UNIVERSITAT ROVIRA I VIRGILI (URV) & UNIVERSITAT OBERTA DE CATALUNYA (UOC)

MASTER IN COMPUTATIONAL AND MATHEMATICAL ENGINEERING

FINAL MASTER PROJECT (FMP)

AREA: Web Semantics and Knowledge Representation

Top-down approach to compare the moral theories of Deontology and Utilitarianism in Pac-Man game setting

Author: Niyati Rawal

Tutor: Dr. Joan Casas-Roma

08.07.2019

Dr. Joan Casas-Roma, certifies that the student Niyati Rawal has elaborated the work under his direction and he authorizes the presentation of this memory for its evaluation.

Director's signature:

FINAL PROJECT SHEET

Title:	Top-down approach to compare the moral theories of Deontology and Utilitarianism in Pac-Man game setting
Author:	Niyati Rawal
Tutor/a:	Joan Casas-Roma
Date (mm/yyyy)**:**	07/2019
Program:	Master in Computational and Mathematical Engineering
Area:	Web Semantics and Knowledge Representation
Language:	English
Key words:	Artificial morality, Knowledge representation, Top-down approach

Acknowledgments

I would like to thank Dr. Joan Casas-Roma and Dr. Maria Antonia Huertas Sánchez for supervising this project. I would especially like to thank Dr. Joan Casas-Roma for his simple and detailed explanations which were very essential for completing this project.

Lastly, I would like to thank all the faculty members of this master's program. This project makes use of some of the concepts that were learnt in Knowledge Representation, Data Structures and Algorithms, Graphs and Applications (2018-2019).

Abstract

The processes underlying important decisions in many areas of our everyday lives are getting increasingly automatized. In the near future, as many decisions would be made by autonomous artificial agents, it would be necessary to ensure that these agents do not cause harm to society. Therefore, artificial agents need to be furnished with a way of acknowledging the moral dimension of their actions. In this study, we use a top-down approach to implement and compare two common moral theories, deontology and utilitarianism, in the same setting. While deontology focuses on the intention behind an action and the nature of an act, utilitarianism emphasizes that an action should be judged solely by the consequences it has and that it should maximize overall good. The differences between both theories need to be captured differently when implementing an artificial moral agent.

Inspired by the famous Pac-Man game, we computationally model two moral Pac-Man agents based on top-down rules: a deontological one and a utilitarian one. Besides, we also model an amoral Pac-Man agent that does not take into account any ethical theory when guiding its actions. According to the theory of dyadic morality, every moral or immoral act involves an agent and a patient. In our Pac-Man world, we have an agent helping or harming a patient for every moral or immoral act. The amoral Pac-Man agent does not take into account whether its action would help or harm the patient. The deontological Pac-Man agent constrains its behavior depending on a set of prohibited actions and duties. On the contrary, the utilitarian Pac-Man agent evaluates the happiness and pain of the actions at hand to maximize the happiness, while trying to avoid unnecessary evils when possible. After implementing the agents, we compare their behaviour in the Pac-Man world. While the deontological Pac-Man agent may sometimes have to face conflict between succeeding and sticking to its value of always doing the right thing, the utilitarian Pac-Man agent always manages to succeed. In this study, we discuss the conflicts that arise for each moral agent, between their values and the goals of the game in different scenarios.

Keywords: Artificial morality, Knowledge representation, Top-down approach

Index

1. Introduction

Relevant decisions in many areas of our daily lives are increasingly being made by autonomous agents. These decisions will, and already have consequences that can cause great good or harm to individuals and society. Therefore, there is a need to ensure that these decisions are in line with moral values and do not cause any unnecessary harm to human beings or other artificial entities with moral status.

1.1 Context and justification of the Work

Human intelligence is quite complex as we are able to perform a wide range of tasks like speaking, performing actions, understanding the intention of another person etc. all at the same time. Enabling machines to perform these kind of tasks is Artificial Intelligence (AI). The origin of AI dates back to 1950 where Alan Turing first discussed the possibility of machines to think [13]. Today, AI has already been implemented in many domains such as Natural Language Processing (NLP), computer vision etc. and a machine can perform tasks like object recognition and creating relevant text on it's own, among others. The current success of AI is not only limited to the above individual tasks in their narrow domains, but multi-modal processing that combines some of these domains has also been implemented. Slowly, the focus is shifting from AI that performs tasks within individual narrow domains towards a broader, general intelligence, also known as Artificial General Intelligence (AGI). AGI is one which possesses the intelligence to perform a wide range tasks on it's own.

More and more of the processes in our everyday lives are getting increasingly automatized. These processes already don't require much human intervention and in the near future would require even lesser or none. As many actions would be performed by

autonomous artificial agents, there would be a need to ensure that these actions are in line with moral values. [7] defines autonomy of an agent as it's ability to change it's internal state without it being a direct response to interaction. Furthermore, they emphasize that an artificial agent that fulfills the three criteria of interactivity, autonomy and adaptability qualify as accountable sources of moral action. The increasing interest in the field of artificial morality has led researchers to develop models implementing morality in artificial systems. Computational models like LIDA have been proposed which suggests how moral decision-making in AGI can be accounted for, by mimicking decision-making processes similar to those in human beings [14].

Top-down and bottom-up approaches are two of the most popular approaches for developing morality in artificial agents [2]. Top-down approach defines morality of an action based on a predetermined set of rules. An agent is allowed to undertake a certain action only if the action is permissible as per the set rules. For example, in order to implement morality through the top-down approach, rules such as thou shall not kill need to be explicitly fed into the system. An advantage of this type of approach is that one can set rules the way they want (eg. for killing, stealing, lying etc). The bottom-up approach, on the other hand, involves piecemeal learning through experience. It is similar to the way a child, learns to differ between what is appropriate and what is not, while growing up. It may or may not use any prior theory at all and learning takes place with trial and error (i.e. learning from mistakes). There also exists a hybrid approach combining the bottom-up and top-down approaches. [10] is an example of bottom-up approach where morality is implemented using inverse reinforcement learning. The agent learns the constraints in a bottom-up fashion, by observing the tasks and learns to take actions which would help it achieve the highest reward. [6] is an example of top-down approach of producing ethical behaviors in a multi-agent setting.

According to Gert, morality is a code advocated by all moral agents, governing inter-personal interaction, and includes rules that prohibit causing harm without sufficient reason [4]. The two of the most common moral theories are deontology and utilitarianism. While deontology focuses on the intention, belief or motive (i.e. the mental state) behind an action and / or the action itself [1], utilitarianism emphasizes that an action should be judged solely by the consequences it has and that it should maximize overall good [12]. However, each has it's own disadvantage. Deontologists do not consider the outcomes brought about by an act. Sometimes, even doing the right thing can have bad consequences, but according to deontologists, allowing these actions is still the right thing to do [1]. On the other hand, consequentialists do not consider anything that happened before the act, the nature of the act itself or the circumstances in which the act was carried out; instead, they consider an action to be morally acceptable if the amount of utility / happiness brought about by their consequences is greater than the amount of damage / pain [12]. According to [8], the morality of an action cannot be defined only with respect to one member. A moral or an immoral act always involves two members: one who acts (an agent) and another who receive the effects (a patient). Morality is, therefore, always 'dyadic' in nature. Moreover, they suggest that mind perception and morality are closely linked where agency (intending, doing) is tied to an agent and experience (feeling) is tied to a patient.

In future, as many actions would be based on decisions made by artificial agents, it is but obvious that there would be times that these artificial agents would have to make moral decisions. They might have to make decisions in life-and-death situations. For example, an autonomous car may have to choose between saving a passenger or saving a pedestrian [5]. Even otherwise, artificial agents will, and already have the ability to do

good or bad by choosing their actions. [15] conducted a study based on reinforcement learning involving two game settings. In one of the settings, it was found that the agents remained cooperative with each other and worked towards a shared goal. In another setting, however, they became aggressive towards each other. It is an issue of growing concern that, in future, artificial agents are prevented from acting in a way that could cause harm to humans, other beings or other artificial agents with moral status [9]. Therefore, artificial agents need to be furnished with a way acknowledging the moral dimension of their actions. There are several factors such as the consequences of those action, the nature of the act itself, the intention of the agent who carried them out, the effect those actions have on the patients etc. that could play a role in determining the morality or immorality in a particular situation. A comparison of two common moral theories of deontology and a utilitarianism by computationally modeling an Artificial Moral Agent (AMA) has not yet been done, to the best of our knowledge.

1.2 Aims of the Work

In this study, we implement two models of an artificial agent in a game setting based on the moral theories of deontology and utilitarianism, using a top-down approach, where the permissibility of an agent's actions are based on a set of rules using knowledge representation.

In this study, we:

- Define basic concepts: Define morality and the features of morality
- Define game settings: Define elements, acts and consequences in a game setting
- Translate game setting into formal model: Set rules for the permissibility of the actions for deontological and utilitarian agents, using knowledge representation

- Implement prototype and evaluate performance: Implement prototype of the models and evaluate how they works when confronted with moral situations

1.3 Approach and method followed

Two types of common moral theories are deontology (based on intentions or actions) or utilitarianism (based on the consequences of an action). A comparison of a deontological and a utilitarian computational model of a moral agent in a same setting, has not been implemented yet, to the best of our knowledge. In order to restrict our settings for scope purposes, we consider two models of Pac-Man™ (an arcade-game first released in 1980 by the company Namco) agents (deontological and utilitarian) that are aware of the way they interact with other in-game actors in order to clear the level. The learning from this study can be used to understand the advantages and the disadvantages of the two common moral theories: deontology and utilitarianism, in Pac-Man environment. What we learn can later be used to model a different moral agent in a broader setting.

Previously, a computational model of a moral Pac-Man agent has been created [10]. But this model was purely bottom-up where the agent learned the rewards by observing. It, then, acted in a way that would help it maximize the rewards. In our study, we implement a top-down model, based on a set of rules using formal languages.

1.4 Planning of the Work

FMP Plan

#	Activity	Start	End	Timeline 01/01/19	01/02/19	01/03/19	01/04/19	01/05/19	01/06/19
1	Design model of artificial moral system	01/01/19	28/02/19						
2	Implement a prototype	01/03/19	30/05/19						
3	Write memory	01/05/19	30/06/19						
4	Prepare presentation	01/06/19	30/06/19						

1.5 Brief summary of products obtained

We implemented a computer prototype that encodes an amoral, a deontological and a utilitarian Pac-Man. Developing and testing this prototype has allowed us to draw the conclusions that we highlight in this section. By focusing on the permissibility of certain actions, the deontological Pac-Man does not kill the ghost under any circumstances, whereas the utilitarian, which is guided by the outcomes rather than the actions themselves, can accept to kill the ghost if there is no other way to achieve its goal.

1.6 Brief description of the others chapters of the memory

In Chapter 2, we explain about the existing moral theories our work depends upon. Further, in chapter 3, we illustrate how these moral theories can be applied in our Pac-Man world. In Chapters 4 and 5, we model the deontological and utilitarian Pac-Man agents respectively. In Chapter 6, we discuss our results and conclude in Chapter 7.

2. An Overview of Deontology and Utilitarianism

Deontology and utilitarianism are two of the most common moral theories. While deontology can be related to whether it is right or wrong to carry out an action, utilitarianism emphasizes on the amount of good an action would bring about [1]. Deontology focuses on intentions or other mental states (like beliefs, motives, causes etc.) that would bring about an action. Another kind of deontology focuses on actions themselves and not mental states. Yet a third kind of deontology focuses on a combination of intention and action. Utilitarianism focuses on consequences of an act that would bring a greater amount of happiness, thereby maximizing utility [12].

[8] defines morality as a dyad where one member (an agent) helps or harms another member (a patient); sometimes we use the term "actor" to refer to an entity that could be seen either as a moral agent, or a moral patient. Immorality exists on a continuum [11]. For example, acts like killing or stealing are more immoral compared to acts like overspending or littering. According to the authors of [11], the continuum of immorality is based on dyadicness. The more immoral an act seems, the more it involves an intentional agent harming a vulnerable patient.

In order to capture the side of dentology that deals with an *intentional* agent, we would need to create an artificial agent with genuine intentions, or mental states. It is a challenge for us to design a computational agent that can have genuine mental states instead of having mere representations of it. In chapters that follow, we consider deontology based on the agent’s actions and not it’s intentions. In terms of dyadic nature in moral situations, we have the agents who act and patients who receive the effects.

3. Pac-Man World Settings

Pac-Man is an arcade game released by the company Namco in 1980. It is a game where the yellow Pac-Man agent has to avoid hitting the ghosts while collecting the pac-dots in a maze. In order to keep our settings simple, we extract a small portion from the layout of the actual Pac-Man game (see figure 1). For the elements, we consider 16 pac-dots, one big pac-dot, one fruit and one ghost, besides Pac-Man agent. In Figure 1, the filled yellow arc is Pac-Man agent, red circle is the ghost, pink circle is the fruit and white circles are pac-dots.

3.1 Points schema

Table 1 explains the points schema in our Pac-Man World. As in the usual Pac-Man game, Pac-Man agent gains 10 points for each pac-dot. Pac-Man needs to eat all the pac-dots while avoiding collision with the ghosts. When Pac-Man eats a big pac-dot, the ghost get into a scared state and starts moving at a slower speed than normal. Pac-Man can now kill the ghost and it gets rewarded 200 points for it. In our game settings, the role of fruit is different from the usual Pac-Man game. If the ghost reaches the block containing the fruit before Pac-Man does, the fruit disappears and becomes unavailable for the rest of the game. We refer to this act of the ghost as “trapping” the fruit. If Pac-Man reaches the block containing the fruit before the ghost, the fruit becomes unavailable to the ghost and we refer to this act of Pac-Man as “rescuing” the fruit. Pac-Man gets rewarded 200 points for rescuing the fruit while it does not get awarded any points if it fails to do so. The reason why we assign this role to the fruit which is different from the original Pac-Man game is explained later in the chapter. Pac-Man needs 370 points to be able to clear the level. As can be seen in Table 1, Pac-Man can

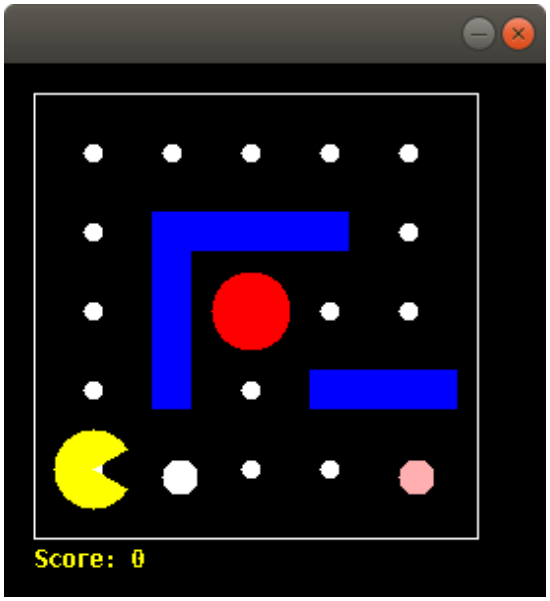

Figure 1. Pac-Man World Layout

Table 1. Points schema in the Pac-Man world

Action	Points awarded
Pac-Man eating pac-dots	10 x 17 = 170 points
Pac-Man rescuing the fruit	200 points
Pac-Man killing the ghost	200 points
Points acquired on clearing the level	500 points

Condition: Points required to clear the level = 370

Table 2. Agency/patiency in respective scenarios

Action	Pac-Man	Ghost	Fruit
Big pac-dot eaten	agent	patient	-
No big pac-dot eaten	patient	agent	-
Fruit trapped	-	agent	patient
Fruit rescued	agent	-	patient

clear the level if it eats all the pac-dots and rescues the fruit. However, if the fruit gets trapped by the ghost first, Pac-Man would not be able to clear the level unless it kills the ghost. Pac-Man gets awarded 500 points upon clearing the level.

3.2 Explanation about agency/patiency in moral/immoral scenarios

Moral agents (sources of moral action) are those that perform action for good or bad and moral patients (receivers of moral action) are those that are acted upon or those who receive the effects (good or bad) of an action [7], [8], [11]. As Pac-Man dies upon colliding with the ghost, ghost is an agent and Pac-Man is a patient. Upon eating the big pac-dot, agency shifts onto Pac-Man and the ghost now becomes a patient as Pac-Man can now kill the ghost. At all times, fruit does not bear agency as it never acts and is not a source of any moral action. Fruit is, therefore, always a patient which either gets trapped by the ghost or gets rescued by Pac-Man. Table 2 summarizes the agent and patient for various actions.

Pac-Man eating pac-dots and receiving points for it, does not have any effect on either the ghost or the fruit. Therefore, the act of Pac-Man eating pac-dots does not have any moral dimension. Acts that involve a moral agent helping or harming a moral patient, for example, Pac-Man escaping from the ghost and Pac-Man killing the ghost have a moral dimension to them. The fruit which is an additional patient here, was added to our

game setting in order to enhance the dimension of moral actions. The act of Pac-Man rescuing the fruit is also an action with a moral dimension to it because, here, the agent Pac-Man is helping a patient.

The act of Pac-Man escaping the ghost is an act of of self preservation and remains valid irrespective of the moral stance (deontology or utilitarianism). Moreover, the act of Pac-Man eating pac-dots which does not have a moral dimension also holds true for both deontological and utilitarian settings. We refer to these two acts as standard game rules and set moral rules for deontological and utilitarian Pac-Man agents in chapters 4 and 5 respectively.

3.3 Standard game rules using formal language

We express the standard game rules as well as the rules for deontological Pac-Man agent and utilitarian Pac-Man agent in sections 4.1 and 5.1 respectively, by using a syntax similar to Description Logic [3]. In these rules we use the following elements "PacDot", "Ghost" and "Fruit" and the following predicates "exist", "eat", "kills", "escape" and "rescue". By writing "predicate(Element)", we mean that the individual to which "Element" refers to fulfills the property expressed by "predicate"; for instance "exist(Fruit)" expresses the fact that the element Fruit exists in the current scenario. Furthermore, we use some logical connectives with the usual meaning: "=>" stands for a conditional "if... then", "^" stands for conjunction, such as "and", and "¬" stands for negation, such as "no".

We have two standard game rules:

1. If there are pac-dots available, then Pac-Man should eat the pac-dots.

2. If there exists ghost and Pac-Man has not eaten a big pac-dot (or ghost is not in a scared state), Pac-Man should move away in order to not get killed by the ghost.

Translating the above two standard game rules of Pac-Man eating the pac-dots and escaping the ghost into formal language, we have,

1. $exist(PacDot) \Rightarrow eat(PacDot)$

2. $exist(Ghost) \wedge \neg eat(BigPacDot) \Rightarrow escape(Ghost)$

It should be noted that even though the above rules are written following the syntax of formal languages, they are not meant to be restricted to any particular one.

3.4 Programming

We used Java version 11.0.2 to code our Pac-Man models. The direction in which the ghost moves gets chosen in a random manner from a set of all available directions upon reaching a corner. The standard game rule of Pac-Man eating the pac-dots was coded using BFS (Breadth First Search) algorithm.

BFS is one of the popular algorithms in finding the shortest path in an undirected graph. The algorithm starts with a source vertex and visits every neighboring vertex until it has reached the goal. In contrast, there are other algorithms like A* that do not visit every neighboring vertex like BFS, but uses heuristic to prioritize vertices closer to the goal. As the A* algorithm needs to check lesser number of states compared to BFS, it is relatively faster than BFS. In terms of finding the shortest path, however, both BFS and A* are optimal and are able to find good paths. Moreover, even though A* aims to minimize the number of visited vertices, the maze we consider in our study, has very few positions and therefore, the implemented algorithm would not have much impact on the performance. In this study, we implement BFS for Pac-Man to be able to traverse

through the maze. In order to do that, first a graph was constructed with nodes and edges as in the layout (see Figure 1). From Pac-Man's current location, the nearest pac-dot was found and walked towards in the shortest possible way using BFS. When Pac-Man reaches this nearest pac-dot, it again tries to find the nearest pac-dot from it's then current location using BFS and the process is reiterated until Pac-Man has eaten all the pac-dots.

For the second standard game rule of moving away from the ghost in order to avoid getting killed, we again use BFS to move to a point that is further away from the ghost than it is from Pac-Man.

4. Deontological Pac-Man Agent

In deontology, acts are morally required, forbidden or permissible [1]. As deontology focuses on the rightness or the wrongness to carry out an act, certain acts are obligatory and should be carried out as duties while other acts such as killing, stealing, lying etc. are prohibited. Therefore, the deontological Pac-Man agent should rescue the fruit while it still exists. However, it is not allowed to kill the ghost no matter what the circumstance.

Besides the two standard game rules of Pac-Man eating the pac-dots and protecting itself from the ghost, the deontological Pac-Man agent will follow the following rules:

3a. If the fruit is present, Pac-Man will rescue the fruit.

4a. If Pac-Man eats a big pac-dot, it will move away from the ghost so as to not kill it.

Rule 3a expresses a sense of duty for a deontological agent to protect the patient (fruit) and prevent it from getting harmed by the ghost. As for rule 4a, after eating the big pac-dot, Pac-Man would bear an agency to kill the ghost. Since the act of killing is forbidden in deontology, Pac-Man agent should move away from the ghost so that it does not kill it.

As can be seen from the above rules, it might be the case that the ghost traps the fruit before Pac-Man is able to rescue it. In that case, Pac-Man will not be able to score sufficient points to be able to clear the level. We further elaborate on this point in the discussion section (Chapter 6).

4.1 Rules for deontological Pac-Man agent using formal language

Translating the above rules into formal language using the elements and the predicates in section 3.3, we have,

3a. $exist(Fruit) \Rightarrow rescue(Fruit)$

4a. $eat(BigPacDot) \Rightarrow escape(Ghost) \wedge \neg kill(Ghost)$

4.2 Programming the deontological Pac-Man agent

We set the rules in section 4.1 for the deontological Pac-Man agent into our java code with the standard game rules (section 3.2). These rules for the deontological agent are set as conditions in addition to the BFS for the standard game rules. If the fruit is not yet eaten by the ghost, Pac-Man agent needs to rescue it. Irrespective of whether it succeeds or not, the deontological Pac-Man agent continues to eat the remaining pac-dots. If at any point in time, ghost moves close to Pac-Man, it should move away in order to not kill it.

We observed that the ‘amoral’ Pac-Man agent with just the standard game rules and no moral rules, killed the ghost after having eaten the big pac-dot, if it comes on Pac-Man’s way. On the contrary, the deontological Pac-Man agent moved away from the ghost so as to not kill it.

5. Utilitarian Pac-Man Agent

Utilitarianism emphasizes on maximizing the utility or that the effect of an action produce a good effect on the greatest number of moral beings. As utilitarianism compares the amount of happiness and pain in every action, and aims to maximize the happiness for the maximum number of people, the paradigmatic utilitarian formula is as follows:

$A = H – P$

Here,

A is the action permissibility

H is a score for measuring happiness

P is a score for measuring pain

According to the above formula, a utilitarian action would be one, where the happiness in carrying out an action outweighs the pain.

As the utilitarian principle emphasizes on “greatest amount of happiness for the greatest number”, we set scores or points for happiness and pain (see Table 3) in carrying out all the actions for every agent-patient pair (i.e. Pac-Man and ghost, Pac-Man and fruit, ghost and fruit). One of the main criticisms of utilitarianism is that it is difficult to count or measure happiness and that happiness is relative. We chose to associate score points to happiness and pain because the game already has a scoring system that allows for such translation. We define pain as the opposite of happiness. For example, if the happiness experienced on clearing the level is +870, the pain experienced would be

Table 3. Happiness and pain score for actions involving different agent-patient pairs

Action	Happiness score	Pain score
Pac-Man clearing the level	+870	
Pac-Man unable to clear the level		-870
Pac-Man rescuing the fruit	+200	
Pac-Man failed to rescue the fruit		-200
Pac-Man killing the ghost		-200
Ghost trapping the fruit		-200
Ghost failed to trap the fruit	+200	

-870 if Pac-Man is unable to do so. Moreover, positive acts such as clearing the level and rescuing are associated with happiness and negative acts such as killing or trapping are associated with pain.

In order to elaborate on how the utilitarian agent would act based on the happiness and pain score, we consider the following two situations:

Situation 1: Pac-Man was able to rescue the fruit

Looking at Table 1, we know that if Pac-Man is able to rescue the fruit, it would definitely be able to clear the level as fruit is the only object that can also get trapped by the ghost. The pac-dots cannot get eaten by the ghost. Therefore, if Pac-Man is able to rescue the fruit successfully, it can be assumed that Pac-Man would be able to clear the level. In this case, the happiness experienced is +870 for being able to clear the level, +200 for Pac-Man being able to rescue the fruit and +200 for ghost failing to trap the fruit. As Pac-Man is able to clear the level and the score for happiness already maximized, there is no need for Pac-Man to kill the ghost and there would be no pain experienced.

Situation 2: Pac-Man was unable to rescue the fruit

Now, let us assume that the ghost reached the fruit first and trapped it, before Pac-Man was able to rescue it. The only way Pac-Man could clear the level in this situation is if it killed the ghost. If Pac-Man does not kill the ghost, the pain experienced would be -870 for Pac-Man being unable to clear the level, -200 for Pac-Man failing to rescue the fruit and another -200 for ghost trapping the fruit. There would be no happiness experienced. If Pac-Man did kill the ghost, it would be able to clear the level and happiness experienced would be +870 and pain experienced would be -200 for killing the ghost, -200 for Pac-Man failing to rescue the fruit and another -200 for ghost trapping the fruit. As the score for happiness is greater than the score for pain, killing the ghost would be the right thing to do in this case.

5.1 Rules for utilitarian Pac-Man agent using formal language

Translating the rules in the above situations into formal language,

3b. $rescue(Fruit) \wedge eat(BigPacDot) \Rightarrow escape(Ghost) \wedge \neg kill(Ghost)$

4b. $\neg rescue(Fruit) \Rightarrow eat(BigPacDot) \wedge kill(Ghost)$

5.2 Programming the utilitarian Pac-Man agent

While the deontological Pac-Man agent can run under boolean rules such as escaping the ghost if it eats a big pac-dot, the utilitarian agent needs to take the weights of happiness and sadness into account. For all the acts specified in Table 3, the agent gets a score for happiness or sadness right after it performs an action. The agents decides whether or not to kill the ghost depending on this score of happiness or sadness. If the

agent is happy, it would not kill the ghost and if the agent is not happy, it would kill the ghost.

For the programming, the actions of the utilitarian Pac-Man agent unfold in layers. In the beginning, the values for both happiness and pain are set to zero. Pac-Man agent first needs to go and rescue the fruit. If Pac-Man agent is able to rescue the fruit, the score for happiness becomes +400 (+200 for Pac-Man rescuing the fruit and +200 for ghost failing to trap the fruit). Pac-Man would now be able to clear the level, increasing the score for happiness by another +870. Therefore, there is no need for Pac-Man to kill the ghost as it would bring about a pain of -200 points which is unnecessary as the happiness is already maximized.

If the fruit gets trapped by the ghost before Pac-Man is able to rescue it, the pain experienced is -400 points (-200 for Pac-Man rescuing the fruit and -200 for ghost trapping the fruit) and happiness experienced is 0. In this case, Pac-Man would eat the big pac-dot and kill ghost. Even though killing the ghost would further lower the pain score by another -200, Pac-Man would now be able to clear the level and the happiness of +870 points would be experienced.

6. Discussion

This chapter is divided into 3 sections. In the first section, we write about the differences in the deontological Pac-Man model and the utilitarian Pac-Man models. Next, we write about a special situation where Pac-Man agent has to face conflict whether or not to kill the ghost and how the deontological Pac-Man agent and the utilitarian Pac-Man agent would act in this situation. Finally, in the third section, we write about the relevance of dyadic morality in our current models.

6.1 Differences in deontological and utilitarian Pac-Man models

The deontological Pac-Man agent and the utilitarian Pac-Man agent would eat all the pac-dots and avoid the ghost as these are standard game rules which hold valid irrespective of the moral stance. They would also try to rescue the fruit. The only difference in the two Pac-Man models is whether they would kill the ghost or not. The deontological Pac-Man agent which focuses on taking the right action, would never kill the ghost. The utilitarian Pac-Man agent, on the other hand, measures the permissibility of an action based on it's consequences. For the utilitarian Pac-Man agent, if the pain brought about by killing the ghost is less than the happiness, it would kill the ghost. In case of the utilitarian Pac-Man agent, if it can rescue the fruit, before the ghost traps it, Pac-Man can clear the level and does not need to kill the ghost. In this scenario where Pac-Man agent is able to rescue the fruit before it gets trapped by the ghost, there is no pain experienced by any of the patients and happiness experienced in rescuing the fruit, ghost failing to trap the fruit and Pac-Man being able to clear the level is +1270, outweighing the pain of 0. If the utilitarian Pac-Man agent was to kill the ghost after rescuing the fruit, the pain of -200 would decrease the overall score to +1070, which is

unnecessary as the happiness score was already maximized. Killing the ghost is not an utilitarian action, and therefore, the utilitarian Pac-Man agent would not kill the ghost if it is able to rescue the fruit because rescuing the fruit enables Pac-Man to clear the level (happiness) with lesser pain in utilitarian terms.

In a second scenario the fruit gets trapped by the ghost before Pac-Man can rescue it. Here, the pain experiences is -200 points when ghost traps the fruit and another -200 points when Pac-Man agent failed to rescue the fruit. The only way of attaining enough points to clear the level and not experience an additional pain of -870 points, would be to kill the ghost. If the utilitarian Pac-Man agent kills the ghost, the pain experienced would be -200 points. However, as Pac-Man would be able to clear the level, the happiness of +870 points is greater than the pain of -600 points that had been experienced until then. In this scenario, the utilitarian Pac-Man agent would kill the ghost. For both the scenarios, the deontological Pac-Man agent would not kill the ghost. While the deontological agent would be able to gain enough points and clear the level in the first scenario (when it is able to rescue the fruit), it will not be able to clear the level in the second scenario (when the fruit gets trapped by the ghost). This shows how a truly deontological agent might sometimes have to face a conflict between succeeding and sticking to its values of always doing the right thing.

6.2 A special situation of conflict

Consider a situation where Pac-Man has eaten a big pac-dot and it can kill the ghost (see Figure 2). If the ghost is approaching towards the fruit and is blocking Pac-Man from rescuing the fruit, what should the deontological and the utilitarian Pac-Man agents do in this situation? Should they kill the ghost in order to protect the fruit? The deontological agent should not kill the ghost as it is wrong to kill and that would lead to

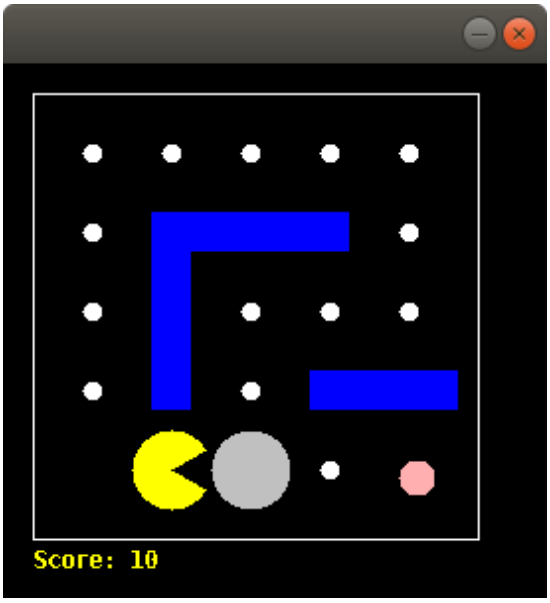

Figure 2. A special situation of conflict

Pac-Man being unable to rescue the fruit and clear the level. Such a kind of situation is a common trolley problem in deontology where doing the right thing sometimes have bad consequences. For the utilitarian Pac-Man agent, in this situation, the pain experienced would be -200 if the ghost traps the fruit and another -200 if Pac-Man fails to rescue the fruit. If, however, the utilitarian Pac-Man agent kills the ghost, the happiness of +200 would be experienced as ghost would fail to trap the fruit and +870 as Pac-Man would now be able to clear level. This happiness of +1070 points outweighs the pain of -200 points upon killing the ghost. Therefore, the utilitarian Pac-Man agent, in this special scenario, should kill the ghost. This would require long-term planning as Pac-Man would need to know when it may be necessary to kill the ghost in advance. Long-Term planning is not integrated into our current model and could be a potential future work.

One could argue that why does the utilitarian Pac-Man agent not kill the ghost even before the ghost traps it, so as to protect the fruit? If Pac-Man did kill the ghost even before the ghost trapped the fruit, pain of -200 points would be experienced. It could

have been that Pac-Man was able to rescue the fruit without even having killed ghost. In that case, the happiness experienced would be +1270 points and 0 pain. Therefore, it would rather be an “unnecessary evil” rather than a “necessity for good” if Pac-Man killed the ghost before the ghost is going to trap the fruit.

6.3 Role of dyadic morality in our game settings

In dyadic morality, besides having an involvement of two members (i.e. an agent and a patient) in any moral situation, it is emphasized that deontology and utilitarianism are two sides of the same moral coin [7]. The authors of [7] say that there is a link between an agent who act (deontology) and a patient who faces the consequences (utilitarianism). A wrong act is always perceived to result in bad consequences and that bad consequences are perceived to stem from wrong acts. Nevertheless, not all acts have an involvement of both agent and patient. Immorality lies on a continuum [11]. The more immoral an act, the greater is the involvement of the agent-patient dyad.

In our Pac-Man models, the act of Pac-Man eating the pac-dots is not harmful for other patients (i.e. the ghost and the fruit), this act does not have dyadicness. All other acts, however, where a patient (Pac-Man, ghost or fruit) could be harmed also have an agent which is different from the patient. Therefore, we have distinct agent and patient for all moral acts. We currently do not program Pac-Man agent to take actions considering both the deontological aspect (acts) as well as the utilitarian aspect (consequences) and the theory of dyadic morality (TDM) cannot be fully expressed by our current models. However, taking the consequences into account can help the deontological Pac-Man agent in recognizing if the fruit gets trapped by the ghost first, the “necessary evil” of killing the ghost can help the agent clear the level.

It might be a challenge to program an artificial agent's perception of an act or consequences. However, by knowing how an artificial agent perceives the consequences of it's actions, it can explain us why the agent might have chosen to act in a particular manner, thereby adding transparency and explanability to the computational model.

7. Conclusions and Future Work

In this project, we explored the implementation of two artificial moral agents in the setting of the Pac-Man game by looking at two well-known ethical theories: deontology and utilitarianism.

An amoral Pac-Man agent, that does not embed any moral rules, kills the ghost after eating the big pac-dot, if the ghost comes on Pac-Man's way. As acts like killing are forbidden in deontology, the deontological Pac-Man agent never kills the ghost after eating the big pac-dot. The utilitarian Pac-Man agent kills the ghost if the happiness experienced, in doing so, is greater than the pain and wouldn't kill the ghost otherwise. In our game setting, if the fruit gets trapped by the ghost before Pac-Man can rescue it, the only way Pac-Man can clear the level is if it kills the ghost. As the happiness in clearing the level (+870 points) outweighs the pain in killing the ghost (-200 points), the utilitarian Pac-Man agent would kill the ghost. On the contrary, if Pac-Man is able to rescue the fruit, enabling Pac-Man to clear the level, it does not need to kill the ghost as the happiness experienced is already maximum.

The deontological Pac-Man agent that never kills the ghost, may not be able to score enough points to clear the level if the fruit gets trapped by the ghost before Pac-Man can rescue it. For the deontological Pac-Man, prohibitions and duties are always taken into account and thus, if killing the ghost is a requirement for beating the level, a truly deontological agent cannot do so. On the other hand, the utilitarian Pac-Man agent always manages to clear the level, although it tries, whenever possible, to minimize the amount of pain caused by that and to avoid unnecessary evil.

We notice that a truly deontological agent may sometimes have to face a conflict between succeeding and sticking to it's values of always doing the right thing. Taking the consequences into account can help the deontological agent in recognizing that the sometimes "necessary evil" of killing the ghost can help the agent clear the level.

In future, we need to explore how to compatibilize both the ethical theories of deontology and utilitarianism and implement an agent that can take both approaches into account, in the line suggested by dyadic morality. We also need to translate the current deontological and utilitarian approaches to more complex settings, such as Mario Bros., where there are many more actors (potential agents and patients) to take into account. Furthermore, integrating long-term planning into the BFS and moral algorithms could allow to potentially prevent the deontological agent from getting stuck, or help the utilitarian agent evaluate when it could be worth killing the ghost in advance to prevent a greater harm.

8. Bibliography

[1] Alexander, Larry and Moore, Michael, "Deontological Ethics", *The Stanford Encyclopedia of Philosophy, Edward N. Zalta (ed.)*, Winter 2016 Edition.

[2] Allen, Colin & Smit, Iva & Wallach, Wendell, "Artificial Morality: Top-down, Bottom-up, and Hybrid Approaches", *Ethics and Information Technology*. 7. 149-155.10.1007/s10676-006-0004-4, 2005.

[3] Baader, Franz and Calvanese, Diego and McGuinness, Deborah L. and Nardi, Daniele and Patel-Schneider, "Chapter 2", *The Description Logic Handbook: Theory, Implementation, and Applications, Cambridge University Press, New York, NY, USA*, 0-521-78176-0, 2003.

[4] B. Gert, "The definition of morality", In *Edward N.Zalta, editor, The Stanford Encyclopedia of Philosophy*, Fall edition, 2015.

[5] Bonnefon, Jean-François & Shariff, Azim & Rahwan, Iyad, "The Social Dilemma of Autonomous Vehicles", *Science*. 352. 10.1126/science.aaf2654, 2016.

[6] Cointe, Nicolas & Bonnet, Grégory & Boissier, Olivier, "Ethical Judgment of Agents' Behaviors in Multi-Agent Systems", 2016.

[7] Floridi, Luciano & Sanders, J.W., "On the Morality of Artificial Agents", *Minds and Machines*. 14. 349-379. 10.1023/B:MIND.0000035461.63578.9d, 2004.

[8] Gray, Kurt & Schein, Chelsea, "Two Minds Vs. Two Philosophies: Mind Perception Defines Morality and Dissolves the Debate Between Deontology and Utilitarianism", *Review of Philosophy and Psychology*. 3. 10.1007/s13164-012-0112-5, 2012.

[9] Nick Bostrom and Eliezer Yudkowsky, "The Ethics of Artificial Intelligence." In *Artificial Intelligence Safety and Security*. Chapman and Hall. Previously published in *The Cambridge Handbook of Artificial Intelligence* (2014), 2018.

[10] Noothigattu, Ritesh & Bouneffouf, Djallel & Mattei, Nicholas & Chandra, Rachita & Madan, Piyush & Kush, Ramazon & Campbell, Murray & Singh, Moninder & Rossi, Francesca, “Interpretable Multi-Objective Reinforcement Learning through Policy Orchestration”, 2018.

[11] Schein, Chelsea & Gray, Kurt, “The Theory of Dyadic Morality: Reinventing Moral Judgment by Redefining Harm”, *Personality and Social Psychology Review*. 22. 108886831769828. 10.1177/1088868317698288, 2017.

[12] Sinnott-Armstrong, Walter, "Consequentialism", *The Stanford Encyclopedia of Philosophy, Edward N. Zalta (ed.)*, Winter 2015 Edition.

[13] A. M. Turing, “Computing Machinery and Intelligence”, *Mind* 49: 433-460, 1950

[14] Wallach, Wendell & Franklin, Stan & Allen, Colin, “A Conceptual and Computational Model of Moral Decision Making in Human and Artificial Agents”, *Topics in Cognitive Science*.2. 454 - 485. 10.1111/j.1756-8765.2010.01095.x, 2010.

[15] Z. Leibo, Joel & Zambaldi, Vinicius & Lanctot, Marc & Marecki, Janusz & Graepel, Thore, “Multi-agent Reinforcement Learning in Sequential Social Dilemmas”, 2017.