

UNIVERSIDAD DE COSTA RICA
SISTEMA DE ESTUDIOS DE POSGRADO

**IMPROVING COOPERATION BETWEEN
HUMAN AND COMPUTER PLAYERS
THROUGH PLAN RECOGNITION IN THE
GAME PANDEMIC**

Tesis sometida a la consideración de la Comisión del Programa de
Estudios de Posgrado en Computación e Informática para optar al
grado y título de Maestría Académica en
Computación e Informática

PABLO SAUMA CHACÓN

Ciudad Universitaria Rodrigo Facio, Costa Rica

2020

Dedication

To Artificial Intelligence,
may we awake the right Basilisk one day.

Acknowledgements

I would like to thank my advisor, Markus Eger, for all the attention and help he provided me in the development of this thesis. From midnight chats to awesome discussions on varied topics, he gave me the “little push” I needed to extend beyond my Shire. I would also like to thank the other members of my committee, Juan José Vargas Morales and José Andrés Guevara Coto, who also supported me in this journey and gave me valuable feedback for my work.

Beyond my committee, I want to thank all the other faculty members that helped me in this endeavor: professors and administratives alike. Special thanks to professor Gabriela Marín Raventós who was always willing to answer all of my questions.

Special thanks to my family, who have supported me my entire life. My parents always inspired me to reach higher and follow my dreams and curiosity which ultimately is what brought me here. A special mention to my sister, who is having her own battle with a master’s at the UK, you are missed. And to my extended family: Pedro, Nacho, Luisa, Alex and Mariana, you know you are family to me.

I want to thank all of my friends, regardless of time and place, you have all left a footprint on my heart. Special mention to Pepe, Moriato, Pancha, Achille and Leo, who have shared laughs with me all this years; my high school friends: Josem, Camacho, Andrix and Tacsan, the members of Team Miaw and Luisito’s company. All my other friends who share memories with me, I keep you always in my mind and heart.

These acknowledgments would not be complete without a mention to Mich, who has been by my side the last eight years. Thank you for standing by me, your commitment, patience (*lots* of patience) and loyalty to bear with me all these years.

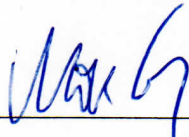
Finally, I just want to say I am grateful for having this opportunity.

Oh Fortuna, velut luna, statu variabilis.

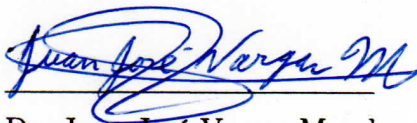
“Esta tesis fue aceptada por la Comisión del Programa de Estudios de Posgrado en Computación e Informática de la Universidad de Costa Rica, como requisito parcial para optar al grado y título de Maestría Académica en Computación e Informática”



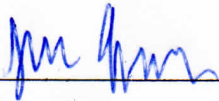
Dra. Kryscia Ramírez Benavides
Representante del Decano
Sistema de Estudios de Posgrado



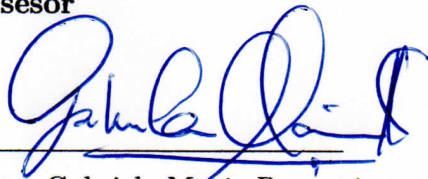
Dr. Markus Eger
Director de Tesis



Dr. Juan José Vargas Morales
Asesor



Dr. José Andrés Guevara Coto
Asesor



Dra. Gabriela Marín Raventós
Directora del Programa de Posgrado



Pablo Sauma Chacón
Estudiante

Contents

Dedication	ii
Acknowledgements	iii
Hoja de Aprobación	iv
Contents	v
Resumen	vii
Abstract	ix
List of Tables	x
List of Figures	xi
Index of Abbreviations	xiii
1 Introduction	1
1.1 Objectives	3
1.1.1 Justification	4
2 Theoretical Framework	5
2.1 Pandemic	5
2.1.1 Starting the game	7
2.1.2 Player turn	8
2.1.3 Overall strategy	11
2.1.4 Complexity of the problem	13
2.1.5 Previous implementations	14
2.2 Planning	14
2.2.1 State-Space planning	15
2.2.2 Problem complexity	17
2.2.3 Planning in games	17

2.3	Plan recognition	18
2.4	Monte Carlo sampling	18
3	Methodology	20
3.1	Game platform	20
3.2	Planning model	21
3.2.1	Restricted Pandemic	23
3.2.2	Planning goals	24
3.2.3	Heuristic	26
3.2.4	Evaluating unknown information	30
3.3	Plan recognition	31
3.4	Data collection and comparison	33
4	Results	35
4.1	Agent evaluation	35
4.2	Experiment results	41
4.2.1	Survey results	41
4.2.2	Performance results	45
4.2.3	Qualitative analysis of the agents	49
5	Conclusions	52
5.1	Game platform	52
5.2	Intelligent agents	52
5.3	Comparing the performance of the agents	53
5.4	Future Work	54
A	Player Experience Questionnaire	56
A.1	Previous game experience	56
A.2	Experience with the AI agent	57

Resumen

Los juegos cooperativos son un desafío importante en el ámbito de la investigación de la inteligencia artificial. El estudio de agentes computacionales, para la cooperación con jugadores humanos, ha tenido un interés creciente en el campo con propuestas de diferentes juegos que presentan una variedad de desafíos. Uno de estos es Pandemic, un juego cooperativo en el que los jugadores deben organizarse para salvar el mundo de cuatro enfermedades representadas con piezas en un tablero. Este juego presenta el desafío de perseguir objetivos múltiples, información incompleta y toma de decisiones, mientras los jugadores se desplazan por el tablero para descubrir la cura a las enfermedades y evitar perder el juego, ya sea al ser sobrepasados por las enfermedades o al quedarse sin acciones restantes. En esta tesis se presenta un agente planificador para el juego Pandemic y se describen dos experimentos que se desarrollan para evaluarlo. En el primer experimento, se evalúa el desempeño del agente al jugar consigo mismo, al variar el nivel de dificultad y la cantidad de jugadores en el juego. En el segundo experimento, se evalúa el desempeño del agente al jugar con jugadores humanos mientras se utilizan dos variantes del agente: una con un módulo de reconocimiento de planes y otra sin este. Este agente utiliza un horizonte limitado y una heurística específica del dominio al efectuar su planificación; además, realiza un procedimiento de muestreo para manejar la información incompleta del juego. Al utilizar estas técnicas, el agente logra ganar alrededor del 34% de los juegos consigo mismo, lo cual es un hito en este ámbito de investigación. El módulo de reconocimiento de planes le permite al agente analizar las acciones tomadas previamente por otro jugador y determinar cuál es su meta actual. Esta información es utilizada por el agente en su proceso de planificación, al incorporar los futuros movimientos potenciales del otro jugador para elaborar planes que les beneficien a ambos. Este experimento demuestra que los jugadores perciben de manera diferente al agente cuando se utiliza reconocimiento de planes, al hacer que

los jugadores humanos lo entiendan con mayor facilidad. Finalmente, se discuten otras aplicaciones para los métodos presentados y para la investigación futura del agente desarrollado.

Abstract

Cooperative games are an important challenge in artificial intelligence research. The study of agents that cooperate with human players has been of increasing interest in the field, with different games proposed as providing a variety of challenges. One of these games is Pandemic, a cooperative game in which the players must cooperate to save the world from four diseases represented by game pieces on a board. This game presents the players with the challenge of pursuing multiple goals, hidden information and decision making while moving across the board to discover the cure of the diseases while avoiding losing the game: getting overwhelmed by the diseases or running out of actions. In this thesis we present a planning agent for the game Pandemic and describe two experiments we performed to evaluate it. In the first experiment we evaluated the performance of the agent while playing with itself with varying difficulty setting and number of players in the games. In the second experiment we evaluated the agent's performance by making it play with human players, using two variants of the agent: with and without a plan recognition module. Our agent uses a limited horizon during planning with a domain-specific heuristic, and performs a sampling procedure to handle hidden information. Using these techniques our agent wins about 34% of the games when playing with itself as partner, which is a milestone in this research domain. The plan recognition module allows our agent to analyze the previous actions performed by the other player and determine what their current goal is. This information is used as part of its own planning algorithm to compute a plan that supports the other player, while incorporating the other player's potential future actions. Our experiment shows that players perceive our agent differently when it uses this plan recognition technique, making the agent easier to understand for the human players. Finally, we also discuss other applications of our techniques and future research for the developed agent.

List of Tables

2.1	List of basic actions that can be performed by all players.	9
2.2	List of specialized actions and effects for the player roles.	10
4.1	Win rate and average number of cures discovered per game by the heuristic, planning and plan recognition agents over 3000 games with p players and e epidemics (up to $\pm 1.70\%$ for the win rate and ± 0.04 for the average number of cures discovered with a confidence of 95%).	38
4.2	Spearman’s rank correlation coefficient between evaluated skill, understanding and helpfulness for the planning agent.	45
4.3	Spearman’s rank correlation coefficient between evaluated skill, understanding and helpfulness for the plan recognition agent.	45
4.4	Observed win rate and average discovered cures for the agents with human participants in the first game played by each participant.	47
4.5	Observed win rate and average discovered cures for the agents with human participants in the second and later games.	49

List of Figures

2.1	Image of the game board of Pandemic depicting the cities of the game, their disease color and their connections with other cities. (Source: boardgamegeek.com , accessed: 11/11/2019)	6
2.2	Pandemic’s outbreak chain example with three cities.	12
2.3	Graph representation of a Pandemic game in state-space planning.	16
3.1	Client-server architecture implemented as the game platform.	22
3.2	Screenshot of the game client for Pandemic depicting the game board, players’ meeples and cards, and other game information.	23
3.3	Comparison of the values obtained for equation 3.2 produced by two different states. The value is reduced as the player moves closer to a nexus of infection.	28
3.4	Comparison of the values obtained for equation 3.4 produced by two different states. The value is reduced as the players concentrate cards of the same color in a single hand.	29
4.1	Observed win rate for the planning agent when using different role combinations. ‘S’ for Scientist, ‘R’ for Researcher, ‘Q’ for Quarantine Specialist and ‘M’ for Medic. Comparison of the results for each combination across different number of epidemics.	37
4.2	Observed win rate for the plan recognition agent when using different role combinations. ‘S’ for Scientist, ‘R’ for Researcher, ‘Q’ for Quarantine Specialist and ‘M’ for Medic.	40
4.3	Survey answers of the participants’ previous experience with the game Pandemic.	42

4.4	Survey answers of the participants' impression on the agents.	44
4.5	Ease of use of the client interface by the participants.	46
4.6	Distribution of the reasons for the game to end for each of the agents. .	48

Index of Abbreviations

In this work the following abbreviations and acronyms are used:

AI	Artificial Intelligence
AIIDE	Artificial Intelligence and Interactive Digital Entertainment
STRIPS	Stanford Research Institute Problem Solver



Autorización para digitalización y comunicación pública de Trabajos Finales de Graduación del Sistema de Estudios de Posgrado en el Repositorio Institucional de la Universidad de Costa Rica.

Yo, Pablo Sauma Chacón, con cédula de identidad 1-1529-0001, en mi condición de autor del TFG titulado Improving Cooperation Between Human and Computer Players Through Plan Recognition in the Game Pandemic

Autorizo a la Universidad de Costa Rica para digitalizar y hacer divulgación pública de forma gratuita de dicho TFG a través del Repositorio Institucional u otro medio electrónico, para ser puesto a disposición del público según lo que establezca el Sistema de Estudios de Posgrado. SI NO *

*En caso de la negativa favor indicar el tiempo de restricción: _____ año (s).

Este Trabajo Final de Graduación será publicado en formato PDF, o en el formato que en el momento se establezca, de tal forma que el acceso al mismo sea libre, con el fin de permitir la consulta e impresión, pero no su modificación.

Manifiesto que mi Trabajo Final de Graduación fue debidamente subido al sistema digital Kerwá y su contenido corresponde al documento original que sirvió para la obtención de mi título, y que su información no infringe ni violenta ningún derecho a terceros. El TFG además cuenta con el visto bueno de mi Director (a) de Tesis o Tutor (a) y cumplió con lo establecido en la revisión del Formato por parte del Sistema de Estudios de Posgrado.

INFORMACIÓN DEL ESTUDIANTE:

Nombre Completo: Pablo Sauma Chacón

Número de Carné: B16195 Número de cédula: 1-1529-0001

Correo Electrónico: pablo.sauma.chacon@ucr.ac.cr

Fecha: 5 de enero del 2021 . Número de teléfono: 8316-5870

Nombre del Director (a) de Tesis o Tutor (a): Markus Eger

BSc

FIRMA ESTUDIANTE

Nota: El presente documento constituye una declaración jurada, cuyos alcances aseguran a la Universidad, que su contenido sea tomado como cierto. Su importancia radica en que permite abreviar procedimientos administrativos, y al mismo tiempo genera una responsabilidad legal para que quien declare contrario a la verdad de lo que manifiesta, puede como consecuencia, enfrentar un proceso penal por delito de perjurio, tipificado en el artículo 318 de nuestro Código Penal. Lo anterior implica que el estudiante se vea forzado a realizar su mayor esfuerzo para que no sólo incluya información veraz en la Licencia de Publicación, sino que también realice diligentemente la gestión de subir el documento correcto en la plataforma digital Kerwá.

Chapter 1

Introduction

In recent years there has been a growing interest in artificial intelligence research applied to games. A reason for this is that games provide a domain in which to try and test different methods and algorithms, as they provide a controlled environment, a well defined set of rules and generally include a scoring system which eases the development of an intelligent agent [BFGM06, BNVB13]. In comparison to the use of datasets (like the *iris* and *MNIST* datasets), games provide a dynamic environment where actions have effects and rewards might not be immediate. They are a different environment that provide different tasks to accomplish when developing an intelligent agent. An intelligent agent can be understood as a computational abstraction which acts with a certain degree of autonomy in a physical or simulated environment [SW10].

Different algorithms can be applied in the development of intelligent agents. Machine learning is the term coined for the algorithms in which the agent or system learns and improves through experience instead of being directly programmed to perform a task. These algorithms have become popular within the field due to the application of neural networks to a variety of problems, including games. Examples of the application of machine learning to games can be seen in the latest developments by Google, like the creation of the AlphaGo artificial intelligence which was able to defeat a human professional player in the game Go in 2016 [SHM⁺16] and the creation of AlphaStar which marked a reinforcement learning milestone when it defeated a professional player in the game StarCraft II five times in a row [VBC⁺19]. Another example of machine learning in games is the Gym project by OpenAI which provides a development platform for reinforcement learning algorithms by enabling the users to test their agents

in different virtual environments and games [BCP⁺16, Ope18]. The previously mentioned examples share the characteristic of the games being adversarial in nature, be it between players or the player versus the environment, nevertheless research regarding cooperative games has grown lately, as demonstrated by the recent attention on the game Hanabi by DeepMind [BFC⁺19].

However, while all these approaches use machine learning to great success, the need for lengthy training periods and large repositories of data make them impractical for developing agents that play with human players when developing a game. Classical AI methods, such as planning, can be considered another approach to the development of intelligent agents, since they allow the development of agents which adapt to the challenges presented by a problem, even in situations that have never been presented with before. Planning is the process by which, given an initial state, a set of goals and a set of actions with their preconditions and effects, a solution is found so that a state satisfying the goals is reached [Wel94]. An example of this is the robot Shakey, developed from 1966 to 1972. Shakey was the first mobile robot with problem solving capabilities which used the Stanford Research Institute Problem Solver (STRIPS) program to solve a variety of problems through planning [Nil84, FN71]. Therefore, planning algorithms can be used to design an intelligent agent for a game if its state, actions, preconditions and effects can be represented logically.

An example of the application of this technique in a game was done by the company Monolith Productions in their game F.E.A.R where they used planning as a tool to create smarter and easier to develop artificial intelligence [Ork06]. This became a milestone as one of the first well documented cases of the application of planning algorithms in a “AAA” game (informal industry classification given to high-end high-cost video games), and demonstrated the great potential this kind of applications could have in systems that interact with users.

An important application of planning can be found in plan recognition which can be explained as the inverse process of planning: instead of finding a set of actions to achieve a goal, the objective is to determine which goal, and plan to reach it, a set of observed actions is likely to serve in pursuit of [RG09]. As such, the recognition of an agent’s intention can be seen as a tool to improve systems that interact with human beings or autonomous agents. A similar application has already been shown to apply for the cooperative game Hanabi [Bau10], in which communication theory was used

to identify the intention of the players’ actions and an agent was created which could understand human players in a way that was understandable and enjoyable for them [EMS⁺20].

In this research we propose an agent design for a cooperative game with no means of communication between itself and the human players. This agent utilizes planning to determine the best course of action. Additionally, we propose a variant of the agent that utilizes plan recognition to test whether its cooperation with the human players is enhanced. We also report the performance of our agents when playing in simulated games with itself, as well as with human players and compare the performance between the two agent variants.

The game in which the interaction between players and the planning agent takes place is the game Pandemic, which is a cooperative game in which players work together to discover a cure for four different diseases while trying to prevent outbreaks¹, the spreading of the diseases and running out of time [Lea08]. The game’s rules and mechanics will be described in detail in section 2.1.

1.1 Objectives

The main objective of this research is *to develop a planning agent for the board game Pandemic that is capable of recognizing the actions taken by human players and cooperate with them in playing the game.*

To accomplish the main objective the following specific objectives were proposed:

1. Create a system that allows human players and an AI agent to participate in a game of the board game Pandemic in a controlled platform.
2. Define a planning model for the game Pandemic that allows an AI agent to accomplish playing the game of Pandemic.
3. Develop a model of plan recognition for the game Pandemic that allows an AI agent to gain insight on the actions taken by human players and incorporate that information to its plan.

¹This work initiated in early 2019, well before and entirely unrelated to, the outbreak of COVID-19.

4. Compare the performance of the agent with and without plan recognition when playing the game with human players and their impact in the players' satisfaction.

The mentioned specific objectives each satisfy the following goals accordingly:

- The creation of a virtual board of the game Pandemic that allows the participation of human and AI agents.
- The definition of the planning model for the game Pandemic.
- The model of plan recognition for the actions of a player in the game Pandemic.
- The results of the comparison of the models given by the metrics of player satisfaction and win rate.

1.1.1 Justification

As previously stated, the application of plan recognition to cooperative environments (in which a human and an autonomous agent must interact and help each other) is a field of research that has not been studied enough. Previous application of plan recognition has been mostly focused on adversarial environments, which do not focus solely on games. Cooperation has mostly been performed through decision trees and finite state machines [Isl05] or through the application of communication theory [EMS⁺20].

This research presents a case of study of the application of plan recognition in human-AI interaction in games and how this application could improve the experience of human players. Also, by taking feedback from the players our work allows better comprehension of what kind of behaviours or characteristics impact the way humans feel towards cooperation and by doing so, provide a new perspective on how to improve systems that cooperate with humans.

The game Pandemic was chosen because of the challenges it presents which include the cooperation between players in a game with non-verbal communication, incomplete information regarding the game state, as well as multiple goals which must be balanced out to succeed. All of these characteristics make Pandemic a challenge different from any other before and we proposed the game as a new domain for human-AI cooperation research, which has been well received by the research community [SCE19].

Chapter 2

Theoretical Framework

In this chapter we will explain some of the key notions required for the understanding of the proposed research. In the next sections the following topics will be explained: the rules of the game Pandemic, planning, plan recognition and Monte Carlo sampling. The rules of the game are necessary to understand the nature of the problem. Planning is used as the central strategy of the agents and plan recognition allows the development of an agent that allows interpretation of the players moves. The Monte Carlo sampling simulates outcomes for a given situation and gives an approximation on the expected result.

2.1 Pandemic

Pandemic is a cooperative board game created by Matt Leacock in 2008. In Pandemic the rules of the game set the players to work together in order to discover the cure to four diseases that are plaguing the world while preventing the diseases from spreading. As it is a cooperative game, all the players win or lose the game as a team. The game was designed to be played by two to four players on a game map representing Earth. This map contains forty eight cities, as shown in figure 2.1. Each city has a specific color defining which of four diseases will emerge in that city and connections to nearby or adjacent cities (see figure 2.1).

The game is comprised by a series of markers that assist the players in tracking the game's progress. There are player meeples with different colors. Meeples assist players in keeping track of their current locations within the game map. Additionally, there

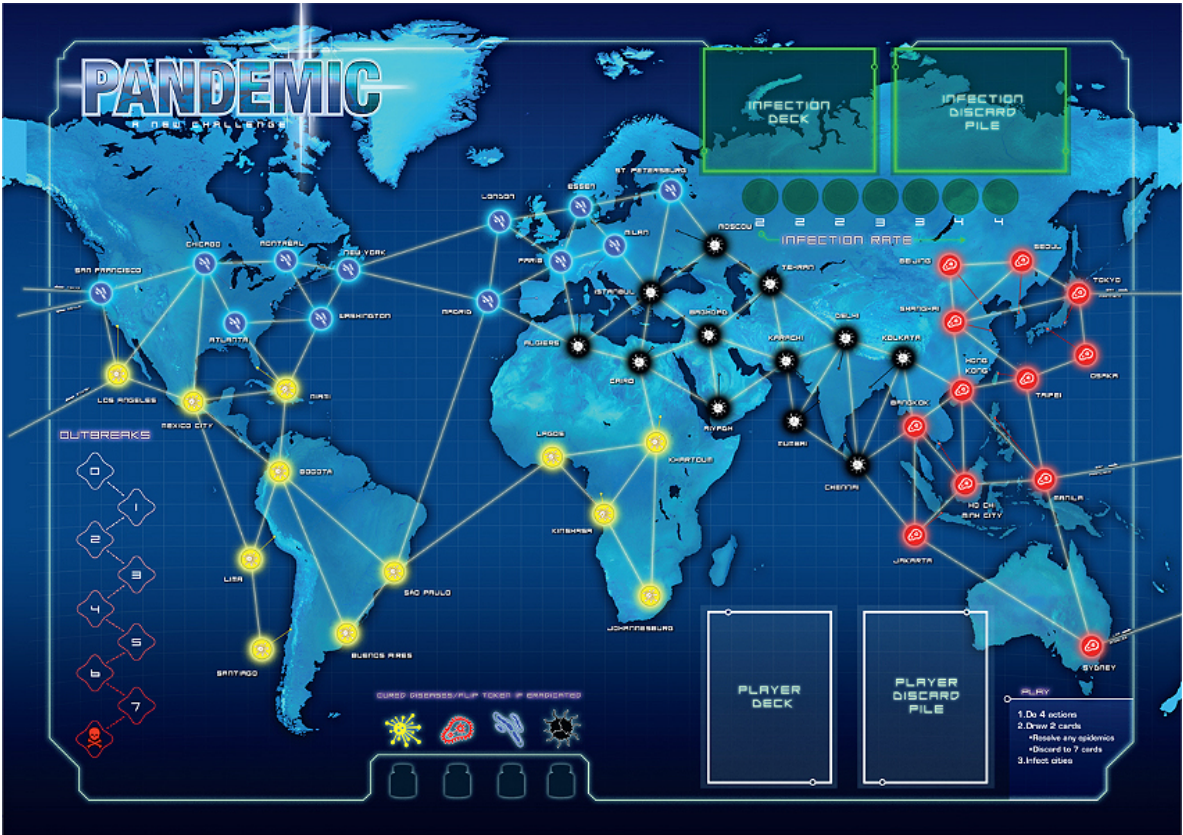


Figure 2.1: Image of the game board of Pandemic depicting the cities of the game, their disease color and their connections with other cities. (Source: boardgamegeek.com, accessed: 11/11/2019)

are colored cubes (twenty four for each of the four colors) representing diseases present on each city. The game also includes an outbreak marker and an infection rate marker. These are used to keep track of the number of outbreaks that have occurred during the game and also of the infection rate throughout the game. The game contains six research station markers which can be placed on cities. These markers indicate the sites in which discovery of the cures can be undertaken. Lastly, there are four different cure markers, one for each disease, that keep track of the diseases that have already been cured.

The game also includes a variety of different cards: role cards which grant them additional specific abilities or actions during their turns; infection cards, that compose the infection deck, which are used to determine which city will suffer an infection of its colored disease at a given moment; city cards which are required to win the game and have a wide range of uses (explained in section 2.1.2), they also have the name of one of the forty eight cities of the game, their population (which has no effect on game play) and their disease color; event cards which grant the players a specific bonus and can be used at any moment; and epidemic cards which cause the beginning of an epidemic event (described below).

As stated previously, the main objective of the game is that the players collaboratively discover the cures to all four colored diseases (in which case they all win). Nonetheless, there are three ways in which the players may lose the game: Eight outbreaks occur during the game (representing a worldwide panic), there are not enough disease cubes left when needed (representing a disease spreading too much) or the players ran out of cards in their player cards' deck when needed (representing the players running out of time).

2.1.1 Starting the game

When the game starts, the outbreak counter is set to zero, no cures have been discovered and the infection rate is set to two. Additionally, an initial research station and the player meeples are placed in Atlanta, where the headquarters of the Center for Disease Control is located. Nine cards with city names on them are drawn from the infection deck. Each card represents a city where infection begins. Each of the cities will have a varying degree of infection: the first three cities get three disease cubes each, the next

three get two disease cubes each and the last three just one each, of their corresponding city color. Each player receives a unique role card and from two to four player cards depending on the number of players.

The player card deck consists of city cards and event cards. It is separated into four to six stacks based on the selected game difficulty and one epidemic card is included into each stack. The stacks are then shuffled individually and piled up to create the player card deck. This way of shuffling and building the player card deck provides a constant rate (or smoothed randomness [Lan18]) to the epidemic events, which means that the events will be spread out somewhat evenly over the duration of the game. Finally, the player which has the city card with the highest population among their player cards starts the game.

2.1.2 Player turn

In Pandemic, for each turn each player will go through the following steps: an action phase, a drawing phase and an infection phase. Each of these will be explained in detail in the following sections.

Action phase

In this phase a player must perform up to four game actions. These are chosen among eight possible basic actions and potentially a special bonus action provided by the player's role. The basic actions any player can perform are listed and explained in the table 2.1.

By discovering cures and treating diseases the players can eradicate a disease. A disease is considered to be eradicated if it has been cured and all of its disease cubes have been removed from the cities. When a disease is eradicated any further infections by that disease to a city will not add any disease cubes.

Additionally, the players may perform specialized actions depending on their role. Each of the specialized actions and effects different roles have on the game are listed and explained in the table 2.2.

Table 2.1: List of basic actions that can be performed by all players.

Action name	Description
Drive/ferry	Move to a neighboring city connected on the map by a white line.
Direct flight	Move to a given city, discarding a card with that city's name on it.
Charter flight	Move to any city, discarding a card with the current city's name on it.
Shuttle flight	Move from a city with a research station to another city that also has a research station on it.
Build research station	Place a research station marker on the current city, discarding a card with the current city's name on it.
Treat a disease	Remove a disease cube or, if the disease has been cured, all of the same color from the current city.
Share knowledge	Give to or receive from another player the a city card if both are in that city.
Discover cure	Discover a cure to a disease by discarding five cards of that disease's color in a city with a research station.

Drawing phase

In this phase, the player must draw two cards from the player cards deck. If at any time the player holds more than seven cards then an exceeding amount of them must be used or discarded.

When a player draws an epidemic card from the player deck an epidemic occurs. The infection rate marker increases its value by one and the current infection rate might increase: it is increased by one at the third and fifth epidemic. The bottom card of infection deck is drawn, the city indicated by the card gets infected by three disease cubes of its color and the card is discarded. Finally, the discard pile of the infection deck is shuffled, put on top of the infection deck and the epidemic card is removed from the game.

Infection phase

In this final phase, a number of cards equal to the current infection rate is drawn from the top of the infection deck. Each of the cities indicated by the cards gets infected by one disease cube of its color. Then the game continues with the beginning of the turn

Table 2.2: List of specialized actions and effects for the player roles.

Player role name	Description
Contingency planner	Can take an event card from the discard pile and store it on their role card. Only one of them may be stored at a time. After use, the card is removed from the game.
Dispatcher	Can move any player's meeple to any other city in which there is another player and also move it as if it was its own.
Medic	Removes all cubes of the same color when treating a disease using only one action. If a disease has been cured then automatically treats the disease when being in a city without consuming an action.
Operations expert	Can build research stations in its current city without discarding any card. Also, once per turn can move from a research station to any city by discarding a city card.
Quarantine specialist	Prevents outbreaks and the placing of cubes in their current city and all neighboring cities, except during the game setup.
Researcher	Can give any city card to other players in the same city and other players can also take the cards in their turn.
Scientist	Requires only four city cards of the same color (instead of five) to discover a cure for a disease.

of the next player to the left.

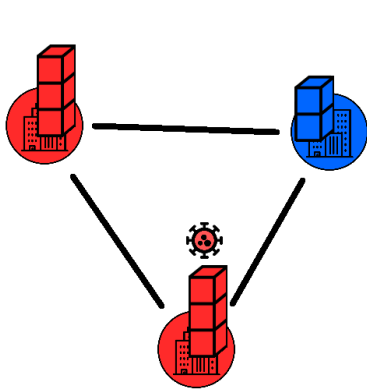
The disease cubes are added to a city by infections and stay in the city until the players remove them through an action. Nevertheless, there can only be a maximum of three cubes of the same color in each individual city. If an infection would add more cubes than the maximum three, then the city stays at three and an outbreak occurs in the city. When an outbreak occurs, the outbreak counter increases by one and all the surrounding cities add one cube of the outbreak's disease color to their own. This may cause another outbreak in one or more of the neighboring cities causing a chain reaction. In a chain reaction, each outbreak must be solved individually, but without causing outbreaks in any cities which have already been part of that same outbreak chain, as shown in figure 2.2.

2.1.3 Overall strategy

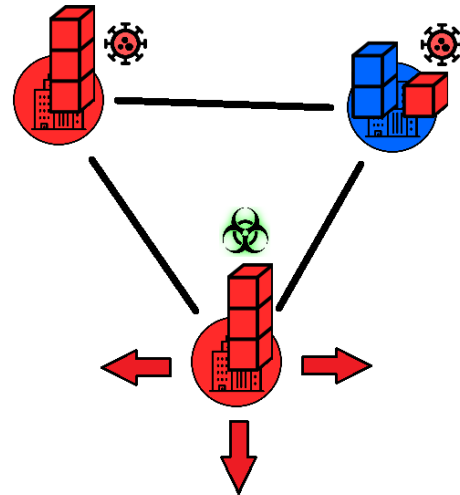
The main characteristic of Pandemic is that it presents a challenge to the players by forcing them to make decisions regarding their current objective. It forces the players to balance between containing diseases that are already spreading and discovering cures. Treating diseases can be considered a simpler task than discovering a cure, because the latter often requires card exchanges that can only occur if players are in the appropriate city. Therefore the main problem faced by the players is their cooperation and the way they decide to manage their actions in an effective way.

The overall strategy of the game [Boa, Jam17] may vary depending on the roles randomly assigned to each player. Usually the players with the medic and quarantine specialist roles are best used to control infections in the cities, moving promptly to prevent outbreaks around the globe. The contingency planner, operations expert and dispatcher take on more of a support role, using their abilities to focus on helping the other players realize their objectives usually involving easing up the movement on the map. Finally, the researcher and scientist roles are best used when focusing on discovering cures, be it through sharing their cards or receiving cards respectively.

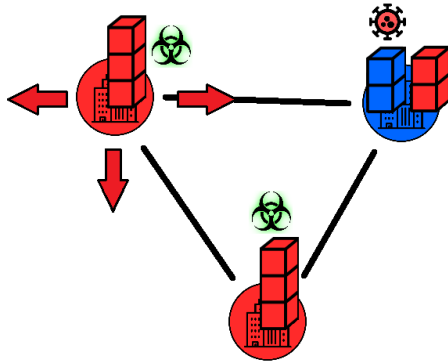
When the game begins, it can be helpful for the players to focus on establishing research stations. These stations also allow players to ease their movement and facilitate disease treatment, which helps prevent outbreaks. The strategic focus of outbreak prevention is a priority since each outbreak implies overwork for the players and it



(a) The city at the bottom gets infected by the red disease but already has three red disease cubes.



(b) An outbreak occurs at the bottom city infecting the neighboring cities with the red disease.



(c) The city on the left outbreaks infecting neighboring cities which are not already part of the outbreak chain.

Figure 2.2: Pandemic's outbreak chain example with three cities.

is easier to treat one disease cube in the current city than one disease cube in each neighboring city. A general rule of thumb is to keep track of cards in both discard piles, since the cards discarded from the player card deck may make it impossible to achieve victory and the ones discarded in the infect pile represent cities which will not be infected until a new epidemic card strikes. On the other hand, when an epidemic card causes the infection cards to be shuffled and set on top of the infection deck the players know for sure which cities will be infected next. The players should try to use the cards instead of discarding them to the maximum hand limit. Additionally, it is important to increase the focus on discovering cures for the diseases as the game grows closer to the end since the amount of remaining actions grows smaller.

While eradicating diseases is not mandatory for winning the game, it can provide some relief in the later stages of the game. Nonetheless, spending too much effort on eradicating a disease might be counter productive so the players must be vigilant of making efficient choices. Discovering an early cure for a disease is also an example of a case in which spending too much effort might be counter productive and the players may want to wait a little longer before making any rash decisions.

2.1.4 Complexity of the problem

The described game poses a challenging problem when trying to create an intelligent agent for it. The generalized problem of the game itself has been proven to be NP-complete [NT12] which suggests that traditional approaches to finding a solution are unlikely to be effective.

The game can be understood as a graph composed by forty-eight vertices, in which multiple variables are accounted for in each vertex. Also, the players enjoy a wide variety of possible actions which increase the complexity of the game giving a massive range of possible final states in just a single player's turn. Another fact to take into account is the incomplete information that the players have during the game, due to the randomness of the two different decks. The effect this unknown information has on the game is that the players always need to rethink their current plans when new information is revealed and also adapt their behaviour to the new circumstances.

2.1.5 Previous implementations

Previous work regarding a virtual implementation for the game Pandemic has been performed. In 2012, an implementation of the tabletop game was developed for the purpose of comparing how different levels of automation impacted the game [WPC⁺12]. This automation was defined as getting rid of the processes which might be routine to the game, like shuffling cards or setting up the board game, and acting as an impartial referee.

An implementation of the game’s mechanics (which does not possess a graphic user interface) was created by Thomas Keefe and it’s available for free on GitHub¹. The game has also been implemented by Asmodee Digital together with Z-Man Games to create a digital version of the game available for sale in the Steam store², but the game allows only human players to participate.

When we started developing this project, we did not find evidence of other previous intelligent agents developed for this game. However, in the international conference on the Foundations of Digital Games 2020, Konstantinos Sfikas and Antonios Liapis published their own agent for the game Pandemic, which uses a Rolling Horizon Evolutionary Algorithm (RHEA) to develop plans for its agent [SL20]. For their implementation the authors used macro-action encoding to reduce the complexity of the plans and used their genetic algorithm to develop a better plan to be taken by the agent. This approach allowed their agent to achieve a 22% win rate in its best setup (three player and four epidemics setup) for randomized games.

2.2 Planning

As mentioned in chapter 1, planning is the process by which, given an initial state, a set of goals and a set of actions with their preconditions and effects, a solution is found so that the set of goals is satisfied [Wel94]. To do this, a model is created that formalizes the problem space into its three main entities which include: the initial world model (what is true and what is false at the beginning of the problem), the goal conditions (conditions a state must satisfy to be considered a goal state) and actions. Actions have

¹<https://github.com/thomaskeefe/pydemic>

²https://store.steampowered.com/app/622440/Pandemic_The_Board_Game/

preconditions, conditions that need to be satisfied for the action to be applicable, and effects which are the changes applied to the world model [FN71]. This way of formalizing a planning problem was first described as part of the Stanford Research Institute Problem Solver (STRIPS) program and has since become of widespread use in the form of multiple mathematical representations and algorithmic implementations.

An important distinction to be made regarding actions is the difference between an specific action and an action schema. An action schema is used to describe an action that may take place with a wide variety of parameters or variables. This allows flexibility when declaring the set of actions by not having to specify the same action for different variables. Actions without free variables (or where free variables have been bound) are also called *ground* actions. An example of an action schema is ‘move to X’, while a *grounded* instance of the schema would be ‘move to London’.

By defining a planning problem as previously described, the planning problem can then be understood as a search or path-finding problem. There are two main representations for this search problem which are known as state-space planning and plan-space planning, the first of which will be explained in the following subsection. The second one will only be mentioned briefly as a representation in which possible plans are used as vertices of a graph with the edges being plan refinement actions (like the addition of an action) [Wel94].

2.2.1 State-Space planning

One of the possible representations of the planning problem is to represent the problem as a finite set of all possible states and a finite set of actions which allow to the transformation from one state into another [BG01]. In this formulation, a planning problem can be interpreted as a graph, where the states are the vertices of the graph (and the starting vertex the initial state) and the actions are the edges of the graph [Wel94]. For the game Pandemic an example of its graph is shown in figure 2.3.

This representation allows the application of known search algorithms and heuristics that may speed up the finding of a solution. An example of a possible heuristic is obtained by solving a simplified version of the problem to obtain an estimate for the complexity of solving it. With an heuristic it is possible to apply the A* (A-star) search algorithm [HNR68] which performs a best-first search on the graph to find an optimal

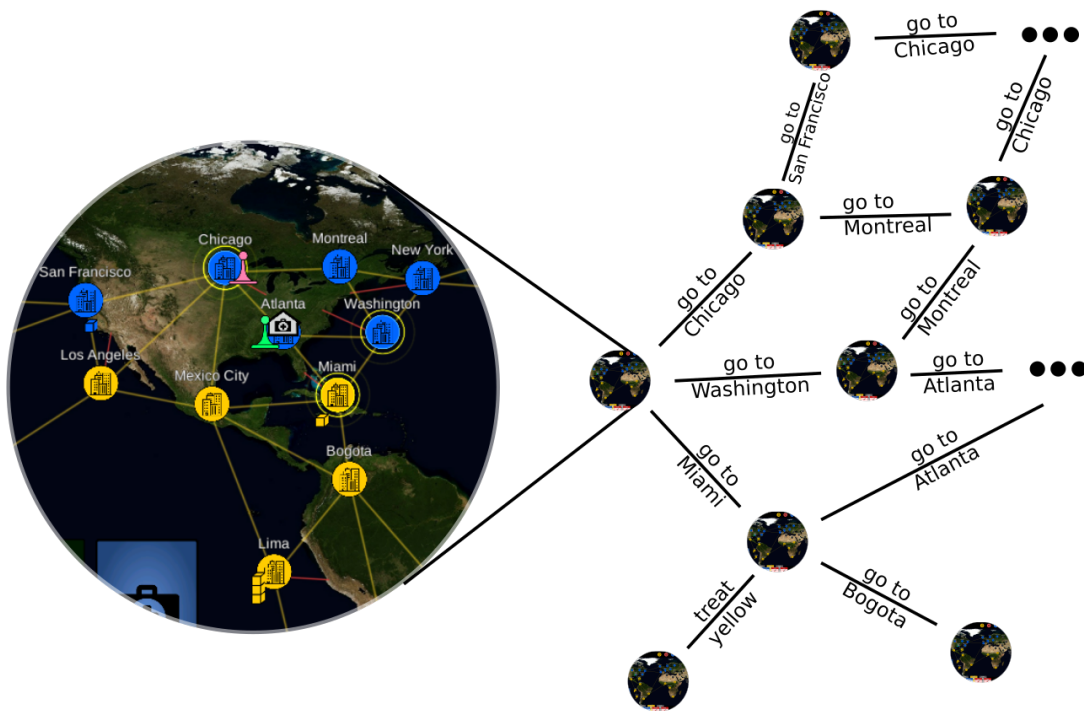


Figure 2.3: Graph representation of a Pandemic game in state-space planning.

solution.

2.2.2 Problem complexity

As explained above, the planning process can be represented as a graph and therefore, the planning problem understood as a path-finding problem. Nevertheless, the main difference between this two problems resides in the complexity of each of them. While path-finding is known to be a polynomial time problem, planning is not, instead being known to be a PSPACE-complete problem. This difference arises from the fact that path-finding problems tend to have an associated specified graph, while planning problems' graphs are implicit to the problem and may be exponentially large [BJ11, By194].

2.2.3 Planning in games

Planning has been used as part of the development of artificial intelligence agents for games in recent years. One of the best examples of planning applied to games was the previously mentioned case of F.E.A.R., which used planning to decide the actions to be performed by the non-playable enemy characters when interacting with the human player [Ork06]. This system was later used by the developing company (Monolith Productions) in the games F.E.A.R 2, Condemned, Condemned 2 and most recently Middle-earth: Shadow of Mordor [Hig15].

Other games which made use of planning are Tomb Raider by the developer Crystal Dynamics [Con15] and Deus Ex: Human Revolution by the developer Eidos Montreal [Rab13]. In both games, planning algorithms were used to decide the actions of the non-playable enemy characters similar to what had previously been developed.

Other companies have also used planning when developing their agents, as is the case of The Creative Assembly, developers of the Total War saga. Planning has been used in their recent releases starting with Empire: Total War³ where the game AI uses planning to manage its troops actions and movements when playing on the battlefield. This approach has been extended to other games of the saga like Napoleon Total War, Total War: Shogun 2 and Total War: Rome 2⁴ with increasing complexity allowing for

³Interview with a developer by Adam Pavlacka. Available at: worthplaying.com/article/2008/12/1/interviews/57018/, accessed: August 10, 2020

⁴As described by Tommy Thompson in the videos series "The AI of Total War" available at [youtube.com/user/tthomps](https://www.youtube.com/user/tthomps) and gamasutra.com/blogs, accessed: August 10, 2020

a more challenging experience for the players.

2.3 Plan recognition

As mentioned in chapter 1, plan recognition is the problem of identifying a set of goals such that the observed actions would be part of an optimal plan for those goals [RG09]. There are two main approaches to solve this problem: the plan library approach and the domain theory approach.

The first approach focuses on the use of recipes that conform a plan library. A recipe encodes partial plans (a set of observable actions) and their corresponding goals. When a set of actions is observed, the plan library is checked to find a partial plan which matches the observed actions and a goal is predicted. The goals in the library can be either final or partial, with partial goals being used as part of other goals' partial plans [Car01].

The second approach focuses on the use of the domain theory used for planning: initial state, goals and actions, but uses them as a means to understand the observations of another agent. This approach uses the observations as part of a new planning problem in which those observations are included as required actions that must be taken to satisfy the final goal [RG09, CRY15]. As such, a planning representation of a problem can be suited to handle plan recognition by modifying the goals. A planning process is applied using the different goals to develop plans, these are used to discover the goal which makes for a more feasible explanation of the observed actions.

2.4 Monte Carlo sampling

Monte Carlo sampling, also called Monte Carlo method, is the process of obtaining information about a problem by performing random simulation modeling. In this process, random instances of an object of study are generated, simulated and evaluated to obtain additional information regarding its result. These results are used to gain information about the problem as, for example, its expected result [MU49].

This kind of methods have been used in a wide spectrum of problems, ranging from engineering and physics problems to economics and finance [KBTB14]. Additional,

making way to further more specialized methods as for example the Monte Carlo Tree Search [[Cou06](#), [YT18](#)].

One of the advantages this kind of algorithms present is that they are an anytime algorithm, which means their execution can be stopped at any moment and the calculated data so far can be used. These methods have been applied previously in various games, including card games like poker [[SPPF09](#), [VdBDR09](#)] and Magic: The Gathering [[CWP12](#)]

Chapter 3

Methodology

In this chapter we will explain the methodology applied to fulfill each of the objectives of this thesis.

3.1 Game platform

The first objective of this thesis is to create a system that allows a human player and a simulated player to participate in a game of Pandemic in a controlled platform or environment.

As mentioned in section [2.1.5](#), previous implementations of the game have been developed as part of different experiments and engines. However, only one of these frameworks is available online for free. While the implementation of the game by Thomas Keefe could be used to develop the game engine, it does not provide a graphic user interface in which human players can play with a simulated player in real time.

The approach taken was the creation of an implementation of the game based on web-technologies to make the game playable in a browser. To this end, a client-server architecture and a user interface to facilitate the player's control and understanding of the game were developed. The client was also used to get feedback from the players' experiences.

The game logic and server were implemented in Python. The server has the capability to display HTML webpages, images and files. The game client was developed using Unity and C#, exporting the final product as a WebGL which could be accessed by the players through the server. The client then communicated with the server through

regular HTTP protocols and received game information from the server using JSON structured data (as shown in the code snippet). A representation of the architecture used is shown in figure 3.1.

JSON data snippet

```
{
  ...
  "players": {
    "p0": {
      "location": "atlanta",
      "role": "RESEARCHER",
      "cards": [
        "lagos",
        "algiers"
      ]
    },
    "p1":{
      ...
    }
  },
  ...
}
```

The obtained result of this development is the client-server system where the client provides the user interface for the players, while the game logic is kept entirely on the server side. A screenshot of the game client can be seen on figure 3.2.

3.2 Planning model

The second objective of this thesis is the definition of a planning model for the game. This was achieved by designing a mapping for the game state, the game goals and actions to an AI planning problem. This representation was then used by an agent to perform planning to win the game.

In this research, a representation of the game state was developed that allows a state-space planning approach to be used. The game logic represents the game as a

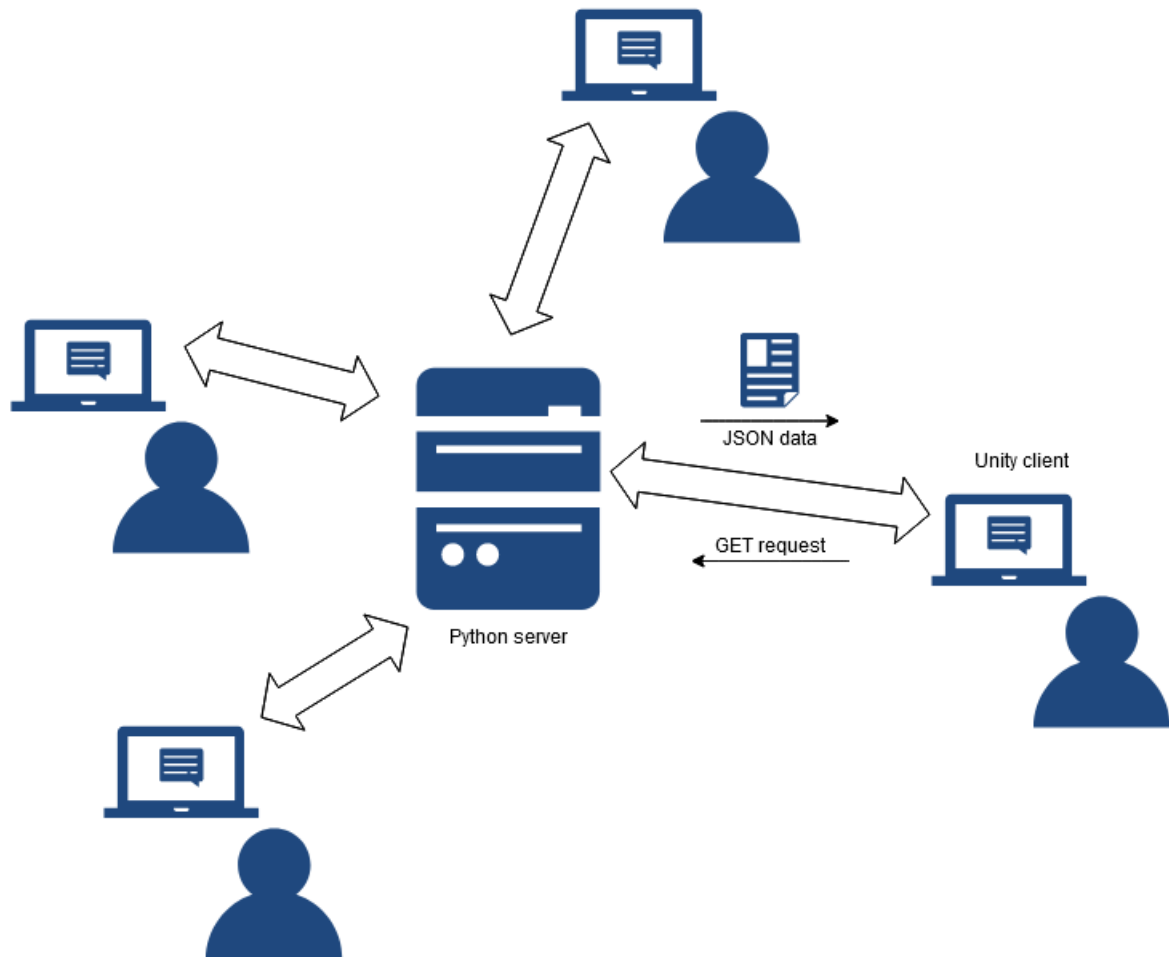


Figure 3.1: Client-server architecture implemented as the game platform.



Figure 3.2: Screenshot of the game client for Pandemic depicting the game board, players’ meeples and cards, and other game information.

game state and provides information regarding all possible actions for the current state. These actions can be used to obtain all neighboring states that would result from taking each specific action, without modifying the original game state.

This representation is used by our planning agent which performs A* search to find a path (sequence of actions) that leads to a goal state. However, given the complexity of the problem, some minor restrictions were necessary to create a simplified version of the game without affecting the core gameplay.

3.2.1 Restricted Pandemic

Compared to the original, our version of Pandemic has some restrictions. These restrictions remove the event cards and three of the seven player roles from the game.

The removal of the event cards was chosen because they could be used at any given moment (even between steps of an effect like an epidemic or between phases) and many of them had a large action space to choose from (for example, adding a research station at any city). Therefore they present a problem in the form of a search space explosion

when performing planning. Nonetheless, their removal doesn't affect any of the core mechanics of the game and the overall goals and actions remain the same. However, this removal adds additional difficulty to the game because the number of maximum turns in the game is reduced (the player deck runs out of cards faster) and they were a powerful tool for the players because they allowed them to perform powerful actions once per game.

The second restriction in our version of Pandemic is that three of the seven player roles were removed. The removed roles are the Contingency planner, the Dispatcher and the Operations expert. The reason for the removal of the Dispatcher and Operations expert lies in the fact that both of them have powerful movement options that caused search space explosions. The Dispatcher can move any other players' meeples as if it was their own and can also rally a player to any other's position, this increases the number of available movement actions greatly (and increases with a higher number of players). The Operations expert had the ability to once per turn move from a city with a research station to any other by discarding a card, this greatly increased the number of possible movement actions, especially when considering the player could also build research stations in any city without discarding any cards. The Contingency planner removal was decided upon grounds that their abilities are related to event cards (in the discard and in its hand). As these cards were removed, it left the role without a purpose so it was removed from the restricted version.

With these restrictions in place, we will move on to the planning goals to be used by the agent.

3.2.2 Planning goals

A goal function allows the evaluation of a state to find out whether it is a goal state or not. In Pandemic players are faced with the problem that there is a main goal (to discover all four cures) but to achieve it they need to avoid losing the game (commonly called a *maintenance* goal). This forces the players to deal with competing goals where they must choose at each moment which of the two goals to work towards to.

Another thing to consider when performing planning, is that planning too far ahead might present different problems. First, the search space of the problem increases about seven-fold with each action taken, with each turn consisting of at least four actions.

This increase is the result of the possible actions the players can perform in their turns. This causes the search space to grow exponentially, making planning until the end of the game infeasible, especially on early states of the game. Also, the hidden information of the game (presented by the order of the cards in the player and infection deck) can make planning too far ahead wasteful, since making plans taking all possible scenarios into account becomes increasingly complex and many of the planned scenarios will not be used at all. Lastly, since the planning agent is expected to be able to play with human players the time the agents takes for their turn also becomes an important consideration, since human players dislike to wait too long for a computer player to take its turn. For these reasons we added an upper limit to the search to be performed. The upper limit cuts off planning after approximately 10 seconds, which after some initial tests was chosen to be after 3500 nodes of the graph have been expanded and use the best plan found so far.

Two specific goals were developed to be used by the planning agent. We named these goals “discover cure” and “survive” respectively. The “discover cure”-goal is satisfied when discovering a previously unknown cure and the “survive”-goal is satisfied if the players manage to avoid losing the game after a given number of turns. Whenever our planning agent perform a planning action one of the two goals is selected and the agent tries to come up with a plan to satisfy the goal. The agent assumes all the other players are currently working on the same goal when planning their actions. In the case of the “survive” goal we used an upper bound of two turns, which means the agent will plan how to survive (treat infections and prevent outbreaks) up to two turns ahead.

When performing planning, to choose the goal to be used the current state is evaluated. If any of the players has the number of cards required to discover a cure or is missing just one card, while being able to receive a card currently in game from another player, then the “discover cure” goal is chosen. In any other case the “survive” goal is chosen.

The agent performs a planning cycle whenever it must take an action and new information has become available to the players, such as when players draw cards or cities become infected. The planning usually happens at the beginning of the agent’s turn, but can also happen if, for example, a player transferred a card which causes the agent to discard in the middle of another player’s turn. After coming up with a plan, the agent will execute its actions until new information becomes available or there are

no more actions left in the plan.

However, there is an issue to address regarding the “survive” goal: how does the agent choose among different plans when all of them manage to arrive at the set goal? The answer to this question lies in an useful parameter of the A* algorithm: the heuristic.

3.2.3 Heuristic

A heuristic is necessary to improve the efficiency of the A* algorithm, since it is used as a means to give priority to nodes which are more likely to lead to the goal quickly. For our implementation of the planning agent, a domain specific heuristic was developed. This heuristic is composed of seven different equations which evaluate different aspects of the game. The purpose behind these evaluations is to give an estimate of the number of actions remaining until the agent can win the game or satisfy a specific goal.

To evaluate a state, the heuristic takes into account the following factors: cures discovered, number of disease cubes in cities, cards in the players’ hands and in the discard pile, distance of the current player to locations of interest and distances between cities. The number of cures discovered (or contrarily, remaining to be discovered) is directly related to the main objective of the game and is the main indicator of how close to winning the game the players are. The number of current disease cubes in cities is an indicator of how much the diseases have spread, which has a direct effect on the number of outbreaks. The distance of a player to the nearest city with a research station establishes the minimum number of actions it would take for the player to be able to discover a cure. The distance in relationship to city with disease cubes allows us to evaluate how many actions in average would take the player to arrive at an infected city. The cards in each players’ hands allow the calculation of how many cards are missing until the players are capable of discovering another cure, and the cards in the discard pile allows to keep track of the number of cards remaining for each color. Lastly, the distances among cities allow us to estimate how costly in terms of actions it is for players to move around the board.

For the description of the heuristic the following expressions will be used. The expression $active(k)$ evaluates whether the k colored disease is currently active (its cure has not been discovered). The expression $distance(p, c)$ calculates the distance between player p and city c . Similarly, $infection(c)$ calculates the number of disease

cubes present in city c . The expression $\text{cards}(p, k)$ counts the number of cards of color k that the player p has in its hand. Meanwhile, $\text{discard}(k)$ counts the number of cards of color k currently in the discard pile. Lastly, the constant R_p is the number of cards required to discover a cure by player p (is always 5 except for the Scientist player with value 4).

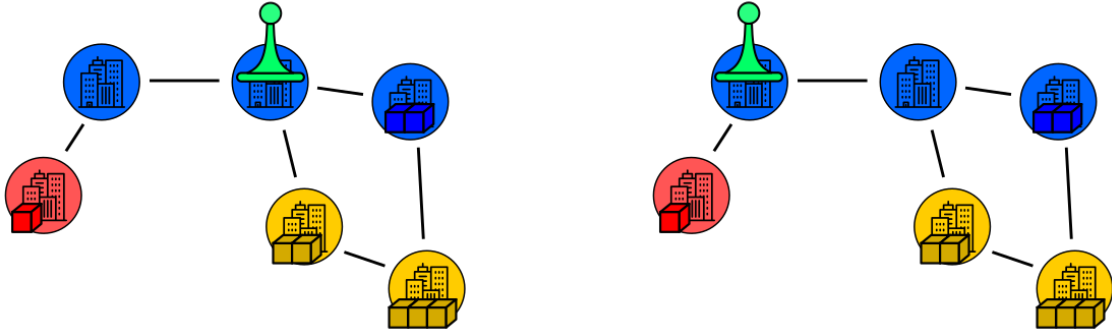
To perform counts among different elements of the game, four sets are used in the expressions. The set *Color* contains the colors of the four diseases, which is used to evaluate whether they are active. The set *City* contains the twenty four cities spread across the map, which is used to calculate distances and infections among them. The set *Player* contains the players participating in the game, which is used to count their cards in hands and their distance to cities. Finally, the set C_{rs} contains the cities with a research station built on them, used in a similar fashion as the *City* set.

The first equation in our heuristic evaluates the number of cures that remain to be discovered in the current state. Equation 3.1 calculates the number of diseases that still require a cure to be discovered to win the game by counting the number of active diseases.

$$h_{cures} = \sum_{k \in Color} \text{active}(k) \quad (3.1)$$

To evaluate the current position of the players, two different equations were developed. The first one, seen in equation 3.2, evaluates the positions of the players in relationship to the cities as the average distance to each city weighted by the amount of disease cubes it has. The purpose of this equation is to motivate planning towards positions closer to nexuses of infection, which is necessary when trying to prevent outbreaks. An example of this equation is shown in figure 3.3 where two different states are compared. The second equation, seen in equation 3.3, evaluates the positions of the players with regard to their proximity to the nearest research station. The purpose of these equation is twofold: it motivates players to stay near research stations, which allows them to quickly discover a cure when the right cards are available, and also motivates the construction of more research stations distributed across the board.

$$h_{dsurv} = \sum_{p \in Player} \frac{\sum_{c \in City} \text{distance}(p, c) \cdot \text{infection}(c)}{\sum_{c \in City} \text{infection}(c)} \quad (3.2)$$



(a) Player is closer to a nexus of infection. The heuristic value for h_{dsurv} in this state is 1.5.

(b) Player is farther away from a nexus of infection. The heuristic value for h_{dsurv} in this state is 2.25.

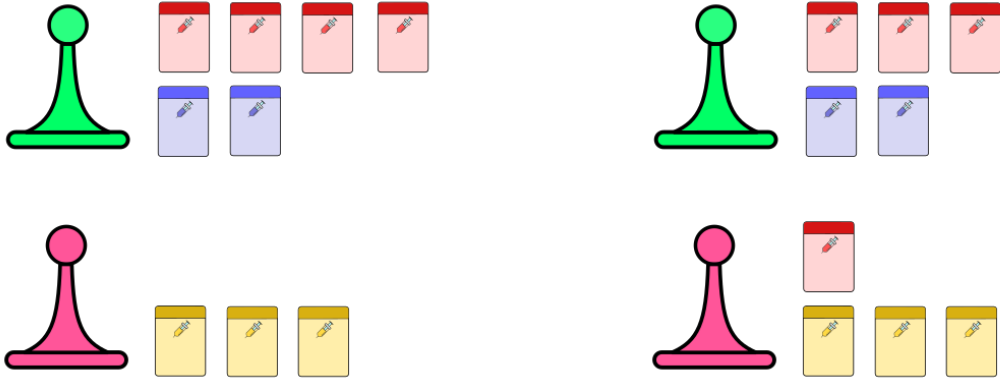
Figure 3.3: Comparison of the values obtained for equation 3.2 produced by two different states. The value is reduced as the player moves closer to a nexus of infection.

$$h_{dcure} = \sum_{p \in Player} \min_{c \in C_{rs}} \text{distance}(p, c) \quad (3.3)$$

To evaluate the value of the cards in the players hands, equation 3.4 is used. This equation calculates the minimum number of cards missing to discover a cure among players' hands for each disease color, while ignoring diseases which have already been cured. The purpose of this equation is to motivate players to concentrate cards of the same color in one of the player's hand to get them closer to discover a cure. An example of this equation is shown in figure 3.4 where two states are compared. Another factor to take into account is the number of cards remaining of each color since, if too few cards of a certain color remain, it might be impossible to discover a cure for the disease. For this reason the equation 3.5 was included, which increases in value by the number of cards in the discard pile for which the cure has not yet been discovered. An increase in value in terms of the heuristic means that the players are farther away from winning the game.

$$h_{cards} = \sum_{k \in Color} \text{active}(k) \cdot \min_{p \in Player} R_p - \text{cards}(p, k) \quad (3.4)$$

$$h_{disc} = \sum_{k \in Color} \text{active}(k) \cdot \text{discard}(k) \quad (3.5)$$



(a) A single player has the red cards. The heuristic value for h_{cards} in this state is 11.

(b) Both players have red cards. The heuristic value for h_{cards} in this state is 12.

Figure 3.4: Comparison of the values obtained for equation 3.4 produced by two different states. The value is reduced as the players concentrate cards of the same color in a single hand.

Another important value to take into account is the total number of disease cubes on the board, as treating the disease cubes is imperative to avoid losing the game. To this end, equation 3.6 calculates the total of disease cubes present across all cities, and thus motivates players to treat the diseases. As for building research stations, this is important not only because it allows players to discover cures for diseases, but also because players can move directly between cities with research stations on them. This reduces the number of actions needed to get from one city to others and has a great impact in the long run. To evaluate these distances, equation 3.7 is used. This equation calculates the average current distance from each city to each of the other cities and then multiplies it by a linearly decreasing value which represents the number of turns remaining. This value is not constant since cities with research stations allow players to travel between as if they were connected. This motivates players to build research stations in cities which would reduce the travelling actions needed to go from one city to another, however, the relevance of this value decreases over time.

$$h_{inf} = \sum_{c \in City} \text{infection}(c) \quad (3.6)$$

$$h_{dist} = \sum_{c_1 \in City} \sum_{c_2 \in City} \frac{\text{distance}(c_1, c_2)}{48 \cdot 47} \cdot \frac{\text{turns}_{remaining}}{\text{turns}_{max}} \quad (3.7)$$

Finally, with the seven equations specified, it is possible to specify the heuristic used to evaluate each state. However each of these values must be multiplied by a specific weight which impacts its relevance in relationship to the others. The final equation can be seen in equation 3.8. Additionally, only one of the distance equations is applied when evaluating a state depending on the goal, with the distance to the nearest research station being used for the “discover cure” goal. The values for the weights used in this final heuristic are the result of a grid search of values in which different values for each parameter were tested. The values to be used in the grid search were chosen from preliminary tests in which for each of the equations of the heuristic different values were tested and chosen.

$$\begin{aligned} h_{state} = & 0.5 \cdot h_{dsurv} + 0.5 \cdot h_{dcure} + \\ & 1 \cdot h_{cards} + 0.5 \cdot h_{disc} + \\ & 0.6 \cdot h_{inf} + 0.6 \cdot h_{dist} + 24 \cdot h_{cures} \end{aligned} \quad (3.8)$$

This heuristic is applied when evaluating the states as part of the A* algorithm. However, there is still a last question regarding how to tackle the planning problem which is how to handle the hidden information of the game.

3.2.4 Evaluating unknown information

The evaluation of hidden information poses a challenge to the planning process because the outcome of a specific plan is unknown. To tackle this problem, a Monte Carlo sampling approach was taken. Our planning agent performs planning using only the currently available information until a goal state is reached. After a goal state is reached, random simulations are performed simulating the possible draws that could happen during the plan. These states are evaluated using the heuristic function to calculate an average expected value for the state resulting from such plan.

When considering the possible draws that may occur, while the exact ordering of the cards is unknown, there is partial information about the order which is used

when generating the possible draws for the algorithm. This information is often used by veteran players to counteract the uncertainty to some extent. The information regarding the player deck comes from the partial distribution of epidemic cards in it. When assembling the player deck, the cards are split into piles and each of the piles gets an epidemic card. While each pile is shuffled individually, the piles are just stacked to form the deck, which gives players certain ability to predict when an epidemic will or will not show up (depending on the number of cards remaining in the deck and whether the current pile's epidemic has already been drawn). The information regarding the infection deck comes from the epidemic mechanic, which shuffles the infection deck's discard pile and puts it on top of the infection deck. What this implies is that, while the exact order is unknown, players can be certain that the next cards can only come from a much smaller pool of possible cards, which are precisely the ones that were previously put into the discard pile.

The information is used by our agent when generating the possible draws for both decks when performing a simulation of the outcome for a specific plan. By simulating the draws using the known information, a more precise estimation of the expected heuristic value for the plan is obtained. This expected value for a state allows our agent to compare between multiple equally valid plans and choose the one with the best expected value. An example of this would be the agent having to choose between treating diseases or transferring cards with its ally: while transferring cards would allow the agents to discover a cure, treating diseases could be equally important. If the players are one or two outbreaks away from losing the game and one of the next cards might trigger a game-losing outbreak, the agent would consider the expected value of that scenario to be much worse than the benefit of discovering the cure.

All of the previously explained mechanics were used to implement the planning agent. The planning agent is the obtained result of creating the planning model for the game Pandemic and fulfills the second objective of this research. The results of the evaluation of this agent will be presented in section [4.1](#).

3.3 Plan recognition

The third objective of this thesis is to develop a plan recognition model for Pandemic. The main goal to achieve with this objective is to allow the planning agent to analyze

and react to the actions of its cooperator and deduce what may be the current goal of that player. The agent can use this information to predict which actions the other player may perform in the future, to be able to cooperate with them.

As explained in section 3.2, a state-space representation is used to describe the planning. When performing the plan recognition we use the same representation used by the planning agent. The agent has knowledge of the actions the other players took and the different goals are evaluated to choose the one which best matches the observed actions. This approach basically transforms the problem of plan recognition to one of planning [RG09].

When performing a planning cycle, before starting with its plan, the plan recognition agent performs a recognition process on the actions taken by the player before it. The agent takes the state in which the previous turn started and performs a planning cycle on it using both the *discover cure* and the *survive* goals. For each of the goals a different action sequence is obtained, these action sequences are the equivalent of the agent figuring out which plan it would have executed were it in the position of the other player. The goal which produces the action sequence most similar to the observed actions taken by the player is chosen as the current goal for that player. The actions are evaluated in terms of their effect in the game, for example, treating the red disease would be considered equal to treating the blue disease in terms of purpose. Also, in case the player discovered a cure last turn, the actions to be considered would be the ones taken after the cure was discovered, since the goal of “discover cure” was already satisfied and what is important is the future goal and plans.

Once the agent recognized the goal the other player is trying to achieve, it performs its planning cycle as normal, with a single exception. When performing its planning process, the plan recognition agent will use the recognized goal as part of its planning when building up the action sequence the other player will do. The prediction of what the other players’ current goals are, allows the agent to make assumptions of the other players future actions and come up with the best plan to fulfill their goals.

This process was implemented in our plan recognition agent and is the obtained result of creating our model for plan recognition for the game Pandemic.

3.4 Data collection and comparison

The final objective of this thesis is the comparison of the developed agents with and without the plan recognition model when playing the game with a human player, measuring its win rate and the player's satisfaction. By the player's satisfaction we mean the player's perception in regard to how well the agent performed in game and how well the agent reacted to the player's actions.

This evaluation allows us to evaluate how well our agents are able to cooperate with a human player, and measure the impact that plan recognition has on the way intelligent agents cooperate and play games with human players. Our hypothesis is that plan recognition will improve the cooperation between the human and computer players: increasing the win rate and giving the human player a more enjoyable experience.

Two treatments were used to perform this evaluation: the intelligent agent using plan recognition and the basic planning agent as a control group. The participants are randomly and blindly assigned to one of these two treatments. Once each game is finished, a questionnaire is presented to the participant with questions regarding the players' previous experience with the game and questions regarding their interaction with the intelligent agent and how the agent reacted to their actions. The questionnaire presented to the participants is available in the appendix [A](#) and uses a 5-point Likert scale for evaluation of the agents.

With the purpose of reducing the noise produced by the randomness of the in game decks, five specific seeds for the random number generator were used to shuffle the card decks. The decks to be used in each game were randomly assigned at the beginning of each game. This process was also done to assign player roles and for the purpose of the experiment only the Scientist, Researcher and Medic roles were used. The Quarantine Specialist role was excluded, because the Quarantine Specialist and the Medic roles serve similar in game purpose, they are roles specialized in fighting the spreading of the diseases. Lastly, out of the three possible combinations available for those roles, only the Scientist-Medic and the Researcher-Medic combinations were used, to reduce to the noise generated by having multiple possible combinations.

The game platform developed in section [3.1](#) was made available online for volunteers to participate in the study by playing the game with an intelligent agent. The game was available online from June 8, 2020 until July 6, 2020 at the address: [pandemic](#).

slothlab.info. Before participating in the study, participants were presented with an informed consent form containing the information related to the experiment which they had to read and accept in order to participate.

Knowledge of the study was propagated through social media, specifically: Twitter, Facebook and Reddit. In Facebook and Twitter the post was made as public, asking volunteers to participate and if they wanted, share the publication or webpage with other people who might be interested. In Reddit a thread was opened in the subreddits *boardgames* and *samplesize* asking the community to participate in the study. These subreddits were chosen because of the areas of interest by the participants of such communities, which made them good subjects for the study.

With this experiment we obtained the results for the final objective of this thesis, which will be presented in the following chapter.

Chapter 4

Results

In this chapter we present the results achieved from the development of this thesis. This chapter is divided in two main sections: first the evaluation of the developed agents in individual tests, and second the results obtained from the experiment run with human participants.

4.1 Agent evaluation

To evaluate the agents developed, we used the same metrics that will be used when evaluating the results with human players: win rate and average number of cures discovered. The tests were performed with varying number of players (two to four) and epidemics (four to six), to see the impact these variables have in the performance of our agents. Additionally, to provide a baseline for comparison we implemented a simplified version of our agent called the “heuristic” agent, which decides its next action using the heuristic function and chooses the action which leads to the best possible state. Another, simpler “random” agent was implemented to see what would be the expected result for a player which just takes random actions each turns. These agents were chosen because there were no other at the time of development implemented for this game in literature which could be used as a direct comparison for our agents.

The agents were made to play 3000 games for each combination of number of players and epidemics. In each game, the players were instances of the agent being evaluated. Therefore for each agent a total of 27000 games were played and their results stored.

As expected, the random agent did not perform well, and did not win a single game

in any setup. It only managed to discover 5 cures over all 3000 games in the “easiest” setup (two players and four epidemics). The results for the remaining setups decrease in the number of cures discovered. Because of this, it can be concluded that Pandemic is not a game that can be won by luck alone and some kind of reasoning is required to win the game.

The results for the heuristic agent are shown in table 4.1. While there is a noticeable improvement compared to the random agent, the agent performed poorly overall which might indicate Pandemic is not a game that can be won by heuristic alone. A trend appears in both metrics for the heuristic agent: as the number of epidemics and the number of players increase, the win rate and average cures decrease. The decrease of both metrics from the increase of epidemics was to be expected as, in terms of the game, the number of epidemics in the game is used to select the difficulty of the game (four for easy, five for normal and six for hard). However, the lower win rate with a higher number of players was unexpected. An explanation for this is that, by increasing the number of players, the number of turns per player reduces and therefore cooperation becomes a bigger requirement to succeed. This, in turn, increases the difficulty of the game and reduces the win rate and average number of discovered cures. Lastly, it is worth noting that small changes in the average number of discovered cures seem to have a strong impact in the win rate: with a change of just 0.08 in the average number of cures making a difference of 1.67% in the win rate.

The results obtained for the planning agent are shown in table 4.1. The first thing to notice is the increase in both win rate and average discovered cures in comparison to the heuristic agent which is used as baseline. The planning agent manages to win around one third of games, while on average discovering 2.80 cures per game in the two-player four-epidemic configuration. Similar to the heuristic agent, it can be observed that the increase in number of players and number of epidemics decrease both metrics.

When analyzing the results of the planning agent in terms of role combinations (the randomly assigned roles the agents had during each game) there are also some trends that appear. This results can be seen in figure 4.1. We will examine these results by the number of players participating in the game.

In the two player games it is possible to see that the games in which the *Scientist* role was present had a higher win rate than the other scenarios. This can be explained by the fact that the Scientist requires fewer cards of the same colored disease to discover

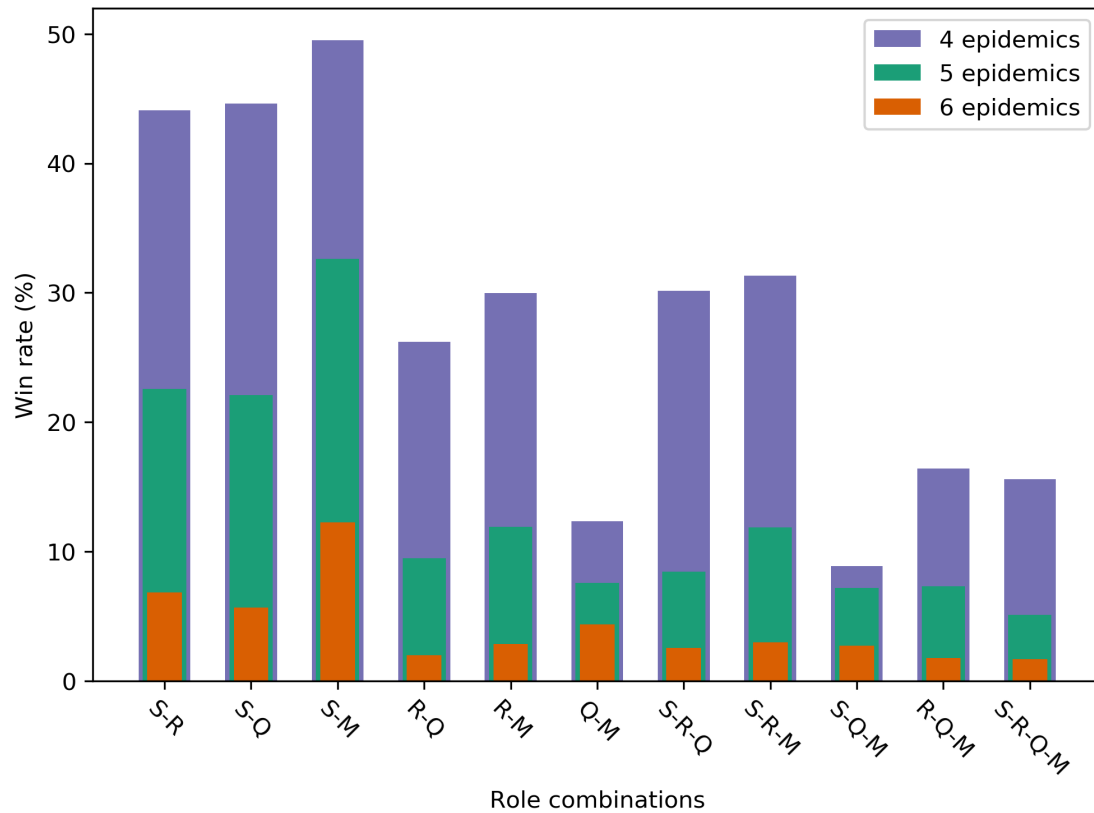


Figure 4.1: Observed win rate for the planning agent when using different role combinations. ‘S’ for Scientist, ‘R’ for Researcher, ‘Q’ for Quarantine Specialist and ‘M’ for Medic. Comparison of the results for each combination across different number of epidemics.

Table 4.1: Win rate and average number of cures discovered per game by the heuristic, planning and plan recognition agents over 3000 games with p players and e epidemics (up to $\pm 1.70\%$ for the win rate and ± 0.04 for the average number of cures discovered with a confidence of 95%).

Setup		Win rate			Average number of cures		
p	e	Heuristic	Planning	Recognition	Heuristic	Planning	Recognition
2	4	2.27%	34.27%	34.00%	0.90	2.80	2.80
2	5	0.83%	17.63%	17.60%	0.69	2.27	2.27
2	6	0.37%	5.67%	5.40%	0.52	1.67	1.66
3	4	0.60%	21.73%	18.07%	0.82	2.41	2.33
3	5	0.10%	8.73%	8.23%	0.57	1.83	1.73
3	6	0.03%	2.53%	2.27%	0.47	1.30	1.24
4	4	0.03%	15.60%	12.07%	0.49	2.22	2.14
4	5	0.00%	5.13%	3.90%	0.32	1.54	1.47
4	6	0.00%	1.70%	1.37%	0.21	0.92	0.88

a cure (four instead of five), therefore reducing the number of requirements to win the game. In the scenarios with the *Scientist* role, the ones with the *Medic* had a higher win rate than the ones with the *Quarantine Specialist* and the *Researcher* (with win rates 49.50%, 44.62% and 44.10% respectively). While the *Scientist-Researcher* combination has two abilities useful at discovering cures (since the *Researcher* can give cards to the *Scientist*), they lack any special abilities which allow them to stop the spread of the diseases which can explain why the other role combinations (*Scientist-Quarantine Specialist* and *Scientist-Medic*) have a higher win rate. As to why the *Scientist-Medic* combination has a higher win rate than the *Scientist-Quarantine Specialist*, this can be explained by the fact that the *Medic* can treat diseases without consuming actions once the cure for the disease has been discovered. Since the *Scientist* can quickly discover cures this can explain why their combination has the highest win rate of them all.

Among the two player combinations that do not have a *Scientist* among them, the ones with the *Researcher* are the ones with the highest win rate. This can be explained because the *Quarantine Specialist-Medic* combination (12.36% win rate) lacks any kind of benefit which assists in discovering cures which makes it difficult for them to discover all the cures to win the game. Among the two combinations with the *Researcher*, the *Researcher-Medic* is the one with the highest win rate (29.98% compared to 26.21%), this can be explained in the same way as for the *Scientist-Medic*, which seems to support

the idea that the *Medic* is slightly superior to the *Quarantine Specialist* role as a disease fighting role.

In the three player games we can divide the role combinations into those that have a double cure discovery focus (*Scientist-Researcher-X*) and the ones with double disease fighting focus (*X-Quarantine Specialist-Medic*). Of these two groups the first one has a higher win rate, this can be explained because of how the *Scientist-Researcher* dynamic helps players discover cures faster and having a third player which focuses solely on fighting the disease helps them keep the diseases at bay. In these combinations, once again, the combination with the *Medic* shows a small advantage (31.33% compared to 30.16%) over the one with the *Quarantine Specialist*. In the combinations with the double disease fighting focus the one with the *Researcher* has a higher win rate than the one with the *Scientist* (16.44% compared to 8.90%). This can be explained by the fact that, with a reduced number of turns per player (because of a larger number of players), a role which has more options to interact with the other players to discover cures has a better chance of discovering all the cures.

Lastly, since our version of *Pandemic* only contains four roles, there is only a single case to examine for the four player games. This four player game produces results similar to the three player games with the *Researcher-Quarantine Specialist-Medic* combination, with a win rate of 15.60%. This is an effect of the reduced number of turns for each player and the fact that each player has to wait three full turns before having the chance to act again, which increases the difficulty of the game since the future state of the game when the turn comes back to a player is harder to predict.

As for the plan recognition agent, the results of its evaluation is shown in table 4.1. One of the first thing to notice is that for the plan recognition agent, the results for two player games are similar to those of the planning agent. However, as the number of players increases, the overall win rate and average number of discovered cures is worse in comparison to the planning agent by up to 3.66%. This can be explained by the fact that, when playing games with more players, assuming players will stick to their original plan after two or three turns is counterproductive. In the cases of games with three or four players, players tend to rethink their plans and goals every time they take their turn, because the game state has changed a lot in between. Because of this, in games with a higher number of players, the plan recognition agent performs worse than the planning agent.

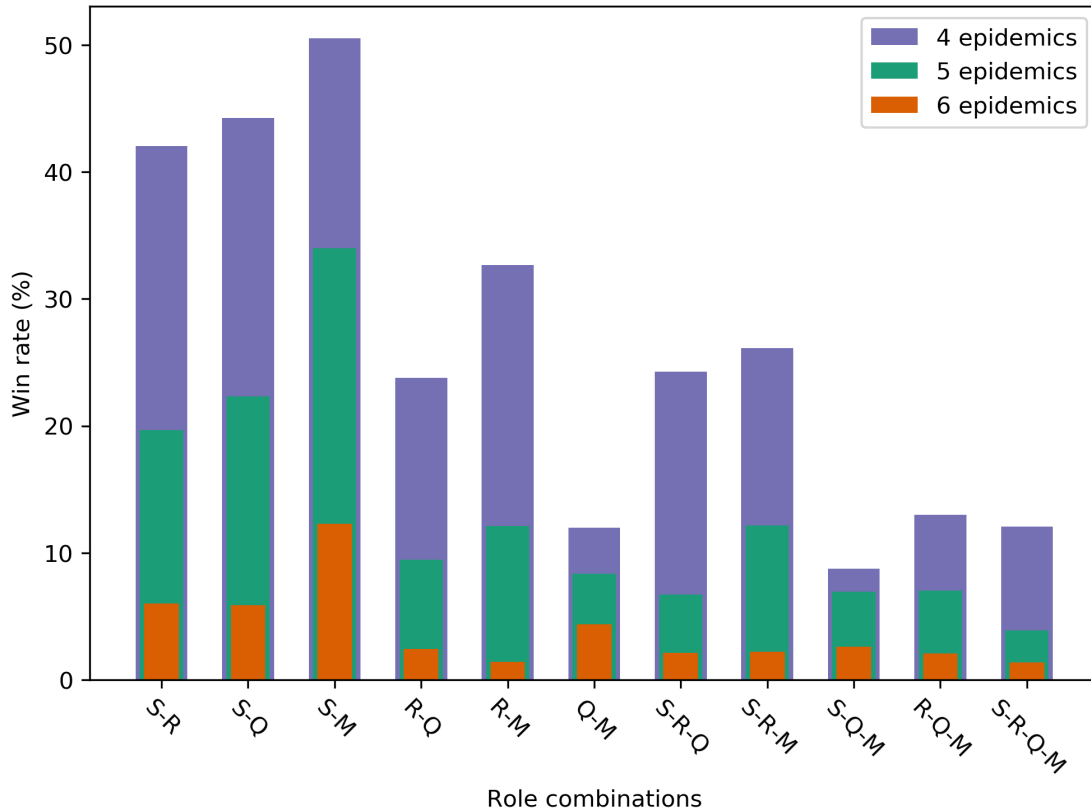


Figure 4.2: Observed win rate for the plan recognition agent when using different role combinations. ‘S’ for Scientist, ‘R’ for Researcher, ‘Q’ for Quarantine Specialist and ‘M’ for Medic.

Finally, the results for the plan recognition agent in terms of role combinations can be seen in figure 4.2. It is possible to observe that these results present the same trends for the two, three and four players games as for the planning agent. This was expected, as both of the agents use the same planning algorithm and heuristics to develop their plan. Additionally, the plan recognition performed by the agent should recognize the action pattern performed by the previous agent easily, as it is the same it would have executed.

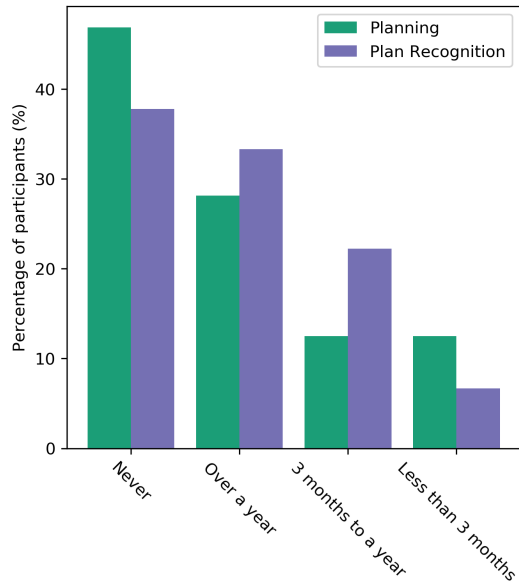
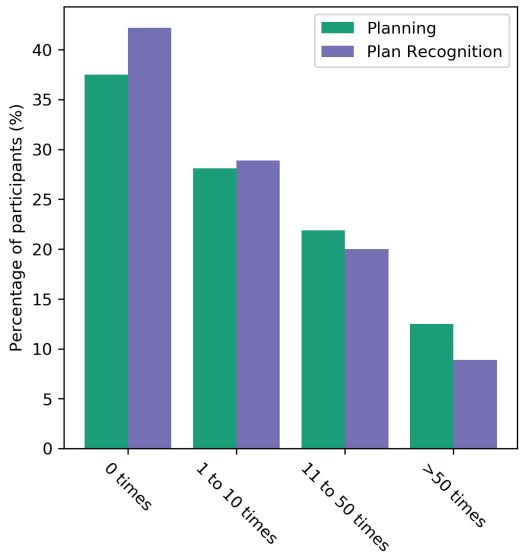
4.2 Experiment results

In this section we will analyze the results obtained from the experiment ran with human participants explained in section 3.4. We were able to recruit a total of 116 participants which played 187 games. However, for the purpose of this analysis we will only take into account the first game played by each participant, for a total of 116 games. Of these 116 games, only 77 participants answered the survey after the game, but not all of these participants answered all the questions. A total of 90 participants were assigned to the planning agent and 97 to the plan recognition agent. However, after the previously mentioned filters, only 32 participants answered the survey for the planning agent and 45 for the plan recognition agent.

As for the demographic of the population, we asked the participants about their previous experience in the game Pandemic. The results to these questions can be seen in the figure 4.3. Participants were asked how many times had they played the game before (figure 4.3a), how long ago was the last time they played (figure 4.3b) and how often do they manage to win the game (figure 4.3c). It is possible to see that the majority of the participant population is formed by people with little to no experience in the game: about two thirds of the population have played fewer than eleven games, with over a third being participants who have never played the game. This division is strongly correlated (Spearman’s rank correlation coefficient of 0.76) by how often the players win the game, with about a third of the players winning above one in four games. Another thing to notice is that, of the players with any previous experience in the game, about half of them have not played the game in over a year.

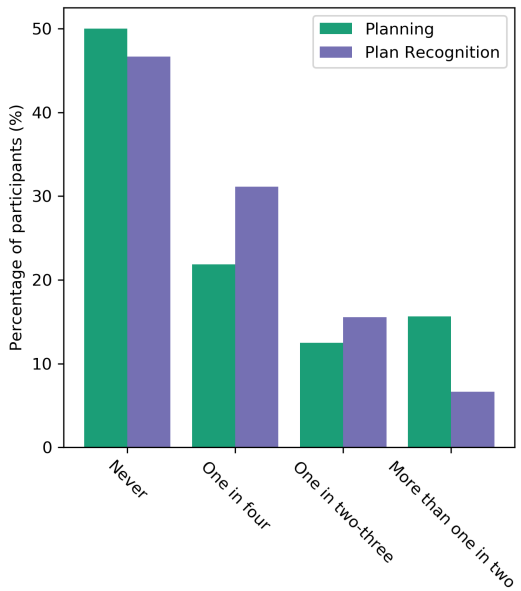
4.2.1 Survey results

In this subsection we present the results of the evaluation done by the players. After each game players were asked to evaluate the agent they played with. The results for this evaluation are shown in figure 4.4. The agents were evaluated in terms of their perceived skill (figure 4.4a), whether the participants understood what the agent was doing (figure 4.4b) and its helpfulness towards the participant (figure 4.4c) using a 5-point Likert scale. It can be observed that while overall the plan recognition agent was rated higher by the participants in many cases, this grading was not consistent. When evaluating



(a) Previous experience with the game by number of times played.

(b) Time passed since the last time the game was played.



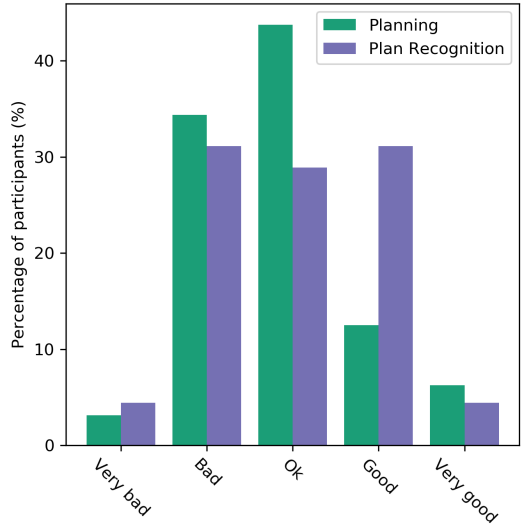
(c) How often the participant wins a game.

Figure 4.3: Survey answers of the participants' previous experience with the game Pandemic.

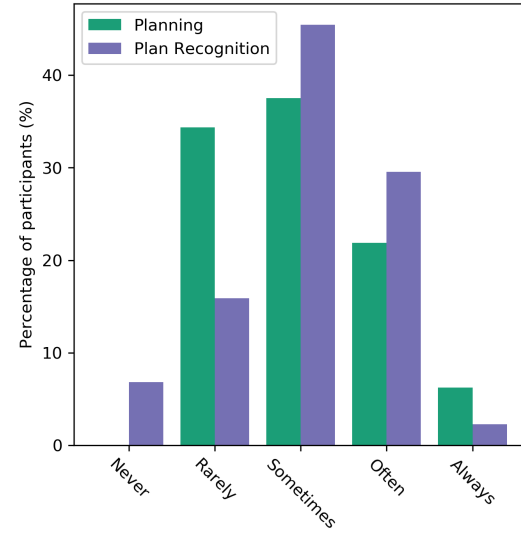
the skill of the agents, the plan recognition agent presents a higher number of ratings on the “good” qualification, however seems to perform worse at both extremes (having more “very bad” and fewer “very good” evaluations than the planning agent). A similar behaviour can be seen in how the participants rated their understanding of the agent, with the plan recognition agent performing worse at both extremes but having better results at the middle-positive qualifications. In contrast, the helpfulness evaluation seems to have a different behaviour, with the plan recognition agent performing better in the lower grades (has less “never” and “rarely” evaluations) and also a higher number of evaluations in the “always” (highest) grading. While this qualitative examination supports our hypothesis, we performed further statistical analysis of the obtained data to get more accurate conclusions.

First, we examined the evaluations performed by the participants for correlation, by calculating their Spearman’s rank correlation coefficient. This coefficient is used to measure the statistical dependence between rankings of two variables. Calculating these coefficients allows us to measure the correlation between the skill, understanding and helpfulness in the evaluations done by the participants. The coefficients for the planning agent and the plan recognition agent are shown in table 4.2 and table 4.3 respectively. From these results it can be concluded that the values for skill and helpfulness show a strong correlation and a moderate correlation exists between understanding and helpfulness. The strong correlation give us insight as to how the participants see helpfulness as a defining characteristic of a skilled player. The participants may consider that a skilled player is more aware of the current game state and more proactive to assist its teammates which would explain the correlation. Regarding the correlation between skill and understanding there is a difference between both agents, with the planning agent showing a moderate correlation and the plan recognition agent a weak one. It is interesting that for the plan recognition agent with the higher correlation between skill and helpfulness (0.76 compared to 0.69), the correlation between understanding and skill is lower. This might imply that, for players, it is more important that a player is helpful than being able to understand it when assessing their skill level.

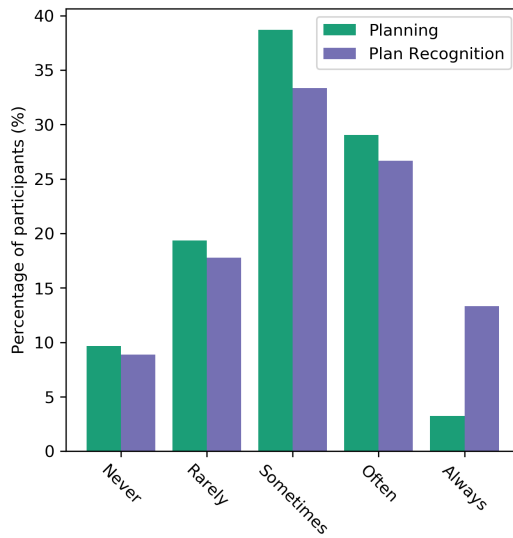
To evaluate the differences between how the participants rated the agents, the data was processed using two different tests. We compared the evaluations for the three different survey questions using a Mann-Whitney U test to test for difference in qualitative value, and a χ^2 test to test for a difference in distribution, using Holm-Bonferroni



(a) Skill level of the agent perceived by the participant.



(b) How often the participant understood the agent.



(c) How often the agent was helpful towards the participant.

Figure 4.4: Survey answers of the participants' impression on the agents.

Table 4.2: Spearman’s rank correlation coefficient between evaluated skill, understanding and helpfulness for the planning agent.

	Skill	Understanding	Helpfulness
Skill	-	0.44	0.69
Understanding	0.44	-	0.53
Helpfulness	0.69	0.53	-

Table 4.3: Spearman’s rank correlation coefficient between evaluated skill, understanding and helpfulness for the plan recognition agent.

	Skill	Understanding	Helpfulness
Skill	-	0.15	0.76
Understanding	0.15	-	0.35
Helpfulness	0.76	0.35	-

method to account for multiple testing. The Mann-Whitney U test showed no statistically significant difference for any of the three metrics. However, the χ^2 test showed that a statically significant difference ($p < \frac{0.05}{3}$) in the “understanding” metric but for the remaining metrics the results were inconclusive.

Finally, even though not relevant to the objective of this thesis, the participants were also requested to evaluate how easy to use the game client was. The results to this question can be seen in figure 4.5. The results show that more than half of the participants didn’t considered the game client difficult to use, but none of them considered it very easy to use. Also, the percentage of participants that consider the client difficult (29.9%) or very difficult (7.8%) to use is considerable and further improvement should be considered.

4.2.2 Performance results

In this section we will analyze the success, in terms of the game, participants had when playing the game with the agents. The success will be measured in terms of the average number of won games and average number of discovered cures per game, which are the most relevant metrics to the performance of the agents. To this end, the first game played by each of the 116 participants will be grouped according to which of the agents were they assigned to. Using this groupings, a total of 51 participants were assigned to

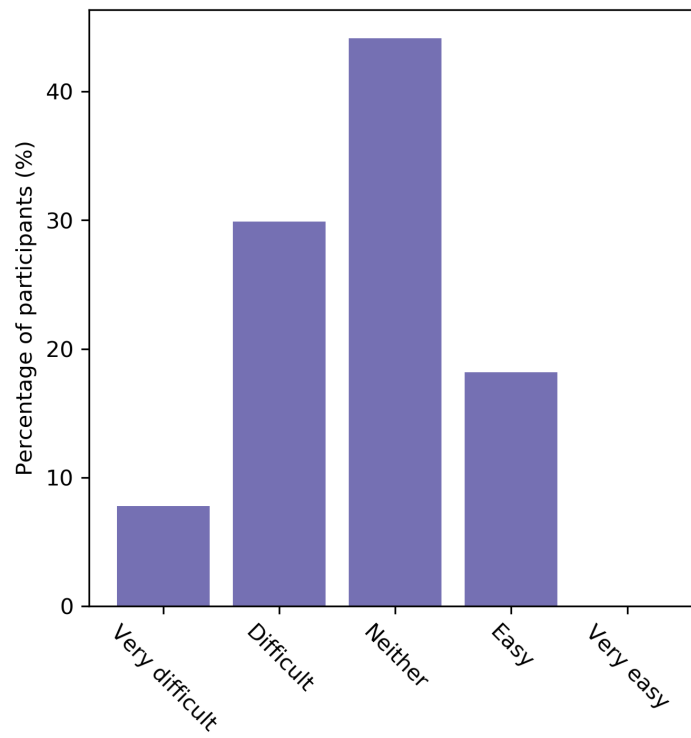


Figure 4.5: Ease of use of the client interface by the participants.

Table 4.4: Observed win rate and average discovered cures for the agents with human participants in the first game played by each participant.

Agent	n	Win rate	Average cures
Planning agent	51	25.49%	2.49
Plan recognition agent	65	27.69%	2.72

the planning agent, while the remaining 65 were assigned to the plan recognition agent.

The obtained win rate and the average number of discovered cures for the agents are shown in table 4.4. The results seem to indicate that the plan recognition agent outperformed the planning agent, even though just slightly, in both of the metrics used. A Student's t-test was applied to determine if there was a difference in the average number of discovered cures between the agents, and did not find a statistically significant difference.

It can be pointed out, that the win rate was lower when the agent played with human participants than when it played with itself: 25.49% in comparison to 34.27% for the planning player and 27.69% in comparison to 34.00% for the plan recognition agent. This can be explained by the fact that a large portion of the participants did not have much experience with the game. Another factor to take into account is that the participants were not been accustomed to the game platform and playing with the agent.

The reasons why each of the games ended for the different agents are shown in figure 4.6. In the observed results there is a difference regarding the number of times the players lost the game due to outbreaks: the number of times the players lost by outbreaks is lower in the plan recognition agent with an increase in the number of times players lost due to running out of time. However, this difference is not statistically significant.

When considering only the second and later games of participants which played more than one game the results change dramatically as shown in table 4.5. For these games the win rate and average discovered cures for both of the agents went up and is even higher than those registered for the agents playing with themselves. The observed win rate for the planning agent is higher than the one of the plan recognition agent, but what is interesting is the fact that the plan recognition agent had a bigger observed average number of discovered cures. Nonetheless, this difference is not statistically

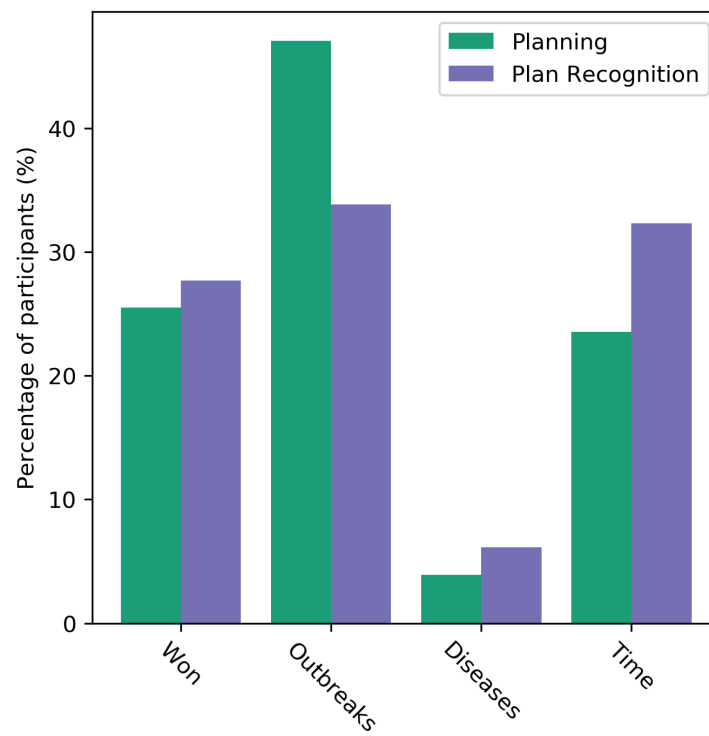


Figure 4.6: Distribution of the reasons for the game to end for each of the agents.

Table 4.5: Observed win rate and average discovered cures for the agents with human participants in the second and later games.

Agent	n	Win rate	Average cures
Planning agent	39	41.02%	2.92
Plan recognition agent	32	40.63%	3.09

significant.

4.2.3 Qualitative analysis of the agents

In this section, the game logs for the worst and best rated games of the planning and plan recognition agent are evaluated. The purpose of this is to gain insight as to the possible reasons why a player gave a specific rating to the agent and, because of this, only participants with previous experience about the game will be considered.

For the planning agent the worst rated game is the game log 167¹. In this game the participant was assigned the Medic role and the planning agent the Researcher role. The random decks #3 were assigned to this game. The game was characterized by an early discovery of the red cure as a result of the agent transferring its cards to the human player and a rush to eradicate the disease by the medic. This was followed by a wave of black outbreaks at the center of the map, which the medic had to attend to and the discovery of the black cure by the planning agent. After the discovery of the latest cure, the medic player proceeded to eradicate the disease, which helped the players reduce the cities they had to worry about. Afterwards the players managed to regroup on the American continent which allowed the planning agent to transfer its yellow cards to the human player, forcing the player to discard its less useful cards. With three cures discovered, the players focused on discovering the last (blue) cure and winning the game. While the game had a successful resolution, the agent was rated poorly which raises questions as to why. However, observing the game log it is possible to observe that the planning agent sometimes “stalled” its turn waiting for new cards, performing actions that the human player could see as wasteful (e.g: moving from London to New York, to treat a disease and then moving back to London). The fact that the agent forced the human player to discard might not have been well received by the participant, since wasting cards is usually a poor strategy in the game.

¹Game ID: 5a939559ed263d82

The highest rated game for this agent is the game log 168². In this game the participant was assigned the Scientist role and the planning agent the Medic role. The random decks #4 were assigned to this game. The game was characterized by a good distribution of assignments with the planning agent focusing on treating the disease while the scientist focused on containing a portion of the map and discovering cures. The scientist managed to discover the cure for the black disease relatively early in game, allowing the medic to eradicate it further ahead. The medic discovered the cure for the red disease, followed shortly by the scientist discovering the cure for the yellow disease. However, at this point both players had used blue cards as part of their movement actions which made it impossible to discover the last remaining cure and the players lost the game. In its last turn the planning agent managed to eradicate the red disease. Analyzing the log, the reason behind the good rating of the agent might come from the fact that it served its role with great dedication. As the medic the agent was expected to focus on treating the diseases, which the agent did diligently and even managed to eradicate two of the diseases. The eradication of diseases is not required to win the game, but allows players to roam more freely without worrying about those diseases and its performance might have impressed the participant to give it a good rating.

For the plan recognition agent the worst rated game is the game log 3³. In this game the participant was assigned the Medic role and the plan recognition agent the Scientist role. The random decks #2 were assigned to this game. This game was characterized by a difficult start with three outbreaks which forced the participant to move quickly to treat the spreading diseases. The scientist remained near the research station at Atlanta fighting diseases and then found the cure for the black disease. The medic then proceeded to fight the disease in the central portion of the map and managed to eradicate it. In the meantime the agent continued its watch near Atlanta and discovered the cure for the red disease before moving to Seoul, regrouping with the medic, and building a research station. The medic used this new research station to discover the cure for the blue disease. However shortly after this, the yellow disease reached a critical point in which it spread without control and caused a chain of outbreaks which caused the players to lose the game. In the case of this game, the main reason behind the rating given by the participant could be from the focus the agent had toward discovering cures

²Game ID: 90bb2d5545cf1ce5

³Game ID: 77bd9d679a9a31e3

instead of helping out the participant with treating the diseases. The agent spent too much time treating diseases in cities which posed little risk instead of engaging with the medic to treat highly affected areas, in the end this allowed diseases to spread dangerously in a specific part of the map causing the players to lose the game.

Finally, the highest rated game for the plan recognition agent is the game log 171⁴. In this game the participant was assigned the Medic role and the plan recognition agent the Researcher role using the random deck #5. This game was characterized by a starting big transfer of cards from the plan recognition agent towards the participant, which allowed the participant to discover the cure for the black disease by its second turn. A series of outbreaks occurred at different locations but the players managed to treat the spreading diseases. Afterwards the agent managed to discover the cure for the red disease and the participant the cure for the yellow disease. Afterwards another series of outbreaks occurred through the map, but the players managed to treat the diseases and then group together for a series of card exchanges to happen. Finally the agent managed to discover the last missing cure and win the game. In this game the agent managed to keep an active role throughout the game, helping the participant fight the diseases and trading cards actively to discover two of the four cures. This behaviour could explain the grading done by the participant which was the highest rated game for this agent.

From the analysis of these games we can conclude what kind of behaviours participants seemed to regard as positive or negative. Participants tended to grade poorly when agents performed actions that could be considered wasteful: either allowing cards to be discarded, not participating actively in what the participant considered the priority goal and also performing actions with perceived low value. On the other hand, agents that used actions more effectively and performed noticeable actions for the participant were rated higher. Transferring cards for the participant to discover a cure seems to improve the rating, as long as it does not force the player to discard any cards.

⁴Game ID: 07f5221758f38062

Chapter 5

Conclusions

In this chapter we will present the conclusions obtained from the work developed in this thesis and how the produced artifacts and results satisfy the specific objectives.

5.1 Game platform

The developed game platform was presented in section 3.1, as well as the specification of how it was implemented. This platform performed satisfactorily as a platform to be used to allow interaction between human participants and artificial intelligence agents as shown in section 3.4.

However, while the platform is successful in allowing the interaction with a large number of players, and collect their respective answers, it was not graded as easy to use by the participants. About 40% of the participants graded the platform as neither easy nor difficult to use, and almost 30% of them graded it as difficult to use. This points towards possible improvements to be done to the game platform to make it easier to use to human players.

5.2 Intelligent agents

The developed agents proved to be good at playing the game with themselves and with human players. While still below human skill level, the agents themselves managed to win about a third of the games playing by themselves which is a milestone in research regarding the Pandemic game. At the time we started our work, no other agent existed

that would have been suitable for our purposes, which is why we had to implement our own versions of a baseline agent (the heuristic agent). In this sense, our agents can provide a baseline for future research on the subject and provide a new example of possible applications to planning as a way of creating intelligent agents to cooperate with human players.

Some of the results presented in this thesis have been published in articles at the Artificial Intelligence and Interactive Digital Entertainment (AIIDE) conference. The problem domain and game platform were presented in AIIDE 2019 as part of the Experimental AI in Games (EXAG) workshop in an article titled *Pandemic as a Challenge for Human-AI Cooperation* [SCE19]. The results for both the heuristic and planning agent were presented in the AIIDE 2020 conference in an article titled *PAIndemic: A planning agent for Pandemic* [SCE20] along with the codes for them. This article and its code was nominated (along with three other papers) to best artifact at the conference and was acknowledge of interest to the community.

5.3 Comparing the performance of the agents

The performance of the planning and plan recognition agents proved satisfactory on the both the standalone tests and the evaluation done by the participants of the experiment. In the standalone tests both of the agents outperformed the heuristic agent used as baseline and obtained similar results for the two player versions of the game. This was to be expected since the interpretation of a possible plan resulting from a planning agent should be the same as the plan the agent would have performed itself.

In the experiment with the human participants, we were unable to demonstrate a significant difference between the two variants of our agent. We collected data from 187 games, but with the restrictions applied, these games only provided us with 77 samples of data from players answering the survey. This posed a challenge for the statistical analysis due to the size of the sample which might have produced more significant results with more samples. Another explanation for the lack of an observable difference between the agents could also be that, while the agents might have different behaviour, the difference is likely small which makes it difficult to measure among the noise provided by other factors (for example, the participants).

One result to highlight, though, is the statistically significant difference between

the two agent types in terms of the participants' understanding of the agents. This difference being positive or negative still was not conclusive, but provides some evidence that plan recognition has an impact on cooperation with human players.

5.4 Future Work

While our agent provides a strong baseline for Pandemic playing agents, our work directly enables several future improvements of our approach.

The implemented agent could be improved either by defining a new heuristic function or by developing a new evaluation function that chooses the weights to be used by the heuristic function at each turn during the game. Further research of this topic could focus on other types of agents, perhaps using neural networks through reinforcement learning or through a genetic algorithm approach (like a rolling horizon evolutionary algorithm). A neural network could also be used together with an heuristic function to evaluate the states in a hybrid approach.

The experiment could be reproduced with a larger population or a particular group of interest (e.g: people which have played the game before) to obtain more precise results which would allow to confirm a difference should there be one. For the purpose of allowing reproduction of the presented methodology and to allow further research to be performed, we have made parts of our code available in Github in the repository: github.com/BlopaSc/PAIndemic. The repository will contain all the needed source code and a guide about how to use the developed software. The missing source code will be added after the publication of the papers relating to them (the developed agents). All of the assets utilized in the development of the game client were obtained from open source with open licenses which request the authors to be attributed for their work (the authors are duly credited inside the game client).

The approach presented in our work could be applied to other cooperative games and problems as future work. The use of a planning agent for a turn-based problem with an heuristic could be used for cooperative games as *The Big Book of Madness* or *Eldritch Horror*. Plan recognition planning agents could be applied to games with a "traitor" mechanic (a feature of cooperative games in which one of the players is a traitor scheming against the other players), which poses an interesting problem regarding belief modeling.

Lastly, further improvements to the platform could be obtained through an evaluation of usability or using a study group to come forth with possible improvements. The idea of reusability was central when developing the artifacts of this thesis to allow further development of agents to be done upon the platforms developed.

Appendix A

Player Experience Questionnaire

A.1 Previous game experience

1. How familiar are you with the board game Pandemic?
 - I have never played before participating in this experiment
 - I have played a few (1-10) times
 - I have played multiple (11-50) times
 - I have played many (>50) times
2. When was the last time that you played Pandemic before this experiment?
 - I have never played before or can't remember when I played the last time
 - The last time I played has been a long time (over a year) ago
 - The last time I played has been some time (between 3 months and a year) ago
 - The last time I played was recently (up to 3 months ago)
3. How often do you win a game of Pandemic with 4 epidemics?
 - I never win or I have never played Pandemic
 - I almost never win the game (about one in 4 or more games)
 - I sometimes win the game (about one in 2 or 3 games)
 - I often win the game (more than one in 2 games)

A.2 Experience with the AI agent

1. How easy to use was the User Interface (game display/controls)?
 - The User Interface was very difficult to use
 - The User Interface was difficult to use
 - The User Interface was neither hard nor easy to use
 - The User Interface was easy to use
 - The User Interface was very easy to use
2. How would you rate the play skill of this AI agent?
 - The AI agent played very badly
 - The AI agent played badly
 - The AI agent played ok
 - The AI agent played well
 - The AI agent played very well
3. How often did you understand what the AI agent was trying to do?
 - I never understood what the AI agent was trying to do
 - I rarely understood what the AI agent was trying to do
 - I sometimes understood what the AI agent was trying to do
 - I often understood what the AI agent was trying to do
 - I always understood what the AI agent was trying to do
4. How often did the AI agent helped you achieve your plans and goals?
 - The AI agent never helped me achieve my plans and goals
 - The AI agent rarely helped me achieve my plans and goals
 - The AI agent sometimes helped me achieve my plans and goals
 - The AI agent often helped me achieve my plans and goals
 - The AI agent always helped me achieve my plans and goals

Bibliography

- [Bau10] Antoine Bauza. Hanabi. <https://boardgamegeek.com/boardgame/98778/hanabi>, 2010. Online; accessed 25 February 2019.
- [BCP⁺16] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *CoRR*, abs/1606.01540, 2016.
- [BFC⁺19] Nolan Bard, Jakob N. Foerster, Sarath Chandar, Neil Burch, Marc Lanctot, Francis Song, Emilio Parisotto, Vincent Dumoulin, Subhodeep Moitra, Edward Hughes, Iain Dunning, Shibl Mourad, Hugo Larochelle, Marc G. Bellemare, and Michael Bowling. The hanabi challenge: A new frontier for AI research. *CoRR*, abs/1902.00506, 2019.
- [BFGM06] Michael Bowling, Johannes Fürnkranz, Thore Graepel, and Ron Musick. Machine learning and games. *Machine Learning*, 63(3):211–215, Jun 2006.
- [BG01] Blai Bonet and Héctor Geffner. Planning as heuristic search. *Artificial Intelligence*, 129:5–33, 2001.
- [BJ11] Christer Bäckström and Peter Jonsson. All PSPACE-complete planning problems are equal but some are more equal than others. In Daniel Borrajo, Maxim Likhachev, and Carlos Linares López, editors, *SOCS*. AAAI Press, 2011.
- [BNVB13] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, jun 2013.
- [Boa] Ultra BoardGames. Tips to win pandemic. ultraboardgames.com/pandemic/tips.php. Online; accessed 3 April 2019.
- [Byl94] Tom Bylander. The computational complexity of propositional strips planning. *Artif. Intell.*, 69(1-2):165–204, September 1994.
- [Car01] Sandra Carberry. Techniques for plan recognition. *User Modeling and User-Adapted Interaction*, 11(1):31–48, Mar 2001.
- [Con15] Chris Conway. GOAP in Tomb Raider. Game Developers Conference, 2015.
- [Cou06] Rémi Coulom. Efficient selectivity and backup operators in Monte-Carlo tree search. In *In: Proceedings Computers and Games 2006*. Springer-Verlag, 2006.
- [CRY15] Rogelio Cardona-Rivera and Robert Young. Symbolic plan recognition in interactive narrative environments, 2015.
- [CWP12] P. I. Cowling, C. D. Ward, and E. J. Powley. Ensemble determinization in monte carlo tree search for the imperfect information card game magic: The gathering. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(4):241–257, 2012.

- [EMS+20] M. Eger, C. Martens, P. Sauma Chacón, M. Alfaro Córdoba, and J. Hidalgo Cespedes. Operationalizing intentionality to play hanabi with human players. *IEEE Transactions on Games*, pages 1–1, 2020.
- [FN71] Richard E. Fikes and Nils J. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. In *Proceedings of the 2Nd International Joint Conference on Artificial Intelligence, IJCAI'71*, pages 608–620, San Francisco, CA, USA, 1971. Morgan Kaufmann Publishers Inc.
- [Hig15] Peter Higley. GOAP from FEAR to Shadow of Mordor. Game Developers Conference, 2015.
- [HNR68] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, July 1968.
- [Isl05] Damian Isla. Handling complexity in the Halo 2 AI. Game Developers Conference, 2005.
- [Jam17] Trevor James. Strategy tips for new players. boardgamegeek.com/thread/1813968, 2017. Online; accessed 3 April 2019.
- [KBTB14] Dirk P. Kroese, T. Brereton, T. Taimre, and Z. Botev. Why the monte carlo method is so important today. *Wiley Interdisciplinary Reviews: Computational Statistics*, 6:386–392, 2014.
- [Lan18] Nathan Langley. Unraveling Riot’s critical strike algorithm. <https://doranslab.gg/articles/crit-strike-algorithm.html>, August 2018. Online; accessed 11 January 2020.
- [Lea08] Matt Leacock. Pandemic. <https://boardgamegeek.com/boardgame/30549/pandemic>, 2008. Online; accessed 25 February 2019.
- [MU49] Nicholas Metropolis and S. Ulam. The monte carlo method. *Journal of the American Statistical Association*, 44(247):335–341, 1949. PMID: 18139350.
- [Nil84] Nils Nilsson. Shakey the robot. *SRI International Technical Note*, 323:135, 04 1984.
- [NT12] Kenichiro Nakai and Yasuhiko Takenaga. NP-completeness of pandemic. *Journal of Information Processing*, 20(3):723–726, 2012.
- [Ope18] OpenAI. Gym: a toolkit for developing and comparing reinforcement learning algorithms. <https://gym.openai.com/>, 2018. Online; accessed 25 September 2018.
- [Ork06] Jeff Orkin. Three states and a plan: The A.I. of F.E.A.R., 2006.
- [Rab13] Steven Rabin. *Game AI Pro: Collected Wisdom of Game AI Professionals*. A. K. Peters, Ltd., USA, 2013.
- [RG09] M. Ramirez and H. Geffner. Plan recognition as planning. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI-09)*, 2009.
- [SCE19] Pablo Sauma-Chacón and Markus Eger. Pandemic as a challenge for human-AI cooperation. In *Proceedings of the AIIDE workshop on Experimental AI in Games*, 2019.
- [SCE20] Pablo Sauma-Chacón and Markus Eger. PAIndemic: A planning agent for Pandemic. 2020.

- [SHM⁺16] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, jan 2016.
- [SL20] Konstantinos Sfikas and Antonios Liapis. Collaborative agent gameplay in the pandemic board game. In *Proceedings of the 15th International Conference on the Foundations of Digital Games*, 2020.
- [SPPF09] Immanuel Schweizer, Kamill Panitzek, Sang-Hyeun Park, and Johannes Fürnkranz. An exploitative monte-carlo poker agent. In Bärbel Mertsching, Marcus Hund, and Zaheer Aziz, editors, *KI 2009: Advances in Artificial Intelligence*, pages 65–72, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
- [SW10] Claude Sammut and Geoffrey I. Webb, editors. *Agent*, pages 35–35. Springer US, Boston, MA, 2010.
- [VBC⁺19] Oriol Vinyals, Igor Babuschkin, Junyoung Chung, Michael Mathieu, Max Jaderberg, Wojtek Czarnecki, Andrew Dudzik, Aja Huang, Petko Georgiev, Richard Powell, Timo Ewalds, Dan Horgan, Manuel Kroiss, Ivo Danihelka, John Agapiou, Junhyuk Oh, Valentin Dalibard, David Choi, Laurent Sifre, Yury Sulsky, Sasha Vezhnevets, James Molloy, Trevor Cai, David Budden, Tom Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Toby Pohlen, Yuhuai Wu, Dani Yogatama, Julia Cohen, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Chris Apps, Koray Kavukcuoglu, Demis Hassabis, and David Silver. AlphaStar: Mastering the real-time strategy game StarCraft II. <https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/>, 2019. Online; accessed 24 February 2019.
- [VdBDR09] Guy Van den Broeck, Kurt Driessens, and Jan Ramon. Monte-carlo tree search in poker using expected reward distributions. In Zhi-Hua Zhou and Takashi Washio, editors, *Advances in Machine Learning*, pages 367–381, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
- [Wel94] Daniel S Weld. An introduction to least commitment planning. *AI magazine*, 15(4):27, 1994.
- [WPC⁺12] James R. Wallace, Joseph Pape, Yu-Ling Betty Chang, Phillip J. McClelland, T.C. Nicholas Graham, Stacey D. Scott, and Mark Hancock. Exploring automation in digital tabletop board game. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work Companion*, CSCW '12, pages 231–234, New York, NY, USA, 2012. ACM.
- [YT18] Georgios N. Yannakakis and Julian Togelius. *Artificial Intelligence and Games*. Springer, 2018. <http://gameaibook.org>.