JAIST Repository
https://dspace.jaist.ac.jp/

Title	Production of Emotion-based Behaviors for a Human-like Computer Game Player [課題研究報告書]
Author(s)	Temsiririrkkul, Sila
Citation	
Issue Date	2015-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/12957
Rights	
Description	Supervisor: 池田　心, 情報科学研究科, 修士

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

Production of Emotion-based Behaviors for a Human-like Computer Game Player

By Temsiririrkkul Sila

School of Information Science

March, 2015

Production of Emotion-based Behaviors for a Human-like Computer Game Player

By Temsiririrkkul Sila (1310046)

A project paper submitted to
School of Information Science,
Japan Advanced Institute of Science and Technology,
in partial fulfillment of the requirements
for the degree of
Master of Information Science
Graduate Program in Information Science

Written under the direction of
Associate Professor Ikeda Kokolo

and approved by
Associate Professor Ikeda Kokolo
Professor Iida Hiroyuki
Associate Professor Shirai Kiyoaki

February, 2015 (Submitted)

Copyright © 2015 by Temsiririrkkul Sila

Abstract

For a long while, the main goal of academic research in computer games was to make a suitable computer opponent for humans. To reach this goal, the first and major step was to improve the strength of the computer players. In the recent years, the computer game players have improved significantly due to the advance of search algorithms and computer technology. The computer players are now strong enough to win against average human players in classical board games such as Chess or Othello, but also in modern computer games such as Starcraft. It is perhaps time to devote more attention to other issues of computer players such as their usage for education or entertainment purposes.

The current performance of computer game players is very promising and efficient in terms of strength, but the behavior of such strong computer players is not so promising for the entertainment of players. For example, in a famous video of a computer player for Infinite Mario Bros, Mario character shows very precise movements, and no hesitation in the decisions. Such behavior looks very mechanical, in other words too strong and then not so entertaining.Furthermore, in the case of multiplayer games, since the computer player may be a partner or an opponent of the human player, such too strong or unnatural behaviors can make human players suspect that they are being cheated, and the entertainment of the game will be harmed. Hence, the production of behavior that looks natural to humans, called human-like, is essential to make computer players able to entertain humans.

Many approaches were proposed to produce human-like behavior. Schrum presented a human-like computer player based on a neural network for First-Person Shooter games. The computer player learning was supervised by human-player data recorded from past games. In 2013 Fujii et al. showed a new approach to produce human-like behavior with human-like mistakes. The biological constraints, including sensory error, perceptual and motion delay, and physical fatigue, were introduced into the path finding algorithm A* and the reinforcement learning Q-Learning. This research tried to imitate the behavior of human players by using a single behavior model, so the transition between different behaviors was not explicitly implemented, though such transition is common for human players. Then one of our goal in this research is to produce such transition between multiple behaviors.

In order to conduct a research on computer players, a suitable test-bed platform is needed. The classical platform of board games such as Chess or Go was the main target over the last 30 years. In order to play well at these board games, good reasoning and planning is required. In modern video games, other capacities such as pattern recognition, navigation and decision making in a short period are required. Furthermore, many games have imperfect information and they are played not only by one or two players but by three or even more. Thus it is a very challenging target. Recently, Sergey et al. published the platform called “Mario AI Benchmark” to evaluate computer players in a side-scrolling video game. The platform provides an API which allows developers to implement their own computer player. Some competitions were held using this platform, targeting not

only strength, but also human-likeness and entertainment. Then we also employ the same test-bed platform in this research.

In many modern video games, multiple goals are given to the players. Sometimes a player might choose to suspend the main goal of the game and challenge himself for another goal. For example in Super Mario Bros, the main goal of the game is to clear the stage within a limited time. In the beginning, the player tries to reach the goal as fast as possible. But after he finds some coins, the player might ignore the main mission and try to collect coins, which is a sub-objective of the game. He might be inspired by his greed or need of enjoyment. Or when he encounters many enemies at once, he might stop moving or collecting the coins and run away from the risk of being killed. Such behavior might be inspired by his fear. Such changes of behavior are inspired by human feelings or emotions. Hence, we believe that the production of behavior transition is important to obtain a human-like behavior.

In this study, we propose a design of human-like computer players with five emotional behaviors : "Safety", "Hurry", "Greedy", "Enjoy" and "Habit". "Safety" reflects anxiousness and fear of the player when he is on guard. "Hurry" shows speedy less careful actions, when the player is anxious about the remaining time. "Greedy" reflects the enjoyment of humans when they find rewards. "Enjoy" reflects the enjoyment and interest such as killing enemies continuously. "Habit" reflects unintended behavior such as pressing repeatedly the jump button.

And also we propose a hand-coded rule-based transition model of these behaviors.

As a preliminary experiment, we carefully investigate the human-likeness of the "Safety", "Hurry" and "Greedy" models, by comparing play videos from these models, from the A* algorithm and from human beginners. We conclude that our proposed models can successfully produce very natural behavior specific to each purpose. In the near future, automatic switching between the behaviors will be investigated with the proposed transition model. The individual performance of the behavior models and the total performance of the transition model with behavior switching will be evaluated by both of a Turing test and an entertainment test.

Contents

1 Introduction **1**

2 Literature review **4**
2.1 Human-likeness 4
2.2 Turing Test 6
2.3 Entertainment of game 8
2.4 Human's emotion and behavior 9
2.5 Area of contribution 10

3 Mario AI benchmark **11**
3.1 Framework 11
3.2 Comptetitions 15
3.2.1 Gameplay Track 15
3.2.2 Turing Test Track 15
3.2.3 Level generation Track 15
3.3 A* Path finding algorithm 17
3.4 Human-like Behavior with Biological Constraints 18

4 Methodology **20**
4.1 Core Concept 20
4.2 Research Framework 21
4.3 Local behavior models 23
4.3.1 Safety Model 23
4.3.2 Hurry Model 24
4.3.3 Greedy Model 25
4.3.4 Enjoy Model and Habit Model 26
4.4 Transition Model 27
4.5 Assessment method 29

5 Experiment Result **30**
5.1 Safety Model 30
5.2 Hurry Model 34
5.3 Greedy model 35

6 Discussion **36**

Chapter 1

Introduction

For a long while, the ideal goal of the academic study of computer games, was to make a suitable computer opponent for humans. To reach this goal, the strength of the computer player was the first and major aim. Many techniques/algorithms were invented and debated in order to optimize the performance in a restricted environment. In 1996, the chess computer "Deepblue" of IBM became the first computer player to win against the world champion Garry Kasparov[8]. Afterwards, research in this field became more popular among academic researchers.

In the recent years, the computer game players have improved significantly due to the advance of computer technology. Not only in classical board games such as Chess or Othello but also in modern computer games such as Starcraft, the computer players have become strong enough to win against average human players. It is perhaps time to devote more attention to other issues of an automatic player such as the education or entertainment purposes.

In the video games (commercial games), sometimes computer players were developed for controlling the character as a partner or an opponent, in order to entertain human players. The suitable design for the computer player's behavior or strategy is difficult to find and often becomes a heavy burden for the developers. Thus, advanced algorithms such as the path finding algorithm A* (Claessens, 2012) or the learning algorithm TD-learning[13] are applied to generate behaviors or strategies in order to reduce the load of developers' work.

The current performance of computer game players is very promising and efficient. For example, many computer chess players are very powerful and it is hard even for master players to win against them. However, the behavior of strong computer players is not so promising for the entertainment of players. For example, a famous video of a computer player for Infinite Mario Bros (the public domain clone of Super Mario Bros of Nintendo) was published via website youtube.com. The video shows Mario character with very precise movements, and no hesitation in its decisions. Such behavior looks remarkably mechanical. In this case, a human observer only acknowledges that the player is a machine. However, in case of two-player games (e.g. fighting game like street fighter) or multi-player games (e.g. Shooter game like Unreal tournament 2004) where the computer player simultaneously plays with human players, humans might suspect that they are being

cheated and the entertainment of the game will be harmed because of unnatural behaviors of the computer player. Hence, the production of behavior that looks natural to humans, called human-like, is essential to make computer players able to entertain human players.

Many approaches were proposed to produce human-like behavior. Schrum et al. present UT$\hat{2}$, a neural network based human-like computer player for the UT2004 game (UT2004 or Unreal Tournament 2004 is a first person 3D shooter game. The goal of the game is to gain the highest possible score by killing other players). The computer player was supervised by human-player data recorded from past games. The computer player participated in 2K Botprize 2010. However, the results of the competition show that the player is still insufficient to be more human-like than human player[4]. In 2013 Fujii et al. showed a new approach to produce human-like behavior with human-like mistakes. The biological constraints, including "Sensory error", "Perceptual and motion delay", and "Physical fatigue", were introduced into the path finding algorithm A* and the reinforcement learning Q-Learning[9]. This research tried to imitate the behavior of human players by using a single behavior model, so the transition between different behaviors was not explicitly implemented, though such transition is common for human players.

In order to conduct a research on computer players, a suitable test-bed platform is needed. The classical platform of board games (e.g. Chess, Shogi, or Go), was the main target over the last 30 years. In order to play them well, good reasoning and planning is required. This is a very difficult and challenging task for researchers, but current computer players such as Monte-Carlo based computer players of computer Go are able to show high performance. In modern video games, other capacities such as pattern recognition, navigation and decision making in a short period are required. Furthermore many games are imperfect information and they are played not only by one or two players but by three or even more. Thus it is a very challenging target. In recent years, Sergey et al. published the platform called "Mario AI Benchmark" to evaluate computer players. The platform is based on "Infinite Mario Bros" which is the clone game of Nintendo's classic 2-dimension side-scrolling "Super Mario Bros". The platform provides an API (Application programming interface) which allows developers to implement their own computer player[14][10].

To be able to evaluate the level of the researcher's computer players, the competitive stage is important. The developers of the platform provide the "Mario AI Championship" which allows researchers to test their computer players and exchange knowledge with other researchers. The competition has been held with the international conference IEEE Computational Intelligence in Games (IEEE-CIG) every year[11][6].

In many modern video games, multiple goals are given to the players. Sometimes a player might choose to suspend the main goal of the game and challenge himself for another goal. For example in Super Mario Bros, the main goal of the game is to clear the stage within a limited time. In the beginning, the player tries to reach the goal as fast as possible. But after he finds some coins and acknowledges that he has enough time, the player might ignore the main mission and try to collect coins, which is a sub-objective of the game. He might be inspired by his greed or need of enjoyment. Such changes of behavior are inspired by human feelings or emotions. Hence, the production of behavior

transition is important to obtain a human-like behavior.

In this study, we propose a design of human-like computer players with emotional behaviors. We selected as a test-bed game the Infinite Mario Bros, for which the API is provided and is known by a lot of people,. Our research aims to produce a computer player with the transition of five specific behaviors including "Safety", "Hurry", "Greedy", "Enjoy" and "Habit". These elements describe how humans often behave while playing Super Mario Bros. All the elements (except "Habit") are inspired by human emotions which can be explained as follows:

- "Safety" reflects anxiousness and fear of the player when he is on guard.
- "Hurry" shows speedy less careful actions, when the player is anxious about the remaining time.
- "Greedy" reflects the enjoyment of humans when they find rewards.
- "Enjoy" reflects the enjoyment and interest such as killing enemies continuously.
- "Habit" presents unintended behavior such as pressing repeatedly the jump button

As a base algorithm, the path finding algorithm A* using a combination of cost function and heuristic, is selected due to the flexibility of the heuristic and easiness of combining multiple behaviors. Our research is conducted under 2 research questions: " how to produce human-like behavior which looks like it is inspired by emotions" and "how to learn the behavior transition trigger from humans". The evaluation of the research contains two directions: "Turing test" and "Entertainment Test". The former is used in order to evaluate the naturalness of the computer player. The test is conducted under the same method as "Mario AI Championship" in 2010. The latter is originally added in order to evaluate the capability to entertain humans, which cannot be evaluated only by the Turing test.

Chapter 2

Literature review

2.1 Human-likeness

The topic of human-likeness in machines is one interesting topic in the field of philosophy of mind and cognitive science. The study in this field has been discussed widely since Alan Turing proposed the Turing-Test in his article "Computing Machinery and Intelligence" [1]. Until now, the definition is still not precise.

In the field of computer games, the study of human-like behavior (also known as believability) has interested many researchers. Every year, new approaches for generating human-like computer players are proposed and implemented. However, as far as we know the definition of human-like behavior is still ambiguous even in the special case of computer games.

Considering the word "believability", the literal meaning is that something can be reasonably believed by someone. Thus in the computer player research domain, the word might be interpreted as "someone believes that some character or computer player is real."

Togelius et al. show two classes of believability as follows:

- Character believability: Someone believes that the character/bot itself is real, i.e. an actual living being.
- Player believability: Someone believes that the player controlling the character/bot is real, i.e. that a human is playing.

Character believability is related to the reality of the shown character itself. When using very high level computer graphic animation, we may be able to believe that the character in the film is a human in costume. However with the current technology of media, it is difficult to achieve such believability.

Player believability is related to the case of video games such as Starcraft and Super Mario Bros, where either a computer player or a human player can play a character in the game. Thus the reality of visualization is not related to the believability. On the other hand, the player observes the behavior and judges whether a human player is playing or not [5].

Effects of believability/human-likeness can be classified into two levels. In the case where humans do not participate to the game (i.e. in one-player games) such as Infinite Mario Bros, the unnaturalness of the computer player may be not so harmful. On the other hand, in two or multiplayer games where players simultaneously play with a computer player and when the computer player is assigned as a partner/opponent of human, unnaturalness will directly harm the entertainment of the game.

2.2 Turing Test

The Turing test was proposed in 1950 by Alan Turing with the question "Can machines think?" and the concept has been discussed many times. The purpose of the test is to evaluate the machine's ability to generate a behavior difficult to distinguish from human's behavior. The model of Turing test was proposed in a concrete manner called Imitation game. The game is played by 3 person Man (A), Woman (B) and Interrogator (C) (refer to fig 2.1). A and B are in the same room but C is in a different room. C has to ask the question to A and B and finally he has to answer what is the gender of A and B.

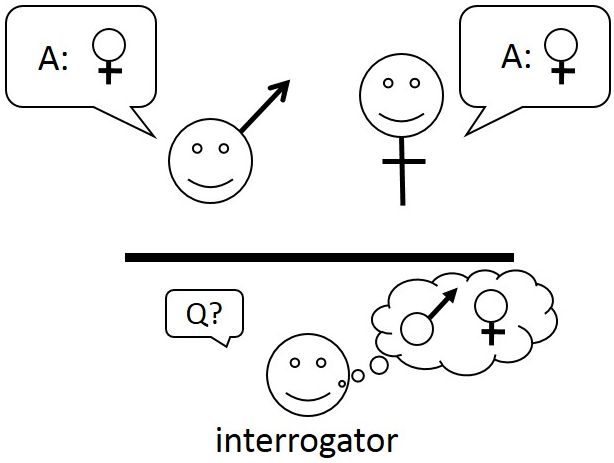

Figure 2.1: Imitation game (human-human)

The method of the game was applied to assess machine behavior. As we show in Fig 2.2, A was replaced with Machine but other rules are still similar to the previous test. But this time C has to identify which one of the two entities is the human.

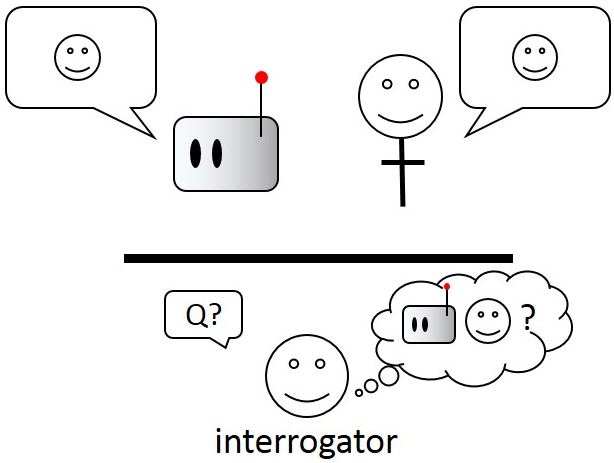

Figure 2.2: Imitation game (machine-human)

The concept of the Turing test has been widely used in computer science research. The actual method has been modified many times, yet it is still far from being complete.

The Turing test is also used to assess the human-likeness of computer game players. Hingston presented the Turing test for computer game players in his 2009 article. The test is based on an imitation game of Turing. There are three participants, an interrogator (human judge), a human player and a computer player. The task of the judge is to distinguish the computer player by observing the behavior of the two players. The test is based on the original Turing test with several differences:

- The participants are not in a two-way interaction but three-way interaction (all three participate to the game).
- The human player is not trying to assist the judge but to win the game.
- The task is more restricted than conversation in natural language.

This method was presented in order to assess human-like computer players in 2K BotPrize Contest. This competition of computer players uses Unreal Tournament 2004 as a test-bed game [12].

Based on the earlier approach, Togelius et al. discussed the difference between first person assessment (the judge also participates to the game) and third person assessment. In the first person assessment where the judge takes part to the game, game-player interactions might influence the judgment. On the other hand, in third person assessment where the judge is independent from the gameplay, the judge can concentrate more on the assessment. Thus the test becomes more accurate [11].

2.3 Entertainment of game

In the last 30 years, video games have significantly improved in many aspects such as story, system or visualization. A large number of commercial games were released. Many games received good feedback while others did not. This left us a question "what is entertainment of games and how we evaluate it".

Malone introduced three factors which make video games interesting: challenge, fantasy and curiosity. Challenge occurs when the situation gives an uncertain result, or in a situation with time constraint, or in a competition with other players. Fantasy is a situation which allows players to face an over-realistic experience [16].

Elements in the game are also important factors to make the game fun. For example, games with high quality visualization are able to make the player feel like it is realistic and enjoy the story of the game. In two or multi-player video games, computer players are introduced either as allies or as opponents of human players. Some years ago, the production of skillful computer players was difficult due to the limitation of technology. In new generations of games, many algorithms were introduced for preparing strong computer players. It can be said that present computer game players have almost sufficient performance to win against standard human players. However, the computer players with advanced algorithm sometimes take too perfect actions, acting extremely precisely, with no mistake, no delay, and no hesitation. From the point of view of common human players, such behavior is hard to accept. Such situation can harm the entertainment of the game.

One of the difficult tasks of entertainment media developer is to evaluate the fun and entertainment of the media. So far, the only method that has been used is to evaluate from feedback questionnaire or data from entertainment web sites [3]. However, it might encounter individual biases and the result might be deviated. Thus a good design of the evaluation model is needed in order to improve the accuracy of the assessment.

2.4 Human's emotion and behavior

Emotions are one unique specification of intelligence that can be qualified as especially human. Human emotions are varied and often change. Furthermore, emotion affects the behavior directly. For example, we can imagine a workplace in Friday afternoon when people start to walk around the room, the work becomes slower. In their mind they are already enjoying the activity after work. As another example, we can imagine someone carrying a pot almost full of boiled water. Because of anxiety and fear, his moves will be slow and careful.

In game-play, some actions are also affected and inspired by human emotions such as fear, anxiety or enjoyment. For example in a situation where Mario character is in invisible state (effect of some game item during a short period) the player enjoys to kill as much enemies as possible. On the other hand, if the character is surrounded by many enemies, the player might hesitate to move forward or backward due to fear and anxiety.

There are some approaches to introduce emotions to machines. Canamero presents his paper about emotion for behavior control. The paper discussed the importance of including emotion into machines so that the system has better communication ability and a flexible behavior [2].

Our work aims to produce computer players with human-like behavior that looks like it is inspired by some emotions. We propose 5 specific models as follows: "Safety" model shows a behavior which reflects anxiety and fear of the player. In this state, the character should move carefully and the output behavior should look like he intends to be safe. "Hurry" model shows anxious behavior when the player is concerned by the remaining time. The computer player acts speedily and takes less careful actions. "Greedy" model represents the enjoyment of collecting rewards. "Enjoy" model also reflects the enjoyment and interest of the player. The enjoyment of this model does not come only from profitable actions but also from non-profitable actions such as killing enemies continuously or destroying all breakable blocks. "Habit" model presents unintended behavior such as pressing repeatedly the jump button.

2.5 Area of contribution

The final aim of this research is a computer player which seems to be controlled by a human. Our hypothesis is that the transition of behaviors inspired by emotions is the key to produce natural behaviors. Under this hypothesis we discuss the following problem statement:

Problem statement

- Can computer players present emotional behaviors and their transition?

To answer the problem statement, we formulated 2 research questions.

As we all know, a computer player is a program which gets inputs as game data to process and returns some pattern behaviors. The computer player has no literal emotion. Thus we formulated the first research question.

- How to produce a human-like behavior which looks like being inspired by an emotion?

Human behaviors are inspired by emotions which often change. To imitate human-like transitions between behaviors, learning from humans might be the direct and the most efficient way. Hence the second question has been formulated.

- How to learn the trigger of the emotion transition from human players?

Chapter 3

Mario AI benchmark

This chapter describes a test-bed platform "The Mario AI Benchmark" which is used in this research. We describe 4 topics about the framework itself, the competition using this framework, the usage of A* algorithm in Mario, and a recent approach for obtaining a human-like computer player with human mistakes.

3.1 Framework

The Mario AI benchmark is a test-bed platform based on Infinite Mario Bros (developed by Markus Persson). The game is a public domain clone of Super Mario Bros, a classic side scrolling game published by Nintendo in 1985. The Infinite Mario Bros is an open source playable program, available on the web. In Super Mario Bros, the player plays a

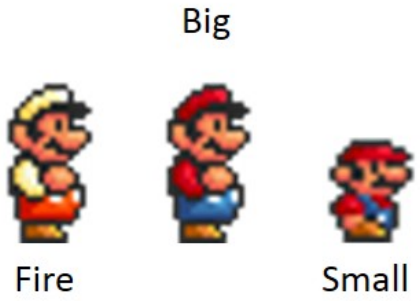

Figure 3.1: Three states of Mario.

character named Mario, travelling through a 2-dimentional stage. Mario has three states: Small, Big and Fire where the "Small" state is the weakest, the "Big" state is able to crush breakable obstacle and the "Fire" state can shoot fireballs to burn enemies (See Fig. 3.1). The character is able to walk and run to left and right, jump, or shoot fire-balls (if he is on fire state). There is gravity in the game thus it is necessary to jump over the obstacles such as cliffs or pipes to go through a level.

Figure 3.2: Examples of Enemies in Infinite Mario bros.

The goal of the game is defined as to clear the stage (usually to reach the right end) within a time limit. However, there are subtasks for the player to challenge, such as collecting coins, or making high scores (score is obtained from killing enemies or collecting coins). There are obstacles: "Enemies" (See Fig 3.2) and "Holes" (See Fig 3.3) scattered along the stage. If Mario touches an enemy, he gets hurt. In case of "Small" Mario, he will instantly be killed. Otherwise he will be degraded from "Fire" or "Big" to "Small". On the other hand if Mario falls into a hole, he will immediately be killed even if on "Fire" or "Big" state.

Figure 3.3: Hole in Infinite Mario bros.

Infinite Mario Bros platform also provides a "stage generator" which can generate original and random stages from given seed parameters and a difficulty parameter. This makes the game more suitable as a tool for testing computer players.The Mario AI benchmark provides an Application Program Interface (API) which allows researchers to create their own computer player easily. The system is independent from the system clock and drawing process, so researchers can consider only the behavior of Mario. At each tick, the controller obtains the environment data and responds with an action. The time of each step is 40 milliseconds since the refresh rate of this game is 25 frames/s.

The API of Mario AI benchmark contains an environment interface and an agent interface. The environment interface allows the controller to access the information of the stage which includes the following items.

- Receptive field: It represents the world around Mario in 2D arrays. The agent controller is allowed to change the level of the object's detail in the array by changing the Z-Level., for example 22×22 or 7×7 (See Fig 3.4). Developers can select convenient settings, considering a trade-off between accuracy and speed.

- Exact position of enemies: The data from the receptive field show the rough position of enemies to the controller, however it might be not enough accurate for some controllers. Then, the exact position of enemies is also available as a list of x and y coordinates.

- Mario state: Binary or discrete variables are available individually with the information about Mario such as the state of Mario (Small, Big, Fire), whether Mario is able to jump, whether Mario is on the ground, and whether Mario is carrying a shell.

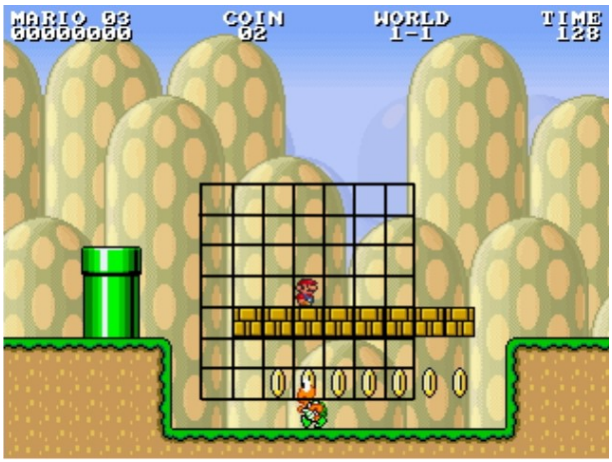

Figure 3.4: Sample of receptive field around Mario.

In order to develop a computer player, the key function "getAction()" should be implemented in the controller class. This function returns 5 Boolean values as an array, corresponding to whether each button (of the game controller) is pressed or not. The different buttons lead to a left move, a right move, ducking, dash, fireball shooting and jumping (See Fig. 3.5). Thus the number of possible actions is $2^5 = 32$, though some of them have no meaning such as pushing left and right at the same time.

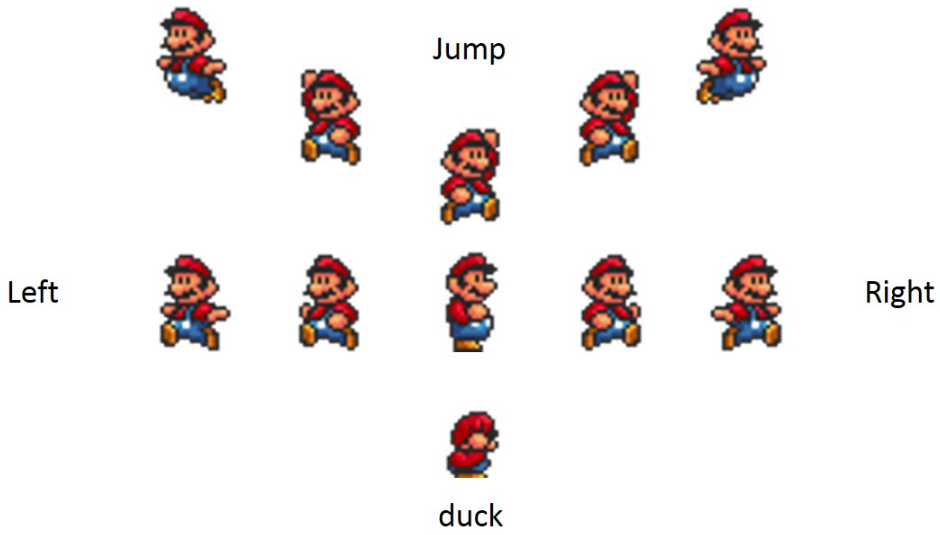

Figure 3.5: Possible actions of Mario.

3.2 Comptetitions

In recent years, many game AI competitions were organized associated with major international academic conferences. For example, StarCraft AI competition is held in IEEE Computational Intelligence in Games (IEEE-CIG) every year. Such competitions allow researchers to evaluate the performance of their computer players AI. For the Mario games, Mario AI benchmark development team provided the Mario AI competition associated with IEEE-CIG in order to expand the platform and exchange knowledge of researchers who are interested in this domain.

The Mario AI competition started in 2009, and surprisingly it provided 3 types of contests.

3.2.1 Gameplay Track

The goal of a controller submitted to "gameplay track" is to clear the stage as much as possible. The aim of this track is the strength of computer players. For scoring, all entries are scored once before the conference. All of them run through 10 levels with increasing stage difficulty. The next part of competition runs in CIG period. The controllers need to play along 40 stages. If there are more than 2 controllers which are able to clear all 40 stages, the winner will be decided by the time left at the end of all 40 stages, the total number of kills, and final Mario state.

3.2.2 Turing Test Track

The "Turing test track" tries to find the most human-like controller. The assessment in the competition is done by a third person Turing-test since the game is a one person game (the judges cannot play the game with a computer player). The judges observe a pair of game-play videos (about 1 minute each). Afterwards, a post-experience questionnaire is presented to the judges. The questionnaire is as follows:

Q1 "Which do you think plays in a more human-like manner?"

Q2 "Which do you think is more expert?"

The possible answers for the judges are: Video A, Video B, both equally or none of them.

3.2.3 Level generation Track

The "Level generation track" is different from the previous two tracks. This track focuses on automatic stage design by a program for human players. The assessment of the track is done by human judges. First, each judge is given a test level to measure his/her skill of Mario playing. The game play is recorded and referred by the level generators of two competitors. The two level generators each generate a stage, then the judge plays and evaluates the two stages. The fun of the stage is ranked by the following question.

Q "Which game of the two is more fun to play ?"

The possible answer is forced to: Stage A or Stage B, following 2-AFC method[10].

3.3 A* Path finding algorithm

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

$$f(current, goal) = g(start, current) + h(current, goal)$$

$f(curren, goal)$ 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, Baumgaten presented a very efficient controller by using 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 very short time, because the performance is excellent, and also because the behavior is far from that of human players.

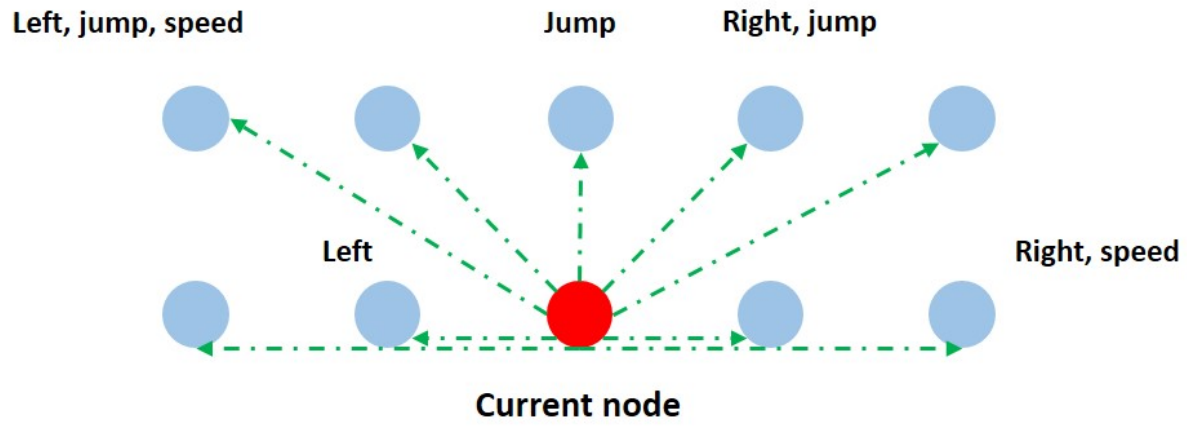

Figure 3.6: Possible nodes for A*

The algorithm expands the path by the 9 actions shown in Fig.3.6 (i.e. left, right, duck, jump, dash/fire). 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.

3.4 Human-like Behavior with Biological Constraints

Recently, an interesting approach was proposed to produce human-like behavior by introducing human mistakes to a computer agent. Fujii et al. proposed a human-like computer player with biological constraints. The constraints include "Sensory error", "perceptual and motion delay" and "Physical fatigue" (See Fig. 3.7).

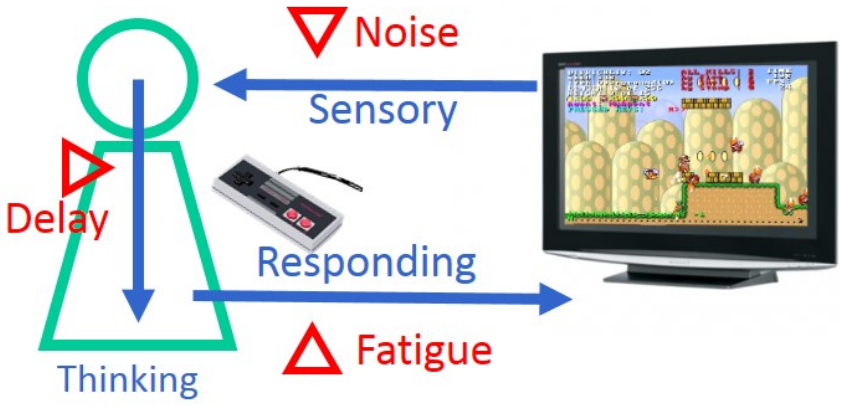

Figure 3.7: Fujii's model considering biological constraints.

"Sensory error"

Contrary to computer players, it is hard for humans to recognize the positions of objects exactly, especially while objects are moving. Humans show some mistakes from such sensory errors, and they can be imitated. The implementation of this error was done by adding Gaussian noise to the recognized positions of objects.

"Perceptual and motion delay"

This feature explains the delay between perception and response of the human body system. While playing computer games, human players first perceive the game information with their eyes, and then they spend some time for thinking, before finally responding, after for example 0.2 seconds. Then, a virtual delay was added into the agent controller, mainly for observation. Thus, the information that the agent observes at the current time will be a situation several frames ago.

"Physical fatigue"

The fatigue from incessant or repeated inputs is frequent for human players, especially untrained players. In order to imitate such fatigue, a virtual penalty is given to the agent when many keys are pressed in a row.

These constraints were applied to a Q-learning and A* algorithm and Infinite Mario Bros. was used as a test-bed game. It was shown that the proposed agent can be more human-like than both novice and expert human players.

Chapter 4

Methodology

In this chapter, we describe the research methodology in 5 sections, core concept, framework, local behavior models, transition model and assessment method.

4.1 Core Concept

In a modern style game, not only a single goal but many sub-goals are given and available for challenge. In the case of Super Mario Bros series, the major goal is to reach within a time limit the stage's goal located on the rightmost of the stage. Players have other optional tasks such as collecting coins or beating enemies, though they are not necessary for clearing the stage.

The player is able to challenge any goal that he prefers, but he has to respect the major goal. Thus, the player's behavior will have some transitions between several local behaviors. For example, in the beginning of the stage, player's movements are at ease, so he can enjoy collecting coins, or the player can control Mario very carefully when he encounters many enemies. After a while, when the time has almost run out his movement becomes faster and more risky in order to clear the stage in time.

Our research interest is to create a human-like computer player with transition of emotional behaviors. The usual practice in this area has been focused on the human-likeness of behavior in the whole game play, whereas our approach produces the transitions of multiple behaviors. Each behavior model produces a specific human-like behavior inspired by human emotions or feelings (e.g. anxiety, fear, greed), and the transition model then decides the appropriate timing to change the behavior, which looks like a human transition.

4.2 Research Framework

We propose a research framework which is described in 3 layers that are the "Behavior model", the "Transition" and the "Evaluation". The test-bed game of this research is Mario benchmark and the implementation was done on java environment. The Behavior model includes 5 elementary models (i.e. "Safety", "Hurry", "Greedy", "Enjoy", and "Meaningless"). Each of them was designed for simulating a specific behavior which is likely influenced by an emotion. In this article, we describe only 3 of the 5 models which are "Safety", "Hurry" and "Greedy". All the models are based on A* path finding algorithm so that it is easier to combine multiple behaviors.

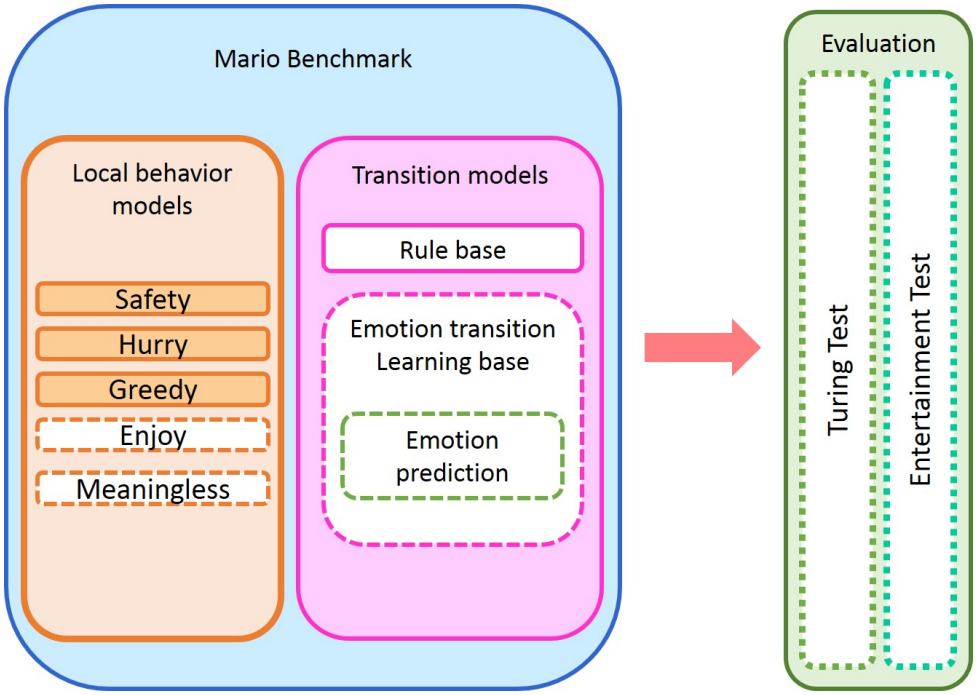

Figure 4.1: Research framework

The selection of a behavior model is done by the transition model. We propose two types of transition models, Rule based and Learning based to automate the transition between behavior models. The rule-based model decides the transition by considering the environment around the character without a learning algorithm, whereas the learning based approach aims to learn the transition triggers directly from human players. The selected model will process and return the actions to the game. In this article, only the Rule based approach will be described.

We consider two kinds of tests that should be done for evaluating our method, the Turing test and the entertainment test. The former will be conducted in order to assess the human-likeness of the model. The test uses post-experience questionnaires and follows the method of 2010 Mario AI Competition. The latter assessment aims to evaluate the ability to entertain players. The Turing test is able to evaluate the naturalness of computer

players from human perspective. However, the result of the test does not prove that the player is able to entertain humans. Thus, we propose the entertainment test to handle this problem.

4.3 Local behavior models

Our idea aims to produce human-like behaviors by simulating the transition of various different behavior models. Each model was designed by referring human actions inspired by a specific emotion such as “enjoy”, “fear” or “anxious”. We proposed 5 models including “Hurry model”, “Safety model”, “Greedy model”, “Enjoy model”, and “Habit model.” However, only the former three models will be described in this article. The implementation of each behavior model is hand-coded, in other words unsupervised, and based on the A* algorithm [7] which was explained in chapter 3.3.

4.3.1 Safety Model

Maslow explained the motivation of humans in a hierarchy of 5 layers of needs. The term “Safety” has been used to describe the needs of health, well-being, and also the safety against adverse impacts. Also, the need of “Safety” can influence the player 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 the evasion from an enemy by 1 pixel are possible. But, beginner and intermediate players are aware of their imperfect control and perception. Thus, safer movement such as keeping a distance to each enemy is preferred.

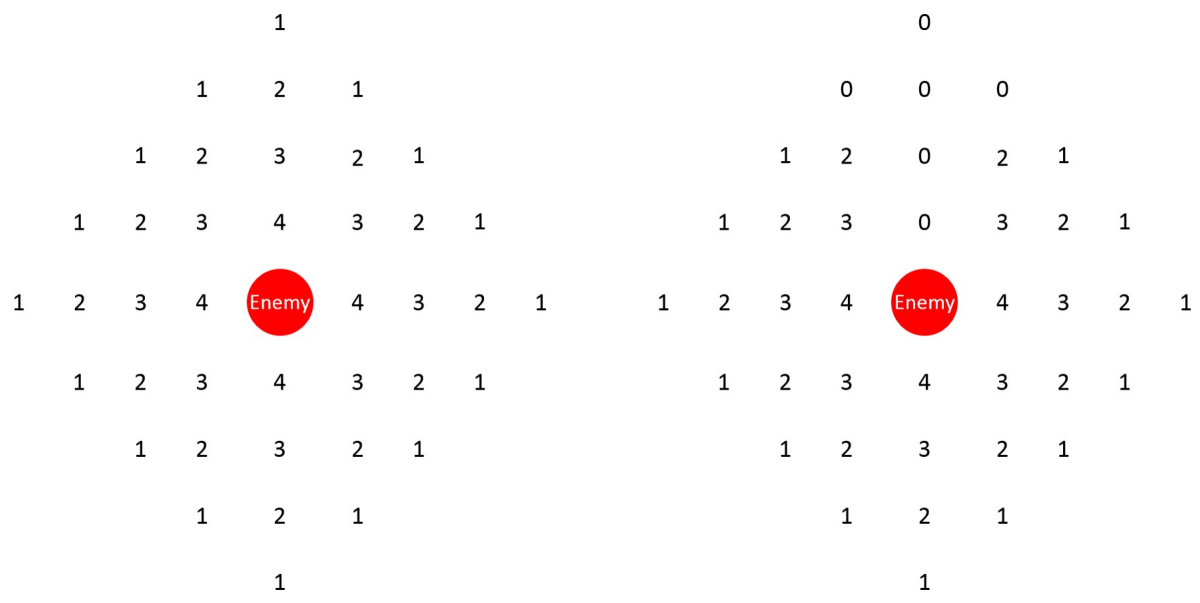

Figure 4.2: Dangerous area

The safety model imitates such behavior by introducing a “dangerous area” to the A* algorithm. The “Dangerous area” is surrounding each harmful object (Fig 4.2), hence the safety model controller intends to avoid the object and “nearby area”.

The heuristic function of A* algorithm is defined as:

$$h'(s_t) : S \to \mathbb{R} | h'(s_t) = RP_t + MP_t - h'(s_{t-1})$$

where s is the state of game at frame t, and 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 4.2.

There are many kinds of 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 4.2, the left one has an isotropic area, on the other hand, the right one has no virtual damage just over the enemy. This is because some enemies can be stomped on and in such case Mario is not damaged.

4.3.2 Hurry Model

The major goal of Super Mario Bros. is to clear the stage, travelling to the right most end of the stage without being killed. In the beginning of the stage, 200 seconds are given to Mario and the player has to clear the stage within the time limit. In Super Mario Bros, after 150 seconds have passed, there will be a warning sound, the time displayed will be shown in red color and the background song will become quicker. Afterwards, the player will be aroused and he will try to clear the stage as fast as possible. Sometimes the player might ignore the remaining coins or give up killing enemies. Or he might also ignore a damage that will not kill him immediately, when Mario is in "Fire" or "Big" state. Thus, we proposed "hurry model" to display such behavior.

The aim of this model is to clear the stage as fast as possible by ignoring other rewards such as coins or damages from enemies. Thus the design of hurry model is almost the same as the original A*. However, there is some difference. The heuristic $h(s_t)$ in the cost function is replaced by $h'(s_t)$ as follows.

$$h'(s_t) : S \to \mathbb{R} | h'(s_t) = \begin{cases} \gamma_s(RP_t + MP_t) - h'(s_{t-1}) & , EnemyNum \leq 2 \\ \gamma_i(RP_t + MP_t) - h'(s_{t-1}) & , EnemyNum > 2 \end{cases}$$

γ_s is the weight of damage penalty when the number of enemy EnemyNum is small or if it is easy to escape, while γ_i is the weight of damage penalty when there are groups of enemies and it is difficult to escape. Normally, γ_s and γ_i are setup in the range $0.7 - 0.8$ and $0.01 - 0.2$ respectively. It means that when there are many enemies, hurrying Mario will prefer being damaged than taking much time to escape.

When the player is anxious about time and hurry to clear the stage, it is not unusual that human recognition of the positions becomes imprecise. We imitate such human mistakes of perception into the model in order to display a more natural behavior. Perception mistakes are produced by adding Gaussian noise to Mario location in the input of the agent, as shown by Fujii et.al. [9].

4.3.3 Greedy Model

In Mario game, coins and items are rewards which give some benefit to Mario. Collecting coins give a score to the player, and for every 100 pieces of coin the player gains an additional life. Items give to the player a score with also status upgrade, Small to Big or to Fire. Sometimes the player attention might be drawn by these rewards. Our greedy model imitates such attention to rewards. This behavior reflects the enjoyment of humans when obtaining a benefit.

The main idea of the Greedy model is slightly different from the [Hurry] model. The goal of path finding is set to coin locations and item locations instead of the real goal (goal of the stage):

$$h'(s_t) : S \to \mathbb{R} | h'(s_t) = \sum \frac{1}{D_i^2}$$

where $\sum \frac{1}{D_i^2}$ is the summation of distance D_i from Mario to each coin i at frame t (See Fig). The main concept of the Greedy model is to sacrifice some time for collecting the rewards. The closer Mario come to the coin, the smaller $h(s_t)$ is.

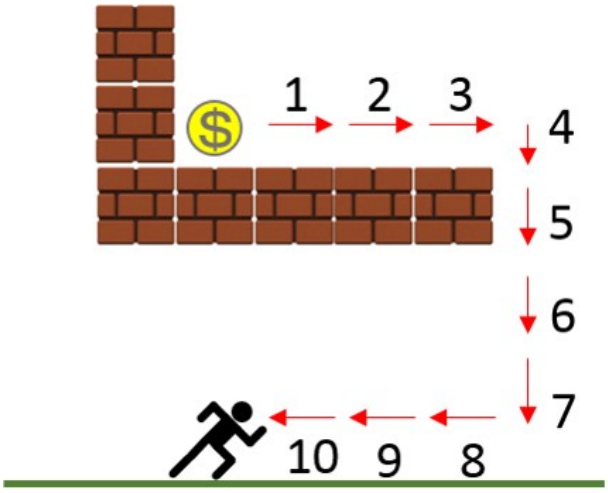

Figure 4.3: Distance from Mario to coin(D_i)

4.3.4 Enjoy Model and Habit Model

We proposed two other models, [Enjoy model] and [Habit model], but have not implemented them.

[Enjoy Model]

Sometimes the player might face some challenging situations which are not necessary to solve in order to clear the stage. But the player might enjoy such situation. In the case of Mario game, if the player is able to continuously stomp on enemies without falling to the ground, the score will be increased twice for each kill. The challenge may give a big score but it does not matter in order to clear the game. We present interest and enjoy emotion in this model.

[Habit Model]

We found that for some behaviors of humans, it is not possible to identify the purpose or even the reason of the behavior. Often, human players show some actions which have 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 an instinct or sometimes with the player's intention. We call such behavior as [Habit mode].

4.4 Transition Model

We conduct this research under the hypothesis that the transition of local behaviors is a key to produce human-like behavior. In section 4.3 we present "how to produce specific human-like behavior" by using 5 models which are inspired by human emotions. However, the production of natural changes between models is needed in order to produce human-like behavior.

The transition model has to decide whether "at the present time the behavior should be changed or not." We call this decision a "transition trigger."

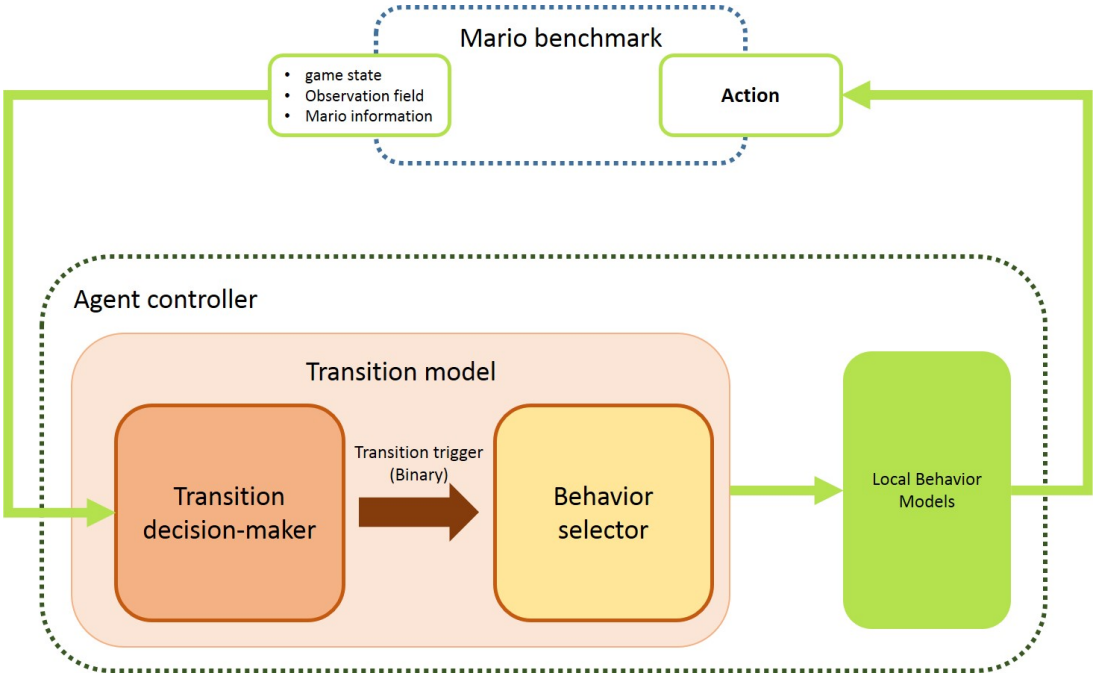

Figure 4.4: Transition model

The transition model will receive inputs from the Mario AI benchmark via the agent interface class. The inputs contain the observation field, the number of enemies, the state of Mario and time left. Once the inputs are sent to the transition model, the "transition decision maker" processes the transition trigger. Afterwards, if the behavior should be changed, one local behavior model will be selected and returns an action to the game.

In the rule based model, the current state of the game will be checked for deciding the transition according to some if-then rules. We describe the rule based model with the following state transition machine (Figure 4.5).

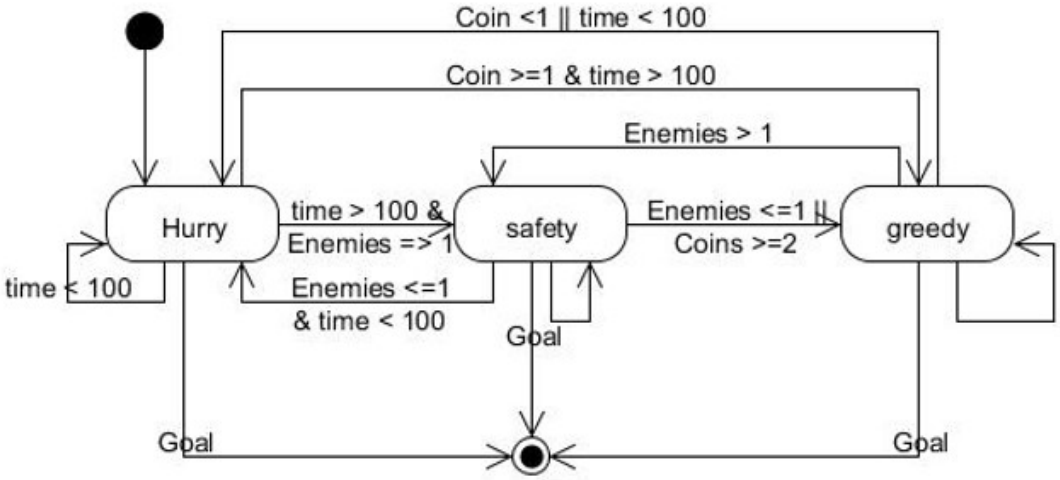

Figure 4.5: State machine diagram of rule based model

At the beginning, the computer player starts in the hurry state. If he encounters one or more enemies and the time limit is still over 100 seconds, the model will automatically switch to the safety model. Or if the time is less than 100 seconds and the number of enemies is less or equal to one, the model will switch back to the hurry model. Similar to the safety model, if there are more than 2 coins available for capture, the player will switch to the greedy state. And if enemies are encountered, the model will switch back to the safety model.

Since such rules are hand-coded, there are maybe not accurate. Then, a learning algorithm should be employed in a near future. We have 5 local behavior models, then we can estimate which part of human actions is based on the hurry model, or the safety model etc. Then, we will be able to learn also the timing when the transition should be done between these models.

4.5 Assessment method

The evaluation in this research is conducted in two directions.The Turing test is used to evaluate the human-like behavior of the model. The test follows the method of Mario AI Championship 2010 as follows (See Fig 4.6)

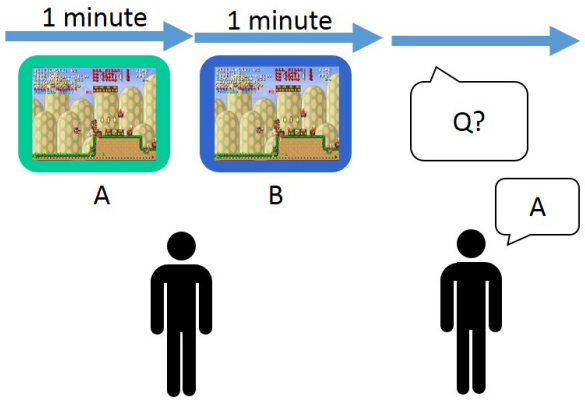

Figure 4.6: Subjective evaluation method

- Three types of video (1 minute) including, our computer player, a beginner player and an intermediate player, will be prepared.
- Human subjects are asked to observe pairs of videos randomly selected from the prepared videos.
- Just after, the post-experience questionnaires are given. The questions are as follows:
  - Q1. Which do you think plays in a more human-like manner?
  - Q2. Which do you think is more expert?

  The possible answers are forced into the 4 following elements as the 4-AFC (4 alternative forced choices) method, Video A, Video B, Both equally, none of them.
- Furthermore, we try to assess the ability to entertain players. Thus we add 3 questions:
  - Q3. Which do you think is more entertaining?
  - Q4. Which do you prefer to play with?
  - Q5. From which do you feel more/clearer emotions?

Chapter 5

Experiment Result

In this Chapter we describe the current results of this research. Before conducting the Turing test, we have to confirm if the naturalness of the behaviors generated by our method is at a sufficient level or not. So, as preliminary experiments, we compared the three behaviors of our computer player, the original A* algorithm and a human beginner. The result of each model will be shown individually.

5.1 Safety Model

The core concept of the safety model is to use a "dangerous area". On Figure 5.1, we show the dangerous area (approximately) by a red color. The computer player tries to avoid the red area, as if a virtual big enemy existed, though in fact it is not dangerous. On the other hand, the A* player goes closer to any enemy without hesitation, and suddenly avoids the enemy by a few pixels (Fig 5.2).

Figure 5.1: Safety model in simple avoidance.

Figure 5.2: A* in simple avoidance.

We show another example in a more complex situation in Figure 5.3.

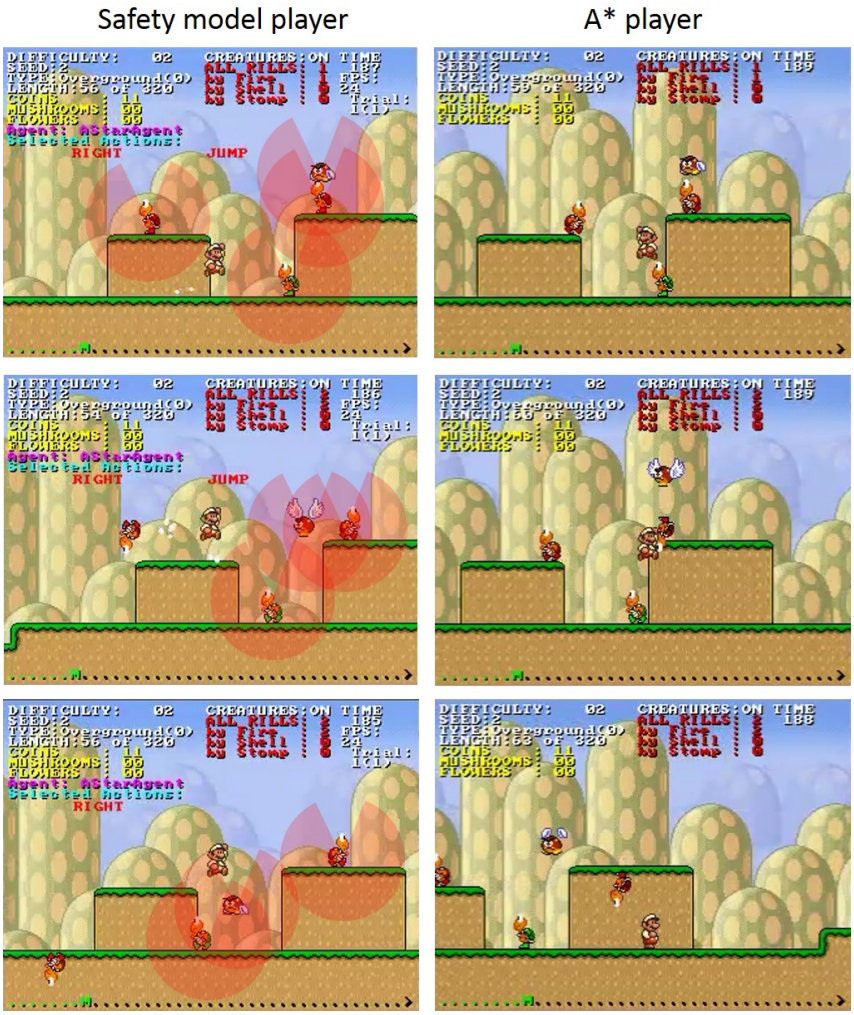

Figure 5.3: Safety model player VS A* player in complex situation.

When the safety model player encounters 4 enemies, some hesitation to move forward is observed. The computer player avoided the enemies at the front, stepped backward,

killed back-side an enemy and then decided to move forward again when a chance to avoid the dangerous areas was found. On the other hand, the A* algorithm selected the best but very risky route and moved fast forward in order to reach the goal as fast as possible.

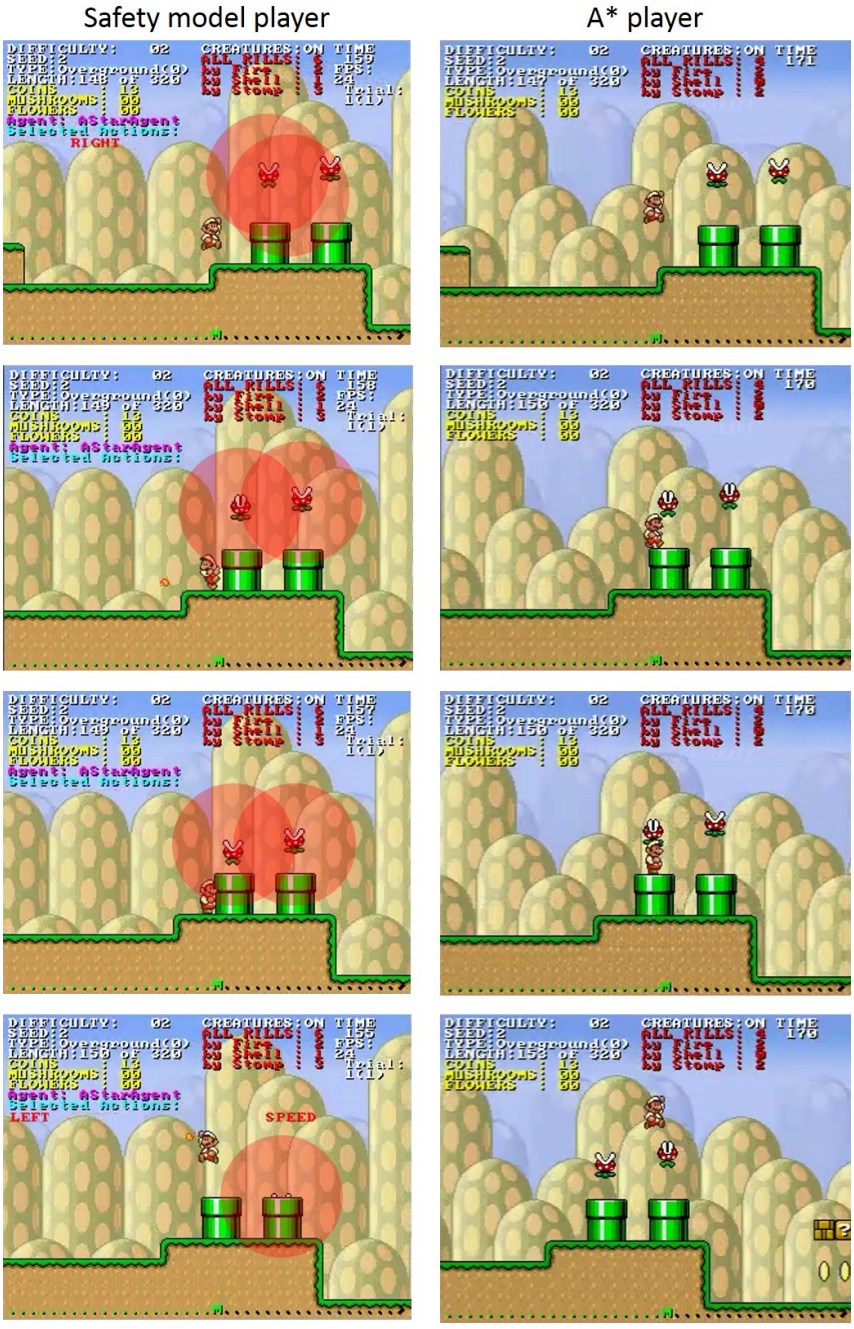

Figure 5.4: Safety model player VS A* player (non stomp-able enemies).

In Figure 5.4, we show the safety model player's behavior when he faces enemies that cannot be stomped on (i.e. that cannot be killed by jumping on them). Flower enemies

in Mario game rise from the pipes and then fall to the pipes slowly, and Mario will be damaged if touched. Usually, the player will wait until the flower falls down into the pipe. The A* player showed very skillful move to pass through the narrow space under the flowers (Figure 5.4, right). Such move is very risky and unnatural as a human behavior. On the contrary, the safety model player came next to the pipe and waited until the flower was gone, and then went through (Figure 5.4, left).

Up to this point, the safety model shows the desirable behavior which is natural for humans. It is sufficient enough to use in the further steps of this study.

5.2 Hurry Model

The behavior of the current hurry model is very similar to that of the A* player. However some points are different.

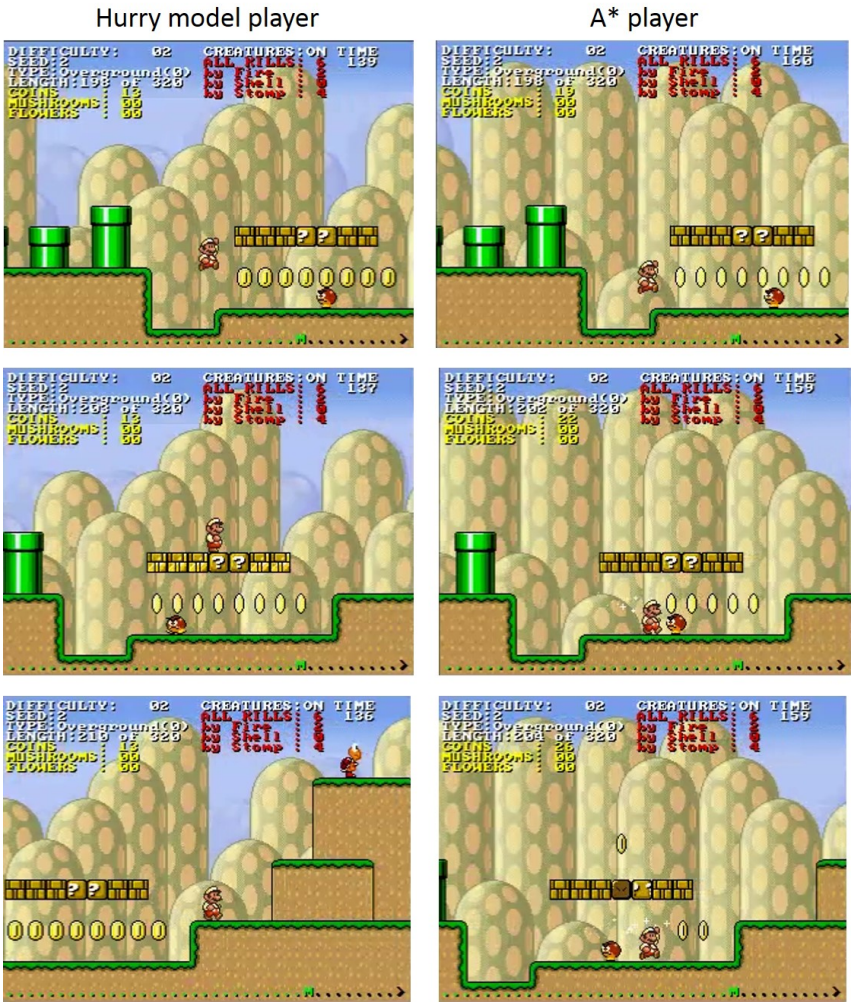

Figure 5.5: Hurry model player VS A* player.

Figure 5.5 shows how the Hurry model handled a situation where a coin and an enemy were in front of the player. For the A* algorithm, it was easy to move forward, collect the coins and move through (right of the figure). But the hurry model showed hesitation to risk the life of the player. The computer player chose instead to avoid the enemy even though he cannot collect the coins (left of the figure). Thus, we can say that the hurry model shows an intention to hurry the clearing of the stage, while also taking care a bit of dangers. In other words, "natural hurry behavior" is produced.

5.3 Greedy model

Among the 3 models, the greedy model is the most complex one. For now, we implemented the greedy model only for coins, not for killing enemies or collecting items.

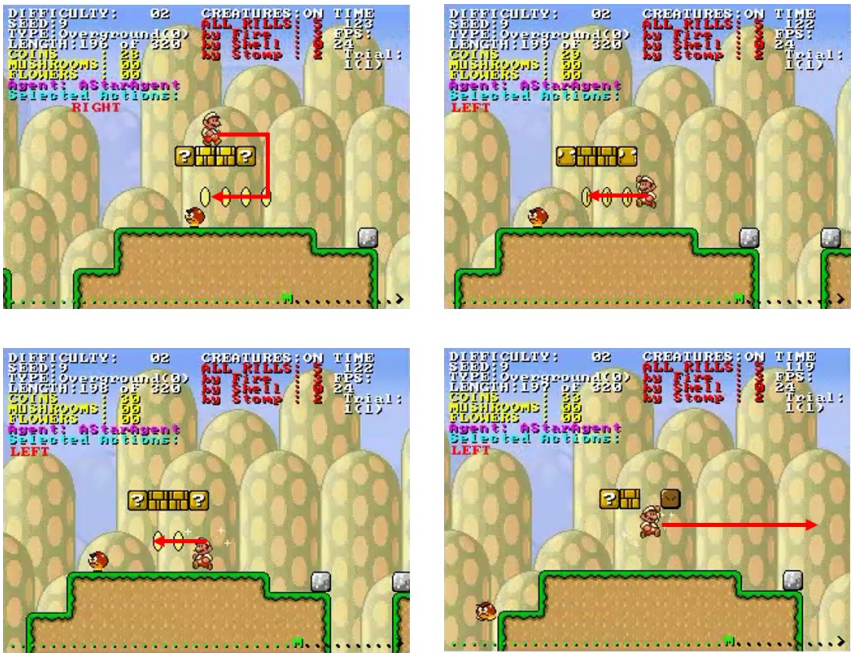

Figure 5.6: Greedy model collecting coins.

In Figure 5.6, we show a situation where Mario just avoided an enemy and its current location is above a set of coins. Mario planned to collect the four coins without conflict with the enemy, by sacrificing some of the given time. Such behavior is very natural among human players.

We showed 3 kinds of models in section 5.1, 5.2 and 5.3, and they successfully produced natural behaviors that seem to be inspired by emotions. We have already prepared a rule-based transition model. However, to combine the three models into the transition model, some modifications are needed, so we left this approach as a future work.

Chapter 6

Discussion

We proposed a computer player with five specific switchable behaviors and a method to trigger the transition between them in a human-like manner. Currently, we have obtained three sufficient behavior models which are able to reflect human emotions and feelings. First, the model of "Safety" shows the need of safety. It represents the anxious feeling and fear of humans. Secondly, the model of "hurry" shows the anxiety of humans under limitation of time. And last, the model of "Greedy" shows the enjoyment of humans when they try to get a reward. Up to this point, we prepared a transition model to be applied to these three models. However, because of the technical difficulty and the limited time, we left this part as a future work. In any case, it is confirmed that a computer player can produce emotional human-like behavior.

In a very near future we plan to conduct a subjective assessment of each behavior model and the rule-based transition model. Afterwards, we plan to implement the remaining two behavior models and then use all 5 behavior models in a learning-based transition approach.

Bibliography

[1] A.P. Saygin, I. Cicekli, V. Akman *Turing Test: 50 Years Later* Minds and Machine, Kluwer Academic Publishers, 2001

[2] D. Canamero *A hormonal model of emotions for behavior control* VUB AI-Lab Memo, Vol.2006, 1997.

[3] H. Desurvire, K. Jegers, C. Wiberg *Evaluating fun and entertainment: Developing a conceptual framework design of evaluation methods* Facing Emotions: Responsible experiential design INTERACT 2007 conference, 2007.

[4] J. Schrum, I.V. Karpov, R. Miikkulainen *UT$\hat{2}$: Human-like behavior via neuroevolution of combat behavior and replay of human traces* Computational Intelligence and Games (CIG), 2011 IEEE Conference on , pp. 329-336, 2011

[5] J. Togelius, G. N. Yannakakis, S. Karakovskiy, N. Shaker *Assessing Believability* Believable Bots, Springer-Verlag Berlin Heidelberg, pp. 215-230, 2012.

[6] J. Togelius, N. Shaker, S. Karakovskiy, G. N. Yannakakis *The Mario AI Championship 2009-2012* AI Magazine 34.3 pp. 89-92 2013

[7] J. Togelius, S. Karakovskiy, R. Baumgarten *The 2009 Mario AI Competition.* Evolutionary Computation (CEC), 2010 IEEE Congress on , pp.1,8, 18-23, 2010.

[8] M. Cambell, A. J. Hoane and F. Hsu, *Deep Blue* Artificial Intelligence, pp. 57-83, 2002

[9] N. Fujii, Y. Sato, H. Wakama, K. Kazai, H. Kitayose *Evaluating Human-like Behaviors of Video-Game Agents Autonomously Acquired with Biological Constraints.* Advances in Computer Entertainment, Springer International Publishing, 61-76, 2013

[10] N. Shaker, J. Togelius, G. N. Yannakakis, B. Weber, T. Shimizu, T. Hashiyama, N. Sorenson, P. Pasquier, P. Mawhorter, G. Takahashi, G. Smith, R. Baumgarten *The 2010 Mario AI Championship: Level Generation Track* Computational Intelligence and AI in Games, IEEE Transactions on , vol.3, no.4, pp.332,347, 2011

[11] N. Shaker, J. Togelius, G. N. Yannakakis, L. Poovanna, V.S. Ethiraj, S.J. Johansson, R. G. Reynolds, L. K. Heether, T. Schumann, M. Gallagher *The Turing Test Track of the 2012 Mario AI Championship: Entries and Evaluation* Computational Intelligence in Games (CIG), 2013 IEEE Conference on , pp.1,8, 11-13. 2013.

[12] P. Hingston *A Turing Test for Computer Game Bots* Computational Intelligence and AI in Games, 2009 IEEE Transection on, Vol.1, No.3, pp.169-186, 2009.

[13] P. H. M. Spronck *Adaptive game AI* UPM, Universitaire Pers Maastricht, 2005

[14] R. Claessens *Mario AI-gameplay track developing a Mario agent* 2012

[15] S. Karakovskiy, J. Togelius *The Mario AI Benchmark and Competitions* Computational Intelligence and AI in Games, IEEE Transactions on , vol.4, no.1, pp.55-67, 2012

[16] U. Ritterfeld and R. Weber *Video games for entertainment and education* Playing Video Games. Motives, Responses, and Consequences. Mahwah, NJ: Lawrence Erlbaum Associates, pp 399-413, 2006.