as a reward for various human behaviors in games.

It is difficult to design a good reward for each game so that the trained agents behave like human players. Inverse reinforcement learning [63] is such a challenge to acquire good rewards by using game records from human players.

2.3.3 Evolutionary computation

Evolution is an adaptation process of species, which may be used to bring some naturalness. For example, in the farm-simulation game Astronōka, the evolution of pests succeeded in giving players aversion (similar to real insects).

In many cases, evolutionary computation is merely used as an optimization method. Evolutionary computation does not require differentiable functions unlike some other methods such as gradient-based methods. Thus, direct optimization can be performed, and by using a large number of individuals (candidate solutions), it is possible to give players various game experience. By using multi-objective optimization, it becomes relatively easy to balance the strength and naturalness of computer players [38].

Van Hoorn et al. [39] focused on steering and acceleration in a racing game. Human-likeness was reproduced by using multi-objective evolutionary computation to minimize the difference between humans' play data and game AI's play data.

The combination of neural networks and evolutionary computation is often used and is called neuroevolution (NE). NE has been used to minimize differences with human behavior history directly, such as Ortega et al.'s research [33] on Super Mario Bros. Another research on the same game [27] used NE to minimize differences with human behavior statistics (the coin acquisition rate, left button press rate, etc.).

2.3.4 Heuristics, search algorithms, and others

Human-like behaviors were also reproduced by other methods, such as rule-based ones where the behaviors of AI are manually described, and search algorithms where human-likeness is also taken into consideration.

Hirai et al. [64] and Sato et al. [55] generated human-like agents in shooting games by adding modifications to search algorithms, such as introducing an influence map to avoid immediate evasion action (which is hard for humans to perform), reducing frequent changes of directions, etc.

In Ikeda's article [16], after defining what kinds of unnatural moves should be avoided for entertainment, values from a static move evaluation function (trained by supervised learning) and expected winning ratios (calculated by Monte-Carlo Tree Search) were both considered. Not-best but natural moves are selected to intentionally weaken the strength of computer players while keeping naturalness.

As in Omori et al.'s work [22], Nakamichi et al.'s work [28] on shogi was based on the Bonanza method, but an adjustment was performed on the evaluation values from $\alpha\beta$ pruning to reduce unnaturalness. In addition, they further investigated the unnaturalness which was originally described by Ikeda [16]. Classifications such as bad moves, unconventional moves, unintended moves, unintended intentions, inconsistent strength, unnatural tactics, and bad luck were proposed.

In this section, thus far, we classified papers by AI techniques or methods. However, expressing human-like behaviors with only a simple method is hard, because human-likeness has many and different perspectives.

2.4 Human-likeness perspective

Human players show human-likeness in various forms. For example, let's consider the situation shown in Fig. 2.1, which simplifies the decision-making process of a human player. The followings discuss three aspects which human players usually have but computer players do not. First, when observing states ((1) in Fig. 2.1), the player may overlook something or miss approximating the position or timing. Second, when thinking ((2) in Fig. 2.1), the player may pause his or her operation, decide to perform biased behaviors due to habit, or even make bad predictions. Third, when reacting ((3) in Fig. 2.1), the player may fail to respond immediately due to fatigue, make operation mistakes, or change behaviors merely because of emotions.

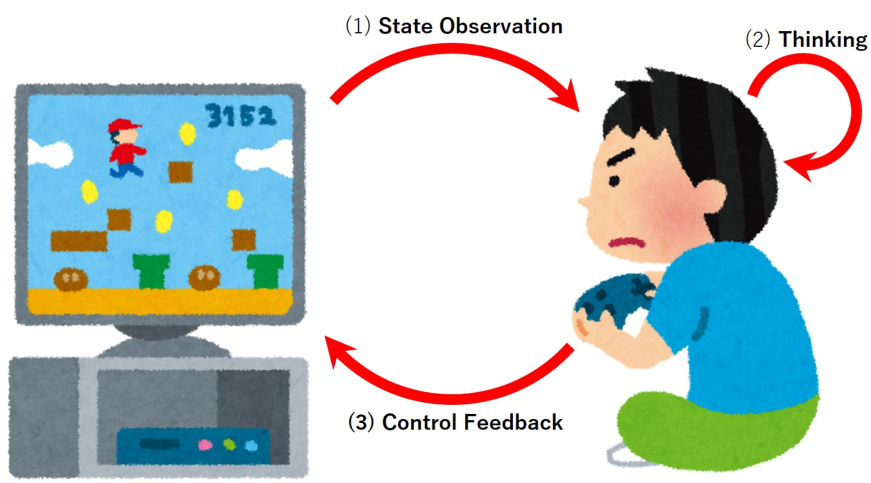


Figure 2.1: A human player plays a game. (1) The player observes the game states with his or her senses (eyes and ears). (2) The player thinks about his or her responses. Then (3) the player responds and makes his or her decided actions via the controller.

There are many possible aspects of human-likeness, but each paper usually only discusses one or some of them. Many papers directly imitated the sequence of actions. In other words, the "outside" aspect of human-likeness is considered.

Also, many papers considered the "inner" aspect of human-likeness, such as cognitive methods or physical limitations, and tried to produce human-like actions by them. In this section, papers are grouped and introduced in terms of such approaches based on perspectives of human-likeness: direct imitation, physical limitation, emotion, mistake/misperception/misunderstanding, and preference. We show only a part of research related to human-likeness; thus, there may be other additional groups to which we do not refer.

2.4.1 Imitation of action sequences

To produce human-like behaviors, direct imitation of action sequences seems to be the simplest approach. Ortega et al. [33] tried to make computer players whose traces for a stage of Super Mario Bros. are similar to those of human players. They implemented and compared three methods (i.e., hand-coded, direct learning, and similarity maximization) to reproduce the traces. For the evaluation, comparisons between the three methods were made and similarity maximization gave the best results.

Kinebuchi et al. [51] pointed out that some typical variations or sequences of shogi are valuable to imitate for improving naturalness from human players' viewpoints. If some variations appear frequently among human professionals but not so frequently among AI players, such variations should be specifically considered by using some biased selection.

Sometimes, human variations/sequences are indirectly imitated. Munoz et al. [31] focused on five elements (i.e., speed, overtaking, overtaking prevention, collision avoidance, and trajectory correction) in car racing games by trying to mimic humans' feature values to imitate human traces.

In a similar way, Phuc et al. [27] applied evolution computing to make the number of collected coins, the frequency of jump, etc. closer to those of humans in the Mario gameplay.

2.4.2 Physical limitation

As a living being, the human body is well made. However, the recognition, transmission, and operation functions are inferior to those of machines. This is an essential aspect of human-likeness and should be recognized whenever we design a computer player as an enemy, a companion, or a test player for procedural content generation.

Fujii et al. [6] realized human-likeness by focusing on humans' physical constraints (tremor, reaction delay, fatigue, etc.). They introduced such constraints into Mario's game AI. However, this method is not only for Mario AI but also for a general-purpose. Various applications can be considered.

As a similar perspective, Tan et al. [40] pointed out that human beings cannot recognize objects at distant places precisely. When designing computer players of a car racing game, the authors tried to realize human-like behaviors by adding noises to state inputs of computer players, depending on the distance to each object such as another car. Consideration of noises in human perception was also shown by Hagelbaeck et al. [42].

In addition, Lu et al.'s work [29] considered that such physical restrictions can be introduced as rules of games. For example, in the Fighting ICE platform, when computer players play, the information transmission is taken into consideration and thus the previous state is passed to the player instead of the current one.

2.4.3 Emotion

Human behaviors are often strongly affected by their inner emotions. This is an essential feature of human beings. Emotions may cause mistakes and inconsistent behaviors. Understanding human emotions and their changes is also important to make computer game players. More specifically, it is important to create not only computer players which have human-like behaviors, but also cooperative or coaching game AI who can understand, respect, and support humans' gameplay.

Mandryk et al. [45] focused on emotions (boredom, challenge, excitement, dissatisfaction, and enjoyment) when playing games. They proposed a method for estimating these five emotional states from physiological information by using the fuzzy theory.

After the work by Mandryk et al., Chanel et al. [21] used a similar emotion estimation method for content generation. Here, the balance of a game is adjusted to an appropriate difficulty for each player, considering their estimated emotions.

In this dissertation, instead of estimating human emotions during gameplay, we try to create a game AI that appears to have emotions. For example, in Mario, when many enemies suddenly pop up, human players are likely to be surprised and frightened, and then move far away from the enemies.

2.4.4 Mistakes and misperceptions

Mistakes and misperceptions are obvious characteristics of human players, and thus are important for human-likeness. Considering the case that computer game players play against/with a human player, if their movements or behaviors are very accurate without any mistake, entertainment of human players (especially casual players) will be harmed. In addition, for game content generation, human mistakes should be well considered. When AI is employed as a test player, if the AI makes no mistakes, there are no guarantees that a human can win/clear the content. Thus, mistakes and misperceptions are a crucial factor of human-likeness.

Ito et al. [15] stated that human errors produce drama in games, and thus contribute to the enjoyment. Also, they proposed a model to explain human mistakes and classified the reasons and types of human errors in shogi.

We discussed physical limitations in section 2.4.2 and misperceptions may be a part of such limitation. For example, the work by Tan et al. [40], introduced in section 2.4.2, reproduced some kind of misperceptions by adding noises to inputs.

In Chapter 6, we discuss that the pseudorandom sequences which are often used in games can make human players dissatisfied, because of humans' mathematical cognitive biases. Normally, algorithms for generating pseudorandom numbers aim to generate the longest non-pattern sequences as the theory in mathematics. However, each human player has their own ways/biases of thinking about random numbers. Thus, even random sequences that are almost mathematically ideal may not satisfy human game players. Further, we show a method for generating random sequences that human players feel more random than those generated by a famous pseudorandom number generator.

2.4.5 Preference

In gameplay, when we compare two players, even if they try to achieve the same goal or even if they have the same skill level, the methods and approaches for achieving the goal may differ. The way of playing is called a playing style. In addition, many games provide sub-goals or rewards that do not directly lead to victory, or often human players invent such sub-goals by themselves. For example, in Super Mario Bros., killing a large number of enemies and collecting many coins are not important for victory. However, human players often prefer to do or achieve this kind of goal or reward. These are also called secondary objectives. By incorporating these objectives into learning and search, human-like behaviors may emerge.

Wada et al. [17] pointed out that when humans play role-playing games (RPG), such as Dragonquest or Final Fantasy, players not only "want to win," but also "want to win with less damage," "want to win and preserve resources," and "want to win as soon as possible." Thus, for making cooperative alliance AI, predicting such a sub-purpose by using the history of the human player's actions and changing behaviors accordingly can improve the degree of satisfaction.

In Chapter 2, we showed much research with various purposes, methods, and perspectives of human-likeness. The purpose of this survey was to understand the current situation of research in this field and whether there is any essential but less studied aspect of human-likeness which is worthy to reproduce. As a result, we found many possibilities to expand the current research area. We explored three topics on human-likeness in particular, which we believe is necessary for enhancing the entertainment of games.

Chapter 3

Approach

The research on human likeness in game AI is varied, yet there is still much human behavior which is unique and should be considered. In Chapter 2, we introduced and summarized many researches related to human-likeness. After intensive survey, we found three interesting topics which are valuable to consider but have been less challenged. In this chapter, their background are explained, our research questions are defined, and our approaches are briefly introduced.

The first topic is about emotion. In the same game and the same stage, throughout the scene, human behavior may change from a style to another style. For example, in Super Mario Bros., sometimes, the player might look like s/he enjoys collecting coins, or sometimes, s/he might look s/he is rushing to clear the stage, due to the situation in the game and player's emotion. These changes are also an essential point of human likeness which should be considered (Fig. 3.1 (1)).The reproduction of emotion-based behaviors and its transition will be valuable for making human-like computer players.

The second topic is about sub-goals. There can be many sub-goals in one player game, for example collecting coins or killing enemies in Super Mario Bros. Further, some interesting examples of human-like behaviors can be found in some massive multiplayer online role-playing games (MMORPGs) in which communication is essential. Sometimes, the action within the game is used to express a message, such as repeating a jump in the same place might be interpreted as "come here." Or, walking around a player character repeatedly often means "what's up?". It has been shown that humans make their sub-goal (in this case, is to notify something) by themselves. The created sub-goal is not directly related or not related to the given goal of the game (Fig. 3.1 (2)).

The third topic is about random number generation. Considering entertainment purposes, human-likeness should be considered carefully not only for making computer players as opponents or allies, but also for making game content such as stages, because stages should be felt to be interesting from the point of human's view. One of such content or environment is "random sequence" used in many games. The pseudo random algorithm which is used for generating randomness in games normally aims for generating a true random

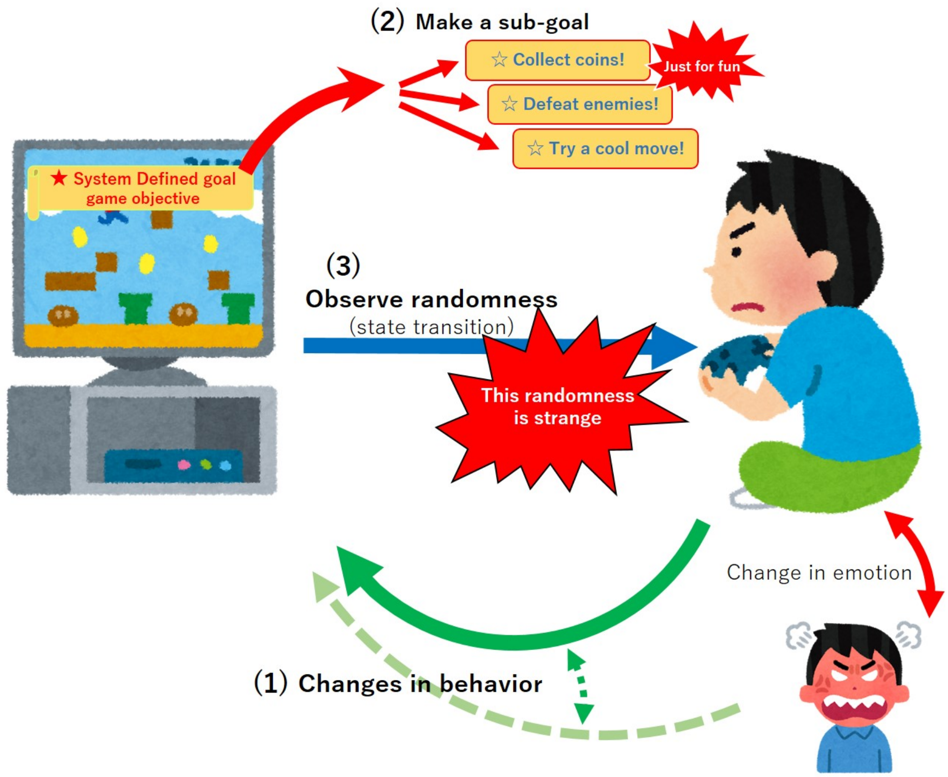


Figure 3.1: Interesting research area

in mathematical terms. However, humans often misunderstood about the randomness, and this seems to be usual for humans. Thus, the content where a pseudo random is used might be unable to satisfy human players. This human likeness is also an important concern (Fig. 3.1 (3)).

In this dissertation, using the three interesting points above, we formulate the following research question: "how to analyze and reproduce/consider human-likeness in computer games". This question is very difficult to solve, but we try to approach to this ultimate question by solving the following sub-questions:

(A) How to create an agent whose behaviors look to be from some emotions

(B) How to understand the reasons and conditions when human players emerge sub-goals

(C) How to generate random number sequences that satisfy human players than usual generators.

For answering the ultimate research question, we should define "what human-likeness is". But after wide survey we understood it is still difficult for researchers to find one simple definition, and it is better to find new aspects of human-likeness, try to reproduce it, or try to consider it for improving human's satisfaction. Then, solving the above three questions can contribute for solving only a part of ultimate question, but we believe they are very important for further understanding of human-likeness.

3.1 Approach for emotion-based behaviors for a human-like computer player

For Problem (A), we focus on the transition of different playing styles for different sub-objectives within a game. Today, computer games provide not only a single ultimate goal but also sub goals or extra rewards to provide diverse gameplay to entertain the player.

For example, in Super Mario Bros., the ultimate goal of each stage is to reach the goal in the farthest right of the stage within the time constraints, but the player also able to choose to collect extra rewards, such as coin, or items, or the player can try to defeat enemies that appear during the stage. However, everything should be completed before the time runs out.

In this case, in the beginning, the human player tries to reach the goal as fast as possible. Nevertheless, after finding some coins and acknowledging there is still enough time, the player might ignore the primary mission, and try to collect coins. After a while, when the time has almost run out, the player's movement becomes faster and riskier to clear the stage in time.

As human players often change their behavior within a game depending on their emotion and the situation (Fig. 3.2), we come up with the hypothesis that "an agent's behavior will look emotional and more human-like if the agent is able to change its behavior due to the current situation." Thus, our approach is guided by the following question: "How to create an agent with behavior transitions which looks like it has emotion.?"

We conduct the experiment in the Mario AI benchmark which is based on the famous action side scroll Super Mario Bros. We classify human player behavior into three types: safe, fast, and greedy. These behaviors will be modeled and implemented by hand-coding, in other words, unsupervised, and based on the A* algorithm. As human likeness is very important in the model behavior, each model will be evaluated with the Turing test individually. Furthermore, we also propose two types of transition models to express the transition between behaviors. The first transition is based on the if-then rule. which will be shown in Chapter 4. The second transition model is based on supervised learning which aims to learn the behavior transition by human data. However, in this dissertation, we will describe only the first model.

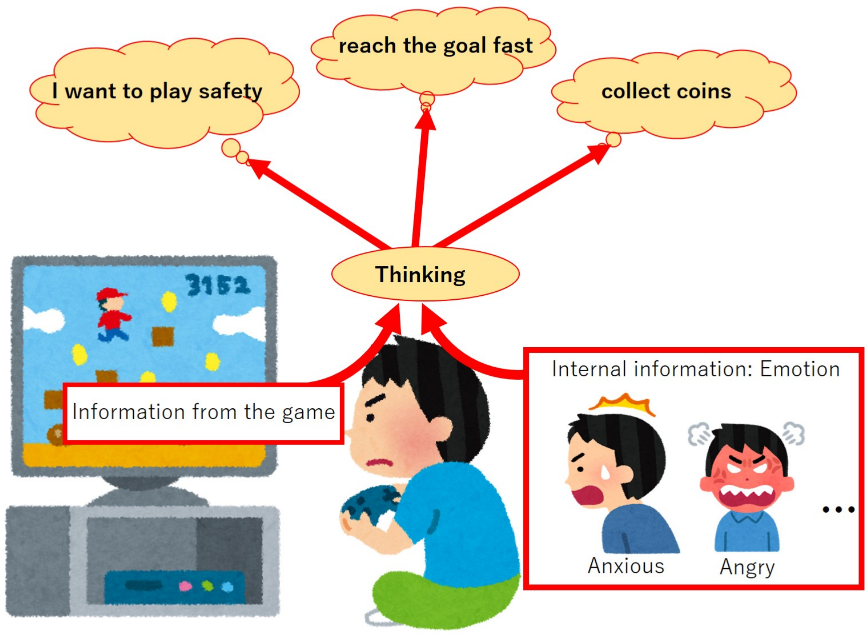


Figure 3.2: Behavior chosen due to external game information and internal emotion information.

3.2 Approach for sub-purposes oriented human like behavior

For appraoching to the question (B), we focus on sub-goals or sub-objectives which are not given by the games but created by human players. Human players sometimes create sub-objectives to achieve a goal or rewards within the game. For example, assume a multiplayer first person shooting (FPS) game, which two teams battles. Sometimes, a human player uses available actions for achieving different purpose; a gun is clearly given to kill the opponent members, but sometimes used to shoot at a wall, as an instant message for allies to alert something(Figure 3.3). Or some players try to use their guns to create illustrations with bullet holes, just for fun. Or in racing games, some players who are far ahead and close to the goal, stop their cars just before the goal, wait until another player comes closer, and then reach the goal just to provoke their opponents. These kinds of actions can often be observed in many types of games, and are unique as human behavior.

As we stated earlier, this kind of behaviors from sub-goals, not directly related to the given primary goal, is often found and seems to be really human-like. Thus, to produce a human-like agent, it is necessary to understand this behavior more deeply.

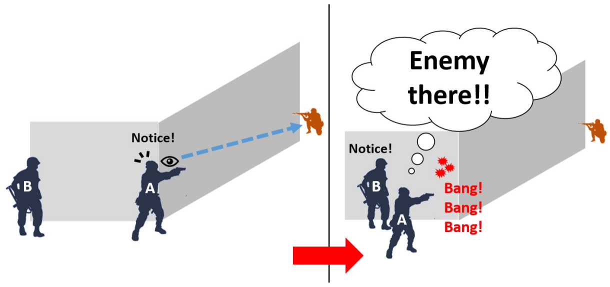


Figure 3.3: Sample of sub-purposes-oriented behavior

To answer the research question, we consider it is valuable to (i) survey and summarize such situations and behaviors, (ii) analyze why/how they are created, and (iii) reproduce such behaviors, for achieving human-like game AIs. (i) We collect a large number of sample cases from a broad type of games and systemically classify them, based on the purposes of the actions, for example warning, provocation, or enjoyment etc. We found many interesting human player behaviors, and we found the tendencies or conditions for such behaviors. Thus, for (ii) we propose several conditions for the emergence of such behaviors. For example, spare time or freedom is a very important factor. A warning, such as shown in Fig 3.3, could not happen if the gun can use a telecommunications application such as Skype. For (iii), we try to reproduce an emergence of a "notice" behavior by using two Q-learning agents. This research will be explained in detail in Chapter 5.

3.3 Approach for biased random sequence generation for making players believe it is unbiased

The question (C) is different from the other two questions. Our purpose is not to make computer players, but to investigate how to generate pseudorandom sequences which look random for human players. Pseudorandom number generators are used in many digital games in which randomness is needed, e.g., Poker and Mahjong. Human players often feel dissatisfied with the given random numbers, especially when they are at a disadvantage due to unlucky. Many players complain about the randomness even though an excellent algorithm (such as Mersenne Twister) is used to simulate the true randomness.

We guess, the reason why many players complain about randomness is that human players have their own cognitive biases, one of important human-likeness (Figure 3.4). Thus we propose a method to generate pseudorandom numbers/sequences where human players believe that it is random. In other words, our approach tries to understand human players' cognitive biases and match it. We firstly let human subjects write down 100 numbers which seem to be

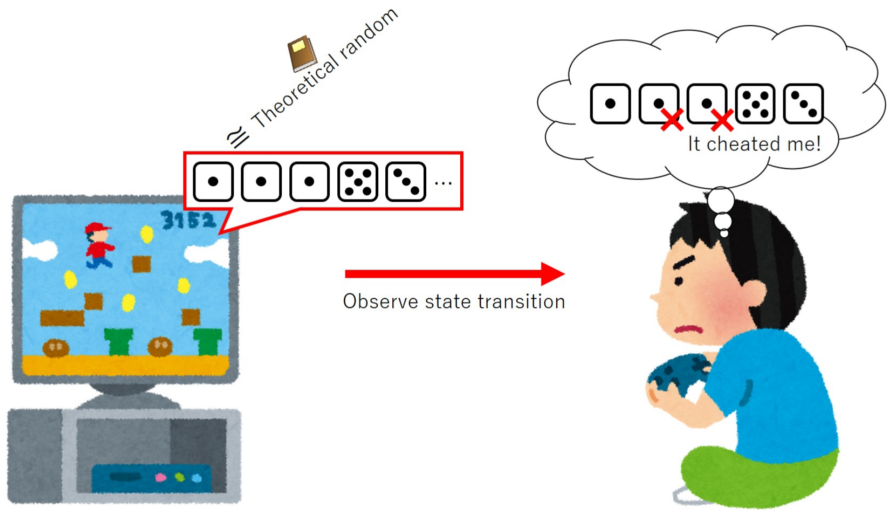


Figure 3.4: Cognitive bias in player

random from their viewpoint. We analyze the obtain sequences by 15 statistical features and compare the values to theoretical ones. Then, we propose a method to generate pseudorandom sequences by a local search, so that each sequence has similar statistical values to human players' values. Also we try to propose an additional technique to adapt such sequences for an actual game, Sugoroku.

Chapter 4

Emotion-based behaviors for a human-like computer player

This chapter is written and modified based on the article "Production of Emotion-based Behaviors for a Human-like Computer Player" which is published in the international conference Game-On 2016 at Lisbon Portugal in 2016.

4.1 Background

There are various uses of a computer player in video games. Sometimes computer players are developed to control another character, such as a partner or an opponent, to entertain human players. The design of a computer player with suitable behaviors or strategies is difficult, and it becomes a heavy burden for developers. Thus, efficient algorithms, such as the path-finding algorithm A* and the learning algorithm TD learning, are introduced to generate behaviors or strategies to reduce the load of developers' work.

We can say that current performance of AI is enough to but the behavior of strong computer players is not promising for the entertainment of players. For example, a popular video of a computer player for Infinite Mario Bros. (the public domain clone of Super Mario Bros. of Nintendo) was published in 2009 on the website YouTube[1]. The video shows an excellent Mario gameplay, which is controlled by a computer. Each of its actions is highly accurate and instantaneous. Such behavior looks remarkably mechanical. In this case, a human observer only acknowledges that the player is controlled by a machine. However, in the case of two-player games (e.g., fighting games like Street Fighter[2]) or multiplayer games (e.g., shooter game like Unreal Tournament 2004), where

[1]https://www.youtube.com/watch?v=DlkMs4ZHHr8
[2]http://www.capcom.co.jp/sfv/

a computer player simultaneously plays with human players, humans might suspect they are being cheated, and the entertainment of the game will be harmed because of the unnatural behaviors of the computer player. Hence, the production of behaviors that look natural to humans, called human-like manner, is essential to enabling computer players to entertain human players.

There are many approaches for producing human-like behaviors, such as directed learning [30],or introducing biological constraints [6]. However, to produce human-like behaviors for computer players, changes in behaviors during the game due to emotions are needed to be concerned.

For example, in Super Mario Bros., the main goal is to clear a stage within a limited time. In the beginning, the player tries to reach the goal as fast as possible. Nevertheless, after some coins are found and the player acknowledges there is still enough time, s/he might ignore the primary mission and try to collect coins, which is a sub-objective of the game. s/he might be inspired by greed or enjoyment. Such a change in behavior is inspired by human feelings or emotions. Hence, the production of behavior transitions is essential to produce a human-like behavior.

Thus, we aim to produce a human-like game AI whose behaviors look like they have emotions. We also aim to create an agent who changes their behavior due to the situation and their current emotions.

4.2 Purpose and goal

In a modern-style game, not only a single goal but also many sub-goals are given and available for challenge. In the case of the Super Mario Bros. series, the major goal is to reach within a time limit the stage's goal located at the rightmost of the stage. Players have other optional tasks of collecting coins or beating enemies, though they are not necessary to clear the stage. The player is able to challenge any goal that s/he prefers, but the player must respect the major goal. Thus, the player will exhibit transitions between several local behaviors.

For example, at the beginning of the stage, the player's movements are at ease, so s/he can enjoy collecting coins, or the player can control Mario carefully when he encounters many enemies. After a while, when the time has almost run out, Mario's movements become faster and riskier to clear the stage in time. Our research interest is in creating a human-like computer player with transitions between emotional behaviors. The usual practice in this area has been focused on the human likeness of behaviors in overall game play, whereas our approach produces transitions between multiple behaviors. Each behavior model produces a specific human-like behavior inspired by human emotions or feelings (e.g. anxiety, fear), and the transition model then decides the appropriate timing to change the behavior, which looks like a human transition.

There are many emotions, and combinations of emotions. Then, there will be a lot of play styles affected by them and we cannot reproduce all. So, we watched much game-play (directly or from video) of Super Mario Bros series,

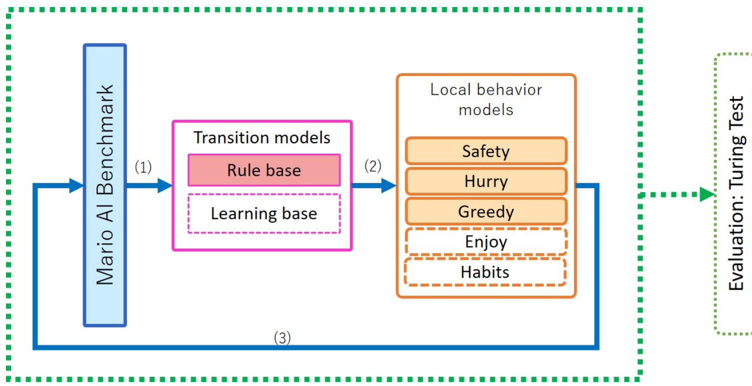


Figure 4.1: Overall research framework (1) information from testbed passing to Transition model. (2)Select one behavior model to do the task. (3) feedback the action from the selected model.

and selected five typical and frequent play styles as follows:

- Some players take some distance from the enemy or dangerous object when approaching to it.
- Some players are attracted by the extra rewards (coin) or upgrade items (mushroom or flower).
- Some players change their behavior when they were forced by time limitation
- Some players produced some behavior just only for expressing their enjoyment.
- Some players show behaviors affected by their individual habits.

These behavior are easily found in the Super Mario game-play. These behaviors are not all representations of human behavior, but just only rough groups of behaviors. But we guess that they may be enough for presenting human-likeness in Super Mario. And also we believe that these groups of behavior are able to express some emotion. Thus, based on the eight base emotions of wheel of emotion which was presented by Robert Plutchik[3], we proposed an idea to represent the selected behaviors as well as a simple transition to produce human-like behaviors. Note that we did not aim to produce emotion itself but Our behavior models provide different play styles to make the AI look like it has emotions. The models are explained as follows:

- "Safety" reflects the anxiety and fear of the player when on guard.

[3]https://psicopico.com/en/la-rueda-las-emociones-robert-plutchik/

- "Hurry" reflects careless speedy actions, when the player worries about the remaining time.
- "Greedy" reflects the enjoyment of humans when they find rewards.
- "Enjoy" reflects enjoyment and interest, such as killing enemies continuously.
- "Habit" reflects unintended behaviors, such as pressing repeatedly the jump button.

We propose a research framework, as described in two layers: the "Behavior Model" and the "Transition." The research is conducted using the Mario AI benchmark as a test bed. The implementation was conducted in the Java environment. The Behavior model includes five elementary models (i.e., "Safety," "Hurry," "Greedy," "Enjoy," and "Habit"). Each was designed to simulate a specific behavior, which is likely influenced by emotion. The transition between model has been designed as two types in preliminary, i.e. if_else hand coding approach and learning based approach. The detail of the hand-coding rule-based approach will be described in the next section.

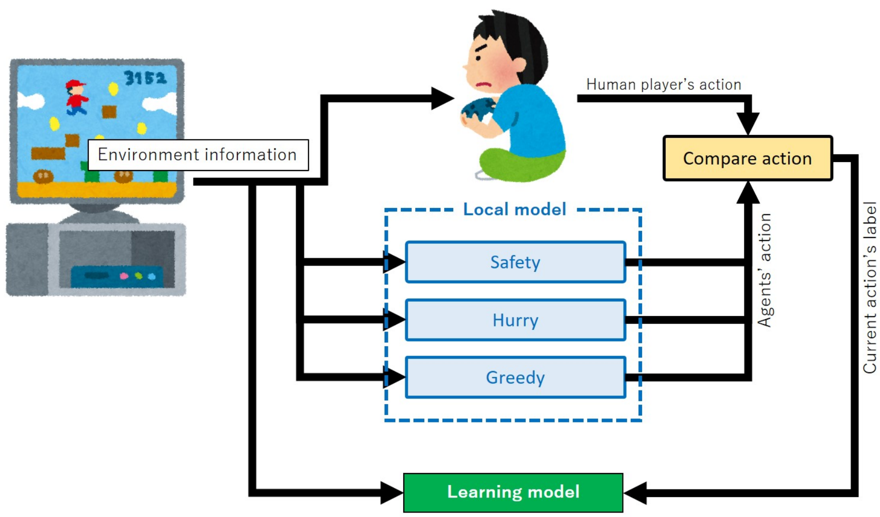


Figure 4.2: Labelling action for SL approach

The other models, we decided to introduce SL in order to reproduce a human-like transition. As the preparation for the learning, gameplay data which contain environment information and the action from a human player will be collected frame by frame. In each frame, action will be compared to the actions from the local models which gain by feeding the same environment information into each local model, and the action will be label by the result of the comparison. The obtained label and environment information will be used as learning data for SL.

The human-likeness of each model is important for the overall human-likeness. Thus we first decided to focus on the Safety model, Greedy and Hurry models. The implementation of each behavior model is hand-coded, in other words, unsupervised, and based on the A* algorithm which showed the best performance of game playing in The Mario AI competition in 2009-2011. The other two behavior models, which are Enjoy and Habit models, are difficult to implement by using such a search algorithm. While we planned to conduct the experiment by using a machine learning algorithm, there are many difficulties in using a machine learning algorithm such as collecting training data. Thus we decided to postpone the implementation of these two models at this time. Also in this article, only rule-based transition model will be described.

4.3 Approach

Our approach based on the A* algorithm which is the best player in the Mario AI competition in the gameplay track where the AIs are contested in term of performance. Thus the ability to clear the stage of A* algorithm is trustable and well enough to modified human-likeness.

A* algorithm in Mario

The A* algorithm is a well-known path-finding algorithm. By using the best-first search, A* finds the path with the lowest cost from a start node to a goal node. To compare traversal paths, a cost function for A* is defined and used:

$$f(current, goal) = g(start, current) + h(current, goal) \tag{4.1}$$

Where f(current, goal) is an estimated total cost of a current node, which is the sum of g(start, current); the actual cost from the start node to the current node; and h(current. goal), the heuristic estimation from the current node to the goal. In the Mario AI competition, Baumgarten presented an efficient controller using

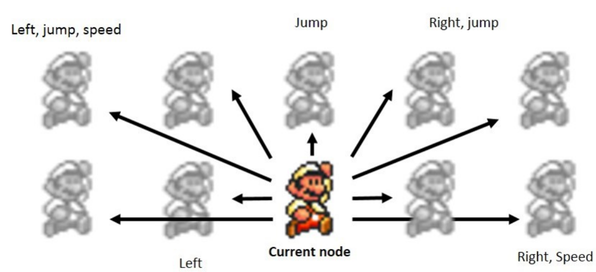


Figure 4.3: Possible nodes for A*

a modified A* algorithm, which computes possible trajectories of Mario. The video of the controller was published and has been viewed over 600,000 times in a short period because the performance is excellent, and the behavior is far more from human players.

The algorithm expands the path by nine actions, as shown in Fig. 4.3 (i.e., left, right, jump, dash/fire), where g(start, current) is defined as the time that the controller used to reach the current position and h(current, goal) is the estimated time from the current node to the goal with the current speed [4].

Our idea of the Safety model and two samples, one being the Greedy model and the other being the Hurry model, is shown in Fig. 4.4.

The first and the most important behavior is the Safety behavior. Most of the time, human players try to play safe, so their character can survive and move toward the goal. Based on this kind of behavior, our Safety model is created, and it allows the character, in this case Mario, to move steadily and carefully. In addition, the character is able to recognize dangerous areas, and it hesitates to move forward until the dangerous turn into a safe area. The area changes from dangerous to safe if the enemies are killed or they disappear. The second behavior model is Greedy. While the Safety model forces the character to pay attention to the enemies, the Greedy model leads the character to the locations of coins. Instead of moving toward the goal, the character moves to the location of a coin. This is only one example related to the Greedy model, where the character will only move to the coin's position.

Similarly, an example of the Hurry model would be making Mario move as fast as possible to reach the goal without paying any attention to enemies or coins. The mechanism of each model will be explained following.

4.3.1 Safety Model

Maslow explained the motivation of humans in a hierarchy of five layers of needs. The term "Safety" has been used to describe the needs of health, well-being, and safety against adverse impacts. Moreover, the need for safety can influence a player's behavior. While playing a game, movements can be affected by anxiety or fear. For computer players with perfect control and information, precise actions, such as evasion from an enemy by one pixel, are possible. Nevertheless, beginner and intermediate players are aware of their imperfect controls and perceptions. Thus, a safer movement, such as keeping distance from each enemy, is preferred.

The safety model imitates such a behavior by introducing a "dangerous area" to the A* algorithm. The "dangerous area" surrounds each harmful object (Fig. 4.5), so the safety model controller intends to avoid the object and the "nearby area." The heuristic function of the A* algorithm is defined as:

$$h'(st) = S \to R | h'(st) = RP_t + MP_t – h'(st-1) \tag{4.2}$$

Where s is the state of the game at frame t, RP_t is the penalty from real damage that Mario takes in frame t, and MP_t is the penalty from the virtual damage from the dangerous area, as shown in Fig. 4.5. There are many kinds of

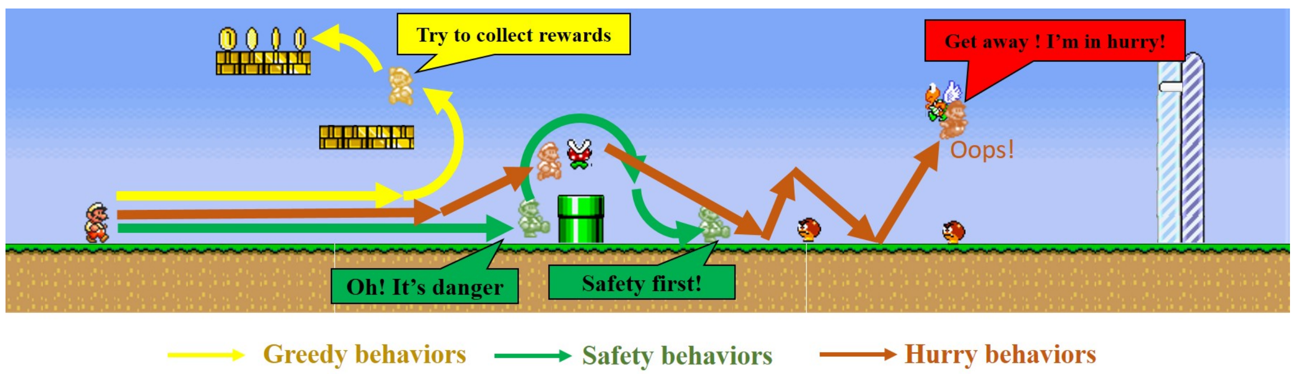


Figure 4.4: Safety, Greedy, and Hurry models

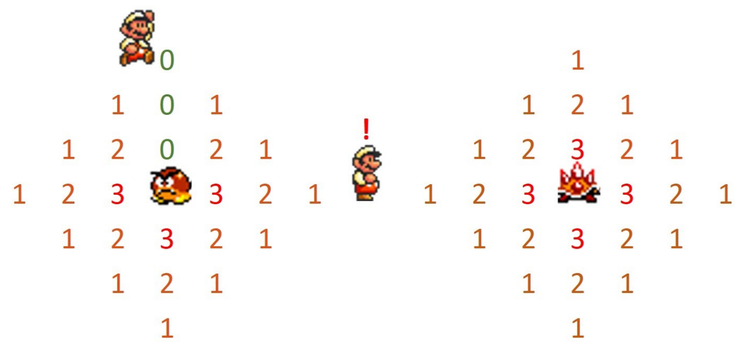


Figure 4.5: Dangerous area

harmful objects, so there are also many kinds of dangerous areas. For example, fast-moving enemies should have a wider area compared to slow or fixed objects. If we compare the two areas shown in Fig. 4.5, the left has an isotropic area. On the other hand, the right one has no virtual damage over the enemy. This is because some enemies can be stomped on, and in such a case, Mario is not damaged.

4.3.2 Hurry Model

The major goal of Super Mario Bros. is to clear the stage by traveling to the rightmost end of the stage without being killed. In each stage, 300 seconds are given, and after 200 seconds have passed, there is a warning sound and the background song will quicken. Afterwards, the player will be aroused and try to clear the stage as fast as possible. Sometimes the player might ignore remaining coins, give up on killing enemies, or even ignore damage that does not kill Mario immediately, such as in the "Fire" or "Big" state. Thus, we proposed the Hurry model to display such behavior. The implementation is based on the A* algorithm of Baumgarten, including the concept of the dangerous area, which smaller than in the Safety model.

4.3.3 Greedy Model

In the Mario game, coins and items are rewards that provide some benefits to Mario. Collecting coins adds to the score of the player, and for every 100 coins, the player gains an additional life. Items give to the player not only a score, but also a status upgrade from Small to Big or to Fire. Sometimes the player's attention might be drawn to these rewards. Our Greedy model imitates such attention to rewards. This behavior reflects the enjoyment of humans when they obtain a benefit. The main idea of the Greedy model slightly differs from the Hurry model. The target of path finding is set to coin locations and item locations, instead of the real goal. For instance, in Super Mario Bros., we assume

(x_g, y_g) and (x_ci, y_ci), where (x_g, y_g) stands for the coordinate of the goal and (x_ci, y_ci) stands for the coordinate of the coin with the index i. As a result, the Mario character will change its target to the coordinate of coin i, and only when coin i is collected will the target change to the next coordinate of coin i+1. When there is no coin left in the area around the character, the target changes back to the coordinate of the goal. Finally, by using the A* algorithm for path finding, we were able to make the character move to the expected target.

4.3.4 Enjoy Model

Sometimes the player might face some challenging situations that are unnecessary to solve to clear the stage. Yet, the player might enjoy such a situation. In the case of the Mario game, if the player is able to stomp continuously on enemies without falling to the ground, the score will double for each kill. The challenge may provide a big score, but it is not necessary to clear the game. We present the interest and enjoy emotions in this model.

4.3.5 Habit Model

We found that for some human behaviors, it is impossible to identify the purpose or even the reason for the behavior. Often, human players produce actions having no aim or benefit, such as the player jumping all the way while running, even though there are no enemies or obstacles in the game scene. The behavior might occur by instinct or sometimes with the player's intention. We defined such a behavior as the Habit model.

4.3.6 Switching Model

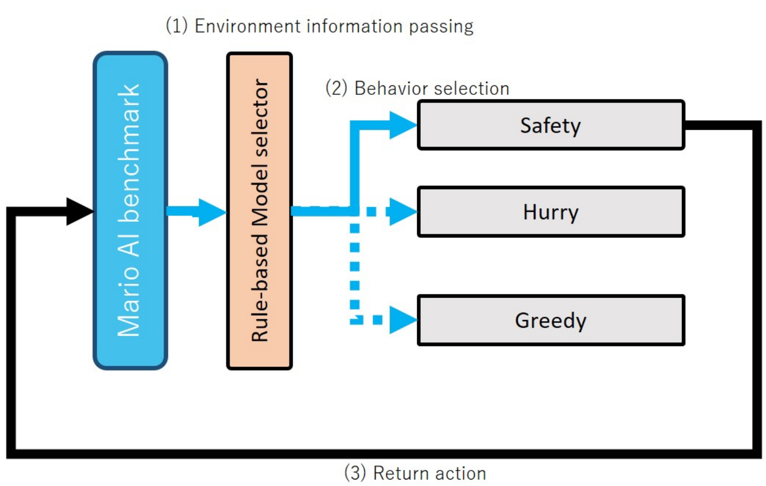


Figure 4.6: Switching model framework

Human behavior is more complicated than we can imagine. To illustrate, at a specific period in the game, a human player uses the Safety style, but after a numerous rewards appear, the player changes to the Greedy style and begins to collect as many rewards as possible. Hence, we propose this model based on the idea that human players change behaviors during the game. The mechanism of this model is simple. It is the combined models of two or more individual models, which we already introduced in the previous part. To change between models, a set of rules is used as a switch. If the information in the environment around a character meets the condition, the character is able to change to an appropriated behavior model. For example, if the number of coins is greater than three, the Greedy model is activated, or if the number of enemies is greater than five, the Safety model is activated. When one model is activated, others are disabled.

4.4 Experiment and analysis

The assessment of each model incorporated the Turing test method of the Mario AI Championship 2010. We want to confirm the human likeness of the Safety model and the possibility of the idea as an extra. The preparations were done by collecting the replay from a human intermediate player. The player was asked to play the game in 10 stages, with four various instructions.

- "Please clear the stage as safely as possible"
- "Please clear the stage as quickly as possible"
- "Please clear the stage and gather as many coins as possible"
- "Please play at will"

In the same set of stages, the replays from the "Safety model player" were collected. We also implement a sample Greedy model that aims to collect coins, a Hurry model that aims to clear a stage as quickly as possible, and simple rule-based switching for these three models. We employed 15 human subjects whose mean age was around 20–30, who have experience with the game, and who know the rules of Super Mario Bros. The subjects were asked to observe pairs of non-label replays. Then, they have to answer the following question: "Q1: How expert is this player?" "Q2: Does the action of this player look natural?" The answers to each question were based on a 5-point Likert scale. Subjects were asked to compare replays one by one, such as to compare a human player with a safety instruction to a Safety model computer player or a human with a greedy instruction to a Greedy model player. The displayed orders are random and each type appears 37–38 times.

Experimental Results

The results of the experiment are shown in following chart(Figure 4.7) and table 4.1. The highest average score is human with greedy instruction by 4.297 where

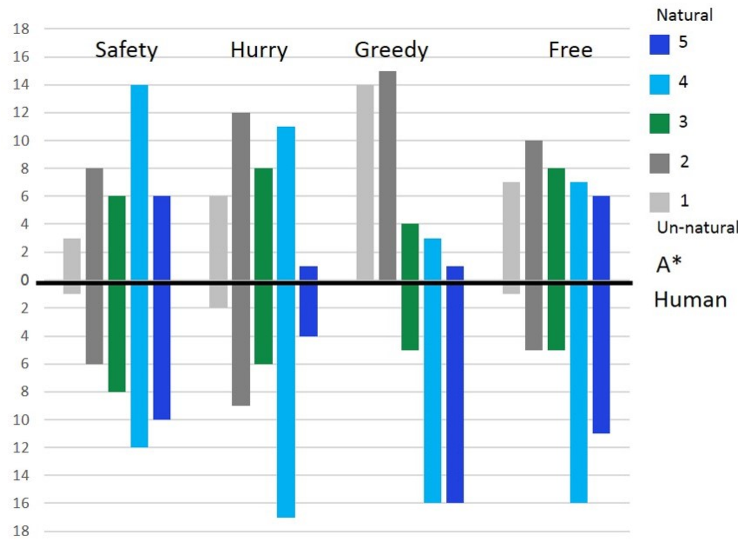


Figure 4.7: The score distributed on human-likeness of each model compared with human play in the same propose.

the lowest is our greedy model(average score = 1.973). By the adjustment of reward of coin in the calculation of algorithm in the algorithm the model seems to pay more attention to extra reward. however, there some ignorance in some upgrade item such as Fire flower where the human will try to get the item even it has no effect (The item will show no effect if character is in the upgraded state). Also the greedy model consider less of any risk or danger, thus the model showed a significantly lower score compared to the human with a greedy instruction.

Considering the average scores of four proposed models, all of them failed to gain equal or better score than human players who are instructed to do safety/hurry/greedy playing. We also conduct Mann–Whitney U test which is a a non-parametric statistical test of null-hypothesis and the result is shown in Table 4.1. Our Hurry, Greedy and Free models are significantly worse than human players (p = 0 and p = 0.001). Only in the result of Safety model, the average score 3.324 is not so bad compared to that of human players 3.649. p-value was 0.145 (z-score = -1.06), it is not significant at p ¡0.05 level. The most common reason for low scoring for the safety model is “there are nearby coins that should be collected but they are not.”

We also implemented rule-based switching to switch among the Safety model, the Hurry model, and the sample of the Greedy model to confirm whether believability will increase if a computer player produces many behaviors. The results show improvement in the Switching model compared to only the Greedy model or only the Hurry model, but it is still lower than the Safety model. The main reason is that overall performance depends on the quality of all individual models.

	Mean(SD)				$z - score$	$p - value$
	A*		Human			
Safety	3.324	(1.226)	3.649	(1.136)	-1.06	0.145
Hurry	2.711	(1.137)	3.316	(1.118)	-2.17	0.015
Greedy	1.973	(1.040)	4.297	(0.702)	-6.55	0.000
Free	2.868	(1.359)	3.816	(1.087)	-2.98	0.001

Table 4.1: The Average scores compared between modified A* and Human in each type of behavior (in parentheses showed standard deviation of each)

Considering human plays, evaluations are high in the case of the free will instruction and the greedy instruction. The reason is that people tend to play in multiple styles (they play safe even when collecting coins). It might be said that the player with multiple behavior types looks more human-like than those with a single behavior do. Thus, behavior transitions are important for making a believable computer player.

To verify the performance of the agent, the comparison with the previous work is necessary and there are many attempts to human-likeness of Super Mario AI. For example one of the famous is reinforcement learning imposes physical constraints by Fujii et al [6]. However, considered to the current result especially "Free" style in the Table , they still showed the inferior performance. Thus, the work will be compared to the other work in the same field after the improvement of the current state in the near future.

4.5 conclusion and future works

We are able to confirm our hypothesis in the Safety model. Staying safe is a significant behavior among human players, which refers to maintaining distance from enemies and avoiding risky play. This important behavior makes a computer player appear more human. The result of the Turing test has shown that the believability of the Safety model is almost closely equal to a human player. However, there is still a claim from subjects about a lack of some behaviors, such as "searching for coins or items." These are related to our original hypothesis, where the human-like behavior contains multiple behavior types. In this article, we introduced Hurry, which involves risky play, and the simple Greedy model, which ignores all enemies. The quality of their believability in the Turing test is low, but after we combine them all using simple rules, there is an improvement in believability.

Our future work will concentrate on the believability of the Greedy model and the Hurry model, as well as on a better transition between these models. Additionally, a learning-based transition will be employed in the near future. Moreover, the evaluation for each behavior where the subjects have been informed about the type of agent, in other words, the evaluation of human-likeness for specific behavior should be conducted.

Chapter 5

Sub-purposes oriented human like behavior

This chapter is written and modified based on the article " Survey of How Human Players Divert In-game Actions for Other Purposes: Towards Human-Like Computer Players" which is published in the international conference IEEE-Entertainment Computing ICEC 2017 at Chiba Japan in 2017.

5.1 Background

Nowadays, game-AIs are strong enough in term of performance to surpass human players in many domains, especially in classical board games such as chess or the game of Go [3]. In video games, which are more complex, DeepMind showed that a computer player was able to play 49 games, among which 29 were at the level or surpassing human record levels or scores [65] [62] . Game-AIs are now strong enough to be a human opponent or partner in terms of performance.

However, performance is not enough to entertain human players. In recent years, generating "human-like" behavior has become an important target among game researchers [66].

For example, one attempt to generate an entertaining game-AI by Ikeda and col-leagues presented a method to entertain human players in the game of Go by letting them win without allowing the players to notice their advantage, by choosing sub optimal actions but avoiding obviously bad ones [67].

Togelius and colleagues introduced the idea of "believability," which refers to the ability of a character or bot to make someone believe that the character is real or being controlled by a human being [11]. Many approaches were proposed to obtain believability and produce human-like behavior, such as in Fujii and colleagues, where a human-like computer player is obtained by simulating biological constraints. This approach considers human-likeness only in relation to the game's main objective [6].

In some games, the human player can perform actions "outside" of the game itself, such as bluffing or using facial expressions in poker games, or natural language communication via VOIP programs such as Skype. This communication is performed "outside" of the game itself in order to achieve the game's main objective. Such actions are also an important target for human-likeness.

However, sometimes, human player actions are not directly related to the game's main objective. For example, in FPS games, some players try to use their guns to create illustrations with bullet holes; in racing games, some players stop just before the goal, wait until another player comes closer, and then reach the goal. Such actions can be observed in many types of games, such as action games, RPGs, MMORPGs, puzzle games, and racing games. They are sufficiently frequent and significant to be a target (or even possibly a necessity) for obtaining human-likeness in computer agents.

In this research, we focus on human players' actions in-game that are not directly related to the game's main objective. We collect study cases from several types of games and classify them into seven types (i.e., warning, notification, provocation, greeting, expressing empathy, showing off, self-satisfaction). We also discuss the context in which these types of actions appear. In addition, we present an experiment that shows how multiple Q agents in an easy hunting game learn to divert game actions from their original goal in a way that we believe is similar to humans.

5.2 Purpose and goal

Recently, there are a number of approaches that aim for creating entertaining computer players. Specifically, computer players with human-like mannerisms are a very popular subject of interest among researchers. The idea of human-like AI was originally proposed by Alan Turing in the Imitation Game, which was the starting point of the Turing test [68]. Togelius and colleagues defined "believability" as the ability to make someone believe that the character/bot is being controlled by a human player [11]. So far, believability can be assessed by conducting a Turing test, which is mainly conducted by observing in-game behavior from a third person perspective.

Many computer players/bots in competitions were assessed according to believability assessment in order to indicate their performance in the believability aspect. For example, an assessment of the competition of computer players in an FPS (first person shooter) game was based on an Unreal tournament [1]; another assessment was based on competition in an action side-scrolling game, Super Mario Bros [69].

In an FPS game, finite state machine which represent state by information of combat and collected item, and another approach using behavior tree combine with Neuro-Evolution, performed good performance to play game with human-like behavior. In a Turing test tracking a Mario AI competition in 2012, top ranking human-like computer players used artificial neural network, influence

[1]The 2k bot prize,http://botprize.org/, Accessed 2017/04/20

map, and nearest neighbor methods [70]. Also, in recent years, Fujii and colleagues presented approaches to creating believable computer players by using biological constraints, which are applied to Q-learning and the A* algorithm [6]. These earlier methods are improved based on action/behavior in-game directly concerning the main objective of the game ((1) in Table 5.1) (e.g., Super Mario Bros: reach the goal at the right-most end of the screen). However, there are many behaviors that are produced which indirectly related to game main objective.

	Directly concerned with main objective	Indirectly concerned with main objective
Inside a game	(1) Normal play	(3) provoking, reminding, etc.
Outside a game	(2) Bluff, Skype, etc.	(4) Screaming, leaning, etc.

Table 5.1: Human player's action categories

Shiratori and colleagues showed another aspect of human-like behavior: communication outside the game, such as through facial expression and bluffing [43]. This behavior is important in some board games or card games such as Poker and Mahjong. They presented computer players in a fighting game with facial expressions outside the game that match the in-game state. Such "outside game" actions are related to a game's entertainment value, similar to shouting when a game character is being attacked ((2) in Table 5.1). As the study of sub purpose, Yee presented the analysis of sub purpose of humans' behavior in a psychosocial perspective for playing MMORPG (Massive Multiplayer Online Role-Playing Game). His findings say that players of MMORPG are not just only playing the game but also there are some sub purposes such as "Interact with unknown player", " do something to satisfy themselves", "pretend to be someone else", or etc. [71]

Current human-like behavior or believability research is focused on action in-game or outside the game that is mostly based on the intention to clear the game's main objective. However, human players also take some actions with intentions that are indirectly related to the main objective of the game. For example, outside the game, human players might scream when their characters get attacked in-game or may lean in the direction of their characters when turning in a racing game. Human players often take these unnecessary actions in order to immerse themselves in the game environment.

In-game, human players might show behaviors with intentions other than to clear the main objective (3) in table 5.1). For example, some players make illustrations using bullet holes in FPS games. These in-game actions are used for reasons other than reaching the game objective, for purposes such as provoking, reminding, or warning the other player (for example, punching when not in attack range to provoke the opponent in a fighting game). We created Table 5.1 to briefly explain the behavior of human players in response to a game. These behavior have been widely discussed in the field of games and culture; for example, Tylor studied the behavior of players in the game World of Warcraft [70]. However, in the study of computer gaming AIs, these behaviors still receive less attention, especially (3), which is the main subject of discussion in this article.

5.3 Clustering of Human Behavior Not Directly Related to The Game's Main Objective

In section 2, four types of action were presented. When playing a game, human players do not only aim to clear the game's main objective (such as getting high scores, clearing the stage, or defeating the enemy), but they also divert game actions for purposes not directly related to the game's main objective. In a game where co-operation with another player is necessary, some actions might be used to transmit a message, such as a notification or warning about something, or to provoke an enemy.

For human players, it is also possible to notify or warn about something with natural language by using an in-game chat system or VOIP programs such as Skype. However, in some situations, these communication channels are unavailable (e.g., chat is not available in the game or "too busy to chat, chat is difficult"), and in that case, actions inside the game itself can be diverted from their original use and used to represent different meanings or intentions.

In this Chapter, we show study cases where human players seem to select their actions not to win, but for another purpose. Many gameplay videos of human are reviewed, and 50 typical cases are selected. In this article, we introduce 15 cases which easy to understand. The cases we show are not a complete or representative set of all such behaviors. There might be other relevant examples, but we believe these cases are the unique behavior of human and valuable to consider as human-like behavior.

5.3.1 Group of The action clustering by the purpose

The study cases we reviewed, are grouped into seven classes according to the purpose of the action, such as warning, provocation, or greeting. Some of these action types, such as the warning type, are highly related to the main objective of winning, and some of them, such as greeting, are less related. The following subsections explain the seven types of indirect action, from highly-related ones to less-related ones.

Warning

In cooperative games, warning is important for developing a strategy to clear the main objective of the game. Vocal warnings, alarms, or simple signals are part of some games. However, in cases where these functions are not available or the player is unavailable at that time, players often use other actions to transmit their messages. Actions with the intention of warning are strongly related to the main objective of the game. For humans, these actions facilitate a feeling of cooperation, so it is important for game-AI to produce and understand such actions in order to increase the satisfaction of human players with game-AI. We show two study cases of warning actions

- Study case 1 (MOBA: League of Legends): Warning a team member about an incoming enemy or enemy action by using the "?" mark available in the game itself instead of the in-game chat, which can be used but consumes more time.
- Study case 2 (FPS: Sudden Attack): When the player notices a sniper, he/she shoots the nearest wall or corner in order to warn allied players (Fig. 5.1)).

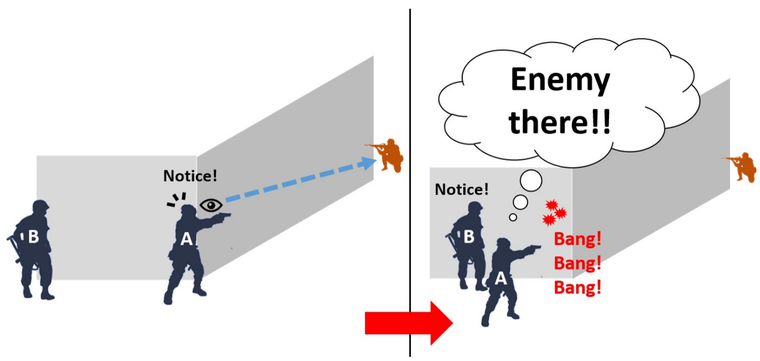


Figure 5.1: Warning action in FPS Game: (Left) Player A moves out from the corner of the building and finds an enemy, then (Right) Player A tries to warn player B by shooting the nearest corner.

Notification

A notification action is defined as an action where the intention is to tell something to an opponent or ally, such as "Let's start the match!", "Please surrender!", or "Hey, come here!" Notification actions are strongly related to the game's main objective, thus the implementation of them in game-AI might be not difficult. The following study cases show some examples of the notification action.

- Study case 3 (MMO: Maple Story): To notify another player that there is a forgotten item on the floor, the player jumps repeatedly over an item with one hand over the item or in the direction of the other player. The intent of this action is to tell another player "there is an item here" and "please pick it up hurry" (Fig. 5.2.).

- Study case 4 (Action fighting: Super Smash Bros): In the character selection lobby in online matching, the match will start when all players press the ready button. In this phase, it is possible to press ready and cancel repeatedly to notify another player to “hurry up.”

- Study case 5 (MMO: Dungeon & Fighter): In a dungeon while the team is co-operating, using attack actions at the door or passage that a player wants to explore indicates the destination to the other teammate.

- Study case 6 (Board Game: Go): In a situation where one player has an advantage on the other player, a clearly sub optimal move is chosen on purpose in order to transmit “I can beat you even if I choose this kind of sub optimal move. Surrender now!”

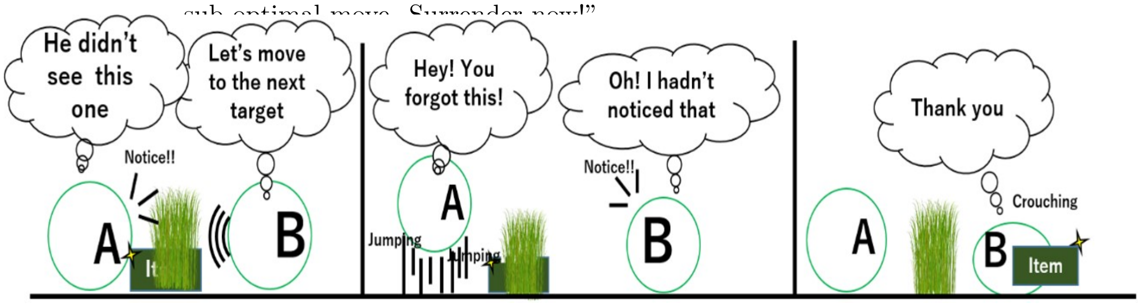


Figure 5.2: Notification action in MMORPG game: (Left) Player A finds an item that Player B didn’t notice; (Middle) Player A jumps repeatedly so that Player B notices the item; (Right) Player B picks up the item and expresses his gratitude by crouching.

Provocation

Provocation (or “trolling”) is an action that tries to frustrate the opponent when the player is in an advantageous situation; this gives some small impediment to the opponent with malice, or a player might put him/herself at disadvantage on purpose. This action often occurs when a player is able to keep superiority in the game continuously. Normally, human players do such actions in order to satisfy themselves. However, sometimes, the goal of provoking or trolling is to lure the opponent into a mistake and might be strongly related to the game’s main objective. Reproducing such actions with Game-AIs might not increase human players’ satisfaction, but these actions are important for human-likeness.

- Study case 7 (FPS: Call of Duty): Moving, jumping, and crouch-standing repeatedly around a defeated opponent character’s (dead body) location to aggravate the defeated player (Fig. 5.3 .)

- Study case 8 (Fighting game: Super Street Fighter II): In fighting games, after a round is finished, a player can punch or kick the dead body of the losing player to provoke the opponent

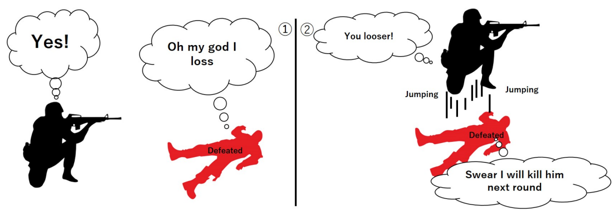


Figure 5.3: Provoking action: In an FPS game, moving, jumping, and crouch-standing repeatedly around a defeated opponent character to provoke the enemy

Greeting

Greeting refers to an action where players communicate something like “Hello”, “Nice to meet you”, “Thank you”, or “My bad” (this category often includes apologizing). Normally, greeting (or apologizing) is done via VOIP programs such as Skype or built-in chat systems in the game. However, in some cases when chat or VOIP are not available, actions in-game are used to express these sentiments. Greeting has a weak relationship with the game’s main objective. In action games where a crouch action is available (normally for evasion from an attack), it is often used to perform greeting or apologizing.

- Study case 9 (Action fighting: Super Smash Bros): Players use crouch-standing repeatedly to express “Nice to meet you” when creating a team battle. The meaning of an action can change depending on when it is performed. After doing something considered as bad manners, a player can apologize to other players by crouch-standing repeatedly.

Expressing Empathy

Expressing empathy refers to actions that expect some response from an opponent or are used to provoke some action from an allied player; these express a “Let’s have fun together” feeling. These actions are done without malice.

- Study case 10 (Fighting game: Super Street Fighter II): Some players enjoy using attack actions or jumping outside attack range for fun, expecting the opponent to do the same thing in response.

- Study case 11 (MMORPG: Final Fantasy XIV): Sometimes, mass numbers of players come together and try to use in-game actions called “emotes” to express some movement or dance at the same time.

Showing Off

Showing off is an action based on appearances rather than performing to meet the game’s objective. Sometimes, this action might be conducted in order to provoke an opponent, but many players try to perform this type of action seriously.

- Study Case 12 (3D Fighting: Soul Caliber, Fate/Stay night) In some fighting game, the combo (series of action) which difficult to perform, afford cost is higher than performance (damage), but the appearance is good, such combo is exist. Some player tries to perform such showoff combo.

- Study Case 13 (Street fighter III) Counter Attack or Blocking are existing in many fighting game. Using such attack allowed player to who encounter the attack avoid and strike back without taking any damages. However, blocking action has to be perform suddenly after opponent perform attack which is very difficult.

Self-Satisfaction

Self-satisfaction actions include actions taken to pursue curiosity or to bind or constrain play. To bind play is to play the game with extra rules stated by the player him/herself, such as clearing the game at a low level, limiting item uses, or clearing a level or area without damage. Bind play is also performed for creating new styles of play.

Another type of action that players take to satisfy themselves is a creativity action. In games with a high degree of freedom such as Minecraft and Mario Maker, players can try to create innovative stages in their own style.

- Study case 14 (Action RPG: The Elder Scrolls V: Skyrim): Some players might enjoy exploring locations in the game that are normally hard to reach, such as the top of a mountain.

- Study case 15 (Action: Resident Evil): Some players implement extra rules such as "clear with limited equipment or weapons" or "stay alive at a low level of life or hit point throughout the game."

5.3.2 Condition of Appearance

We provided examples of study cases and clustered them into seven groups by the purpose of action. Also, the conditions that cause this behavior to appear are very significant.

In-game behavior that is indirectly related to the main objective can be observed in many types of games. However, some of these actions require minimum knowledge of or skill at the game or the type of game in order to be interpreted correctly. For example, in study case 7, the action of moving around the dead body of a defeated player might not be understood by a beginner. However, along with the improvement of player skill, action comprehension becomes deeper. When reproducing such actions in a computer player's behavior, it is necessary to take into account the skill and knowledge of the human player.

Limiting the information that players have affects their comprehension of a behavior's intention. For example, in card games, we can help or impede an opponent by discarding a card, However, if the opponent only knows that a card has been discarded and not for what purpose it can be discarded, it may be difficult to distinguish the other player's intention. Thus, when information is limited, these actions rarely appear. On the other hand, in games with complete information such as fighting games (where health, time, and/or a special attack energy gauge are shown), the intention of actions is easier to understand.

However, in some kinds of games, even when information is limited, human players can estimate the situation by using action information. For example, in FPS games, if enemy attacks suddenly stop or weaken, then the player might assume that opponents have changed their strategy. In this situation, the player is able to fill in the missing information and estimate the intentions of actions.

Another factor is the amount of spare time a player has in game, which relates directly to the difficulty of the game and the situation in game. In a game in which chat is allowed and the player has spare time to use a chat, the game's main objective may become temporarily irrelevant while this action is performed. The degree of freedom of action regarding game tasks directly affects the appearance of this type of action. In games with busy tasks or games where every action in-game affects the score or victory such as Go or Tetris, this type of action will not be performed.

5.4 Experiment and analysis

Most of the actions we introduced in this chapter are unique to humans, though it is possible for some of them to emerge from systems without humans. As we mentioned in section 5.3.2, normally, game AIs are implemented in order to achieve the main objective provided by the game system. Some of sub-goals such as notification have stronger relationship with the main objective than others such as provocation. "Provoking opponent by waiting before goal" or "perform a very hard technique or cool technique to make good visualization" are normally make a disadvantage for achieving the main objective of the game. Thus, it is hard for game AI to learn by a simple learning process. There are many possibilities for making AI that is able to learn these kinds of behavior for example by making some extra modifications or giving an extra reward such enjoyment or fun. However, in this research, we firstly decided to implement AI with notification action by using two reinforcement learning agents, as a typical example.

Thus, we carried out two experiments to observe how actions not directly related to the game's main objective emerge from interactions between reinforcement learning agents.

5.4.1 Basic setting

The two experiments share a common setup. Two reinforcement learning agents with limited views try to catch a target while co-operating with each other, as illustrated in Fig 5.4. We expected these (limited sight) agents to substitute

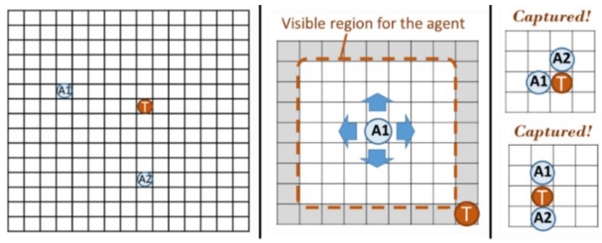


Figure 5.4: Overview of the environment. Two agents try to catch a target. Each agent has a limited view. In this case, when both of the agents touch the target, the task is accounted a success.

their sequential movement actions for a signal that the target is located near the agent.

- Environment:
  - The field consists of 15x15 grid spaces in which agents can locate the target.
  - Two agents and one target are randomly arranged on the grid initially.
  - Each agent can move to an adjacent grid space in a compass direction each turn.
  - The field has a torus structure. Agents will appear in the right-most when they go left on the left-most space and appear on the top side when they go down on the bottom of the grid.
  - More than one character (agent or target) cannot be located in the same grid space at the same time.
  - The target does not move.
  - In the case that both agents are located in grid spaces adjacent to the target before 100 turns passes from the initial state, the search and chase task is accounted a success. The task is accounted a failure otherwise. In both cases, the game state will be reinitialized.
- Agent: Each agent decides its action using a one-step Q-learning algorithm. The game state observed by each agent is a combination of feature values as below.
  - (F1) Coordinate of the target relative to the agent. The agent can find the target only when it is located inside the 7x7 grid area whose center coordinate is the agent; therefore, 49 values are possible for this feature.
  - (F2) Coordinate of the other agent relative to the agent. The limited eyesight of agents does not affect this information; thus, 224 (= 15 x 15 - 1) values are possible for this future.

- (F3) The number of turns during which the agent does not see the target. Once the agent finds the target, this value is set to zero. MaxT +1 values are possible for this feature, where MaxT is a parameter value of the agent (if the number of such turns becomes greater than MaxT, this feature value is set to MaxT).
- (F4) The last MaxH actions taken by the other agent, where the MaxH is a parameter value. The number of possible actions an agent can take is five (go up, down, left, right, or stay), therefore, 5 MaxH values are possible for this feature.

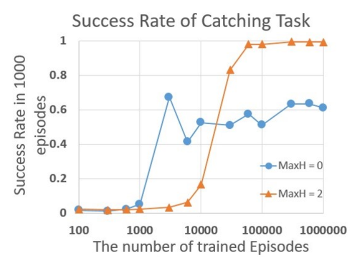
Our agents have two parameter values for observing game states, MaxT and MaxH, as stated above. Additionally, there are other parameter values related to the learning algorithm. The reward is 100 for reaching a terminal state by succeeding in the catching task, 0 for failure.

$$\begin{aligned} &\gamma = 0.8 \\ &\alpha = 0.1 \times \frac{1{,}000{,}000}{1{,}000{,}000 + Number of episodes trained} \end{aligned} \tag{5.1}$$

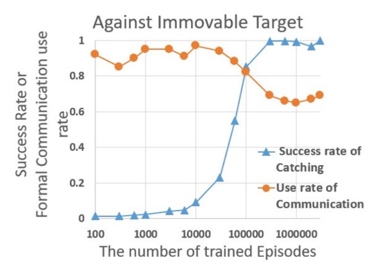
The agent hold above setting where γ is discount factor and α is learning rate. In addition the agent adopts the ϵ-greedy policy with ($\epsilon = 0.1$) as its behavior policy.

5.4.2 Experiment 1: Observing Emergence of Action Substitution

- **Setting**: We compared the movement patterns of agents under two parameter settings, that is MaxT= 0, MaxH= 0 and MaxT= 0, MaxH= 2. In the MaxT= 0, MaxH= 0 case, each agent must decide its action according only to the current positions of the other agent and the target (if it is located within eyesight). Therefore, agents cannot give their partner any clues to find out the target's location. On the other hand, in the MaxT= 0, MaxH= 2 case, each agent is able to pay attention to the movement patterns of its partner. Therefore, agents have a chance of telling their partner the location of the target by showing the partner some characteristic movement patterns



(a)



(b)

Figure 5.5: (a) Success rate of the task. The success rate was enhanced by 40% when each agent memorized the last two moves of the other agent. (b) Success rate and the increasing use rate of the communication function to signal the immovable target. Through the whole training process, agents came to rely on the function at a rate of 70% (i.e., any agent used the function in more than 700 episodes out of the last 1000 episodes on each plot).

- Result: We observed the movement patterns in both settings after 3,000,000 training episodes. In the case of MaxH:0, each agent moved chaotically until the target came into sight. After that, the agent rushed at the target. In the case of MaxH:2, each agent used regular zigzag movement patterns for exploring (e.g., goes up, left, up, left, . . .) until the target came into sight. Whenever an agent found the target, it changed its movement pattern, avoiding moving away from the target (e.g., goes up, down, up, down, . . .) to inspire its partner to approach. Fig. 5.5. shows the performance of agents under these conditions. Introducing the information about the action history of the partner agent enhanced the success rate by 30% in the end, even though that information does not contain any direct clue for the target location. Therefore, we think the enhanced performance was caused by the emergence of an action substitution, that is, agents substituting their movement action to signal their partner.

5.4.3 Experiment 2: Encouraging Action Substitution

In the experiment described below, we aimed to show that even if there is a formal way to communicate with partners, agents prefer to communicate through action substitution if the situation is urgent and the formal method requires more time.

- Setting: We added two rule options for the system.
  - Escaping target: The target can also move in this rule. It moves away from agents once every four turns and in the case that any agent catches the target in its (limited) field of view.
  - Formal communication: Each agent can inform the other agent of the precise current location of the target at the cost of becoming immovable during the following four turns.

Figure 5.6: Success rate and use rate of the communication function while chasing the escaping target. Eventually, agents began to avoid using the communication function in all but 20% of episodes.

We compared the movement patterns of agents under two option settings: Escaping target: Off, Formal communication: On and Escaping target: On, Formal communication: On. In either case, the agent parameter setting is MaxT= 21, MaxH= 2. That means agents can use both action substitution and formal communication to tell their partner the location of the target.

Compared with action substitution, formal communication requires a larger number of turns to communicate and inform the partner of the more precise location of the target. Thus, the formal method imitates text chat or Skype in actual multiplayer video game situations. In this experimental setting, we observed how the frequency of formal communication adopted by the agents varies.

- Result: The success rate of capturing the target and frequency of use for formal communication are shown in Fig. 5.5 (A) and Fig 5.5 (B). In the Escaping target: Off case, the rate of use for formal communication is around 70%. We think this is because the communication method is useful in capturing the target. Meanwhile, in the Escaping target: On case, agents adopt the communication method in only 20% of episodes. We think the reason why is that the "capturing escaping target task" does not allow agents enough time for formal communication. On the other hand, the success rate of tasks after training is similar in both settings. This means that agents use action substitution more frequently in the scenario with an escaping target than with an immovable target, but they are able to use this method with comparable effectiveness to capture the target. Otherwise, the success rate would have largely dropped off in Fig. 5.6.

Experiment 1 showed the emergence of action substitution, in which movement actions are used as signals between agents. The agents obtained this method automatically through reinforcement leaning, without any specific if-then routines for action substitution emergence. Therefore, we insist that agents in a system without humans can automatically obtain a type of in-game action that does not directly achieve the main goal (or at least an action pattern that appears to fall in such a category).

Experiment 2 demonstrated how the degree of urgency affects the probability of action substitution emergence. A higher degree of urgency makes agents less likely to use a formal method of communication and encouraged them to use substituted actions for their communication.

The work in this chapter including the collecting sample, analysis, classify action, and design the primary experiment, was under collaboration with Nakagawa Kento. The implemented and analyzed of the experiment were done originally.

5.5 conclusion and future works

In this research, we showed new aspects of human-like behavior that it is possible to categorize in two sets of categories and in four ways: inside or outside the game, related or not directly related to the game's main objective. We focused on actions without the intention to clear the main objective of the game, which we think are a significant behavior specific to humans and necessary to achieve human-like computer players. So far, 50 study cases of human actions were collected. These study cases were classified into seven types of behavior (i.e., warning, notification, provocation, greeting, expressing empathy, showing off, and self-satisfaction) and we discussed the occurrence conditions and possibility of reproduction by computer players. Furthermore, we conducted an experiment that shows the natural emergence of such behavior by learning between multiple Q-learning agents. This experiment successfully demonstrates the emergence of actions that are similar to the communication of humans.

Chapter 6

Biased random sequence generation for making players believe it is unbiased

This chapter is written and modified based on the article ”Biased Random Sequence Generation for Making Common Player Believe it Unbiased” which is published in the international conference IEEE- Game Entertainment & Media GEM2014 in Toronto Canada in 2014. Contents from section 6.1 to section 6.3.3 are collaboration work with Hisamitsu Nomura, an elder colleague. Contents from section 6.3.4 are my completely original work.

6.1 Background

Pseudorandom is a well-known problem that aims to create random sequences with regards to “distribution quality”, “length of sequence period”, and “speed of generation” [72]. The pseudorandom generators are used in many processes such as stochastic optimization, Monte-Carlo method [73], and reinforcement learning. In recent years Mersenne Twister (MT) has interested researchers and developers due to its ability to produce good quality random sequences. The Mersenne twister has demonstrated outstanding performance by able to produce a uniformly distributed sequence in 623 dimensions with an enormous size of period of up to 219937 . It has been dramatically improved in terms of mathematical requirements mentioned above when compared to the old generation algorithms such as linear congruential [74]. It can be said that this algorithm is very close to theoretical randomness and sufficient for the usual purposes.

In computer games, pseudorandom algorithms have been used to generate randomness such as in the rolling virtual dice in board games and Trumpshuffling and drawing in card games (fig.6.1 (1)).For example, in the popular game series the Pokemon, a pseudorandom generator (PRG) is used in many processes such as in the "monster egg hatch" [1]. Another example can be found in the game "World of Warcraft". The system called "roll" which controls the priority of looting items in the game uses a pseudorandom algorithm [2]. In these two games, the quality of randomness had a significant effect on the players' satisfaction. High quality, accurate random sequences are required in order to make the game unpredictable and thus entertaining to players. For example, in card games like poker and blackjack, drawing cards should be based on good randomness in order to be fair to the players. If the drawn cards often appear to be too bad or frequently appear to be good, the entertainment quality of the game will suffer.

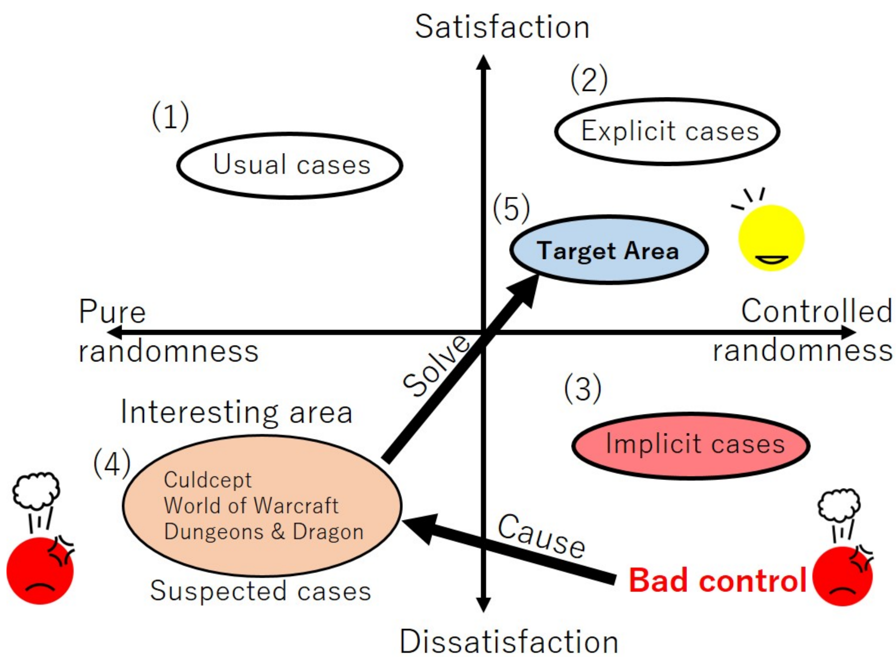


Figure 6.1: Relationship between the uses of random and satisfaction of player.

However, In these games, all the processes are entirely controlled by the program. Thus it is easy to modify or control the randomness (or to cheat players). In fact, many old generation games used this advantage to eases the weakness of computer players. Because of resources limitation, it was hard to develop smart computer players. The arbitrary modification of random sequences such

[1]http://bulbapedia.bulbagarden.net/wiki/Pseudorandom number generation in Pokemon, Accessed May. 12, 2019

[2]http://us.battle.net/wow/en/, Accessed May 12, 2019

as this has often been noticed (fig.6.1 (2)). Such modification affected to the feedback of the game directly, and it might affect to doubt of other games.

In some cases, players accepted the modification of randomness (fig.6.1 (3)). For example, in some computer board games (e.g., “Momotaro Dentetsu” [3]), randomness is adjusted by the system, due to the story-line or characteristic of a character in the games. Such as the character’s ability to force the dice to only appear as “5” or “6”. In such cases, players could accept the modified randomness because it is explicitly shown the intention.

In earlier cases, the satisfaction of players conforms to the modification of randomness. However, there is some exception (fig.6.1 (4)). “Culdcept” [4], a Japanese computer board game series are very popular even in Europe. However, players complained that the random sequences in the game appeared to be modified in order to disadvantage players. Even though the developer declared that there was no modification in this game, many players still were not convinced [5]. Another game, “Dungeons & Dragons online” [6] also faced to the same credibility problem. Many users complained that the dice roll generally gave low values.

World of Warcraft has also faced the same problem. The roll system which random the priority number for looting items in the game, seems to be fair for every player. However, many players are still unsatisfied. In such cases, the randomness generated by a good algorithm might be the best way to deal with problems. However, the algorithm which gives a good quality of randomness does not necessarily guarantee players’ satisfaction. Even though the developers used the new generation algorithm such as MT, complaints about the randomness persisted.

The judgment of randomness (or non-randomness)depends on players’ perception. Normally, human often predicts some events which are random, by referring to previous patterns and trends. Such as, it is very natural that we predict the weather tomorrow by referring yesterday and today’s weather. Such behaviors strongly affect their randomness believability, and they might misperceive real probability and make the judgment in randomness deviated too. These misperceptions which deviate judgments are called cognitive biases [75].

Cognitive bias is a phenomenon in which the context and framing of knowledge or information influence individuals’ judgment and decision-making [76]. Human’s information processing capabilities are limited. Therefore, it accepts a certain amount of error, performs quick perception and judgment to reduce the amount of information. This method is called heuristics (a simple way of thinking unconsciously) and it shows the result of a decision in a short time but not necessarily correct. And here is where a cognitive bias has appeared [77]. Cognitive bias is very diverse. Here, some famous cognitive biases are briefly

[3] “Momotaro Dentetsu” is a famous Japanese computer board game. The game is based on “Sugoroku” Japanese classical board game that similar to “Monopoly”

[4] https://www.culdcept.com/, Accessed May 12, 2019

[5] https://www.omiyasoft.com/jnote07.html, Accessed May. 12, 2019

[6] https://www.ddo.com/forums/showthread.php/437332-Is-the-Daily-Dice-really-random, Accessed May. 12, 2019

explained:

- Confirmation bias: Human favor information which confirm their beliefs. This kind of preference is called confirmation bias. For example, if there is a person who believes that left-handed people are more creative than right-handed people. When this person meets the both left-handed and creative person, he/she will quickly notice and give importance to support his/her believes [78]. In the cases of the dice in a computer game, if a player once makes a hypothesis that the dice has been manipulated, the player will notice sub-sequence, which convenient for deepened further conviction.

- Bandwagon Effect: Humans tend to be influenced by other people's actions and opinions. This is called " synchronization " in psychology. In particular, Japanese people have a strong sense of synchrony. They simply buy products by being introduced to television and magazines as "popular" instead of choosing according to their preference [79]. In some cases, the same phenomenon may occur in game reviews.

- The law of small number: The law of small numbers is a cognitive bias where people show a tendency to believe that even a small number of random samples should represent the characteristics of the population [80]. For example, normally the frequency of numbers in a long random number sequence is uniform, but it is easy to think that any extracted sub-sequences should hold the same property. However, in practice, in a narrow area of sequence often show some biases of an amount of number.

- Gambler's Fallacy: We tend to believe that previous events influence future outcomes and put a weight on them. The classic sample is the coin tossing. After flipping head four consecutive times, it is normal for people to predict that the next toss will be tails [81]. The probability that the head continues for 4 times is $1/2 \times\ 1/2 \times\ 1/2 \times\ 1/2 = 1/16$, and the probability that the head will appear again is $1/32$, about 3 times out of 100 times of probability. However, no matter how many times the head appears, the probability is still 50 by 50. This kind of fallacy is called Gambler's Fallacy.

- Clustering illusion: When certain events that should occur at random occur together, people often create the illusion that they are not random. For example, many people will be surprised if a coin-tossing show heads or tails four times in a row. However, if you throw 20 times consecutively, the probability that heads or tails will appear 4 times is 50% [82].

Previous literature also showed that the randomness from a human perspective is different from the mathematical definition. In 1960, Bakan experimented with 70 undergraduate students who were requested to simulate 300 coin tosses. The experiment proposed to analyze the misunderstanding of the frequency of the change from H to T or T to H (Head / Tail). As a result, the subject showed more than 176 times by average, in the other hand, the theoretical value is only

150 times [83]. Schilling had shown the longest possible consecutive appear of heads or tails such as HTTTTHT or TTHHHHHT. Theoretically, for 200 tosses, the consecutively run of head or tail might be possible to be 7-10 times. However, in human randomness perception, the consecutive runs are less than five times that is much balancing of frequencies. It shows the difference between personal beliefs and actual randomness [84] [85].

Putting into conduction the hypothesis that the ideal randomness in mathematics might not look natural to common players. This research aims to decrease the dissatisfaction of player due to the pseudo-randomness in games by imitating the characteristic of randomness in players' belief by incorporating with a pseudo random algorithm (fig.6.1 (5)).

In this research, common players' perceptions, misunderstandings, beliefs, and bias of randomness were analyzed. As a result of analyzing, the difference between players' believed randomness and theoretical randomness were demonstrated.

Therefore, a particular method to generate natural random sequences from the viewpoint of common players by modifying a pseudorandom sequence is presented. In order to decrease players' negative feeling, pseudorandom sequences were modified in order to make them similar to the players' randomness perception. In other words, the method imitated players' cognitive biases. For confirmation of the capability of the method, the evaluation was done in the game of "Sugoroku". As a conclusion, it is shown that natural random sequences can decrease the dissatisfaction of player in practical uses.

6.2 Approach and goal

We propose the approach in 4 steps. (1) Analysis of players' cognitive biases, (2) generation of believable random sequences by imitating players' biases, (3) evaluation of generated random sequences in the game of Sugoroku, (4) individual randomness generation. The process was shown in fig 6.2) and explained the following.

1. Preliminary questionnaires: trends of misunderstanding, bias, and belief in random numbers and its' distribution in common players were surveyed by preliminary questionnaires.

2. Generation of natural random sequences: a method for generating intended sequences was proposed. According to the trend of cognitive bias retrieved in step (1), sequences were optimized to make usual players "feel" them unbiased. This method was evaluated through a simple experiment using human subjects.

3. Sugoroku: a straightforward board game using dice is employed. In the game, not only the sequence of dice numbers, but also the sequence of "whether being trapped in undesirable cells" must be controlled to decrease complaints. For example, assuming the possibility of being trapped

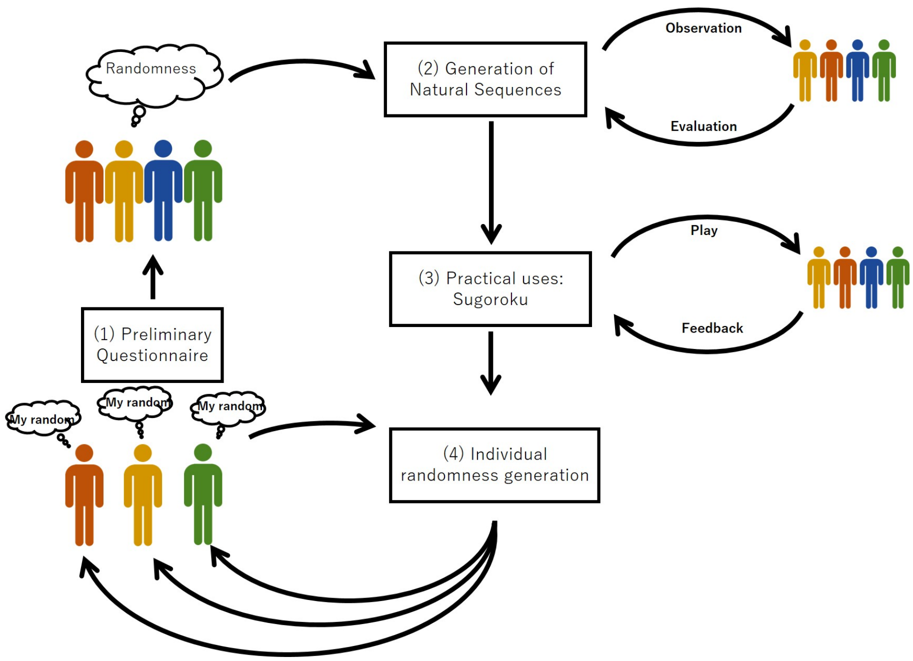


Figure 6.2: Approach for generate natural randomness

is $\frac{1}{3}$, a player will get angry if trapped **three times** in a row, or a player will be bored if not trapped **six times** in a row. A method for controlling both sequences is proposed.

4. Individual randomness: the previous steps are the generation of natural sequences for average players. However, our method based on common players' biases, it might be unnatural for experienced players or players who have mathematical knowledge. The online approach can be proposed in the near future to solve this problem, by analyzing players' trend of biases online and then generating random sequences for each player.

The first three approaches were done and are presented in this dissertation.

6.3 Experiment and analysis

We approach our purpose step by step as we state in the previous section. This section we present the process in each step.

6.3.1 Preliminary Questionnaires

Humans have different biases of randomness. One of our goals is to generate believable random sequences for common players. Preliminary questionnaires were conducted in order to identify common biases, misunderstanding, and belief in the randomness of players. At first, the human biases were investigated to create the criteria for random sequences. Hundred digit length sequences of numbers in the range [1-6] were simulated by subject players (not using real dice) as shown in fig 6.3.

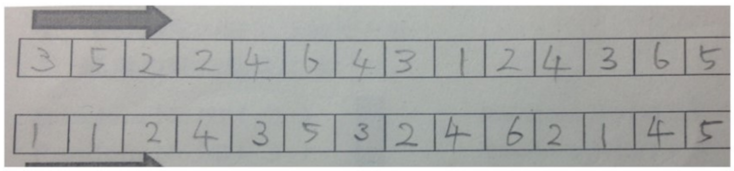


Figure 6.3: Sample of the manual generated sequence

The first 40 digits of 2 sample sequences simulated by subject players are:

- 4525143326144641355542665654121422351611
- 1523645326413253412156362436152342615243

The first sequence frequently showed consecutive parts (such as 33, 44, 555) but no such part was found in the second sequence that is far from theoretical randomness. However, it was assumed that generating random sequences without repeating might look "natural" in this subject's view. Though this is an extreme case, it has been found that almost all subjects tend to avoid such repetitions.

Random sequences' bias feature

As the criteria for analyzing sequences, fifteen characteristic pattern features were defined. These features are manually selected. Some of the features were selected by referring to Cognitive bias references, some of them were selected from our experiences. The features F1-F5 present the distribution of the number in each range of a sequence. The features F6-F15 present pattern of consecutive numbers in the previous 2-7 digits. The maximum range is set as seven digits because a human short-term-memory can be memorized up to seven items. Regarding these features, theoretical random sequences and subjects' artificial random sequences were analyzed and compared. Fifteen features and theoretical sequences' values (for a hundred digits numerical sequences) are captured as follows.

F1	χ^2 distribution of sequence	(5.0)
F2-F5	χ^2 distribution of about quarter length subsequences; 1st-30th, 24th-53rd, 48th-77th,70th-99th	(5.0)
F6	The frequency of the flips from even to odd (or vice versa).	(49.5)
F7	The frequency of the same numbers appearing consecutively **two times**.	(16.5)
F8	The frequency of the same numbers appearing consecutively **three times**.	(2.7)
F9	The frequency of the same numbers appearing consecutively **four times**.	(0.45)
F10	XXYY, XYXY, XYYX (Two pairs); the frequency of four consecutive numbers which comprise two kinds of numbers.	(6.7)
F11	XXYYY (Full house); The frequency of a pair of the same numbers and a three of the same numbers occur consecutively	(1.5)
F12	XYXX, The frequency of subsequences in which 3 of 4 digits are the same numbers.	(4.5)
F13	XYXZX, The frequency of subsequences in which 3 of 5 digits are the same numbers.	(5.6)
F14	XYXZXX, The frequency of subsequences in which 4 of 6 digits are the same numbers.	(1.8)
F15	XXYXZWX, The frequency of subsequences in which 4 of 7 digits are the same numbers.	(2.5)
	(W, X, Y, Z represent the number which appear in subsequence)	

Table 6.1: Bias feature of random sequences

Natural sequences' features

The questionnaires obtained sequences from sixteen subject players were analyzed to identify common cognitive biases about the naturalness of random sequences. The summary of the analysis is shown below in Table 6.2 (the result of F3-F5 resemble F2).

F1 and F2 show the distribution of numbers in the sequence. The average value of artificial sequences was a bit lower than the theoretical values. It is interpreted that players prefer evenly appears of each of the digits.

F6 is revealed that the change from even to odd or vice versa was often shown. These changes sound to be "Random" in the subjects' view. This result confirmed the experiment by Bakan (1960) [83].

F7 to F9 represents the consecutive occurrence of the same number. The practical values are significantly lower than the theoretical values. After the generation experiment, subjects were requested to answer the following question (How many times should 3 consecutive same digits appear?). More than half of the subjects' answers are close to theoretical values. However, players tend to feel that scatter numbers are random, thus the appearance of the same number

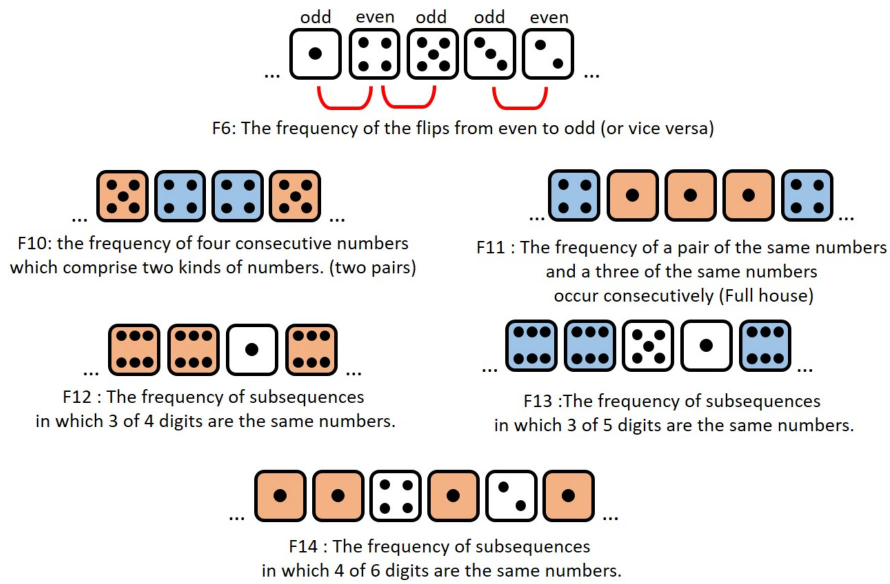


Figure 6.4: Example of Feature F6, F10-F14

Feature	Theoretical value	Average value	Average of the eighth highest value	Average of the eighth lowest value
F1	5.0	2.4	3.3	1.5
F2(-F5)	5.0	2.2	3.0	1.3
F6	49.5	55.6	62.0	49.3
F7	16.5	10.9	15.9	6.0
F8	2.7	0.8	1.5	0.1
F9	0.45	0.06	0.1	0.0
F10	6.7	3.3	5.3	1.3
F11	1.5	0.4	0.8	0.0
F12	4.5	1.4	2.8	0.0
F13	5.6	1.8	2.8	0.9
F14	1.8	0.3	0.5	0.0
F15	2.5	0.5	1.0	0.0

Table 6.2: Value of feature from subject players' sequences

as previous was avoided.

F10 to F15 represents the frequency of pattern appearance. It is significantly different from the theoretical values. The same trend as features of F7 to F9 appeared. The average of the top 8 values is far from theoretical values. It is shown that subject try to avoid the pattern with or without intention.

Regarding the random sequences simulated by subject players compared to the theoretical values, the significant differences are necessary for the generation of "natural" random sequences. However, the difference of personal belief which shown in the difference between the average of the top 8 values and average of the bottom 8 values was significant too.

6.3.2 Natural Random Number Generation

According to the result showing in section 6.3.1 and the hypothesis, an artificial sequence might look natural when it shows the same or similar features as generated by the subject. A method for generating random sequences that make common players feel "natural" or "unbiased" was proposed. The symbols are defined as follow.

- s : pseudorandom sequences
- $f_i(s)$: statistic value of sequence s of i-th feature
- $[\alpha_i, \beta_i]$: favorable range of feature f_i
- err_i : the amount of deviation from range when

$$err_i(x) = \begin{cases} \alpha_i - x \text{if } x < \alpha_i \\ x - \beta_i \text{if } \beta_i < x \\ 0 \text{if } \alpha_i < x < \beta_i \end{cases} \tag{6.1}$$

- γ_i : weight of deviation

For sequence s, the following equation was defined.

$$err(s) = \sum\nolimits_i (\gamma_i err_i(f_i(s))) \tag{6.2}$$

The values of $err(s)$ is minimized by using optimization algorithm, and $err(s) = 0$ only when all feature values are in the favorable ranges.

Example of $err(s)$

The length of random sequences in the experiment is fixed to 50 digits. Regarding the result of the first experiment, α, β, and γ were defined and shown in table 6.3.

Feature	Theoretical value	Lower bound α	Upper bound β	Weight γ
F1	5.0	2	5	3.0
F2(-F5)	5.0	2	5	3.0
F6	24.7	27	30	1.0
F7	8.3	5	8	1.0
F8	1.3	0	1	3.0
F9	0.2	0	0	10.0
F10	3.4	1	3	4.0
F11	0.9	0	0	4.0
F12	2.2	0	1	4.0
F13	2.8	1	2	4.0
F14	0.9	0	0	4.0
F15	1.2	0	0	4.0

Table 6.3: Evaluation Criteria

300 standard pseudorandom sequences were prepared by using the pseudorandom generator (i.e. Mersenne twister). The best sequence gave $err()$ as 2.7 and the worst gave $err()$ as 239:0. The average $err()$ is 52:1. The best sequence and the worst sequence are shown as follows.

- The best Sequence($err(s) = 2.7$) :
  52166232635543316563431216615323356642653154342315
- The worst sequence($err(s) = 239.0$) :
  31443255533554554656246445591543564454542566656544

According to the table 6.3, the practical values of all features except F6, were lower than theoretical values, such as F11, "Full house" was prohibited even though the theoretical average is 0.9 (not extraordinary). Thus the best sequences in which values of feature F7 to F15 are very small, supposed to look natural in players ‘ view. On the other hand, the worst sequences held many groups of consecutive same numbers. That the pseudorandom generator might be able to give such "natural" and "unnatural" output is demonstrated.

Optimization algorithm

We employed the local search algorithm in order to find the sequences with a minimum of $err(s)$. the summary of the algorithm is shown following.

1. Initialize 50 digits sequences by the standard pseudorandom algorithm. (in this case, MT was adopted)
2. s_0 is generated by changing a randomly selected digit of s

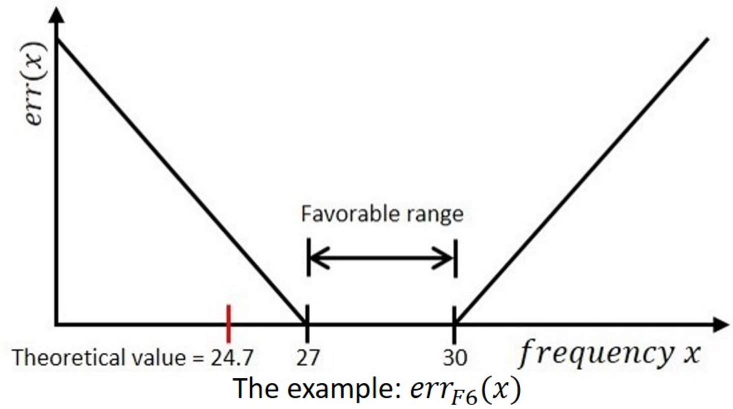


Figure 6.5: The example: errF6(x) :The frequency of the flip from even to odd or vice-versa.

3. If $err(s_0) < err(s)$ then assign $s_0 \rightarrow s$.

4. Repeat (2) and (3) until $err(s)$ is becoming 0 or process over 1000 times.

In preliminary design, simulated annealing and local search were proposed. However, the setting at this time was the local search which could be able to give the err(s) = 0 as well. Also, the time spent in optimization was around 0.02 seconds, exhibiting the acceptable performance with a standard PC and good enough to be used in the usual video games. The simulated annealing could be necessary if the number of features or limit upper or lower bound was increased.

6.3.3 Naturalness evaluation

In order to confirm that the obtained optimized pseudorandom number sequences which err(s) = 0, looks "natural" for common players, the subject experiment was performed.

Evaluation preparation

First, sets of sequence, three types of pseudorandom sequences were prepared as follows.

1. Standard random sequences [Standard]: sequences generated by a pseudorandom algorithm without modification, Random build in C♯.net.

2. Low-Rank Random Sequences [Low rank]: the value of err() sorted 60 bad sequences from the last 20% of 300 pseudorandom sequences.

3. Optimized Random Number [Optimized]: Optimized pseudorandom sequences with err(s) = 0.

48 sequences were prepared for each [Standard] [Low-rank] and [Optimized].

Evaluation method

Sixteen subjects were employed to evaluate the naturalness of each [Standard],[Low-rank], and [Optimized] Sequences. Each subject received nine experimental programs (based on the Microsoft Windows Operating system, see fig. 6.6) which display random sequences. One new number would appear every second, and the last six numbers were displayed. Subjects ran an experimental program twice per program (the same sequences) and gave a score for the program. The definitions of the score are:

1. This random sequence has a Bias.
2. This random sequence might have a Bias.
3. Do not know.
4. This random sequence might be a natural random sequence.
5. This random sequence is a natural random sequence.

The set of nine programs contained three of each [Standard], [Low-rank], [Optimized] sequences which were prepared in section 6.3.3 , then $3 \times 16 = 48$ sequences were generated per each type. The order of 9 programs was shuffled, and the group being shown was unknown to subjects.

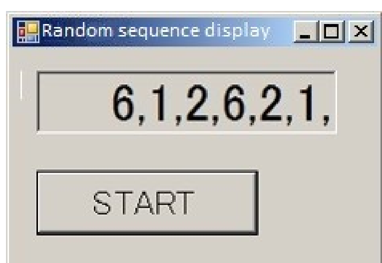


Figure 6.6: Screen capture of Random sequences display program.

Evaluated result

The result of the experiment, classified by groups of sequences, were summarized in Table 6.4.

[Low rank] was the worst 20% of 300 sequences ordered by the value of err(). According to the values of err(), these sequences are far different from randomness in players' perception. Thus the feelings of players "this sequence is biased!" was ordinary. It is outstanding shown in the experiment by the median of the score is 2 and Standard deviation is 1.01.

[Optimized] got the best evaluation of the three groups. Over 50% of the sequences got 4 to 5 scores in the experiment (SD = 1.29, Median = 4). It was concluded that sequences in this group look natural (look like real random sequences) from the players' point of view. In other words, our ***biased*** sequences successfully made the players believe that the sequences are ***unbiased***.

[Standard] got a better result than [Low rank] but lower than [Optimized]. Because [Standard] provided sequences which have err() from 2.7 to 238.0. This phenomenon might be interpreted as "some sequences look natural whereas some sequences did not."

Three groups of sequences showed statistically significant differences in statistical test. By using Kruskal-Wallis Test, it is shown significantly difference at p ¡ 0.5 ($p = 0.0001$.) And each pair also show differences in Mann-Whitney U Test of each pair as we showed in Table 6.5.

The experiment exhibited that our designed err() can be used to evaluate and simulate the naturalness of random sequences from the viewpoint of common players. However, there are some questions such as "are all fifteen feature necessary?" or "is there some necessary features missed?", which need further research and study. The work until this section was under collaborating with Nomura Hisamitsu.

Group	Evaluated score					
	1	2	3	4	5	Average (SD/Median)
[Standard]	11	19	2	11	5	2.58 (1.35, 2)
[Low-rank]	17	23	2	5	1	1.96 (1.01, 2)
[Optimized]	7	9	4	23	5	3.21 (1.29, 4)

Table 6.4: The evaluation result

Pair	z-score	p-value
{[}Standard{]}:{[}Low-rank{]}	2.136	0.016
{[}Standard{]}:{[}Optimized{]}	−2.085	0.019
{[}Low-rank{]}:{[}Optimized{]}	−4.312	0.000

Table 6.5: Mann-Whitney U Test between each pair

6.3.4 Utility test in Sugoroku

The naturalness of random sequences was directly evaluated via display programs in the previous section. However, in practical uses the naturalness of sequences might be affected by other factors. Here, we assessed the utility of the modified method by applied into Japanese tradition simple dice game called

Sugoroku and investigated player's satisfaction about dice. this section was done originally.

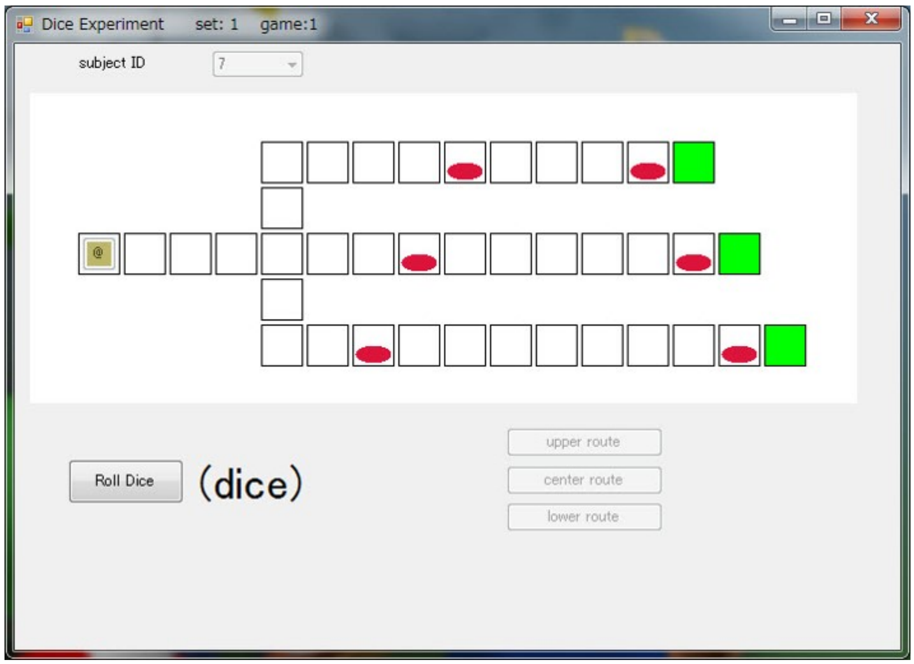


Figure 6.7: Screen capture of simple Sugoroku program.

Sugoroku program

Fig. 6.7 shows a screen capture of the Sugoroku program. The experiment procedures are shown following:

1. The token was placed at the start point (left-most cell).
2. "roll the dice" button was pushed by the player, and dice digit was selected and shown, from 1 to 6 by using a specific algorithm.
3. If there was no branch, the token was moved automatically by the showing digit cells.
4. If there are branches, a player can select one of upper/right/lower routes.
5. If the token stops at a red cell (Trap) then the player lost. If the token reached green seal on the right side, the player wins.
6. 10 games per set and four sets will be done. After each set (10 games) is played, players will be announced that the random sequence is changed.

Each player will be asked to play in totally 40 games. And the four following algorithms control the digit of dices.

1. Using the [Optimized] sequence generated in section 6.3.3 .
2. Using the [Optimized] sequence and "Trap Control" algorithm. (will be explained in next subsection.)
3. Using the [Low Rank] sequence.
4. Using the [Low Rank] sequence and "Trap Control" algorithm.

The order of sets is randomly shuffled, and it is hidden to players. Seventeen subject players were employed in the experiment, and 1000 JPY were rewarded to each subject player to incent them. Each player must decide the path while doing trial and error by themselves because their optimal policy is distinct.

Trap control

In the games of Sugoroku, the belief of randomness might have deviated while playing. The risen of numbers of being traped induce a negative feeling to players. For example, if a player is trapped four times in a row, the player will suspect that the dice is intentionally controlled, even if the dice sequence itself seems not to be biased. Then, a method to control the frequency and continuity of being trapped, with keeping the naturalness of the sequence would be proposed.

To control the number of being trapped, "trap sequences" were introduced. These sequences are randomly generated for each player. The following sequence is an example of trap sequences.

- 2,2,2,2,2,1,2,2,2,1,2,2,2,2,1,2,1,2,2,2,2,2,1,2

In the sequences "2" means "by the next dice roll the player should not be trapped" and "1" means "by the next dice roll the player should be trapped". Trap sequence is optimized by the optimize algorithm shown in section 6.3.2. The appearance rate of "1" is about 1/6, and the consecutive part is minimized.

The trap control will be activated when a trap exists in 6 cells ahead, and the player is in the situation that unable to avoid the trap; after a token has passing crosspoint (fig 6.8). When the trap control is activated, the next trap control number is picked from a trap sequence. Note that, c is the distance between a token and the next trap which is in the range $1 \leq c \leq 6$, t is a digit retrieved from a trap sequence, d_0 is the last displayed digit of the dice, d_1 is the next scheduled digit in the prepared dice sequence, and d_2 is a digit the next to d_1.

When $t = 2$ and $d_1 \neq c$, or $t = 1$ and $d_1 = c$, the dice schedule is already agreed with the trap control, in this cases d_1 shows just as it is.

When $t = 2$, $d_1 = c$ and $d_0 / neq d_2$, d_2 is display instead of d_1, to make the player untrapped according to the trap sequence. In fact, d_1 and d_2 are

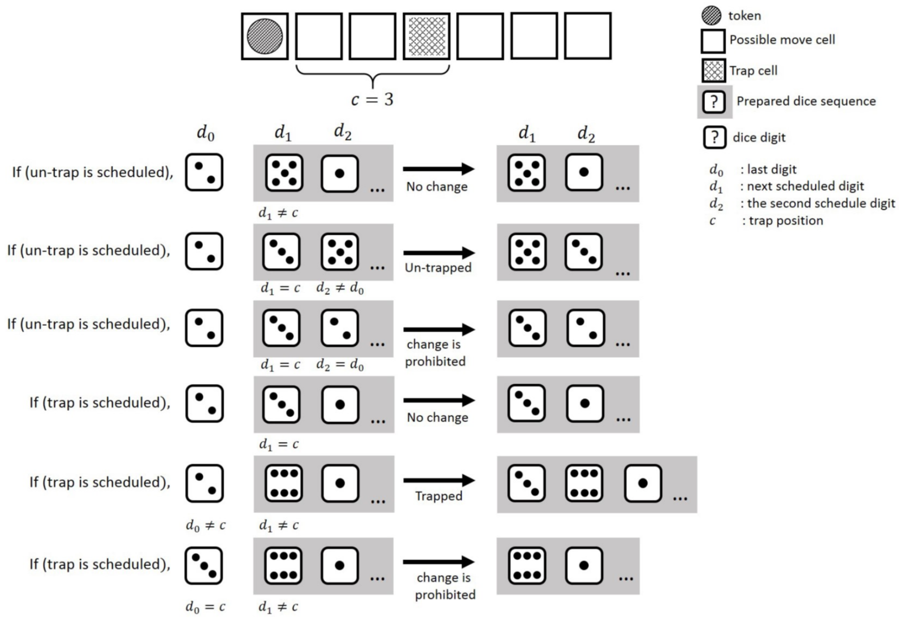


Figure 6.8: Example of trap control method.

swapped, then the next digit is scheduled to d_1 instead of d_2. The condition $d_0 \neq d2$ is introduced to avoid the continuation of the same digit.

Finally, when $t = 1$, $d_1 \neq c$ and $d_0 \neq c$, c is shown instead of d_1, to make the player trapped according to the trap sequence. Here too, the condition $d0 \neq c$ is also introduced to avoid the continuation of the same digit too.

A trap control is activated for each chance to be trapped, not for each game. If the trap schedules are set up in each game, the result will be independent of the route selected by the player. Even if a player selects an optimal route or a bad route, the result will still be the same. This feature is not suitable and should be avoided.

Evaluation of naturalness in practical uses

Similar to the method in section 6.3.3, the naturalness of dice sequences that were used in each set of games were evaluated by human subjects. The averaged score of each set are shown in Table 6.6. [Optimized] sequences seemed significantly more natural than [Low Rank] sequences which are the same trend was shown as a result in table 6.4, though the difference was smaller. One reason might be that the last six digits were displayed in the previous experiment.

One more important result is the naturalness not decreased by using Trap Control when [Optimized] sequence was used. As the goal was to propose a method to control the frequency and continuity of being trapped, and keeping the naturalness, the satisfaction of the following condition was confirmed.

Group	Average score
[Optimized]	3.12
[Optimized] + Trap Control	3.24
[Low-rank]	2.71
[Low-rank] + Trap Control	2.41

Table 6.6: The naturalness evaluation results in Sugoroku

frequency and dissatisfaction

The number of being trapped in each set (10 games) was also recorded. Considering the goal, the number is neither too big nor too small.

In the case of using the [Optimized] sequence without trap control, four players (of 17) were trapped only twice, and two players were caught as often as eight times. The average number was 4.23 whereas its standard deviation was 1.97.

In the case of using the [Optimized] sequence with trap control, one player was trapped only twice, and no player was trapped over six times. The average number was 4.53 that bigger than the previous case, but its standard deviation was 1.14 that fairly smaller. It was concluded that trap control is sufficient to avoid the "too lucky" or "too unlucky" cases, and keeping the naturalness.

After each set of 10 games question was asked to the subjects, "Do you think that dice was controlled to force you to be trapped?". Subjects could answer 1-5, 1 is "Strongly yes" and 5 is "Strongly no". In the case of using the [Optimized] sequence, the number of "strongly yes" is 4, and in the case of that with trap control only 1. This result confirmed that this method is able to reduce the deviation of random numbers' naturalness and dissatisfaction.

6.4 conclusion and future works

In this research, a method for generating a biased random sequence that making players feel unbiased, according to the common cognitive biases of players, was shown. Fifteen features were employed to measure the "randomness from the viewpoint of common players", and sequences are optimized to satisfy a given condition. Generated sequences were evaluated and compared to standard random sequences, and it was proved that our method is effective.

Moreover a trap control algorithm has been proposed and employed to decrease dissatisfaction which can occur while playing dice games like monopoly. The experiments have been conducted in a short period, we did not confirm the ability to reduce dissatisfaction in long usage. However, it should be clearly noted that the method is not decreasing the number of being trapped for better evaluation, but control the numbers to an adequate range, and control the patterns to be trapped. So, we guess that dissatisfaction such as "I am too lucky, boring" will not be so frequent.

Also, please note that this work will be the first attempt to make such sequences. This is the reason why we compared our proposed method only to usual random sequences.

However, this method might not be effective for some players such as scientists or expert game players, because the optimized sequence may be far from the true randomness. To handle this problem, an adaptation for individual cognitive biases needs to be done. This approach will be proposed in the near future.

This time we only conducted a utility experiment on only the Sugoroku, not generally describe in other domains. However, "an event occurs with a certain probability" will frequently happen inside a game. Thus we believe that "the idea of controlling the good and bad events to not occur much" is also able to be applied to other game domains.

Chapter 7

Conclusion

“Can machines think” Since this great question has been raised by Alan Turing [68] for almost seventy years, “Human-likeness” is still a challenging problem for researchers in the field of artificial intelligence. To date, “What is human-likeness.” is difficult to be clearly defined.

Some human-likeness is necessary for entertainment. But “Does human-likeness always lead to entertainment?”. The answer to this question is “ Not always”. There are many cases to support this answer. For example, in Chapter 5 we discussed about “provocation”. Such behaviors can be allowed if they are done among friends, but will not improve satisfaction of human players if they are done by computer players. Thus the selection of human-likeness should be done carefully.

A large number of current research related to this field interprets the human-likeness in their way in which there is almost no common point in each of them. The literature review is one of the tough tasks for the researcher in this field. Thus, for further research in this field, we collected, summarized, systematically clusterized as we show in chapter 2. We publish the work in “The 36th meeting of the Game Informatics Research Group SIG-GI 2018” by the “Game Informatics research group which is a part of the Information processing society of Japan” [12]. We hope that this work will inspire and encourage especially young Japanese researcher to conduct the research and make advancement in this field in the future.

Also, we presented three approaches in human-likeness which are significant and unique behavior of human-being. First, we showed the hand-coded method for representing three types of behavior which are “play carefully,” “give a priority on sub-reward,” “pass the stage as fast as possible” and the switching between this behavior to represent human-likeness. It is successfully performed in some individual models, but some are not. By rule-based switching between model, the AI shows improvement in the evaluation of human-likeness over the unsuccessful model but is still far from the assessment of the human player. Thus we believe there is a possibility of improvement in further research. This research was published in 17th Conference on Intelligent Games and Simulation [7].

Next, we presented research on behavior that is not directly related to the defined goal of the game. The work shows a novelty point of view in human-likeness which is often found in human gameplay. We clustered various cases of study from different games. Besides, we successfully reproduce the action which aims to communicate to the alliance player, without using and supervised. In other words, game-AIs try to communicate with each other by themselves without any particular connection way. Such behavior is very significant and directly related to the entertainment of the games. By introducing some of these behaviors, the interaction between game-AI and human-player might be more efficient and able to give human-player a feeling of being a part of a game which related to the entertainment of play directly. This research was published in Entertainment Computing – ICEC 2017 [86].

Finally, we proposed the method of making human-like cognitive bias on randomness which we claim that is one of the characteristics of humans. We statistically analyzed a random number by an understanding of human-player and tried to reproduce random sequence whose hold a similar statistical feature to human belief. At the result, we successfully produced random sequences which humans believe it is random without biased. We assess the naturalness of sequences directly and practical use in a dice game which showed a similar successful conclusion. We believe that applying the method might able to reduce the dissatisfaction that occurs by the cognitive biased of a player and help players be able to enjoy the games more. This research was published at 2014 IEEE Games Media Entertainment(GEM2014) [87].

The focused three viewpoints of human-likeness we showed in this dissertation are only one portion of human-likeness for game AI. However, we believe that these three points are unique to human players and important for making human-like game AI. And this study showed the advancement of human-likeness especially for entertainment purpose. In the other domain such as a serious game where the purpose of the game is not entertaining player but to support learning something from the game, there are many discussions in order to apply in such a domain. But, there is some connection between the serious game and entertainment game such as the fun of the game might be rising the performance of education or learning. Serious games with a human-like partner or opponents might be necessary. This topic should be discussed in further research.

Appendix

We surveyed many academic papers related to human likeness, and checked their properties, in other words the purposes, the point of view and the employed methods. They are listed in the following long table.

No.	Paper name	Purpose					Point of view	AI method					
		Ent	PCG	ADF	EDU	Etc.		SL	RL	EC	Heu	Sea	Etc
1	Sensitivity to the proportions of faces that vary in human likeness [46]	○					Physical limitations						
2	A Study on Human like Characteristic in Real Time Strategy Games [42]	○					Physical limitations				○		
3	ビデオゲームエージェントの自律的行動獲得と観測情報の信頼性に着目した獲得行動パターンの分析 [44]	○					Physical limitations						
4	次の一手課題に基づく囲碁と将棋の特徴比較 [52]				○		Physical limitations						
5	Evaluating Human-like Behaviors of Video-Game Agents Autonomously Acquired with Biological Constraints [6]	○					Physical limitations		○		○	○	
6	進化計算と UCT による Mario を人間らしくプレイする AI [41]	○					Physical limitations			○			
7	Learning believable game agents using sensor noise and action histogram [40]	○					Physical limitations			○			
8	Assessing Human Likeness by Eye Contact in an Android Testbed [57]					○	Physical limitations						
9	弾幕の認識に人間の視覚特徴を取り入れたシューティングゲーム AI の研究 [64]					○	Physical limitations				○	○	
10	Robust player imitating using multiobjective evolution [39]	○					Behavior imitation			○			
11	将棋における棋風を感じさせる AI の試作 [47]	○					Behavior imitation						

continued ...

No.	Paper name	Purpose					Point of view	AI method					
		Ent	PCG	ADF	EDU	Etc.		SL	RL	EC	Heu	Sea	Etc
12	囲碁 AI 作成に向けた棋風を形成する要素に関する統計的分析 [88]	○					Behavior imitation						
13	UTˆ2: Human-like behavior via Neuroevolution of Combat Behavior and Replay of Human Traces [30]	○					Behavior imitation	○		○			
14	Imitating human playing style in Super Mario Bros [33]	○					Behavior imitation	○	○	○	○		
15	Believable bot Navigation via Playback of Human Traces [34]	○					Behavior imitation	○					
16	Towards Imitation of Human Driving Style in Car Racing Games [31]	○					Behavior imitation	○					
17	手の流れを考慮して自然な手を選ぶ将棋 AI の試作 [51]	○			○		Behavior imitation	○			○		
18	機械学習を用いた将棋における棋風の学習の研究 [22]				○		Behavior imitation	○				○	
19	将棋における棋風を学習するための棋譜分析の取り組み [89]	▲					Behavior imitation	○				○	
20	Comparing Different Control Approaches to Implement a Human-like Virtual Player in the Mirror Game [90]					○	Behavior imitation						▲
21	Creating Human-Like Non-Player Game Characters using A Memetic Multi-Agent System [48]	▲	▲			▲	Behavior imitation	○	○				

continued . . .

No.	Paper name	Purpose					Point of view	AI method					
		Ent	PCG	ADF	EDU	Etc.		SL	RL	EC	Heu	Sea	Etc
22	Developing Game AI Agent Behaving Like Human by Mixing Reinforcement Learning and Supervised Learning [37]	○					Behavior imitation	○	○				
23	Making Racing Fun Through Player Modeling and Track Evolution [19]		○				Preference	○					
24	Towards Automatic Personalized Content Generation for Platform Games [20]		○										○
25	Experience-Driven Procedural Content Generarion [49]		○										▲
26	Emotion Assessment From Physiological Signals for Adaptation of Game Difficulty [21]		○	○									▲
27	Experience-Driven Procedural Content Generarion(Extended Abstract) [50]		○										▲
28	A Turing Test for Computer Game Bots [25]					○	Human-likeness Assessment method						
29	The Challenge of Believability in Video Games: Definitions, Agents Models and Imitation Learning [53]					▲	Human-likeness Assessment method						
30	Assessing Believability [11]					○	Human-likeness Assessment method						
31	エンタテイメント系システムの主観評価実験におけるユーザ統制及び実験手法の検討 [58]					○	Human-likeness Assessment method						

continued . . .

No.	Paper name	Purpose					Point of view	AI method					
		Ent	PCG	ADF	EDU	Etc.		SL	RL	EC	Heu	Sea	Etc
32	A Conceptually Different Approach to the Empirical Test of Alan Turing [26]					○	Human-likeness Assessment method						
33	A Neuronal Global Workspace for Human-like Control of a Computer Game Character [35]	△					Human thinking process						
34	Believability Through Psychosocial Behaviour: Creating Bots That Are More Engaging and Entertaining [14]	○					Emotion, Personality, Psychosocial						
35	ランキング学習による流れを考慮した自然な指し手の選択方法 [32]	○					*Behavior imitation	○					
36	The Challenge of Constructing Psychologically Believable Agents [36]					○	Motivation, Emotion, Etc.						
37	A fuzzy physiological approach for continuously modeling emotion during interaction with play technologies [45]	▲		▲		▲	Emotion.						○
38	Believability Through Psychosocial Behaviour: Creating Bots That Are More Engaging and Entertaining [14]	○					Emotion, Personality, Psychosocial						
39	ゲームにおけるヒューマンエラー　ー将棋における考察ー [15]	○					Mistake						
40	将棋 AI における棋力の調整が不自然さに与える影響 [28]	○					Difference between Human and AI				○	○	
41	戦略型カードゲームのための戦略獲得法 [91]	○					Human Strategies	○					

continued . . .

No.	Paper name	Purpose					Point of view	AI method					
		Ent	PCG	ADF	EDU	Etc.		SL	RL	EC	Heu	Sea	Etc
42	将棋における個人に適応した着手推定モデルの構築 [54]					▲	Personality, Preference.	○				○	
43	Human-Like Combat Behaviour via Multiobjective Neuroevolution [38]	○					Preference			○			
44	Constructing a Human-like agent for the Werewolf Game using a psychological model based multiple perspectives [56]	▲				▲	Psychological based model				○		
45	Building Machines That Learn and Think Like People [92]						imitate learning process of human.						
46	人間とコンピュータの思考の違い〜囲碁の次の一手問題による考察〜 [93]						Difference of Thinking between Human and AI						○
47	Modifying MCTS for Human-like General Video Game Playing [94]						Habit					○	
48	Creating a Personality System for RTS Bots [13]	○					Personality				○		
49	Rethinking the Human-Agent Relationship: WhichSocial Cues Do Interactive Agents Really Need to Have? [95]					△	Social activity						
50	Survey of How Human Players Divert In-game Actions for Other Purposes: Towards Human-like Computer Players [86]	○					Behavior not directly related to the game purpose		○				

continued . . .

No.	Paper name	Purpose					Point of view	AI method					
		Ent	PCG	ADF	EDU	Etc.		SL	RL	EC	Heu	Sea	Etc
51	Learning Human-like Behaviors using NeuroEvolution with Statistical Penalties [27]	○					Behavioral tendency			○			
52	Detection and Labeling of Bad Moves for Coaching Go [23]				○		Human-like education	○					
53	連鎖構成力向上のためのぷよぷよの問題作成 [24]				○		Preference				○	○	
54	少数の記録からプレイヤの価値観を機械学習するチームプレイ AI の構成 [17]	○					Preference prediction	○					
55	Biased Random Sequence Generation for Making Common Player Believe it Unbiased [87]	○					Mistake,mispreception					○	
56	Influence Map を用いた経路探索による人間らしい弾避けのシューティングゲーム AI プレイヤ [55]	○							○		○	○	
57	Production of Emotion-based Behaviors for a Human-like Computer Player [7]	○					Emotion,sub-objective				○	○	

Acknowledgement

I wish to express my deepest gratitude to my supervisor, Professor Ikeda Kokolo, who he convincingly guided and encouraged me to be professional and do the right thing even when the road got tough. Without his persistent help, I would not make this far until graduation. Also, I would like to express my sincere appreciation to my second supervisor, Professor Iida Hiroyuki, who gave me much advice including life, and attitude.

Also, I would like to pay my respect and gratitude to the other professors who spend their time to be my committee for the dissertation.

Moreover, I gave my thanks to all my academic crew seniors and juniors and everyone whom I ever met though these five years who supported my life in Japan, taught me Japanese and the way to behave, and help my research.

Last but not least, I wish to show whole-heartedly appreciated to my mother, father, anties, who always supported me for my whole life.

Bibliography

[1] Stone Peter, Brooks Rodney, Brynjolfsson Erik, Calo Ryan, Etzioni Oren, Hager Greg, Hirschberg Julia, Kalyanakrishnan Shivaram, Kamar Ece, Kraus Sarit, Leyton-Brown Kevin, Parkes David, Press William, Saxenian AnnaLee, Shah Julie, Tambe Milind, and Teller Astro. "artificial intelligence and life in 2030." one hundred year study on artificial intelligence: Report of the 2015-2016 study panel. *Stanford University, Stanford, CA*, September 2016.

[2] Campbell Murray, Hoane A.Joseph, and Hsu Feng-hsiung. Deep blue. *Artificial Intelligence*, 134(1):57 – 83, 2002.

[3] Silver David, Huang Aja, Maddison Chris J., Guez Arthur, Sifre Laurent, van den Driessche George, Schrittwieser Julian, Antonoglou Loannis, Panneershelvam Veda, Lanctot Marc, Dieleman Sander, Grewe Dominik, Nham John, Kalchbrenner Nal, Sutskever Ilya, Lillicrap Timothy, Leach Madeleine, Kavukcuoglu Koray, Graepel Thore, and Hassabis Demis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484 EP –, Jan 2016. Article.

[4] Togelius Julian, Karakovskiy Sergey, and Baumgarten Robin. The 2009 mario ai competition. In *IEEE Congress on Evolutionary Computation*, pages 1–8, July 2010.

[5] Ontañón Santiago., Synnaeve Gabriel, Uriarte Alberto, Richoux Florian, Churchill David, and Preuss Mike . A survey of real-time strategy game ai research and competition in starcraft. *IEEE Transactions on Computational Intelligence and AI in Games*, 5(4):293–311, Dec 2013.

[6] Fujii Nobuto, Sato Yuichi, Wakama Hironori, Kazai Koji, and Katayose Haruhiro. Evaluating human-like behaviors of video-game agents autonomously acquired with biological constraints. In Dennis Reidsma, Haruhiro Katayose, and Anton Nijholt, editors, *Advances in Computer Entertainment*, pages 61–76, Cham, 2013. Springer International Publishing.

[7] Temsiririrkkul Sila, Hu Phuc Luong, and Ikeda Kokolo. Production of emotion-based behaviors for a human-like computer player. In *GAME-ON2016*, 09 2016.

[8] 瀧澤 武信. The history of the world computer shogi championship (wcsc). In ゲームプログラミングワークショップ *2015* 論文集, volume 2015, pages 92–99, oct 2015.

[9] Browne Cameron B., Powley Edward, Whitehouse Daniel, Lucas Simon M., Cowling Peter I., Rohlfshagen Philipp, Tavener Stephen, Perez Diego, Samothrakis Spyridon, and Colton Simon. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43, March 2012.

[10] P. Hingston. A new design for a turing test for bots. In *Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games*, pages 345–350, Aug 2010.

[11] Togelius Julian, Yannakakis Georgios N., Karakovskiy Sergey, and Shaker Noor. *Assessing Believability*, pages 215–230. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[12] テンシリリックン シラ, 高橋 一幸, サンギュ ナム, and 池田 心. コンピューターゲームプレイヤにおける人間らしさの調査. Technical Report 7, 北陸先端科学技術大学院大学, jun 2018.

[13] Mańdziuk Jacek and Szałaj Przemysław. *Creating a Personality System for RTS Bots*, pages 231–264. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[14] Bailey Christine, You Jiaming, Acton Gavan, Rankin Adam, and Katchabaw Michael. *Believability Through Psychosocial Behaviour: Creating Bots That Are More Engaging and Entertaining*, pages 29–68. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[15] 伊藤 毅志, 杵渕 哲彦, and 藤井 叙人. ゲームにおけるヒューマンエラー ー将棋における考察ー. In ゲームプログラミングワークショップ *2014* 論文集, volume 2014, pages 196–201, oct 2014.

[16] 池田 心. 楽しませる囲碁・将棋プログラミング. オペレーションズ・リサーチ *:* 経営の科学 *= [O]perations research as a management science [r]esearch*, 58(3):167–173, mar 2013.

[17] 和田 尭之, 佐藤 直之, and 池田 心. 少数の記録からプレイヤの価値観を機械学習するチームプレイ ai の構成. Technical Report 5, 北陸先端科学技術大学院大学, feb 2015.

[18] Goodfellow Ian, Pouget-Abadie Jean, Mirza Mehdi, Xu Bing, Warde-Farley David, Ozair Sherjil, Courville Aaron, and Bengio Yoshua. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[19] Togelius Julian, De Nardi Renzo, and M. Lucas Simon. Making racing fun through player modeling and track evolution, 2006.

[20] Shaker Noor, Yannakakis Georgios N., and Togelius Julian. Towards automatic personalized content generation for platform games. In *6th Artificial Intelligence and Interactive Digital Entertainment · AIIDE 2010*, 12 2010.

[21] Chanel Guillaume , Rebetez Cyril , Bétrancourt Mireille , and Pun Thierry . Emotion assessment from physiological signals for adaptation of game difficulty. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 41(6):1052–1063, Nov 2011.

[22] 大森 翔太朗 and 金子 知適. 機械学習を用いた将棋における棋風の学習の研究. Technical Report 6, 東京大学大学院総合文化研究科, jun 2015.

[23] Ikeda Kokolo, Viennot Simon, and Sato Naoyuki. Detection and labeling of bad moves for coaching go. In *2016 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 1–8, Sep. 2016.

[24] 高橋 竜太郎 and 池田 心. 連鎖構成力向上のためのぷよぷよの問題作成. Technical Report 10, 北陸先端科学技術大学院大学, feb 2018.

[25] Hingston Philip. A turing test for computer game bots. *IEEE Transactions on Computational Intelligence and AI in Games*, 1(3):169–186, Sep. 2009.

[26] Vorobiev Ilya and V. Samsonovich Alexei. A conceptually different approach to the empirical test of alan turing. *Procedia Computer Science*, 123:512 – 521, 2018. 8th Annual International Conference on Biologically Inspired Cognitive Architectures, BICA 2017 (Eighth Annual Meeting of the BICA Society), held August 1-6, 2017 in Moscow, Russia.

[27] Huu Phuc Luong, Kanazawa Naoto, and Ikeda Kokolo. Learning human-like behaviors using neuroevolution with statistical penalties. In *2017 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 207–214, Aug 2017.

[28] 仲道 隆史 and 伊藤 毅志. 将棋 ai における棋力の調整が不自然さに与える影響. In *ゲームプログラミングワークショップ 2014 論文集*, volume 2014, pages 167–170, oct 2014.

[29] Lu Feiyu, Yamamoto Kaito, H. Nomura Luis, Mizuno Syunsuke, Lee YoungMin, and Thawonmas Ruck. Fighting game artificial intelligence competition platform. In *Consumer Electronics (GCCE), 2013 IEEE 2nd Global Conference on*, pages 320–323, 10 2013.

[30] Schrum Jacob, V. Karpov Igor, and P Miikkulainen Risto. Ut2: Human-like behavior via neuroevolution of combat behavior and replay of human traces. In *2011 IEEE Conference on Computational Intelligence and Games (CIG'11)*, pages 329–336, Aug 2011.

[31] Muñoz Jorge, Gutierrez German, and Sanchis Araceli. *Towards Imitation of Human Driving Style in Car Racing Games*, pages 289–313. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[32] 杵渕 哲彦 and 伊藤 毅志. ランキング学習による流れを考慮した自然な指し手の選択方法. In ゲームプログラミングワークショップ *2016* 論文集, volume 2016, pages 119–122, oct 2016.

[33] Ortega Juan, Shaker Noor, Togelius Julian, and Yannakakis Georgios N. Imitating human playing styles in super mario bros. *Entertainment Computing*, 4(2):93 – 104, 2013.

[34] Karpov Igor V., Schrum Jacob, and Miikkulainen Risto. *Believable Bot Navigation via Playback of Human Traces*, pages 151–170. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[35] Fountas Zafeirios, Gamez David, and Fidjeland Andreas K. A neuronal global workspace for human-like control of a computer game character. In *2011 IEEE Conference on Computational Intelligence and Games (CIG'11)*, pages 350–357, Aug 2011.

[36] Schonbrodt Felix and Asendorpf Jens. The challenge of constructing psychologically believable agents. *Journal of Media Psychology*, 23:100–107, 01 2011.

[37] Miyashita Shohei, Lian Xinyu, Zeng Xiao, Matsubara Takashi, and Uehara Kuniaki. Developing game ai agent behaving like human by mixing reinforcement learning and supervised learning. In *2017 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, pages 489–494, June 2017.

[38] Schrum Jacob, V. Karpov Igor, and P Miikkulainen Risto. *Human-Like Combat Behaviour via Multiobjective Neuroevolution*, pages 119–150. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[39] Van Hoorn Niels, Togelius Julian, Wierstra Daan, and Schmidhuber Jurgen. Robust player imitation using multiobjective evolution. In *2009 IEEE Congress on Evolutionary Computation*, pages 652–659, May 2009.

[40] Tan Chin Hiong, Tan Kay Chen, and Shim Vui Ann. Learning believable game agents using sensor noise and action histogram. *Memetic Computing*, 6(4):215–232, Dec 2014.

[41] 中野 雄基, 美添 一樹, and 脇田 建. 進化計算と uct による mario を人間らしくプレイする ai. ゲームプログラミングワークショップ *2013* 論文集, pages 81–88, nov 2013.

[42] Hagelbäck Johan and Johansson Stefan J. A study on human like characteristics in real time strategy games. In *Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games*, pages 139–145, Aug 2010.

[43] 白鳥 和人, 塩入 健太, and 星野 准一. 感情表現生成を行う仮想ゲームプレイヤ. 芸術科学会論文誌, 7(2):65–74, 2008.

[44] 佐藤 祐一 and 片寄 晴弘. ビデオゲームエージェントの自律的行動獲得と観測情報の信頼性に着目した獲得行動パターンの分析. Technical Report 17, 関西学院大学理工学研究科情報科学専攻, mar 2012.

[45] Mandryk Regan L. and Atkins M. Stella. A fuzzy physiological approach for continuously modeling emotion during interaction with play technologies. *Int. J. Hum.-Comput. Stud.*, 65(4):329–347, April 2007.

[46] Green Robert D., MacDorman Karl F., Ho Chin-Chang, and Vasudevan Sandosh. Sensitivity to the proportions of faces that vary in human likeness. *Computers in Human Behavior*, 24(5):2456 – 2474, 2008. Including the Special Issue: Internet Empowerment.

[47] 生井 智司 and 伊藤 毅志. 将棋における棋風を感じさせる ai の試作. 研究報告ゲーム情報学（*GI*）, 2010(3):1–7, jun 2010.

[48] Hou Yaqing, Feng Liang, and Ong Yew-Soon. Creating human-like non-player game characters using a memetic multi-agent system. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 177–184, July 2016.

[49] Yannakakis Georgios N. and Togelius Julian. Experience-driven procedural content generation. *IEEE Transactions on Affective Computing*, 2(3):147–161, July 2011.

[50] Yannakakis Georgios N. and Togelius Julian. Experience-driven procedural content generation (extended abstract). In *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 519–525, Sep. 2015.

[51] 杵渕 哲彦 and 伊藤 毅志. 手の流れを考慮して自然な手を選ぶ将棋 ai の試作. Technical Report 12, 電気通信大学情報・通信工学専攻, feb 2015.

[52] 高橋 克吉, 猪爪 歩, 伊藤 毅志, 村松 正和, and 松原仁. 次の一手課題に基づく囲碁と将棋の特徴比較. In ゲームプログラミングワークショップ *2012* 論文集, volume 2012, pages 1–8, nov 2012.

[53] Tencé Fabien, Buche Cédric, De Loor Pierre, and Marc Olivier. The challenge of believability in video games: Definitions, agents models and imitation learning. *CoRR*, abs/1009.0451, 2010.

[54] 山内 智晴 and 鶴岡 慶雅. 将棋における個人に適応した着手推定モデルの構築. In *ゲームプログラミングワークショップ 2016 論文集*, volume 2016, pages 112–118, oct 2016.

[55] 佐藤 直之, Temsiririrkkul Sila, Huu Phuc Luong, and 池田 心. Influence map を用いた経路探索による人間らしい弾避けのシューティングゲーム ai プレイヤ. In *ゲームプログラミングワークショップ 2016 論文集*, volume 2016, pages 57–64, oct 2016.

[56] Nakamura Noritsugu, Inaba Michimasa, Takahashi Kenichi, Toriumi Fujio, Osawa Hirotaka, Katagami Daisuke, and Shinoda Kousuke. Constructing a human-like agent for the werewolf game using a psychological model based multiple perspectives. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8, Dec 2016.

[57] Cowley Stephen J. , Ishiguro Hiroshi, Itakura Shoji, MacDorman Karl F., Minato Takashi, and Shimada Michihiro. Assessing human likeness by eye contact in an android testbed. In *Conference: XXVII Annual Meeting of the Cognitive Science Society*, 2005.

[58] 藤井 叙人, 福嶋 良平, and 片寄 晴弘. エンタテインメント系システムの主観評価実験におけるユーザ統制及び実験手法の検討. In *エンタテインメントコンピューティングシンポジウム 2015 論文集*, volume 2015, pages 429–434, sep 2015.

[59] Tsuruoka Yoshimasa, Yokoyama Daisaku, and Chikayama Takashi. Game-tree search algorithm based on realization probability. *ICGA Journal*, 25(3):145–152, 2002.

[60] 野村 久光, テンシリリックン シラ, and 池田 心. 標準的なゲームプレイヤにとって自然に見える疑似乱数列の生成法. In *ゲームプログラミングワークショップ 2013 論文集*, pages 27–34, nov 2013.

[61] Bishop Christopher. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, 1 edition, 2006.

[62] Mnih Volodymyr, Kavukcuoglu Koray, Silver David, Rusu Andrei A., Veness Joel, Bellemare Marc G., Graves Alex, Riedmiller Martin, Fidjeland Andreas K., Ostrovski Georg, Petersen Stig, Beattie Charles, Sadik Amir, Antonoglou Ioannis, King Helen, Kumaran Dharshan, Wierstra Daan, Legg Shane, and Hassabis Demis. Human-level control through deep reinforcement learning. *Nature*, 518:529 EP –, Feb 2015.

[63] Ng Andrew Y and Russell Stuart J. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2, 2000.

[64] 平井 弘一 and Grimbergen Reijer. 弾幕の認識に人間の視覚特性を取り入れたシューティングゲーム ai の研究. In *ゲームプログラミングワークショップ 2016 論文集*, volume 2016, pages 158–161, oct 2016.

[65] Mnih Volodymyr, Kavukcuoglu Koray, Silver David, Graves Alex, Antonoglou Ioannis, Wierstra Daan, and Riedmiller Martin A. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013.

[66] Soni B. and Hingston P. Bots trained to play like a human are more fun. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 363–369, June 2008.

[67] Ikeda Kokolo and Viennot Simon. Production of various strategies and position control for monte-carlo go — entertaining human players. In *2013 IEEE Conference on Computational Inteligence in Games (CIG)*, pages 1–8, Aug 2013.

[68] Mathison Turing Alan. I.—COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, LIX(236):433–460, 10 1950.

[69] Shaker Noor, Togelius Julian, Yannakakis Georgios N., Poovanna Likith, S. Ethiraj Vinay, Stefan J. Johansson, Reynolds Robert, K. Heether Leonard, Schumann Tom, and Gallagher Marcus. The turing test track of the 2012 mario ai championship: Entries and evaluation. In *2013 IEEE Conference on Computational Intelligence in Games (CIG)*, pages 1–8, 08 2013.

[70] T.L. Taylor. The assemblage of play. *Games and Culture*, 4(4):331–339, 2009.

[71] Yee Nick. Motivations for play in online games. *Cyberpsychology & behavior : the impact of the Internet, multimedia and virtual reality on behavior and society*, 9:772–5, 01 2007.

[72] Frederick E. James. A review of pseudorandom number generators. *Computer Physics Communications*, 60(3):329 – 344, 1990.

[73] Preis Tobias, Virnau Peter, Paul Wolfgang, and Schneider Johannes J. Gpu accelerated monte carlo simulation of the 2d and 3d ising model. *Journal of Computational Physics*, 228(12):4468 – 4477, 2009.

[74] Matsumoto Makoto and Nishimura Takuji. Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.*, 8(1):3–30, January 1998.

[75] Tversky Amos and Kahneman Daniel. Judgment under uncertainty: Heuristics and biases. *Science*, 185(4157):1124–1131, 1974.

[76] Haselton Martie G, Nettle Daniel, and Murray Damian R. *The Evolution of Cognitive Bias*, chapter 41, pages 1–20. American Cancer Society, 2015.

[77] Gilovich Thomas, 守 一雄, and 守 秀子. 人間この信じやすきもの-迷信誤信はどうして生まれるか. 1993.

[78] Jonathan St. B. T. Evans and 中島 実. 思考情報処理のバイアス-思考心理学からのアプローチ. pages 51–81, 1995.

[79] 樺 旦純. ダマされる人・ダマされない人一心のワナにかからないための心理学. PHP 文庫, 2002.

[80] Tversky Amos and Kahneman Daniel. Belief in the law of small numbers. *Psychological bulletin*, 76(2):105, 1971.

[81] Dvorsky George. The 12 cognitive biases that prevent you from being rational. 2013.

[82] Carroll Robert T. Clustering illusion, 2014.

[83] Bakan Paul. Response-tendencies in attempts to generate random binary series. *The American Journal of Psychology*, 73:127–131, 1960.

[84] Bar-Hillel Maya and Wagenaar Willem A. The perception of randomness. *Advances in Applied Mathematics*, 12(4):428 – 454, 1991.

[85] Schilling Mark F. The longest run of heads. *The College Mathematics Journal*, 21(3):196–207, 1990.

[86] Temsiririrkkul Sila, Sato Naoyuki, Nakagawa Kenta, and Ikeda Kokolo. Survey of how human players divert in-game actions for other purposes: Towards human-like computer players. In Nagisa Munekata, Itsuki Kunita, and Junichi Hoshino, editors, *Entertainment Computing – ICEC 2017*, pages 243–256, Cham, 2017. Springer International Publishing.

[87] Temsiririrkkul Sila, Nomura Husamitsu, and Ikeda Kokolo. Biased random sequence generation for making common player believe it unbiased. In *2014 IEEE Games Media Entertainment*, pages 1–8, Oct 2014.

[88] 渡邉 裕 and 中村 貞吾. 囲碁 ai 作成に向けた棋風を形成する要素に関する統計的分析. ゲームプログラミングワークショップ *2011* 論文集, 2011(6):62–67, oct 2011.

[89] 大森 翔太朗 and 金子 知適. 将棋における棋譜から棋風を学習するための研究. 情報処理学会論文誌, 57(11):2374–2381, nov 2016.

[90] Alderisio Francesco, Antonacci Dario, Zhai Chao, and Bernardo Mario di. Comparing different control approaches to implement a human-like virtual player in the mirror game. In *2016 European Control Conference (ECC)*, pages 216–221, June 2016.

[91] 藤井 叙人 and 片寄 晴弘. 戦略型トレーディングカードゲームのための戦略獲得手法. 情報処理学会論文誌, 50(12):2796–2806, dec 2009.

[92] Lake Brenden M., Ullman Tomer D. , Tenenbaum Joshua B., and Gershman Samuel J. Building machines that learn and think like people. *CoRR*, abs/1604.00289, 2016.

[93] 伊藤 毅志, 高橋 克吉, 猪爪 歩, 加藤 英樹, 村松正和, and 松原 仁. 人間とコンピュータの思考の違い～囲碁の次の一手問題による考察～. In ゲームプログラミングワークショップ *2012* 論文集, volume 2012, pages 9–16, nov 2012.

[94] Khalifa Ahmed, Isaksen Aaron, Togelius Julian, and Nealen Andy. Modifying mcts for human-like general video game playing. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 2514–2520. AAAI Press, 2016.

[95] Astrid Weiss and Manfred Tscheligi. *Rethinking the Human–Agent Relationship: Which Social Cues Do Interactive Agents Really Need to Have?*, pages 1–28. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.