

TECHNICAL UNIVERSITY OF CRETE  
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING



**A serious game approach to promote  
cooperative civic actions at local level**

DIPLOMA THESIS

***Athina Georgara***

COMMITTEE:

Advisor : GEORGIOS CHALKIADAKIS, Associate Professor  
Committee Member : AIKATERINI MANIA, Associate Professor  
Committee Member : PANAGIOTIS PARTSINEVELOS, Assistant  
Professor



## **Abstract**

*Cooperative games are undoubtedly a very significant part of game theory. This games' category deals with situations where more than one individuals/agents/robots needs to form a group and collaborate with others in order to perform a particular action. There is a great deal of research on coalition formation and its stability. Despite the wealth of research on cooperative games, a trending and unresolved question is what someone should do in situations where there is partial or no information, i.e. under uncertainty.*

*This thesis proposes a novel hybrid model which combines techniques from hedonic games, a non-transferable utility class of cooperative games, and  $k$ -weighted voting games with transferable utility. We transform the concept of coalition formation used in hedonic games to a  $k$ -weighted voting game. Specifically, we target settings modeled by a novel class of cooperative  $k$ -weighted voting games with hedonic preferences, in which players have partial information; in these settings each player has 'empathy' beliefs on other players and therefore preferences over coalitions, while assuming no knowledge regarding others players' weights or the coalition's utility.*

*Furthermore, we have designed and implemented an example serious game application using our proposed model. Our game models a local community where players form coalitions and cooperate in order to perform several civic activities, and each activity can be represented as a hedonic  $k$ -weighted voting game.*

## **Abstract in Greek**

Τα συνεργατικά παίγνια αναμφισβήτητα αποτελούν σημαντική ενότητα της θεωρίας παιγνίων. Αυτή η κατηγορία παιγνίων ασχολείται με τις περιπτώσεις, κατά τις οποίες περισσότεροι από ένα άτομα / πράκτορες / ρομπότ καλούνται να σχηματίσουν ομάδες και να συνεργαστούν μεταξύ τους, προκειμένου να εκτελέσουν μια συγκεκριμένη ενέργεια. Έχουν διεξαχθεί αρκετές έρευνες σχετικά με τον σχηματισμό συνασπισμών και τη χρησιμότητά τους. Ωστόσο, το καίριο ερώτημα είναι το εξής: τι θα μπορούσε κανείς να κάνει στην περίπτωση όπου διαθέτει μερική ή και καθόλου πληροφόρηση, δηλαδή υπό αβεβαιότητα; Σημαντικό κομμάτι της έρευνας έχει αφιερωθεί και σε αυτό το ανοιχτό ερώτημα.

Η παρούσα διπλωματική εργασία προτείνει ένα καινοτόμο υβριδικό πρότυπο που συνδυάζει τεχνικές από τα ηδονιστικά συνεργατικά παίγνια (hedonic games) και τα  $k$ -σταθμισμένα συνεργατικά παίγνια ψηφοφορίας ( $k$ -weighted voting games). Εφαρμόζουμε τη λογική του σχηματισμού συνασπισμών που χρησιμοποιείται στα ηδονιστικά παίγνια, μια κλάση παιγνίων χωρίς μεταβιβάσιμη ωφέλεια, σε ένα  $k$ -σταθμισμένο παίγνιο ψηφοφορίας, μια κλάση παιγνίων με μεταβιβάσιμη ωφέλεια. Στοχεύουμε σε περιπτώσεις που μοντελοποιούνται με μια νέα κλάση ηδονιστικών  $k$ -σταθμισμένων παιγνίων ψηφοφορίας που εισάγουμε, στα οποία οι παίκτες έχουν μερική πληροφόρηση. Στις περιπτώσεις αυτές, ειδικότερα, κάθε παίκτης έχει πεπονηθείσεις ενσυναισθησίας για τους υπόλοιπους παίκτες, και ως εκ τούτου προτιμήσεις για τους πιθανούς συνασπισμούς, ενώ παράλληλα, δεν έχει καμία γνώση για τα βάρη των λοιπών παικτών, ούτε και για τη χρησιμότητα του συνασπισμού.

Επιπλέον, έχουμε σχεδιάσει και υλοποιήσει ένα “σοβαρό” παιχνίδι - παράδειγμα, που εφαρμόζει στο προτεινόμενο πρότυπο. Το παιχνίδι μας προσομοιώνει μια τοπική κοινότητα, όπου παίκτες σχηματίζουν συνασπισμούς και συνεργάζονται, προκειμένου να εκτελέσουν διάφορες δραστηριότητες, οι οποίες μπορούν να αναπαρασταθούν ως ένα ηδονιστικό  $k$ -σταθμισμένο παίγνιο ψηφοφορίας.

# Contents

<b>Abstract</b>	<b>iii</b>
<b>Abstract in Greek</b>	<b>iv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation and Contributions . . . . .	2
1.2 Related Game-Theoretic Work . . . . .	3
1.3 Outline . . . . .	3
<b>2 Theoretical Background</b>	<b>4</b>
2.1 Serious Games . . . . .	4
2.2 Cooperative Games . . . . .	7
2.2.1 Transferable Utility Games . . . . .	8
2.2.2 Hedonic Games . . . . .	9
2.3 Weighted Voting Games . . . . .	10
2.3.1 k-Weighted Voting Games . . . . .	11
2.4 Solution Concepts . . . . .	12
2.4.1 The Shapley Value . . . . .	12
2.5 Risk Profiles/Preferences . . . . .	13
2.6 Civic Actions . . . . .	14
<b>3 Hedonic k-Weighted Voting Games</b>	<b>16</b>
3.1 Formal Definition . . . . .	16
3.1.1 Stability . . . . .	17
3.2 Repeated hedonic k-weighted voting games . . . . .	20
3.3 Applications . . . . .	20
<b>4 Serious Game Implementation</b>	<b>23</b>
4.1 Game Play . . . . .	23
4.1.1 Skills . . . . .	23
4.1.2 Tasks . . . . .	24
4.1.3 Task's time window . . . . .	25
4.1.4 Participation . . . . .	25
4.1.5 Skills' Binding . . . . .	26

4.1.6	Coalition's Empathy . . . . .	26
4.1.7	Task's Outcome . . . . .	26
4.1.8	Skills' Ranking . . . . .	27
4.1.9	Reward . . . . .	27
4.1.10	Players' Statistics . . . . .	28
<b>5</b>	<b>Simulations</b>	<b>30</b>
5.1	Agent Profiles . . . . .	30
5.1.1	Empathy Beliefs Updating . . . . .	33
5.1.2	Skills Ranking Updating . . . . .	34
5.2	Game Versions . . . . .	34
5.2.1	Version 0 . . . . .	34
5.2.2	Version 1 . . . . .	35
5.2.3	Version 2 . . . . .	36
5.2.4	Version 3 . . . . .	36
5.3	Technical Aspects . . . . .	37
5.3.1	DataBase . . . . .	37
5.3.2	Computation of Shapley Value . . . . .	37
5.3.3	Simulations Pseudocode . . . . .	38
5.4	Results . . . . .	41
5.4.1	Comparing performance across risk profiles . . . . .	41
5.4.2	Comparing agents' performance across game versions . . . . .	42
5.5	Conclusion . . . . .	70
<b>6</b>	<b>Interface</b>	<b>71</b>
6.1	Django Framework . . . . .	71
6.2	Distributed Task Queue Celery . . . . .	72
6.3	Basic Views . . . . .	72
<b>7</b>	<b>Conclusions</b>	<b>79</b>
7.1	Summary . . . . .	79
7.2	Future Work . . . . .	80
7.2.1	Enhance Definition . . . . .	80
7.2.2	Repeated Hedonic k-Weighted Voting Game . . . . .	80
7.2.3	Stability on Hedonic k-Weighted Voting Game . . . . .	80
7.2.4	Interface Integration . . . . .	80
	<b>Bibliography</b>	<b>81</b>
	<b>Appendices</b>	<b>84</b>
<b>A</b>	<b>Voting Theory</b>	<b>85</b>

# Chapter 1

## Introduction

People from a young age use games as a fun way to explore, discover and question complicated concepts and notions [35]. Through playing children develop strategic thinking, exercise several physical and intellectual/mental skills and socialize in groups of peers. In an era of technologically advanced media and computers, the development of **digital serious games** industry and digital game-based learning industry comes as a natural outcome. Over the past few decades, the use of gaming to educate, motivate, change behavior and train has become of a great interest. More and more professional fields, e.g. education, military, government, corporate, health care, adopt digital serious games in their training and educating efforts [30].

There is a plethora of definitions in literature, however, we could say that serious games are the games designed and used for purposes *beyond pure entertainment*. These games share the characteristics we find in common games (entertaining games), such as competition, curiosity, collaboration and individual challenge. They also use game media, especially when referring to digital games, such as avatars and 3D immersion, to enhance the motivation of participants to engage in complex or boring tasks. Serious games use game environments and techniques to train or educate users or to promote a product or service in an engaging and entertaining way. The “serious” aspect comes from the fact that they are used in a broad spectrum of professional application areas like defense, education, scientific exploration, health care, emergency management, city planning, engineering, religion, and politics. Therefore serious games apply game thinking and mechanics to “serious” subjects.

Serious games can be used to promote and encourage concepts such as volunteerism among young people. Volunteerism is a popular mechanism for young people to bring positive changes in society, and therefore engage them to sustainable human development. Through gaming, children can become familiar with project of common interest and public good. In recent years, more and more non-profit organizations have been set up to promote civic actions. To give an example, the open lab of *Digital Civics* which is “a Research Council UK funded research initiated to reimagine digital local government and communities” [15]. Digital Civics applications and designs are addressed to citizens interacting with local authorities, rather than customers. Digital

Civics intend to “create a space in which both citizens and local government can explore the potential for digital technologies to underpin new and sustainable models of service provision”.

One key motivation and objective of this thesis is to create an game-based application, that will encourage students and young children to engage in large team projects. We aspire to help introduce to young people concepts such as collaboration, volunteerism, public good and civic actions, through our application. We strove to provide a pleasant and entertaining way for students and children to approach notions such as collective action, social conscience, and environmental consciousness. In the application we have in mind, we focus on the formation of teams. This touches on the issue of forming *coalitions*, a topic widely studied in *cooperative game theory* [11] [24]. In our work in this thesis, we coin a novel class of *cooperative games*, and employ it to design a serious game that promotes civic actions at local level.

## 1.1 Motivation and Contributions

Cooperative games and coalition formation have been extensively studied over the years. A very popular model of cooperative games in conjunction with voting theory is the *weighted voting games (WVG)* with *transferable utility (TU)* and so is their generalization, the *k-weighted voting games (k-WVG)*. In such games, players possess a weight, and a coalition of players wins if the sum of their weights meet or exceed a given threshold. Research, usually, assumes that the players have perfect information, meaning that players know the value of the formed coalition.

In an attempt to deal with the reality of possessing partial information within a k-weighted voting game, we focus on *hedonic games*, which are *non-transferable utility (NTU)* games [1]. In hedonic games, players are indifferent to the value of the coalition, and they care only for the identity of their co-players within the coalition.

Getting inspired from the way social norms tend to influence people, we intend to show that certain groups’ actions provoke the changing of behavior of other people, without focusing their attention on the potential gain they may obtain. As usually happens in social norms, recurrent behavioral patterns of some people affects the way the rest of the people, within a community, behave towards a specific subject.

In this thesis, we present a innovative hybrid model which combines techniques from hedonic and k-weighted voting games, and introduce a novel class of cooperative games that of **hedonic k-weighted voting games**. We adapt the coalition formation concept from hedonic games to overcome uncertainty due to partial information, while we implement a k-weighted voting game and assign a fair pay-off, according to power indices, to each member. Then we proceed on implementing our example serious game application using our proposed model, and study the way people’s behavior change after a number of repeats of hedonic k-weighted voting games over the same set of people.

## 1.2 Related Game-Theoretic Work

There has been a lot of research on serious games and decision making under uncertainty (see, e.g., references listed in [12]). Here we refer only to three related papers, which we got inspiration from. These study mainly coalition formation under uncertainty and “coalitional skills”.

Chalkiadakis and Boutilier [11] proposed a model that uses Bayesian reinforcement learning, in order for the players to reduce uncertainty towards coalitions’ values and other players’ capabilities. The model they proposed, suggest that agents “must derive coalitional values by reasoning about the types of other agents and the uncertainty inherent in the actions a coalition may take”. In this model, the agents have specific beliefs about the types of others, and make their decision about actions and coalition formations according to their “value of information”.

Then, Mamakos [24] focuses on players’ uncertainty on resource contributions of potential co-players, and proposes methods obtaining probability bounds on coalitional task completion.

Finally, Bachrach and Rosenschein [3], introduced Coalitional Skill Games, where players have a set of skills that are required to complete various tasks. The authors study the computational complexity of several problems in this model, such as ‘dummy’ and “veto agent” testing, ‘core’ computation and ‘core’ emptiness, and computation of power indices.

## 1.3 Outline

In Chapter 2 we provide the required theoretical background. We present the basic aspects of cooperative games and voting theory. We also address the issue of players’ behavior towards uncertainty, also known as risk profiles. In Chapter 3 we introduce the structure of our proposed model, the Hedonic  $k$ -Weighted Voting Games. We present the necessary definitions and notions, such as stability, along with example applications where the model can be used. Next, in Chapter 4 we present the application we designed and implemented; we explain the game play and the game rules and options. Then in Chapter 5 we proceed with the results from the simulations we ran using our setting. Last but not least, in Chapter 6 we present the front end/user interface for our application, which is fully functional.

## Chapter 2

# Theoretical Background

In the following chapter we provide the theoretical background required for this thesis. We discuss about *Serious Games*, *Cooperative games* and the basic aspects of *Voting theory*. Furthermore, we scratch the concept of player's *risk profile* and the notion of *civic actions*.

### 2.1 Serious Games

Over the past decades, digital serious game development has been well established as a thriving research field and also business. More and more professional areas turn to serious games mainly for education and training purposes [18][30]. A large number of serious application areas such as education, military, health care, government, etc, exploit the potential of serious games to induce users to engage in boring and complicated tasks in a fun and pleasant way. Looking for a formal definition, the literature provides us with more or less as many definitions as the application areas in which serious games are applied in. However, despite their deviations, all definitions seem to agree on the core meaning of serious games, that is, **serious games are the digital games designed and used for purposes other than mere entertainment**.

Serious games' origins appear to derive from *edutainment*. Edutainment is a concept designed to educate through entertainment [30]. It usually targets preschool children and students, and it focuses on providing an entertaining way to exercise in reading, dictation, mathematics, science, etc. Edutainment was quite popular in 1990s along with the technological outburst in video games and computer multi-media [30]. However, the concept of edutainment was quickly abandoned, as it was described as "boring games and drill-to-kill learning" [31].

Taking a look at literature, one can find a large variety of serious games applications. Koivula et al. in [23] showed that children at ages 5 and 6 years old, learned *social-emotional* skills through playing *Emotion Detectives*; that is they developed various skills such as recognizing and naming emotions and the ability to behave in a socially responsible way. Kenneth McAlpine in [22], highlights "the importance of play as a

vehicle for learning, harnessing both physical and conceptual aspects of play to test and explore the limits of our skills and understanding”. In [7] the authors discuss about the use of games as “an effective way of addressing the problem of engagement in therapy”, and they also provide a number of serious games that they have developed for upper limb rehabilitation following stroke. Di Loreto, Mora and Divitini in [14] analyze ten state of the art serious games for crisis management training which have the potential to develop different skills such as “the ability of anticipate; an enhanced capability for teamwork; learning to cope with short response times; a better understanding of the value of/need for stress management; sharpened business judgement skills; enhanced lateral thinking and creative skills; greater sensitivity to weak signals of abnormality; and better acceptance of change”.

There has been a lot of discussion on both the positive and the negative affects of using gaming in serious aspects of life. One of the most discussed ‘negative’ impacts, is the one of social isolation. Computer and video games are usually accused for young people isolation, since children are likely to spend too many hours in front of a screen instead of being and playing with their friends. On the other hand, serious games can be applied in the concept of socialization of children and students into civil societies. Through serious games, students can get to know and participate in civil societies, following a set of civic values and rules. In an imaginary and harmless environment, children can engage with the public (in a virtual way), make decisions and observe their outcomes. Hence, young people have the opportunity to explore and test many, and most likely conflicting, behaviors within a civil society and to observe the consequences of their decisions and actions. Bers in [5] discusses how video games can engage players in civically oriented experiences, help players to learn about problems in society, explore social, moral and ethical issues, and “make decisions about how a community, city or nation should run”.

Another aspect of socialization that can be promoted through engaging in serious games, is that of team working and cooperation. As it is well understood, in a civic society the interactions among the citizens are of most importance, since these interactions defines directly or indirectly the society itself. Students can use serious games to learn how to work together to achieve a deeper goal, to work smoothly within a group and to make compromises and concessions. Within a democratic society, it is common many and various large team projects to be organized which need citizens to participate in. By using a suitable serious games, children may learn to collaborate with their partners, take responsibilities and carry out their commitments.

To sum up serious games are applied in broad spectrum of professional areas; a few indicating examples are:

**Education** People are usually quite reluctant when digital games, especially video games, are used as an educational tool. Parents and educators often misunderstand the role of serious games and perceive them as “invaders” that may replace traditional educational means [30]. However, a large survey on this subjects has shown that video games offer a unique opportunity for students to develop strategic thinking

and several skills. Players with the aim to win the game, are prone to explore a variety of strategies while they also develop their skills. As it is indicated in [18] “in schools, students usually work alone; gamers, on the other hand, work in community, searching actively for new information, posting FAQs, discussing in forums and criticizing information they get”. Thus players can develop “their awareness to real activities and social practices”, providing a great opportunity to experience many aspects of real life. The use of serious games in education appears to have noticeable results such as students increased motivation for learning, increased collaborative learning and learning through experience.

**Health Care** The domain of health care is perhaps the most well-applied professional area for serious games. When referring to health care, issues such as physical fitness, recovery and rehabilitation are included. “Patients that undergo treatments are motivated, distracted and rewarded through games” [30]. Serious games in the health domain can provide a pleasant environment for patients to train atrophic muscles through recurrent movements, or treat several mental illnesses such as phobias and anxiety. Combined with the field of educations, serious games can inform people about several illnesses and ways for prevention and treatment. Moreover, serious games can be used for professional practice by medical students. A few indicating examples are the *Pulse!* and *Virtual ED* which were designed for training critical care. *Burn Center* was developed to train treatment of burn injuries. The games *Total Knee Arthroplasty* and *Off-Pump Coronary Artery Bypass (OPCAB)* provide a virtual operating room where medical students can train decision making during surgical operations, triage and basic life support [19].

**Politics** In several video games that present virtual worlds, one can find well established communities. These communities share rules and politics found in real life. Thus players (students and teenagers) interact with one another in political way, creating bonds, declaring leaders and facing the plausible problems of their virtual society. A game designed to teach and initiate students into the basics of the U.S. political mechanisms is *The New Alexandria Simulation: A Serious Game of State and Local Politics* [21], even though it is not a digital serious game. PlayGen developed the serious game *SeriousPolicy*, a game that “places citizens in the shoes of a senior civil servant dealing with the Prime Minister, leader of the opposition and the Chancellor” [28].

**Crisis Management** There are situations such as earthquakes, floods, fires, weather disasters, etc., that require well established response units, while the reproduction of such phenomena is practically impossible. Serious games come as a fun way for extreme phenomena simulation, that can train the personnel of various services such as police, fire brigade and army, but also civilians to respond efficiently in such situations. In the game *DREAD-ED* [20] the users are members of an emergency management team that is dealing with an evolving emergency (fire, flood, ect.). *FloodManager* [28]

is designed for teaching and reinforcing basic floodplain management principles to groups of local land development decision makers. In 2010 PlayGen presented *London Crisis* to train citizens in emergency situations such as a terrorist bombing attack [28]. Players take the role of the Incident Commander, they must assess the situation carefully and help prevent mass chaos.

## 2.2 Cooperative Games

In game theory, a cooperative game is a game where players work together in order to achieve a specific goal. In such games players form groups either because they cannot reach the desired goal on their own, or because it is more profitable, they can obtain larger pay-off, if they collaborate with other players.

A cooperative game is a model of interacting decision-makers that focuses on the behavior of groups of players. Every group of players is associated with a set of actions. We refer to each group of players with the term **coalition**, and to the coalition of all player as **grand coalition**. A coalition consisting of only one member is called singleton coalition.

An outcome of a cooperative game is a **coalition structure (CS)**, that is, a partition of all players into disjoint coalitions, usually along with an action for each coalition - the action is not always provided. At one end, the coalition structure may consist of singleton coalitions, in which players act on his/her own; at the other end, may consist of only one coalition, the grand coalition, in which players act as a whole [27].

Giving a more formally perspective (as it is given in the book “An introduction to multiagent systems” by Michael Wooldridge ):

**Definition 1.** ([34]) *We model a cooperative game (or coalitional game) as a pair  $\mathcal{G} = \langle N, v \rangle$ , where  $N$  is a set of agents and  $v : 2^N \rightarrow \mathbb{R}$  is called the characteristic function of the game.*

“The idea of the characteristic function is that it assigns to every possible coalition a numeric value, intuitively representing pay-off that *may* be distributed between the members of that coalition. That is, if  $v(C) = k$ , then this means that coalitions  $C$  can cooperate in such a way that they will obtain utility  $k$ , which *may* then be distributed among team members” [34].

A well-studied subclass of cooperative games is that of simple games:

**Definition 2.** ([12]) *A game  $\mathcal{G} = \langle N, v \rangle$  is said to be simple if it is monotone ( $v(C) \leq v(D) \forall C, D \subseteq N | C \subseteq D$ ) and  $v(C) \in \{0, 1\} \forall C \subseteq N$ . In a simple game, coalitions of value 1 are said to be winning and coalitions of value 0 are said to be losing. Such games can model situations where there is a task to be completed: a coalition is winning if and only if it can complete the task .*

## 2.2.1 Transferable Utility Games

*Transferable Utility games (TU)* is a subclass of cooperative games. Transferable utility games have a characteristic function, also known as utility function. The utility function assigns a numeric value, a utility, to every possible coalition. The outcome of a TU game is pair  $\langle \mathcal{CS}, \mathbf{x} \rangle$ , where  $\mathcal{CS}$  is a coalition structure and  $\mathbf{x}$  is a pay-off vector. The pay-off vector is notated as  $\mathbf{x} = \{x_1, \dots, x_n\}$  and is an allocation of numeric values to the players. TU games usually coincides with *Characteristic Function games (CFG)*.

Now we provide the formal definition of CF games given in:

**Definition 3.** ([12]) A Characteristic Function game  $\mathcal{G}$  is given by a pair  $\langle N, v, \rangle$ , where  $N = \{1, \dots, n\}$  is a finite, non-empty set of agents and  $v : 2^N \rightarrow \mathbb{R}$  is a characteristic function, which maps each coalition  $C \subseteq N$  to a real number  $v(C)$ . The number  $v(C)$  is usually referred to as the value of the coalition  $C$ .

An outcome of a CF game consists of two parts:

- a partition of players into coalitions, called a *coalition structure (CS)*
- a *pay-off vector*, which distributes the value of each coalition among its members

**Definition 4.** ([12]) Given a characteristic function game  $G = \langle N, v \rangle$ , a coalition structure  $\mathcal{CS}$  over  $N$  is a collection of non-empty subset  $\mathcal{CS} = \{C^1, \dots, C^k\}$  such that

- $\bigcup_{j=1}^k C^j = N$ , and
- $C^i \cap C^j \neq \emptyset$  for any  $i, j \in \{1, \dots, k\}$  such that  $i \neq j$ .

A vector  $\mathbf{x} = \{x_1, \dots, x_n\} \in \mathbb{R}^n$  is a *pay-off vector* for a coalition structure  $\mathcal{CS} = \{C^1, \dots, C^k\}$  over  $N = \{1, \dots, n\}$  if

- $x_i \geq 0$  for all  $i \in N$ , and
- $\sum_{i \in C^j} x_i \leq v(C^j)$  for any  $j \in \{1, \dots, k\}$ .

An outcome of  $\mathcal{G}$  is a pair  $\langle \mathcal{CS}, \mathbf{x} \rangle$ , where  $\mathcal{CS}$  is a coalition structure over  $\mathcal{G}$  and  $\mathbf{x}$  is a *pay-off vector* for  $\mathcal{CS}$ .

Each player in a CF game, and therefore in a TU game, obtains a *non-negative value*<sup>1</sup> and the sum of the values of all members within a coalition does not exceed the value of the coalitions, as it is defined by the utility function  $v$ .

---

<sup>1</sup>This is for convenience. See details in “Computational Aspects of Cooperative Game Theory” [12].

## 2.2.2 Hedonic Games

In many real-life applications needed to establish a coalition, the main criterion for the coalition's formation is the identity of its members rather than the utility which can be obtained itself. For example, consider a class of students in primary school, their teacher request the kids to form groups in order to carry out a team project; the kids will prefer to be grouped with their friends rather than the best student. They have little interest on forming a group which may achieve the highest mark, but they mostly care about working with their friends. In such situations we say that players have preferences over the coalitions. We model this kind of applications with *hedonic games*, a subclass of *non-transferable utility (NTU) games*.

Non-transferable utility games are cooperative games in which the utility obtained by a coalition member  $X$  cannot be transferred to another coalition member  $Y$ . The key idea in NTU games is that “each coalition has available to it a set of *choices*, or *consequences*, drawn from some overall set of choices  $\Lambda = \{\lambda, \lambda_1, \dots\}$ ” [12]. In correspondence to TU games, let  $v(C)$  denote the set of choices available to  $C$ . We say that players have preferences over choices, which we capture in *preference relations*:

**Definition 5.** ([12]) *A preference relation on  $\Lambda$  is a binary relation  $\succsim \subseteq \Lambda \times \Lambda$ , which is required to satisfy the following properties:*

1. *Completeness : For every  $\{\lambda, \lambda'\} \subseteq \Lambda$ , we have  $\lambda \succsim \lambda'$  or  $\lambda' \succsim \lambda$ ;*
2. *Reflexivity : For every  $\lambda \in \Lambda$ , we have  $\lambda \succsim \lambda$ ; and*
3. *Transitivity : For every  $\{\lambda_1, \lambda_2, \lambda_3\} \subseteq \Lambda$ , if  $\lambda_1 \succsim \lambda_2$  and  $\lambda_2 \succsim \lambda_3$  then  $\lambda_1 \succsim \lambda_3$ .*

The intended interpretation of  $\lambda \succsim \lambda'$  is that the choice  $\lambda$  is preferred at least as much as choice  $\lambda'$ . We let  $\lambda \succ \lambda'$  denote the fact that  $\lambda$  is *strictly* preferred over  $\lambda'$  (i.e.  $\lambda \succsim \lambda'$  but not  $\lambda' \succsim \lambda$ ). Also, we write  $\lambda \sim \lambda'$  if the agent is *indifferent* between  $\lambda$  and  $\lambda'$ , i.e.,  $\lambda \succsim \lambda'$  and  $\lambda' \succsim \lambda$ .

Having formally defined the term of preference relation, we provide the definition of NTU games given in “Computational Aspects of Game Theory” [12]:

**Definition 6.** ([12]) *A non-transferable utility game (NTU game) is given by a structure  $\mathcal{G} = \langle N, \Lambda, \succsim_1, \dots, \succsim_n \rangle$ , where  $N = \{1, \dots, n\}$  is a non-empty set of players,  $\Lambda = \{\lambda, \lambda_1, \dots\}$  is a non-empty set of choices,  $v : 2^N \rightarrow 2^\Lambda$  is the characteristic function of  $\mathcal{G}$ , which for every coalition  $C$  defines the choices  $v(C)$  available to  $C$ , and, for each player  $i \in N$ ,  $\succsim_i \subseteq \Lambda \times \Lambda$  is a preference relation on  $\Lambda$ . We will assume that  $v(\emptyset) = \emptyset$ .*

In order to define the outcome of an NTU game, first we need to define the term *choice vector*:

**Definition 7.** ([12]) *A choice vector with respect to a coalition structure  $\mathcal{CS} = \{C^1, \dots, C^k\}$  of NTU game  $\mathcal{G} = \langle N, \Lambda, v, \succsim_1, \dots, \succsim_n \rangle$  is a vector  $\mathbf{c} = \{l_1, \dots, l_k\} \in \Lambda^k$  that satisfies  $l_i \in v(C^i)$  for all  $i = 1, \dots, k$ .*

**Definition 8.** ([12]) An outcome for an NTU game  $\mathcal{G} = \langle N, \Lambda, \succsim_1, \dots, \succsim_n \rangle$  is a pair  $\langle \mathcal{CS}, \mathbf{c} \rangle$ , where  $\mathcal{CS}$  is a coalition structure over  $N$  and  $\mathbf{c}$  is a choice vector for  $\mathcal{CS}$ . Where  $\langle \mathcal{CS}, \mathbf{c} \rangle$  is an outcome for an NTU game, and  $i \in N$ , we let  $\mathbf{c}_i$  denote the choice corresponding to the coalition of which  $i$  is a member.

Back to hedonic games, we say that in such games players have *hedonic* preferences over coalitions. Hedonic preferences indicates that every player is only interested in which players are in his/her coalition, and have no interest on the coalition's utility. The hedonic preferences can be interpreted as “enjoying the pleasure of each other's company”.

Formally hedonic games are defined as:

**Definition 9.** ([12]) A hedonic game is given by a structure  $\langle N, \succsim_1, \dots, \succsim_n \rangle$ , where  $N = \{1, \dots, n\}$  is a set of players, and for each player  $i \in N$  the relation  $\succsim_i \subseteq \mathcal{N}_i \times \mathcal{N}_i$  is a complete, reflexive, and transitive preference relation over the possible coalitions of which  $i$  is a member<sup>2</sup>. The intended interpretation is that if  $C_1 \succsim_i C_2$ , then player  $i$  prefers to be in coalition  $C_1$  at least as much as in coalition  $C_2$ . We define an indifference relationship  $\sim_i$  by setting  $C_1 \sim_i C_2$  if and only if  $C_1 \succsim_i C_2$  and  $C_2 \succsim_i C_1$ .

A characteristic of hedonic games is that the set of choices available to each coalition is a singleton. Thus, the **outcome** of a hedonic game consists of **only a coalition structure**  $\mathcal{CS}$ : in hedonic games there is no need to give a choice vector as an output since each coalition has a single choice available to it.

## 2.3 Weighted Voting Games

Weighted Voting Games, for short “WVGs”, is a class of cooperative games that can also be seen as modeling “Weighted Voting Systems” (see Appendix A). Each voter, or player, possesses a numeric value (weight), which indicates the “importance” of this player's view. Weighted voting systems are usually preferred in situation such as shareholder meetings, where votes are weighted by the number of shares that each shareholder owns.

Thinking beyond electoral systems, Weighted Voting Games can be used to model settings where each player has a certain amount of a given resource (say, time, money, or manpower), and there is a goal that can be reached by any group of players that possesses a sufficient amount of this resource [12].

A formal definition of Weighted Voting Games (with transferable utility) is :

**Definition 10.** ([12]) Let  $N = \{1, \dots, n\}$  be a finite set of players and  $C \subseteq N$  be a coalition. A Weighted Voting Game (WVG)  $\mathcal{G}$  is a tuple  $\langle N; \mathbf{w}; q; v \rangle$ , where  $\mathbf{w} = \{w_1, \dots, w_n\} \in \mathbb{R}^N$

<sup>2</sup> $\mathcal{N}_i$  denotes the collection of all subsets of  $N$  that contain  $i$  :  $\mathcal{N}_i = \{C \cup \{i\} | C \subseteq N\}$

is a list of weights,  $q \in \mathbb{R}$  is a quata and  $v : 2^N \rightarrow \{0, 1\}$  is the utility function. The utility function is given by :

$$v(C) = \begin{cases} 1, & \text{if } \sum_{i \in C} w_i \geq q \\ 0, & \text{otherwise} \end{cases}$$

Outcome of  $\mathcal{G}$  is a pair  $\langle CS, \mathbf{x} \rangle$ , where  $CS$  is a coalition structure and  $\mathbf{x} \in \mathbb{R}^N$  is a pay-off vector.

It is not hard to see, that each weighted voting game is a simple game.

**Definition 11.** [12] A coalition  $C \subseteq N$  with  $v(C) = 1$  is said to be **winning**, while a coalition  $C$  with  $v(C) = 0$  is said to be **losing**.

**Definition 12.** [12] Given a WVG  $\mathcal{G} = \langle N; \mathbf{w}; q; v \rangle$  and a coalition  $C \subseteq N$ , a player  $i \in N$  is said to be **pivotal** to the coalition  $C$  if  $v(C \cup \{i\}) - v(C) = 1$ .

### 2.3.1 k-Weighted Voting Games

As we mentioned above, every weighted voting game is a simple game. However, the reverse does not stand: weighted voting games are not a *complete* representation for simple games. Some simple games cannot be represented using weighted voting games. The natural extension of weighted voting games, though, is a complete representation. This extension is known as *Vector Weighted Voting Games* or *k-Weighted Voting Games (k-WVGs)* [12][34].

The key idea of k-weighted voting is that we take a collection of k different weighted voting games over the same set of players. Then a coalition is winning if and only if it is winning in each of the k different weighted voting games. Otherwise it is losing. More formally:

**Definition 13.** ([12]) A *k-weighted voting game (kWVG)*, or *vector weighted voting game (VWVG)*, is given by a set of players  $N$ ,  $|N| = n$ , for each player  $i \in N$ , a *k-dimensional weight vector*  $\mathbf{w}_i = \{w_i^1, \dots, w_i^k\}$  whose entries are non-negative integers, and *k non-negatives quotas*  $q_1, \dots, q_k$ ; we write  $\mathcal{G} = \langle N; \mathbf{w}_1, \dots, \mathbf{w}_n; q^1, \dots, q^k \rangle$ . A coalition  $C \subseteq N$  is deemed to be *winning in  $\mathcal{G}$*  if and only if  $\sum_{i \in C} w_i^j \geq q^j$  for all  $j = 1, \dots, k$ .

In line with weighted voting games,

**Definition 14.** ([12]) A coalition  $C \subseteq N$  with  $v(C) = 1$  is said to be **winning**, while a coalition  $C$  with  $v(C) = 0$  is said to be **losing**.

And as far as criticality is concerned:

**Definition 15.** ([12]) Given a *k-WVG*  $\mathcal{G} = \langle N; \mathbf{w}^1, \dots, \mathbf{w}^k; q^1, \dots, q^k; v \rangle$  and a coalition  $C \subseteq N$ , a player  $i \in N$  is said to be **pivotal** to the coalition  $C$  if  $i$  is pivotal to at least one component game  $\mathcal{G}^j$  and  $C$  is winning in the rest of the component games.

## 2.4 Solution Concepts

A **solution concept**, in game theory, is a formal rule for predicting how a game will be played. These predictions are called “solutions”, and describe which strategies will be adopted by rational players and, therefore, the result of the game. (A rational player is a player that gets to maximize her own utility or in strict accordance to her preference profile). **Solution concept** in cooperative game theory capture the notion of forming a pay-off allocation. A pay-off allocation with respect to a coalition  $C$  is described by a *pay-off vector*, which maps a non-negative numeric value to each member of  $C$ . As stated already, any partition of players into coalitions, that is to say, any coalition structure, along with a pay-off vector, forms an outcome of a cooperative game. The utility function defines the gain a coalition may achieve, however it does not determines how this gain will be shared amongst the members of the coalition. It is not hard to see, that not all outcomes are equally desired. An outcome is usually evaluated according to the following criteria: (i) *fairness* and (ii) *stability*.

### 2.4.1 The Shapley Value

The solution concept we will further discuss is the **Shapley Value**. The Shapley value focuses in fairness, rather than stability. This solution concept assigns to each player a pay-off proportional to his/her contribution to the game, that is the Shapley value of each player depends on his/her *marginal contribution* to possible coalition permutations.

In order to give a formal definition of solution concept Shapley Value, we need to provide some extra notation:

**Definition 16.** ([12]) Given a cooperative game  $\mathcal{G} = \langle N; v \rangle$ , let  $\Pi_N$  denote the set of permutations over  $N$ . We denote with  $S_\pi(i) = \{j \in N | \pi(j) < \pi(i)\}$ , where  $i \in N$  and  $\pi \in \Pi_N$ , the set of predecessors of  $i$  in  $\pi$ .

Now, the Shapley value is defined as:

**Definition 17.** ([12]) Let  $\mathcal{G} = \langle N; v \rangle$  be a cooperative game and  $C \subseteq N$  be a coalition. The **Shapley Value** of player  $i \in C$  is denoted by  $\phi_i(C, v)$  and is given by

$$\phi_i(C, v) = \frac{1}{|C|!} \sum_{\pi \in \Pi_C} [v(S_\pi(i) \cup \{i\}) - v(S_\pi(i))]$$

The amount  $v(S_\pi(i) \cup \{i\}) - v(S_\pi(i))$  is called *marginal contribution* of player  $i$  with respect to a permutation  $\pi$ .

We now provide a few properties of the Shapley value:

**Definition 18.** ([12]) It can be proved that the sum of all the Shapley values of the members within a coalition is equal to the value that the utility function assigns to this coalition:

$$\sum_{i \in C} \phi_i(C, v) = v(C)$$

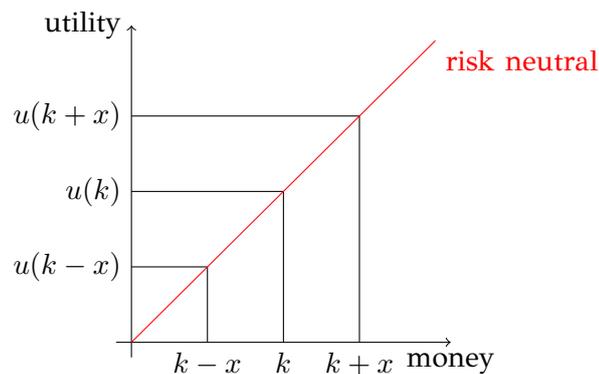
The Shapley value does not allocate any pay-off to players who do not contribute to any coalition.

**Definition 19.** ([12]) Given a cooperative game  $\mathcal{G} = \langle N; v \rangle$  a player  $i \in C$  is said to be **dummy** if  $v(C) = v(C \cup \{i\})$  for any  $C \subseteq N$ . Intuitively, if a player  $i \in N$  is a **dummy** player in  $\mathcal{G}$ , then  $\phi_i(\mathcal{G}) = 0$ .

## 2.5 Risk Profiles/Preferences

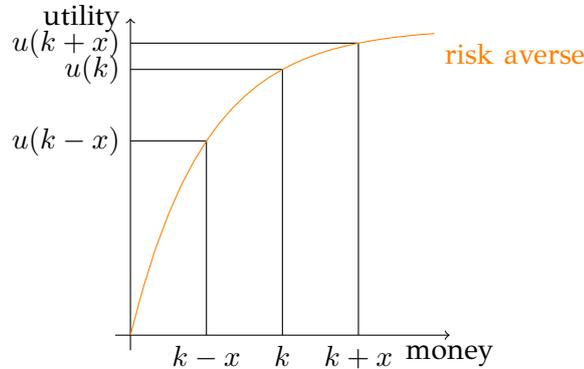
The behavior of player is often a subject of research in game theory and economy. Especially, the most interesting aspect is their behavior when exposed to uncertainty. In such situations, people react differently and they tend to lower the uncertainty at different levels. The literature distinguishes three main behavioral profiles: *risk averse*, *risk neutral* and *risk seeking* [29].

- **Risk Neutral:** An agent who simply wants to maximize his/her expected revenue is said to be risk neutral. As shown in the below graph, this kind of players have a linear value for money. In a fair lottery that half of the time the award is  $k + x$  and the other half of the time is  $k - x$ , a risk neutral player is indifferent about whether participating or not in the lottery, since the expected value is  $u(k) = \frac{1}{2}u(k - x) + \frac{1}{2}u(k + x)$ .

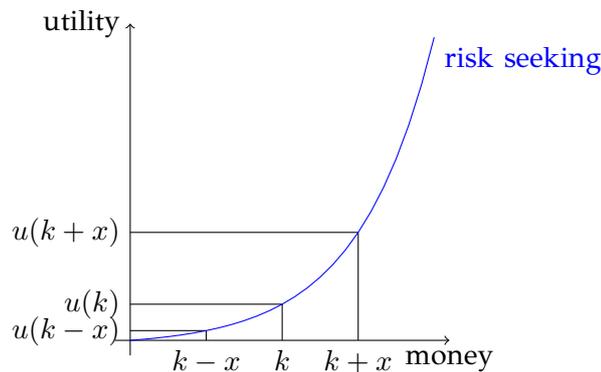


- **Risk Averse:** In the contrary, an agent who prefers a “sure thing” to a risky situation with the same expected value is called risk averse. In the corresponding graph below is illustrated that a risk averse player has a sublinear value for money. Considering the fair lottery described above, for a risk averse player the

marginal disutility of losing  $x$  units of money is greater than the marginal utility of gaining the same units of money ( $x$ ) :  $u(k) > \frac{1}{2}u(k-x) + \frac{1}{2}u(k+x)$ .



- **Risk Seeking:** The quite opposite of risk averse, is an agent who prefers engaging in lotteries to a sure thing with the same expected value. The agents having this kind of behavioral profile are called risk seeking. Looking at the graph below, we see that a risk seeking player has a superlinear value for money.



Choice under uncertainty is often characterized as the maximization of expected utility. Looking at the above figures of value for money for each profile of behavior, we can see that the utility function for a risk neutral agent is linear, for a risk averse agent is convex and for a risk-seeking agent is concave.

## 2.6 Civic Actions

The behavior of individuals within a civic society is determined by the civilians' interaction with each other. Social action refers to an act which takes into account the actions and reactions of individuals. A civic action can be considered as a social

action which aims at the wider public good. The concept of civic action is not clearly defined in the literature, and its meaning is not yet well-established. However, civic action can be defined as a "form of citizenship practice consisting in mainly collective initiatives aimed at implementing rights, taking care of common goods or empowering general public" [26].

Civic actions contain the sense of responsibility and concern for the problems and injustices of society, and they intend to prevent and solve them by using specific tools on the citizens' side beyond the exercise of the right to vote. It implies the exercise of citizens' powers in the public realm, such as the powers to produce information and knowledge, to change the common awareness, to give the "social license to operate", to constrain public institutions to effectively work, to change material conditions, etc. [26].

Another way to perceive the notion of civic actions is that of facing issues which 'hurt' society and require immediate and most likely humanitarian action; this is the way the non-profit, non-partisan organization *CivicAction* uses the very same term [9]. Among others, civic actions include organization of humanitarian aid campaigns, alleviation of environmental disasters, and organization of public good projects. From that point of view, we could say that the term 'civic action' is close to the term 'activism'.

An institution that conjugates civic actions with digital technologies is *Digital Civics*. Digital Civics [15] is an open lab which intent to use "digital technologies to underpin new and sustainable models of service provision". In [32] the authors study "how digital technologies can scaffold a move from transactional to relational service models, and the potential of such models to reconfigure power relations between citizens, communities and the state"; while [33] "examines the ways by which digital technologies can support 'on-the-ground' activist communities in the development of social movement".

## Chapter 3

# Hedonic k-Weighted Voting Games

We now proceed to introduce the novel hybrid game model we propose in this thesis, namely that of *Hedonic k-Weighted Voting Games*.

Hedonic k-weighted voting games similar to k-Weighted Voting Games, but we assume that players have either partial information about the value of a formed coalition, or are indifferent about it at the time the coalition is formed. For this reason, we borrow the concept of preference relations from hedonic games: intuitively, agents will form coalitions given their preferences, but will also receive numeric reward as a result of their actions.

In more details and to further motivate hedonic k-weighted voting games, we suppose that players are only aware of a few of their co-players' characteristics, and have no knowledge of each player's weights. Therefore, they have no knowledge about the utility that can be obtained from a given coalition.

Each player has preferences on coalitions, which are formed on the basis of the other participants in each coalition, and not on the imminent pay-off he/she is about to gain. Once a coalition is formed, a hedonic k-weighted voting game behaves as a k-WVG.

### 3.1 Formal Definition

We now provide a formal definition for hedonic k-weighted voting games:

**Definition 20.** Let  $N = \{1, \dots, n\}$  be a finite set of players and  $C \subseteq N$  be a coalition. A hedonic k-Weighted Voting Game  $\mathcal{G}$  is a tuple  $\langle N; \succsim; \mathbf{w}^1, \dots, \mathbf{w}^k; q^1, \dots, q^k; v \rangle$ , where  $\succsim$  is a preference relation,  $k \in \mathbb{N}$ ,  $\mathbf{w}^j \in \mathbb{R}^N$  is the list of weights for component game  $\mathcal{G}^j$  with  $j \in \{1, \dots, k\}$ ,  $q^j \in \mathbb{R}$  is the corresponding quota for component game

$\mathcal{G}^j$ , and  $v : 2^{k \times N} \rightarrow \{0, 1\}$  is the utility function. The utility function is given by :

$$v(C) = \begin{cases} 1, & \text{if } \sum_{i \in C} w_i^j \geq q^j \text{ for } j = 1, \dots, k \\ 0, & \text{otherwise} \end{cases}$$

An outcome of  $\mathcal{G}$  is a pair  $\langle CS, \mathbf{x} \rangle$ , where  $CS$  is a coalition structure (formed given the preference relation  $\succsim$ ); and  $\mathbf{x} \in \mathbb{R}^{k \times N}$  is a pay-off vector. The pay-off  $x_i^j$  is a function of the utility functions of the component games (indicated by  $u^j$ ) and the utility function of the hedonic  $k$ -WVG (indicated by  $u$ ), that is  $\mathbf{x} := f(u, u^1, \dots, u^k)$ .

According to the above definition, we can see that the pay-off vector of a hedonic  $k$ -weighted voting game assigns to each agent  $i \in N$  a numeric value for each one of the  $k$  component games. Each pay-off is related to the utility of the component games and the utility of the game as a whole. In line with  $k$ -WVGs, we call a coalition  $C$  of a hedonic  $k$ -weights voting game **winning** when  $v(C) = 1$  and **losing** when  $v(C) = 0$ . Similarly, a player  $i \in N$  is **pivotal** to  $C$  in a given coalition  $C$ , if  $i$  is pivotal to at least one component game  $\mathcal{G}^j$ , and  $C$  is winning in the rest of the component games.

### 3.1.1 Stability

Since coalition formation in our setting is based solely on hedonic preferences, hedonic  $k$ -weighted voting games demonstrate the stability principles of pure hedonic games. There is a variety of stability concepts used in hedonic games, and we list them as introduced by Aziz and Bradl in [2].

- A partition  $\pi$  is *individual rational (IR)* if no player has an incentive to become alone i.e., for all  $i \in N, \pi(i) \succsim_i \{i\}$ .
- A partition  $\pi$  is *Nash stable (NS)* if no player can benefit by moving from his coalition to another (possibly empty) coalition  $T$ .
- A partition  $\pi$  is *individually stable (IS)* if no player can benefit by moving from his coalition to another existing (possibly empty) coalition  $T$  while not making the members of  $T$  worse off.
- A coalition  $S \subseteq N$  *blocks* a partition  $\pi$ , if each player  $i \in S$  strictly prefers  $S$  to his current coalition  $\pi(i)$  in the partition  $\pi$ . A partition which admits no blocking coalition is said to be in the *core*( $C$ ).
- A coalition  $S \subseteq N$  *weakly blocks* a partition  $\pi$ , if each player  $i \in S$  weakly prefers  $S$  to  $\pi(i)$  and there exists at least one player  $j \in S$  who strictly prefers  $S$  to his current coalition  $\pi(j)$ . A partition which admits no weakly blocking is in the *strict core*( $CS$ ).

- A partition  $\pi$  is *Pareto optimal (PO)* if there is no partition  $\pi'$  with  $\pi'(j) \succ_j \pi(j)$  for all players  $j$  and  $\pi'(i) \succ_i \pi(i)$ .
- For partition  $\pi, \pi' \neq \pi$  is called *reachable* from  $\pi$  by movements of players  $H \subseteq N$ , denote by  $\pi \xrightarrow{H} \pi'$ , if  $\forall i, j \in N \setminus H, i \neq j : \pi(i) = \pi(j) \Leftrightarrow \pi'(i) = \pi'(j)$ .  
A subset of players  $H \subseteq N, H \neq \emptyset$  *strong Nash blocks*  $\pi$  if a partition  $\pi' \neq \pi$  exists with  $\pi \xrightarrow{H} \pi'$  and  $\forall i \in H : \pi'(i) \succ_i \pi(i)$ .  
If a partition  $\pi$  is not strong Nash blocked by any set  $H \subseteq N$ ,  $\pi$  is called *strong Nash stable (SNS)*
- A subset of players  $H \subseteq N, H \neq \emptyset$  *weakly Nash blocks*  $\pi$  if a partition  $\pi' \neq \pi$  exists with  $\pi \xrightarrow{H} \pi', \forall i \in H : \pi'(i) \succ_i \pi(i)$  and  $\exists i \in H : \pi'(i) \succ_i \pi(i)$ .  
A partition which admits no weakly Nash blocking coalition is said to satisfy *strict strong Nash stability (SSNS)*
- A non-empty set of players  $H \subseteq N$  is *strongly individually blocking* a partition  $\pi$ , if a partition  $\pi'$  exists such that:
  1.  $\pi \xrightarrow{H} \pi'$  (as for SNS),
  2.  $\forall i \in H : \pi'(i) \succ_i \pi(i)$ , and
  3.  $\forall j \in \pi'(i)$  for some  $i \in H : \pi'(j) \succ_j \pi(j)$ .

A partition for which no strongly individually blocking set exists is *strongly individually stable (SIS)*.

Since formation in hedonic k-weighted voting games is simply the result of hedonic preferences, all theoretical results related to hedonic games carry over to hedonic k-WVGs in a straightforward manner. For instance, the theorems below on top and bottom responsiveness [2] stand:

Let  $Ch(i, S) = \{S' \subseteq S : (i \in S') \wedge (S' \succ_i S'' \forall S'' \subseteq S)\}$  be the *choice sets*<sup>1</sup> of player  $i$  in coalition  $S$ , and  $\mathcal{N}_i$  denote the set of all coalitions over  $N$  which contains player  $i : \mathcal{N}_i = \{S \subseteq N : i \in S\}$ .

We say that a game  $\mathcal{G} = \langle N; \succ; \mathbf{w}^1, \dots, \mathbf{w}^k; q^1, \dots, q^k; v \rangle$  satisfies **top responsiveness** if  $\forall i \in N$ :

1. for each  $X \in \mathcal{N}_i$ ,  $|Ch(i, X)| = 1$  ( $ch(i, X)$  denotes the *unique* maximal set of player  $i$  on  $X$  under  $\succ_i$ )
2. for each pair  $X, Y \in \mathcal{N}_i$ ,  $X \succ_i Y$  if  $ch(i, X) \succ_i ch(i, Y)$
3. for each pair  $X, Y \in \mathcal{N}_i$ ,  $X \succ_i Y$  if  $ch(i, X) = ch(i, Y)$  and  $X \subset Y$

---

<sup>1</sup>Sets of players which each player wants to be with

A game satisfying top responsiveness additionally satisfies **mutuality** if  $\forall i, j \in N, X \in \mathcal{N}_i \cup \mathcal{N}_j : i \in ch(j, X) \Leftrightarrow j \in ch(i, X)$ .

**Theorem 1.** *Top responsiveness and mutuality together guarantee the existence of an SSNS partition.*

Let  $Av(i, s) = \{S' \subseteq S : (i \in S') \wedge (S' \prec_i S'' \forall S'' \subseteq S)\}$  be the *avoid sets*<sup>2</sup> of player  $i$  in coalition  $S$ .

We say that a game  $\mathcal{G} = \langle N; \succ; \mathbf{w}^1, \dots, \mathbf{w}^k; q^1, \dots, q^k; v \rangle$  satisfies **bottom responsiveness** if  $\forall i \in N$ :

1. for each pair  $X, Y \in \mathcal{N}_i, X \succ_i Y$  if  $X' \succ_i Y'$  for each  $X' \in Av(i, X)$  and each  $Y' \in Av(i, Y)$
2. for each  $i \in N$  and  $X, Y \in \mathcal{N}_i, Av(i, X) \cup Av(i, Y) \neq \emptyset$  and  $|X| \geq |Y|$  implies  $X \prec_i Y$

A game  $\mathcal{G} = \langle N; \succ; \mathbf{w}^1, \dots, \mathbf{w}^k; q^1, \dots, q^k; v \rangle$  satisfies **strong bottom responsiveness** if it is bottom responsive and if for each  $i \in N$  and  $X \in \mathcal{N}_i, |Av(i, X)| = 1$  ( $av(i, X)$  denotes the *unique* minimal set of player  $i$  on  $X$  under  $\prec_i$ ).

A game satisfying strong bottom responsiveness additionally satisfies **mutuality** if  $\forall i, j \in N$  and  $X$  such that  $i, j \in X, i \in av(j, X)$  if and only if  $j \in av(i, X)$ .

**Theorem 2.** *Bottom responsiveness guarantees the existence of an SIS partition.*

**Theorem 3.** *Strong bottom responsiveness and mutuality together guarantee the existence of an SNS partition.*

Finally, it can be proved that if the preference relation satisfies either of the following properties, it guarantees top or bottom responsiveness and therefore stability:

- $\mathcal{G}$  is *additively separable* [1][2]: each player  $i \in N$  has value  $v_i(j)$  for player  $j$  being in the same coalition as  $i$  and if  $i$  in coalition  $S \in \mathcal{N}_i$ , then  $i$  gets utility  $\sum_{j \in S \setminus \{i\}} v_i(j)$ . For coalitions  $S, T \in \mathcal{N}_i, S \succ T$  if and only if  $\sum_{j \in S \setminus \{i\}} v_i(j) \geq \sum_{j \in T \setminus \{i\}} v_i(j)$ .
- $\mathcal{G}$  is  *$\mathcal{B}$ -hedonic games*: players express preferences over players and these preferences are naturally extended over coalitions. For coalitions  $S, T \in \mathcal{N}_i, S \succ_i T$  if and only if:
  - $\forall s \in \max_i(S \setminus \{i\})$  and  $t \in \max_i(T \setminus \{i\}), s \succ_i t$ , or
  - $\forall s \in \max_i(S \setminus \{i\})$  and  $t \in \max_i(T \setminus \{i\}), s \sim_i t$  and  $|S| < |T|$ .

---

<sup>2</sup>Sets of players which each player wants to avoid having in his/her coalition.

Nonetheless, there is *room for study* of stability based on different considerations, once the game outcomes include pay-offs that are in fact related to utility. Hence, Hedonic k-Weighted Voting Games provide a richer framework for the study of stability. This is interesting future work. Moreover, *learning* the (unknown) utility function is a challenge - and can also potentially lead to updates of the players' preferences.

### 3.2 Repeated hedonic k-weighted voting games

When a player repeatedly interacts with the same set of co-players, it is expected for him/her to change preferences over coalitions. As we have already said, in hedonic k-weighted voting games, each coalition has a utility and therefore players actually obtain a pay-off from it. Based on the pay-off gained by each player, we assume that the players in a subsequent repeat of the game might show greater or lesser preferences over a particular coalition than they had shown in the past. So in an extensive form of the game, players are expected to *update* their preferences over coalitions, and adjust them based on the outcomes of previous repeats of the game.

### 3.3 Applications

We now give examples where hedonic k-weighted voting games might be used.

Suppose there is an online council of experts, where members have to vote for or against a proposal consisting of several chapters. The council votes separately for each proposal chapter and for the proposal to be adopted, all chapters must pass. The vote of each council member has a weight that is not known to the other members and it is different in every new proposal. The members announce their intention to vote on the overall proposal, but not for each individual chapter. By knowing only who tend to be against the proposal and who to be in favor of it, members adjust their final decision according the other members' vote intentions. After the voting process, the members have formed two disjoint sets, those who have voted in favor of at least one proposal chapter and those who vote against all chapters. The final outcome of the voting results from the total of the votes cast in each of the proposal chapters. Depending on the result of the voting process and the players' weights, each player receives some "power units" that can be added to his/her existing weights; thus, the players can strengthen their voting weights for a future voting.

The above example can be modeled by hedonic k-weighted voting games. Each proposal chapter may be a component game of a k-weighted voting game. The fact that the council members might 'move' their position according to the other members vote intentions, though without knowing the final outcome in advance, allow us to claim that they have preferences over the two imminent coalitions (the ones who are against and the ones who are in favor). Finally, the council members participating in a winning coalition in the above setting are rewarded with power units to enhance

their voting power within the council; that is the members receive pay-offs (one for each component game), as defined in hedonic  $k$ -weighted voting games.

Another situation our proposed model can be adopted in, is civic *crowdfunding* public projects. In such cases, citizens are expected to fund specific public projects and there is a number of crowdfunding platforms that allow them to do so (e.g. Spavehive, Citizeninvestor, Neighborly). Assuming that there is need for a public hospital to be built in a small town, the funding of this undertaking is to be generated by the citizens. However, there is a funding requirement for the building's construction and a second funding requirement for the hospital's equipment. People are asked contribute on this project, without knowing whether the required amount of money will actually be gathered (we suppose there is a refund policy if the funding target is not achieved by the deadline). They will probably be aware of which citizens have already contributed to the funding, but they will neither have any knowledge of how much money the others have offered, nor will they know in which of the two causes the others contributed (the construction or the equipment). According to the information about who has already contributed, citizens adjust their decision to contribute or not to the building of the hospital. If this public project is successfully conducted, and the hospital is both built and fully equipped, then the contributors earns some privileges indicative of their contributions.

Crowdfunding projects are supposed to belong in hedonic games since people are prone to participate depending on who is already participating in the project. Although such public good projects can be modeled by simple weighted voting games (meaning that there is only one target to be achieved by a given deadline), hedonic  $k$ -weighted voting games provide a generalization.

Moreover, in line with the application we designed and will present in the next chapter, let us assume that the members of a local community are taking the initiative to collect the garbage from a park. Such an activity requires some skills, e.g. 'organization' and 'teamwork'. Each member who wants to participate in the above campaign, should contribute at least one skill. Most of the citizens will be probably motivated to take part in the activity by their friends and their acquaintances who are already participating in. It is clear enough, that the participants have no information of with which skill each of the rest of the participants contributes, neither whether the above undertaking will be successfully performed. If enough people gather (in a pre-determined date), and the skill requirements are met, then the activity is performed successfully. In addition to the moral satisfaction of participants from cleaning the park, they acquire some further experience of the skills.

This example is designed to be modeled by a hedonic  $k$ -weighted voting game. Each required skill correspond to a component game of a hedonic  $k$ -weighted voting game. By taking into account that the citizens are likely to be motivated by others and participate in the given activity, without having any knowledge of the outcome, we can infer that the citizens have hedonic preferences over the two disjoint coalitions:

those who participate, and those who don't. After a successful activity, we assume that each participant gains some further experience, defining a **pay-off** as it is described in hedonic k-weighted voting games.

## Chapter 4

# Serious Game Implementation

### 4.1 Game Play

In this section we will present the tactical aspects of our proposed game, such as its plot and the way it is played. Beginning with the plot, we are in a virtual local community, with players being the members of this community. The game organizes several *activities* in which players are requested to participate. Thus players are to form groups and work together in order to fulfil the given activities. When an activity is performed successfully, its participants receive a reward for contributing in it.

#### 4.1.1 Skills

A skill is the ability to carry out a task with pre-determined results often within a given amount of time, energy, or both. In our setting, we define eight (8) different skills. These skills are namely  $\{teamwork, organization, painting, music, sports, literature, science, technical\}$ . Each player has all eight skills, but at different **experience levels**. In order to grasp the concept of experience level, we assign a positive integer number. The higher this number is, the higher the experience level is considered to be. Therefore, each player possesses a number for each one of the eight skills, indicating his/her experience level on this particular skill.

For example, given a player  $i$ , he/she has the set of skills  $skills_i = \{teamwork, organization, painting, music, sports, literature, science, technical\}$  in experience levels  $\{50, 43, 55, 19, 27, 36, 22, 57\}$ ; meaning that his/her experience level for skill teamwork is 50, for skill organization is 43, for skill painting is 55, and so on. As we elaborate below, skills' experience levels correspond to weights in a hedonic k-weighted voting game.

Later on we will see that activities also require some skills in specific experience levels.

	1 <sup>st</sup> skill	2 <sup>nd</sup> skill	3 <sup>rd</sup> skill
<b>painting school</b>	teamwork	organization	painting
<b>waste collection</b>	teamwork	organization	-
<b>recycle</b>	teamwork	organization	science
<b>reforestation</b>	teamwork	organization	science
<b>painting exhibition</b>	teamwork	organization	painting
<b>sporting activity</b>	teamwork	organization	sports
<b>music concert</b>	teamwork	organization	music
<b>study group:literature</b>	teamwork	organization	literature
<b>study group:science</b>	teamwork	organization	science
<b>study group:technical</b>	teamwork	organization	technical

Table 4.1: Set of skills required per different type of task.

#### 4.1.2 Tasks

From now on we refer to activity with the term *task*. In our setting, we define ten (10) types of tasks. Each task corresponds to a real-life activity. These real-life activities are  $\{\textit{painting school}, \textit{waste collection}, \textit{recycle}, \textit{reforestation}, \textit{painting exhibition}, \textit{sporting activity}, \textit{music concert}, \textit{study group:literature}, \textit{study group:science}, \textit{study group:technical}\}$ .

For a task to be fulfilled, some skills are required. Specifically, each task requires a predetermined set of skills; this set of skills contains 2 or 3 skills (only one type of tasks requires 2 skills, while the rest nine require 3 skills). In Table 4.1 we see which set of skills is required for each task.

The set of skills for each task may be predetermined, however each skill for a given task is required in different experience levels, as we have already mentioned. To give a short example, suppose we have the task *recycle* which requires the set of skills  $\{\textit{teamwork}, \textit{organization}, \textit{science}\}$  in experience levels  $\{28, 34, 25\}$ ; meaning that this specific task requires skill *teamwork* in experience level 28, skill *organization* in experience level 34 and skill *science* in experience level 25. Another task of the same type (*recycle*) always requires the same subset of skills ( $\{\textit{teamwork}, \textit{organization}, \textit{science}\}$ ) but in different experience levels. The experience levels of a task results from a uniform distribution  $\mathcal{U}(25, \textit{sum\_of\_all\_players\_exp\_level\_for\_corresponding\_skill})$ ; the lower bound in our experiments is arbitrarily set to 25; the upper bound denotes that the required experience level will always be potentially achievable, that is a task will never require an experience level such that the grand coalition contributing all the experience level they possess cannot achieve.

As we elaborate below, experience levels correspond to quotas in a hedonic k-weighted voting game.

### 4.1.3 Task's time window

Every task has a time window. This represents the time that the task needs to be organized and carried out. In other words, it is the time given to the players to form a coalition in order to perform this specific task.

Within the time window, players can either join the task or withdraw from it. If a player is not currently participating in a given task, he/she evaluates the currently formed coalition and decides whether he/she wants to take part in it or not. On the other hand, when a player is already participating in the task, once again he/she evaluates the currently formed coalition and either deviates from it (withdraws) or not (remains in the task).

Each player may evaluate the formed coalition and make his/her decision as many times as he/she likes to, as long as it is within the task's time window. When the time window elapses, the task "comes to an end", players can no longer change their mind about participating in this particular task, and the task is evaluated as a success or a failure.

### 4.1.4 Participation

A participation is determined by a pair  $\langle player, task \rangle$ . A player  $i$  is said to participate in a task  $t$ , when  $i$  contributes to  $t$ 's requirements. As mentioned already, every task requires a subset of skills in some experience level; we say that player  $i$  *grants* an amount of his/her experience level of a particular skill to the task's  $t$  respective required skill.

Players have two degrees of freedom in their participation. Firstly, a player may contribute to any *non-empty* subset of a task's skill requirements. For example, let task  $t$  be of type *recycle* which requires the skills {teamwork, organization, science}, player  $i$  can contribute to any subset  $S \subseteq \{teamwork, organization, science\} : S \neq \emptyset$ ; meaning that player  $i$  can contribute to one, two or all three of the required skills.

The second degree of freedom concerns the 'quantity' of experience each player contributes to a required skill. When player  $i$  contributes to task's  $t$  required skill  $r_1$ ,  $i$  can 'lend' an amount less or equal to the experience level of his/her  $r_1$  skill.

Continuing our previous example, let  $t$  require experience level {28, 34, 25} and let player  $i$ 's experience level of skills *teamwork*, *organization*, *science* be 50, 43 and 22, respectively;  $i$  can contribute to required skill *teamwork* any amount in the range  $[0, 50]$ , to *organization* any amount in the range  $[0, 43]$  and to *science* any amount in the range  $[0, 22]$ <sup>1</sup>. Notice that there is no restriction for a player to contribute an amount of experience level greater than the one required, even though it is not rational (since no greater utility will be derived from exceeding the quota).

All valid participations for a given task must be submitted within the task's time window, as we have previously mentioned.

---

<sup>1</sup>We include the value of 0 in the allowed range, taking into account the first degree of freedom.

#### 4.1.5 Skills' Binding

While a player participates in a given task, he/she contributes an amount of experience level for at least one required skill. The amounts offered are being bound by the task until the task's time window elapses or the player withdraws his/her participation. This means that the player cannot re-use these amounts of experience level, that is the player cannot offer them to another task while they are bound.

Players are to retrieve the amounts of experience level they contributed in a certain task when its time window elapses. Players can also retrieve their contributions whenever they withdraw their participation, however, as we will see later on, participation's withdrawal worsens a player's statistics and reputation.

#### 4.1.6 Coalition's Empathy

Social cohesion arises when bonds link members of a social group to one another and to the group as a whole. Although cohesion is a multi-faceted process, it can be broken down into four main components: social relations, task relations, perceived unity, and emotions. Members of strongly cohesive groups are more inclined to participate readily and to stay with the group.

In an attempt to take into consideration the concept of social cohesion and to estimate how well the players will cooperate, we measure the empathy within a coalition.

Each player defines a positive real number  $e_i^j \in [0, 1]$ , which answers the question "how much player i likes player j". When  $e_i^j = 1$  then player i completely likes player j, while when  $e_i^j = 0$  then player i completely dislikes player j. When  $e_i^j$  is not yet defined, it is set to 0.5. Given a coalition C, we measure the empathy within C as:

$$empathy(C) = \frac{\sum_{i,j \in C | i \neq j} e_i^j}{|C|^2 - |C|}$$

The way the empathy within a coalition is calculated, it is not hard to see that:

$$0 \leq empathy(C) \leq 1$$

This means that each potential coalition has a real number always in range  $[0, 1]$ , which indicates the degree of cohesiveness the coalition has; we can also see this as a percentage of total cohesiveness that can be achieved within a given coalition.

In our setting we use coalition's empathy as a probability of a coalition winning in a particular task; that is, given a task and a formed coalition, we say that if the coalition meets the task's requirements, the coalition has probability equal to coalition's empathy to successfully fulfil the task.

#### 4.1.7 Task's Outcome

When a task's time window elapses, the task is being evaluated. It can either be a *success* or a *failure*. Each task is viewed as a hedonic k-weighted voting game, so

a task is a success if the coalition formed to perform this task has utility  $v(C) = 1$ , otherwise it is a failure.

A coalition  $C$  in a hedonic  $k$ -WVG has utility  $v(C) = 1$  when all the quotas are met or exceeded. However, as we mentioned above, we take into account the coalition's empathy to determine its probability of success, since strongly cohesive coalitions are more likely to work together in good terms and fulfil successfully a task, thus achieving the coalition's win.

Consequently, we say that a given coalition  $C$  is winning in the task  $t$  with probability  $p = \text{empathy}(C)$ , if and only if

$$\sum_{i \in C} w_i^j \geq q^j, \quad \text{for } j = 1, \dots, k$$

where  $w_i^j$  is the amount of experience level player  $i$  contributed in required skill  $j$ , and  $q^j$  is the task's required experience level of skill  $j$ . Otherwise, the coalition  $C$  is losing in the task  $t$ .

#### 4.1.8 Skills' Ranking

Each player possesses all eight skills, as we previously mentioned, in (possibly) different experience level. Players define an ordering of preferences over their skills. In this way, players show their interest towards their skills.

The skill preference list of a player is taken into account when his/her reward is defined after a successfully completed task, as we will see in next subsection. Players, can change the ordering of their skills as many times as they wish. If this preference list is made carefully, a player can maximize his/her gained reward.

#### 4.1.9 Reward

If a task is a success, each participant gains two types of rewards. These are:

- **The skill reward**  
Each player receives an extra amount of experience level for each skill required by the task.
- **The total reward**  
In order to have a single metric that reflects the overall performance of the players in the game, each player has a quantity that we call **total reward**.

Firstly, for each member of the coalition we compute his/her power according to Shapley value<sup>2</sup>. Then we multiply it with the player's weight on each skill, that is the amount of experience level the player contributed to each skill.

---

<sup>2</sup>As we will see in Chapter 5, we studied alternative methods for calculating the skill rewards.

The *skill rewards* are the above products:  $r_i^j = \phi_i(C) \cdot w_i^j$ , where  $\phi_i(C)$  is the Shapley value of player  $i$  in coalition  $C$ , and  $w_i^j$  is the amount of experience level player  $i$  contributed to required skill  $j$ .

The *total reward* is the weighted average of the above products. The weight of product  $r_i^j$  is considered to be the quantity  $8 - \text{skill\_preference\_list}(i, j)$ , where  $\text{skill\_preference\_list}(i, j)$  is the position of skill  $j$  in player's  $i$  skill preference list. The constant 8 in the above expression corresponds to the total number of skill a player has in our working example and experiments; we subtract the position (indicated by a integer in  $[1, 8]$ ) from the constant in order to give a greater weight to a higher position; we assume that the first place is denoted with the integer 1 and it is higher than the second place denoted with the integer 2, et cetera. So the total reward is:

$$r_i = \frac{\sum_j r_i^j \cdot (8 - \text{skill\_preference\_list}(i, j))}{\sum_j (8 - \text{skill\_preference\_list}(i, j))}, \quad \text{for } j = 1, \dots, k$$

#### 4.1.10 Players' Statistics

For each player we keep some statistics. We count the number of player's participations and withdrawals. We also count the number of tasks a player has participated in that were a success, and the number of tasks the player was pivotal in, as the criticality was defined in section 2.3.1.

We define the following percentages:

- **Consistency :**  
Is the fraction  $\frac{\#participation - \#withdraws}{\#participations}$ , which indicates the player's tendency to withdraw from a task; the higher consistency is, the less likely is for the player to withdraw from a task in the future.
- **Criticality :**  
Is the fraction  $\frac{\#pivotal}{\#participations - \#withdraws}$ , which shows the percentage of the tasks a player was pivotal for.
- **Success/Failure:**  
These are the fractions  $\frac{\#success}{\#participations - \#withdraws}$  and  $(1 - \frac{\#success}{\#participations - \#withdraws})$ , respectively. They show the percentage of the tasks the player participated in, that were a success (the former) and that were a failure (the latter).
- **Friendly/Popular:**  
The percentage of friendliness is  $\frac{\sum_j e_i^j}{|\sum_j e_i^j|}$ , where  $e_i^j$  is the empathy player  $i$  has on player  $j$ ; the percentage of popularity is  $\frac{\sum_j e_j^i}{|\sum_j e_j^i|}$ , where  $e_j^i$  is the empathy player  $j$  has on player  $i$ .

Having the above statistics, we also define player's reputations as:  
 Let  $\alpha(i) = 0.75 \cdot Consistency(i) + 0.2 \cdot Criticality(i) + 0.05 \cdot Friendly(i)$  then

$$Reputation(i) = \begin{cases} \text{Beginner,} & \text{if } \alpha(i) < 20, \\ \text{Amateur,} & \text{if } 20 \leq \alpha(i) < 40, \\ \text{Regular,} & \text{if } 40 \leq \alpha(i) < 60, \\ \text{Great,} & \text{if } 60 \leq \alpha(i) < 75, \\ \text{Excellent,} & \text{if } 75 \leq \alpha(i) < 90, \\ \text{Master,} & \text{if } 90 \leq \alpha(i) \end{cases}$$

As we can see, a player that want to maintain a good reputation should be careful with the withdrawals he/she makes, and should be 'friendly' to others. However, the player cannot influence in any way the factor 'criticality'.

# Chapter 5

## Simulations

In order to see the game in progress, and to study the behavior of different types of players, we ran some experimental simulations. In the following section we present the agents used in the game, and the different versions of the game we studied. We also show the results of the simulations, along with some conclusions.

### 5.1 Agent Profiles

We have created three basic agent profiles, focusing on their risk behavior. This was done in accordance with the risk profiles presented in section 2.5. An agent's profile determines the player's behavior when making decisions that is, when the player has to decide if he/she wants to take part in a certain task, and when the player has to decide whether he/she wants to withdraw from a certain task.

**Risk Averse Player** In line with the behavior identified in section 2.5, a player is risk averse if he/she tend to avoid risky tasks, that is tasks with low probability of winning. The thresholds a risk averse player  $i$  sets in order to determine participation in a coalition  $C$ , are the following:

- $EF = X \sim \mathcal{N}(0.75, 0.02^2)$  for the formed coalition's  $C$  empathy : that is for a risk averse player to participate,  $C$ 's empathy should exceed  $EF$ .
- $SR = X \sim \mathcal{N}(0.7, 0.02^2)$  for the formed coalition's success rate (this is the average percentage of success of the players within the coalition) : that is, a risk averse player will not participate in  $C$  if  $C$ 's success rate is lower than  $SR$ .

In Figure 5.1 we provide the pseudocode describing the decision making of a risk averse agent.

**Risk Neutral Player** In line with the behavior identified in section 2.5, a player is risk neutral if he/she neither risk averse or risk seeking; neutral risk players have the following thresholds:

```

makeDecision(player , task ):
    if !alreadyParticipating (player , task ):
        decide2join (player , task )
    else :
        decide2withdraw (player , task )

decide2join (player , task ):
    if coalitionSuccessRate (task ) >= SR OR
        coalitionSuccessRate (task , player ) >= coalitionSuccessRate (task ):
        if coalitionEmpathy (task ) >= EF OR
            coalitionEmpathy (task , player ) >= coalitionEmpathy (task ):
            join (player , task )
        else :
            join (player , task ) with probability 0.4
    else :
        join (player , task ) with probability 0.2

decide2withdraw (player , task ):
    if coalitionEmpathy (task ) < EF:
        withdraw (player , task )
    else if coalitionSuccessRate (task ) < SR:
        withdraw (player , task ) with probability 0.6
    else :
        do not withdraw

```

Figure 5.1: Pseudocode describing the decision making of a risk averse agent

```

makeDecision(player , task ):
    if !alreadyParticipating (player , task ):
        decide2join (player , task )
    else :
        decide2withdraw (player , task )

decide2join (player , task ):
    if coalitionSuccessRate (task )>=SR OR
        coalitionSuccessRate (task , player )>=coalitionSuccessRate (task ):
        if coalitionEmpathy (task )>=EF OR
            coalitionEmpathy (task , player )>=coalitionEmpathy (task ):
            join (player , task )
        else :
            join (player , task ) with probability 0.65
    else :
        join (player , task ) with probability 0.45

decide2withdraw (player , task ):
    if !alreadyWithdrawOnce (player , task ):
        if coalitionEmpathy (task ) < EF:
            withdraw (player , task ) with probability 0.35
        else if coalitionSuccessRate (task ) < SR:
            withdraw (player , task ) with probability 0.2
        else :
            do not withdraw

```

Figure 5.2: Pseudocode describing the decision making of a risk neutral agent

- $EF = X \sim \mathcal{N}(0.6, 0.02^2)$  for the formed coalition's  $C$  empathy ; that is for risk neutral player to participate,  $C'$  empathy should exceed  $EF$ .
- $SR = X \sim \mathcal{N}(0.5, 0.02^2)$  for the formed coalition's  $C$  success rate : that is, a risk neutral player will not participate in  $C$  if  $C'$ 's success rate is lower than  $SR$ .

In Figure 5.2 we provide the pseudocode describing the decision making of a risk neutral agent.

**Risk Seeking Player** Risk seeking players are prone to participate in situations that have high risk index. In our setting, risk seeking players have the following threshold:

- $EF = X \sim \mathcal{N}(0.5, 0.02^2)$  for the formed coalition's  $C$  empathy ; that is for a risk seeking player to participate,  $C'$  empathy should exceed  $EF$ .
- $SR = X \sim \mathcal{N}(0.4, 0.02^2)$  for the formed coalition's  $C$  success rate : that is, a risk seeking player will not participate in  $C$  if  $C'$ 's success rate is lower than  $SR$ .

In Figure 5.3 we provide the pseudocode describing the decision making of a risk seeking agent.

```

makeDecision(player, task):
  if !alreadyParticipating(player, task):
    decide2join(player, task)
  else:
    do not withdraw

decide2join(player, task):
  if coalitionSuccessRate(task) >= SR OR
    coalitionSuccessRate(task, player) >= coalitionSuccessRate(task):
    if coalitionEmpathy(task) >= EF OR
      coalitionEmpathy(task, player) >= coalitionEmpathy(task):
      join(player, task)
    else:
      join(player, task) with probability 0.75
  else:
    join(player, task) with probability 0.7

```

Figure 5.3: Pseudocode describing the decision making of a risk seeking agent

Note that even when the required thresholds are not met, players (regardless their risk profile) join a task with some probability. This is to motivate players to begin participating in tasks. Table 5.1 summarizes the risk characteristics of each player profile.

	Empathy Factor	Success Rate	Pr if EF not met		Pr if SR not met	
			Join	Withdraw	Join	Withdraw
Risk Averse	$X \sim \mathcal{N}(0.75, 0.02^2)$	$X \sim \mathcal{N}(0.7, 0.02^2)$	0.4	1	0.2	0.6
Risk Neutral	$X \sim \mathcal{N}(0.6, 0.02^2)$	$X \sim \mathcal{N}(0.5, 0.02^2)$	0.65	0.35	0.45	0.2
Risk Seeking	$X \sim \mathcal{N}(0.5, 0.02^2)$	$X \sim \mathcal{N}(0.4, 0.02^2)$	0.75	0	0.7	0

Table 5.1: Thresholds and probabilities per risk profile. “Pr” stands for “probability to join/withdraw”.

### 5.1.1 Empathy Beliefs Updating

All agents, regardless of risk profile, follow the same empathy beliefs updating policy. When a task is evaluated as a failure, each participant decides with probability  $\sim \mathcal{N}(0.65, 0.02^2)$  to reduce their empathy towards players who have withdrawn by a value  $\sim \mathcal{N}(0.05, 0.01^2)$ . When a task is evaluated as a success, each participant

increases their empathy towards his/her partners by a value  $\sim \mathcal{N}(0.1, 0.01^2)$  if a partner was pivotal in the task or by a value  $\sim \mathcal{N}(0.05, 0.01^2)$  if a partner was not pivotal.

This is admittedly an ad-hoc and simplistic method to update beliefs. However, given that we are ignorant of the exact process for updating empathy beliefs used by people, this is also a *uniform* and *low complexity* way to perform this update in simulations. Using alternative beliefs' updating methods is future work.

### 5.1.2 Skills Ranking Updating

All agents, regardless of risk profile update their skill preference ordering in the same way. As mentioned earlier, if a player chooses carefully his/her skill preference ordering, the player can maximize his/her imminent total reward. All agents intend to maximize their imminent total reward, thus each player change his/her ordering as follows:

1. Choose the task that is to expire (reach an end) the earliest.
2. Rank the contributions the player has made to this task in descending order.
3. Set first in skill preference list the skill that corresponds to the highest contribution; set second in preference list the skill that corresponds to the second highest contribution; and so on. The skills not corresponding to a contribution get a randomly chosen position in the skill preferences list.

## 5.2 Game Versions

We ran simulations over four versions of the game. The versions differ in the way the players' rewards are calculated after a successful task. As already discussed, a player's rewards mainly depend on the power that player has within the formed coalition; that is, the player's Shapley value. Let  $w_i^j$  be the amount of experience level of skill  $j$  that player  $i$  contributed to a given task  $t$ , and  $q^j$  be the required experience level by  $t$  for the skill  $j$ .

Next we present the different versions of the game.

In all versions, each task is viewed as a hedonic  $k$ -weighted voting game, as mentioned.

### 5.2.1 Version 0

By definition a  $k$ -WVG, and therefore a hedonic  $k$ -WVG, is a collection consisting of  $k$  WVGs over the same set of players. In this version, a player's skill reward depends on the component game corresponding to this particular skill. To be more specific, we find the Shapley value of the player in each one of the component games:

$$\phi_i^j(C, v^j) = \frac{1}{|C|} \sum_{\pi \in \Pi} \left[ v^j(S_\pi(i) \cup \{i\}) - v^j(S_\pi(i)) \right]$$

where  $C \subseteq N$  is the coalition participating in this task and  $v^j$  is the utility function<sup>1</sup> of component game  $j$ . Having  $k$  different Shapley values, we define the  $k$  skill rewards as:

$$r_i^j = \lceil \phi_i^j(C, v^j) \cdot w_i^j \rceil$$

where  $w_i^j$  is the amount of experience level of skill  $j$  that player  $i$  contributed in the task.

The total reward is defined as:

$$r_i = \left\lceil \frac{\sum_j r_i^j \cdot (8 - \text{skill\_preference\_list}(i, j))}{\sum_j (8 - \text{skill\_preference\_list}(i, j))} \right\rceil, \quad \text{for } j = 1, \dots, k \quad ,$$

where  $\text{skill\_preference\_list}(i, j)$  is an integer number in the range  $[1, 8]$  which indicates the position of skill  $j$  in player's  $i$  skill preference list.

Referring to the Definition 20, we see that the skill reward  $r_i^j$  corresponds to the pay-off  $x_i^j$  related to the utility function of the component game  $\mathcal{G}^j$ :  $r_i^j = f(w_i^j)$ .

### 5.2.2 Version 1

In game version 1, we change our policy towards the zero and dummy contributions. With  $\mathcal{R}$  being the set of the task's required skills, a player can contribute to any non-empty subset of the required skills ( $S \subseteq \mathcal{R} : S \neq \emptyset$ ), so we imply that the player's contribution to the rest of the skills ( $D = \mathcal{R} \setminus S$ ) is zero. A dummy contribution, results from a dummy player within a game, as the dummy player is defined in Definition 19. We remind the reader that the Shapley value of a dummy player is equal to zero ( $\phi_i(C, v) = 0 \Leftrightarrow i$  is *Dummy*).

While in game version 0 a zero or dummy contribution has as a result a zero skill reward, in version 1 we define the skill reward as :

$$r_i^j = \begin{cases} m & , \text{ if } \phi_i^j(C, v^j) = 0 \text{ or } w_i^j = 0 \\ \lceil \phi_i^j(C, v^j) \cdot w_i^j \rceil & , \text{ otherwise} \end{cases}$$

where  $m \sim \mathcal{U}[0, 2]$  is a uniformly chosen integer in range  $[0, 2]$  and it is the same for all players participating in the task. The definition of total reward has not changed:

$$r_i = \left\lceil \frac{\sum_j r_i^j \cdot (8 - \text{skill\_preference\_list}(i, j))}{\sum_j (8 - \text{skill\_preference\_list}(i, j))} \right\rceil, \quad \text{for } j = 1, \dots, k$$

---

<sup>1</sup>The utility function of a component game  $j$  is defined as :  $v^j(C) = \begin{cases} 1 & , \text{ if } \sum_{i \in C} w_i^j \geq q^j \\ 0 & , \text{ otherwise} \end{cases}$

### 5.2.3 Version 2

In game version 2, we change again the policy towards zero and dummy contributions. This time, instead of assigning a constant number to all players, we take into account the other contributions each player has made in this task. We find the minimum non-zero skill reward of a player, and we assign it to all skill rewards that derive from a zero or dummy contribution.

So we define the skill reward as :

$$r_i^j = \begin{cases} \min r_i^a |_{\{a \neq j \wedge r_i^a > 0\}} & , \text{ if } \phi_i^j(C, v^j) = 0 \text{ or } w_i^j = 0 \\ \lceil \phi_i^j(C, v^j) \cdot w_i^j \rceil & , \text{ otherwise} \end{cases}$$

In this way, each player influences his/her skill reward which comes from a zero or dummy contribution. Once again, the definition of total reward remains unchanged:

$$r_i = \left\lceil \frac{\sum_j r_i^j \cdot (8 - \text{skill\_preference\_list}(i, j))}{\sum_j (8 - \text{skill\_preference\_list}(i, j))} \right\rceil, \quad \text{for } j = 1, \dots, k$$

### 5.2.4 Version 3

Last but not least, in the final game version we do not view the hedonic k-WVG as a union of WVGs, but as a single entity. In this version, we find a single Shapley value which reflects the total power of the player within the game:

$$\phi_i(C, v) = \frac{1}{|C|} \sum_{\pi \in \Pi} [v(S_\pi(i) \cup \{i\}) - v(S_\pi(i))]$$

As far the policy towards zero and dummy contributions is concerned, we use the policy introduced in version 2. That is, we assign to skill rewards that correspond to zero or dummy contributions, the minimum non-zero skill reward they have obtained. To be exact, this policy is applied to zero contribution, and not to the dummy ones, since a dummy contribution results to a zero Shapley value. Therefore, a dummy contribution has as a result zero skill rewards.

The skill reward is defined as :

$$r_i^j = \begin{cases} \min r_i^a |_{\{a \neq j \wedge r_i^a > 0\}} & , \text{ if } w_i^j = 0 \\ \lceil \phi_i(C, v^j) \cdot w_i^j \rceil & , \text{ otherwise} \end{cases}$$

Once again, the total reward's definition remains unchanged:

$$r_i = \left\lceil \frac{\sum_j r_i^j \cdot (8 - \text{skill\_preference\_list}(i, j))}{\sum_j (8 - \text{skill\_preference\_list}(i, j))} \right\rceil, \quad \text{for } j = 1, \dots, k$$

In this case, referring to the Definition 20, the skill reward  $r_i^j$  corresponds to the pay-off  $x_i^j$  related to the utility function of the game  $\mathcal{G}$ :  $r_i^j = f(u)$ .

In Table 5.2 we see the differences among the four simulated versions.

	Shapley Value	Skill Reward	Total Reward
version 0	$\phi_i^j(C, v^j)$	$\lceil \phi_i^j(C, v^j) \cdot w_i^j \rceil$	$r_i = \lceil \frac{\sum_j r_i^j \cdot (8 - \text{skill\_preference\_list}(i,j))}{\sum_j (8 - \text{skill\_preference\_list}(i,j))} \rceil$
version 1	$\phi_i^j(C, v^j)$	$r_i^j = \begin{cases} m, & \text{if } \phi_i^j(C, v^j) = 0 \text{ or } w_i^j = 0 \\ \lceil \phi_i^j(C, v^j) \cdot w_i^j \rceil, & \text{otherwise} \end{cases}$	$r_i = \lceil \frac{\sum_j r_i^j \cdot (8 - \text{skill\_preference\_list}(i,j))}{\sum_j (8 - \text{skill\_preference\_list}(i,j))} \rceil$
version 2	$\phi_i^j(C, v^j)$	$r_i^j = \begin{cases} \min r_i^a   \{a \neq j \wedge r_i^a > 0\}, & \text{if } \phi_i^j(C, v^j) = 0 \text{ or } w_i^j = 0 \\ \lceil \phi_i^j(C, v^j) \cdot w_i^j \rceil, & \text{otherwise} \end{cases}$	$r_i = \lceil \frac{\sum_j r_i^j \cdot (8 - \text{skill\_preference\_list}(i,j))}{\sum_j (8 - \text{skill\_preference\_list}(i,j))} \rceil$
version 3	$\phi_i(C, v)$	$r_i^j = \begin{cases} \min r_i^a   \{a \neq j \wedge r_i^a > 0\}, & \text{if } w_i^j = 0 \\ \lceil \phi_i(C, v^j) \cdot w_i^j \rceil, & \text{otherwise} \end{cases}$	$r_i = \lceil \frac{\sum_j r_i^j \cdot (8 - \text{skill\_preference\_list}(i,j))}{\sum_j (8 - \text{skill\_preference\_list}(i,j))} \rceil$

Table 5.2: Different simulated versions

## 5.3 Technical Aspects

For all the agents and the necessary code needed to run our, we used the programming language *Python*, which is a widely used high-level programming language for general-purpose programming. For the necessary database *MySQL* was used, which is an open-source relational database management system (RDBMS).

### 5.3.1 DataBase

For our system's needs we use a database of four tables. The Player table contains characteristics of each player such as his/her name, the experience level of each skill the player possesses, the ordering of his/her skills and his/her risk profile. The Task table keeps all organized tasks. For each task we keep its id, its name, its requirements (meaning the experience level of each skill required), its time window (we note it as duration), the remaining time and the task's status ('Success', 'Failure', 'Pending'). The Participation table relates players with tasks, each player may have as many participations as he/she likes and each task may have more than one participants. The main characteristics composing a participation are the pair  $\langle \text{player}, \text{task} \rangle$ , along with the player's contribution, the rewards gained if the task's time window has elapsed, and an indicator if the player has withdrawn. We also have the Empathy table, which shows the empathy  $e$  one player has towards another.

In Figure 5.4 we can see the entity relationship diagram used in our system.

### 5.3.2 Computation of Shapley Value

The computation of Shapley Value, as derived from the Definition 17, is known to be computationally hard. Given a coalition of players  $C$  and  $|C| = n$ , the complexity of computing the Shapley value for *just one* player is  $\Theta(n!)$ . Consequently, the complexity of computing the Shapley value for all the players within the coalition  $C$  is  $\Theta(n \cdot n!)$ . It is not hard to see, that even for coalitions with  $n$  as low as 6, in order to compute all the Shapley values we need to sum up 36288000 values. The above complexity is forbidding. Thus, we turn to an approximating algorithm introduced in [4]. This algorithm is based on randomly sampling coalitions. As shown in [4],

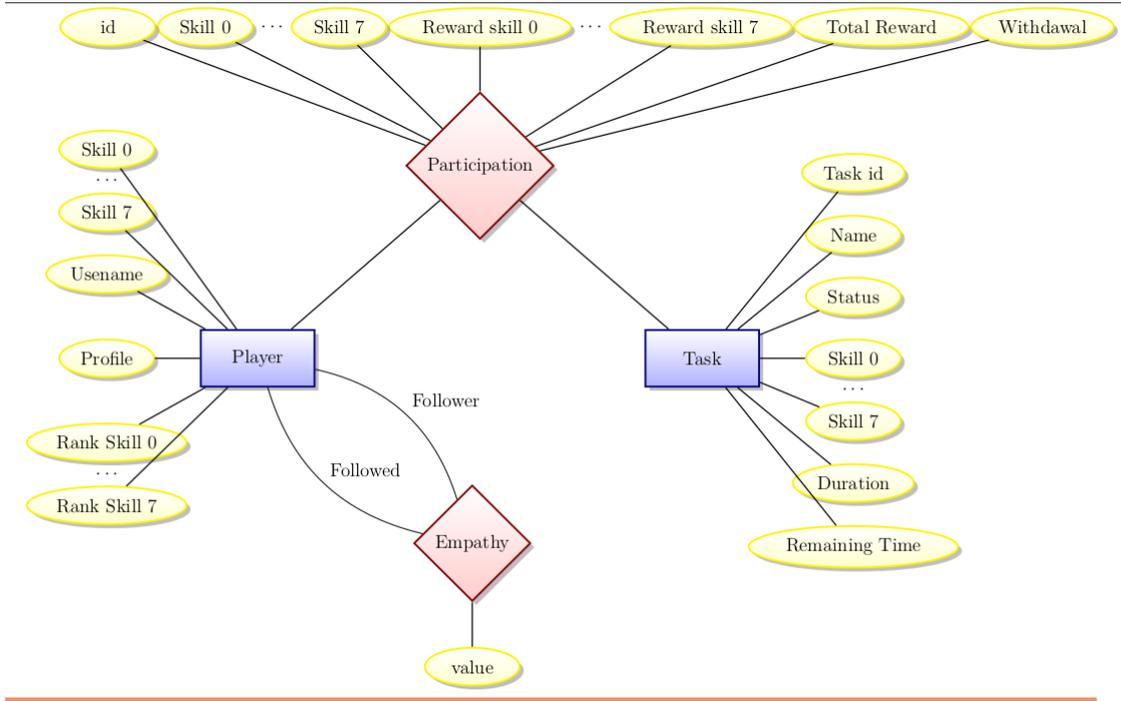


Figure 5.4: ER Schema

the Shapley value of a player  $i$  coincides with the probability of player  $i$  being pivotal (Definition 15) in a random permutation. The approximating algorithm guarantees that given an *accuracy*  $\epsilon$  and a *confidence level*  $1 - \delta$ , the correct Shapley value lies in the interval  $[\hat{\phi}_i - \epsilon, \hat{\phi}_i + \epsilon]$  with probability  $1 - \delta$ . The required number of samples is:  $samples = \frac{\ln(\frac{2}{\delta})}{2\epsilon^2}$ . In Figure 5.5 we present the approximating algorithm for computing the Shapley value.

In our application we set the parameters accuracy  $\epsilon = 0.06$  and confidence level  $1 - \delta = 1 - 0.000001 = 0.999999$ , which requires  $samples = \frac{\ln(\frac{2}{\delta})}{2\epsilon^2} = 2015.09135257$ .

### 5.3.3 Simulations Pseudocode

In all four simulated versions, we used the exact same set of players, that is in each version each player starts with the same experience level of his/her skills, and the same set of tasks. We use a set of thirty players, ten for each risk profile described above. The tasks are loaded in bunches of twenty. That is, we load the first 20 tasks, players carry out these tasks, then we load the next 20 tasks and so on.

The tasks are being ‘organized’, meaning that we load them, with the same or-

```

ApproximatingShapleyValue(delta , epsilon ):
  X=0, k=0
  while k >= ln(2/delta) / (2 * epsilon ^2):
    Randomly choose coalition C which contains i
    k += 1
    if i pivotal in C: X += 1
    phi = X/k
  return phi

```

Figure 5.5: Approximating algorithm for the Shapley value's computation

dering in all four versions. In every time unit<sup>2</sup> the players are to make a decision on each task about joining or withdrawing. We tested both simulations where players were to decide always in the same order, and simulations where players decided in randomly chosen order, without noticing any noteworthy differences. In Figure 5.6 we provide the simulation pseudocode.

The function `evaluateTask`, evaluates a task as 'Success' or 'Failure', computes the approximation of the Shapley value for each participant, and assigns the players' rewards according to the simulated version of the game.

---

<sup>2</sup>For the sake of the simulations, the time window of a task is perceived as a number of loops, i.e. if a task was time window 5, it takes 5 loops for the task to reach to an end.

```

simulation(version , db):
    players = loadPlayers(db)
    tasks = loadTasks(db,20)
    completedTask = 0

    while completedTask<200:
        for t in tasks:
            if t.remaining_time>0:
                t.remaining_time -= 1
                # players are chosen in random order
                for p in players.random_order:
                    c = makeDecision(t,p)
                    if c.do is 'join':
                        join(t,p,c.contribution)
                    elif c.do is 'withdraw':
                        withdraw(t,p)
                elif t.status is 'Pending':
                    evaluateTask(version , t)
                    tasks.remove(t)
                    completedTask += 1
            else:
                task.remove(t)
        if len(tasks)<1:
            tasks = loadTasks(db,20)

```

Figure 5.6: Simulation Pseudocode

## 5.4 Results

We focus on two aspects of the simulations' results. Firstly, we compare the agents' performance across their risk profiles. Then, we compare the agents' performance across on the different game versions. In order to compare the agents' performance we use the following four metrics *consistency*, *critically*, *success rate* and *cumulative total reward*.

The three first metrics (consistency, criticality and success rate) are the percentages we have defined in section 4.1.10. The cumulative total reward is the total reward a player has gathered, that is the sum of all total rewards the player gained by all the successful tasks the he/she participated in.

As we have already mentioned, we have a set of thirty players, ten of each risk profile. For all four metrics, we find the average curve the ten agents corresponding to each risk profile form. We mentioned that we load the tasks in bunches of twenty. To clarify that, we only process the tasks in twenties; the metrics reflect the performance of a player when  $x$  number of tasks have been completed, meaning that the metrics assess the performance the player had when  $x-y$  number of tasks had been completed.

All results depicted in the figures below are averages over 10 runs.

### 5.4.1 Comparing performance across risk profiles

In Figures 5.7-5.10, 5.11-5.14, 5.15-5.18, 5.19-5.22 we present the curves resulting from simulated game version 0, game version 1, game version 2 and game version 3, respectively.

As we can see, the results are more or less as one would have expected. The risk seeking players have higher consistency than neutral risk players, who have higher consistency than risk averse players. The success rate of risk averse players is higher in the beginning, since they are more restrained in their participations. Then however, risk seeking players increase their success rate, since more players, regardless of risk profile, take part in the organized tasks.

Looking at the metric of criticality, risk averse players appear to have higher percentage of criticality in the beginning, similarly to the success rate, since they are more careful about their participations. However, as more and more tasks are completed, risk seeking 'lure' risk averse and risk neutral players to participate in more tasks. Thus, risk averse players, on the one hand, reach their participating thresholds more easily and tend to stabilize their criticality percentage. Risk neutral players, on the other hand, tend to stabilize their criticality percentage in a higher value, but they need more completed tasks to do so. As far as we can see, risk-seeking players show an increasing curve, that is we infer that they need a larger number of completed task in order to stabilize in some percentage value.

The curves of the metric of the total reward, are commulative so they keep rising, as expected. The total reward of risk averse players seems to be always lower than

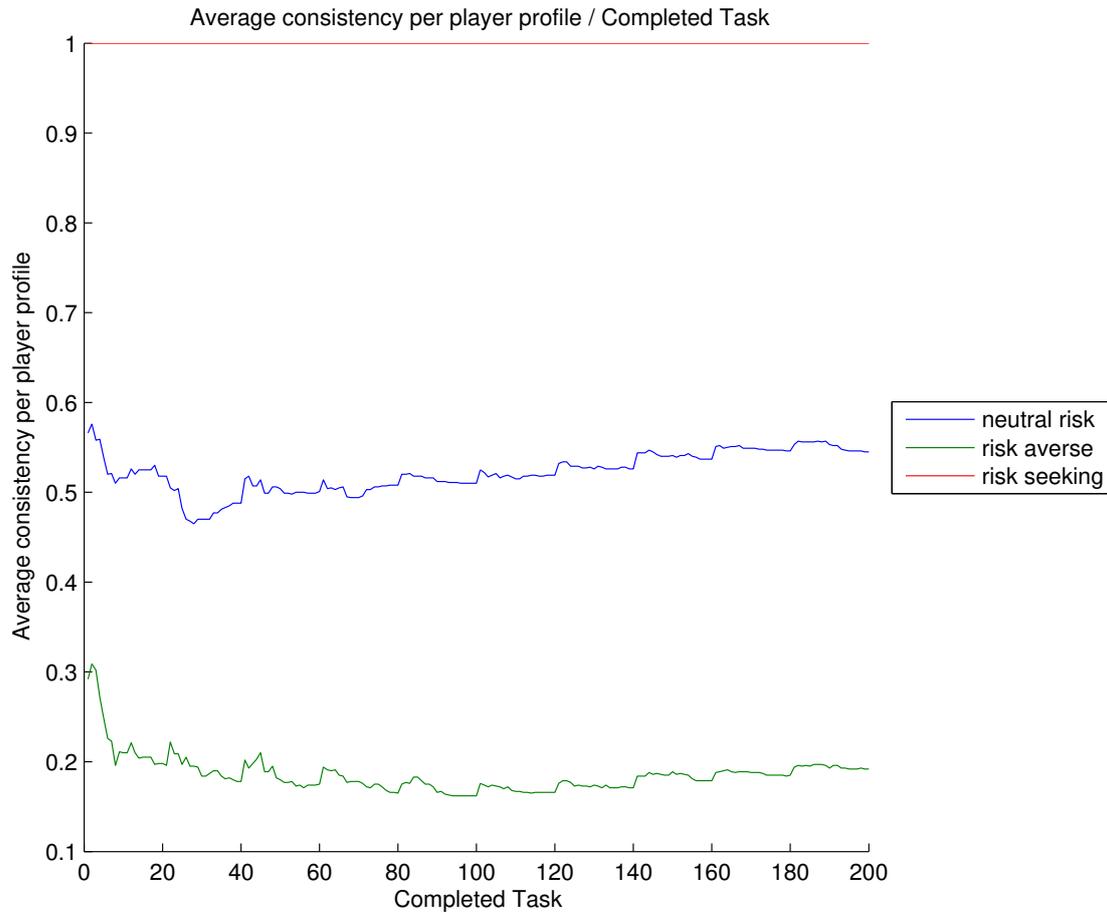


Figure 5.7: The metric of consistency for game version 0 (Note that the curve for risk seeking players is a straight line of value 1).

the risk seeking and risk neutral players, which makes sense since both type of agents have higher success rate, and therefore gain total rewards.

#### 5.4.2 Comparing agents' performance across game versions

In Figures 5.23-5.25, 5.26-5.28, 5.29-5.31, 5.32-5.33 we present the curves of metrics criticality, total reward, success rate, and consistency. In Figure 5.34 we present the number of successful and failed task per simulated game version, while in Figure 5.35 we show the curve of the average percentage of withdrawals per simulated game version.

We focus on the results depicted by the curves of success rate and total reward. Looking at the definitions of skill rewards for each game version, it is not hard to see that version 2 provides a higher potential pay-off than the other versions. However, looking at the above results, we see that since agents do not take into account their

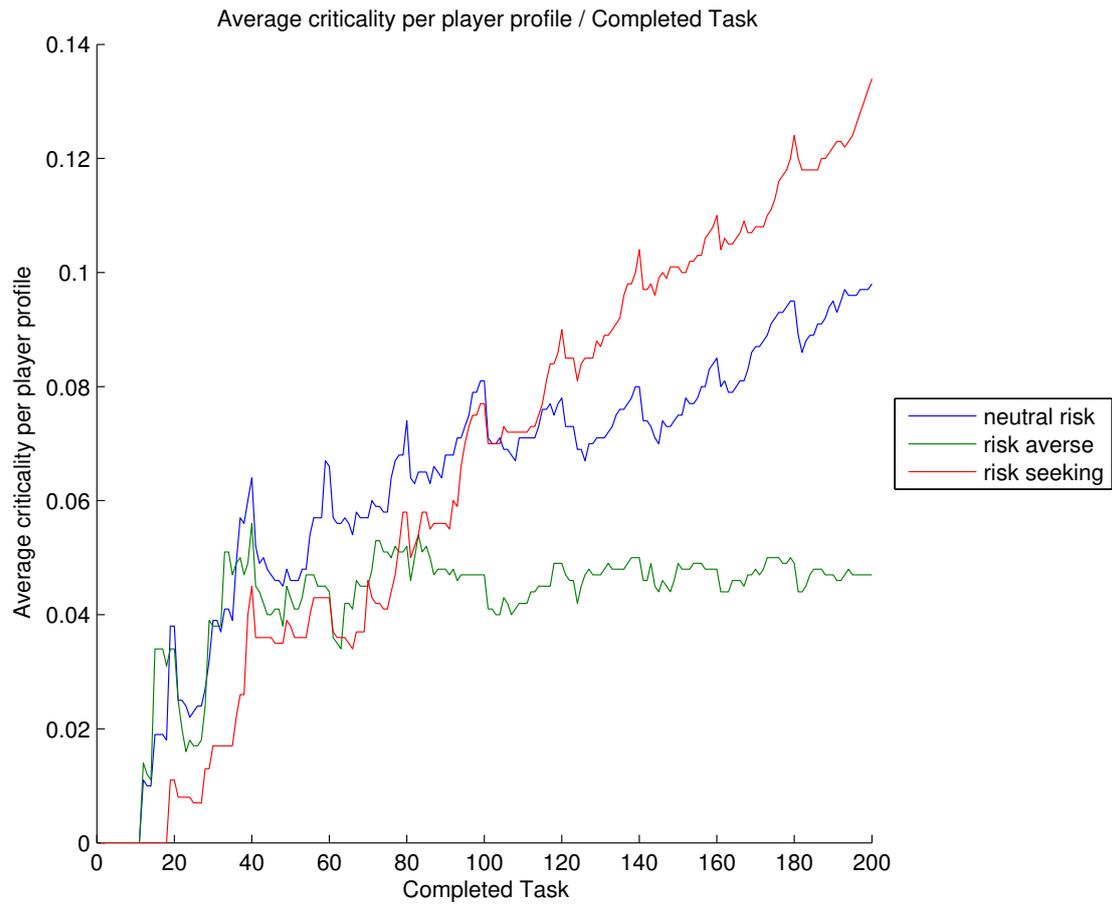


Figure 5.8: The metric of criticality for game version 0.

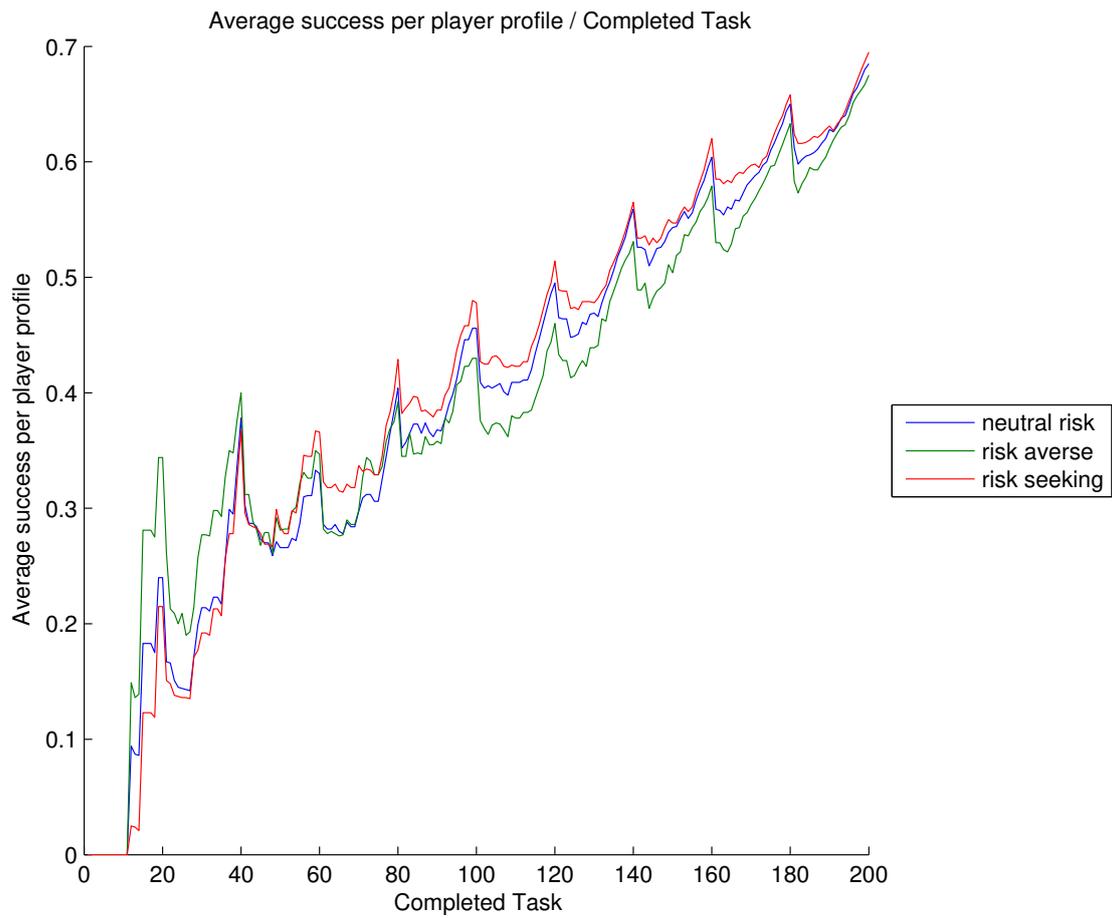


Figure 5.9: The metric of success rate for game version 0.

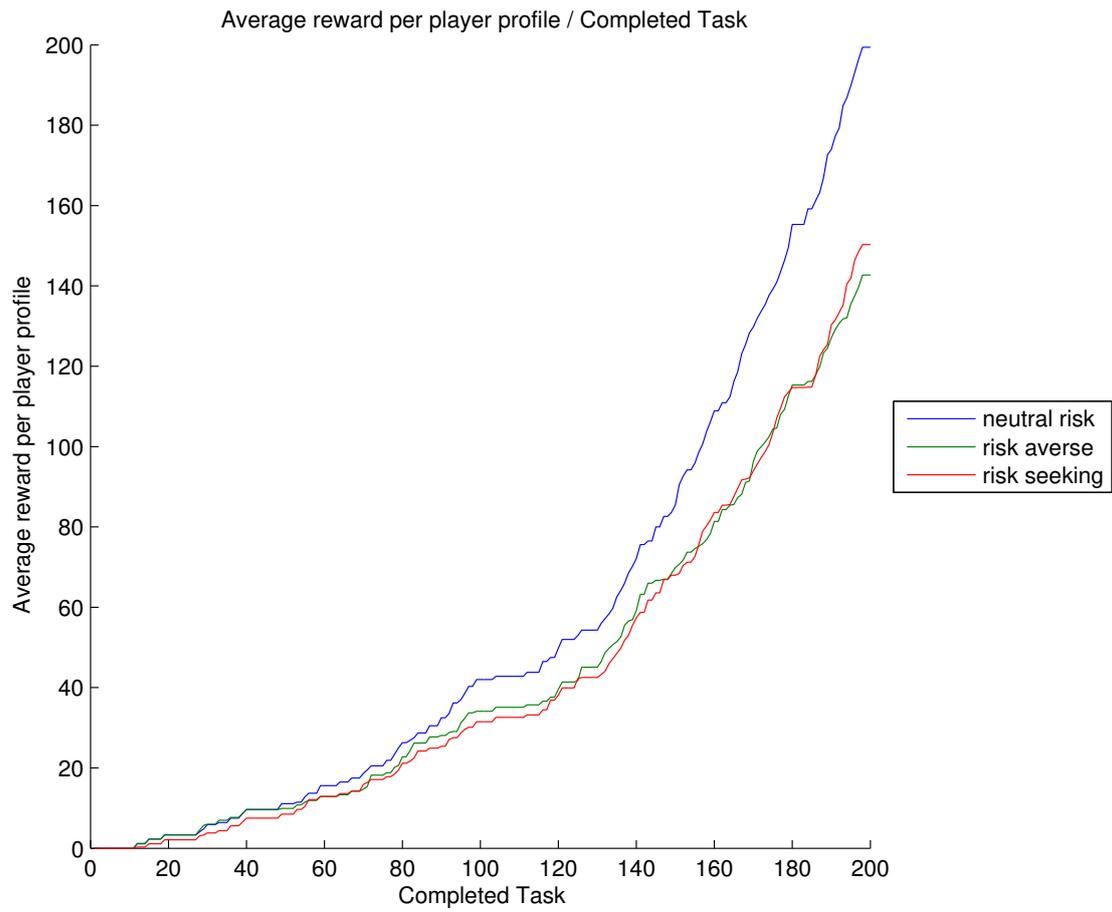


Figure 5.10: The metric of total reward for game version 0.

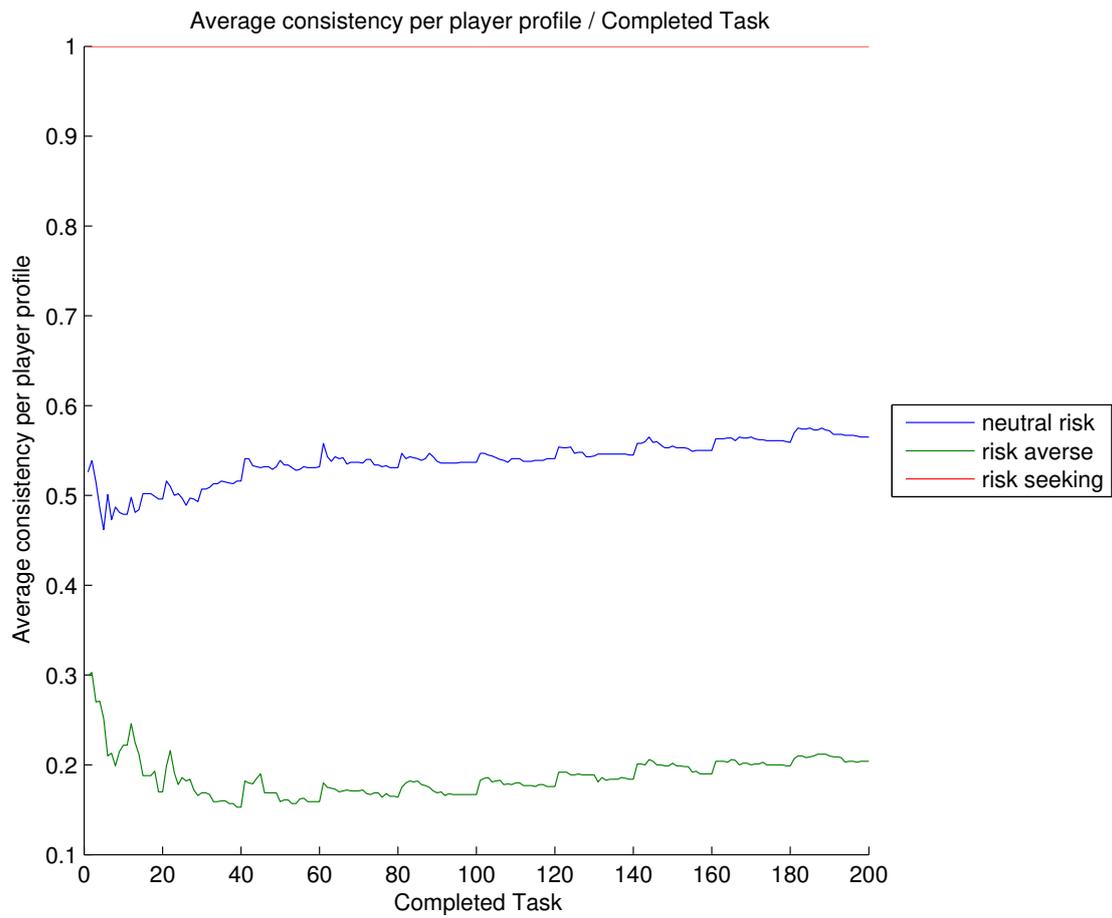


Figure 5.11: The metric of consistency for game version 1 (Note that the curve for risk seeking players is a straight line of value 1).

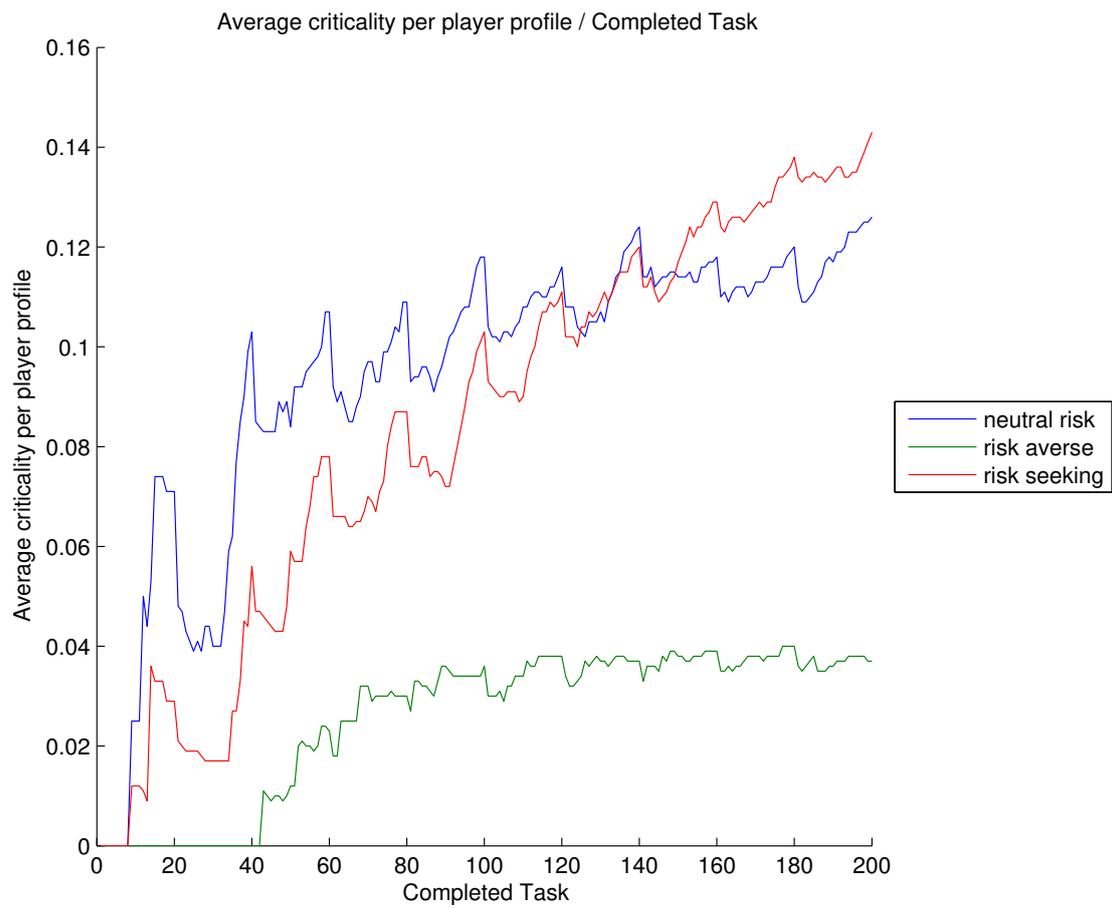


Figure 5.12: The metric of criticality for game version 1.

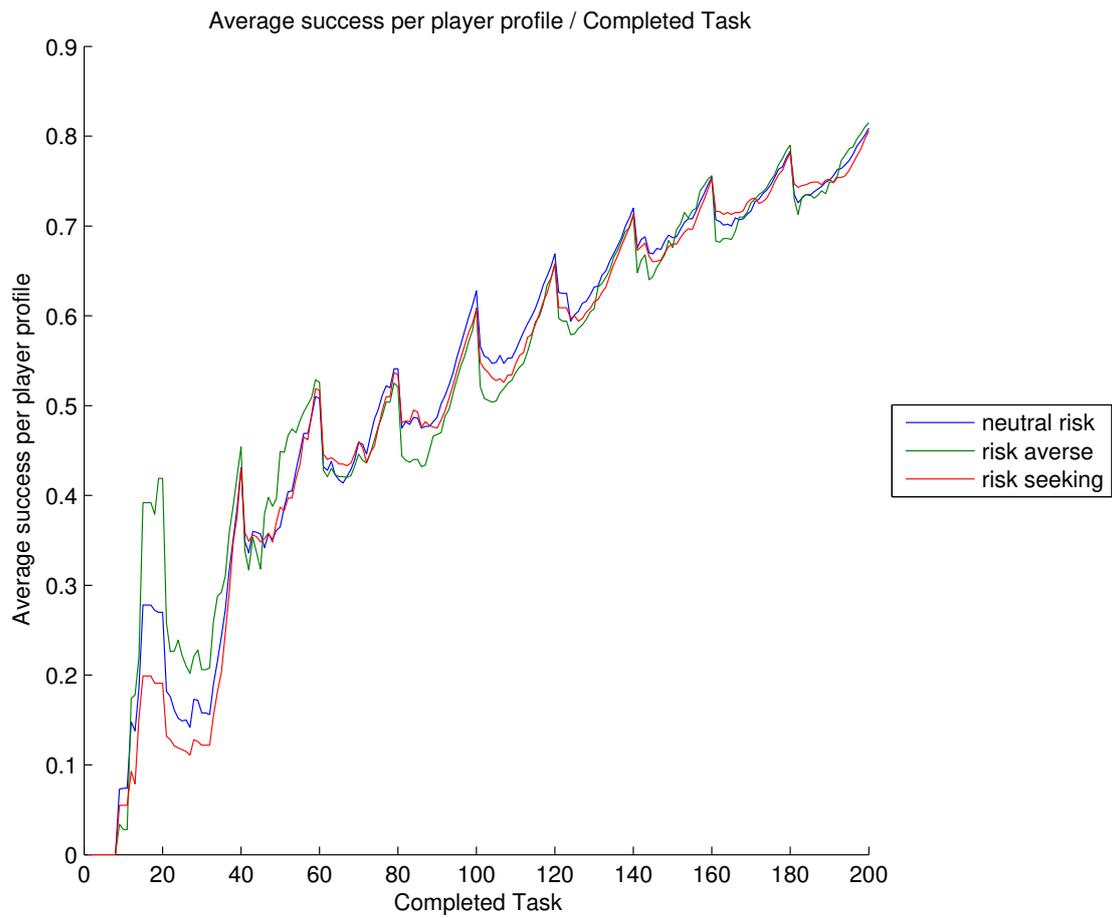


Figure 5.13: The metric of success rate for game version 1.

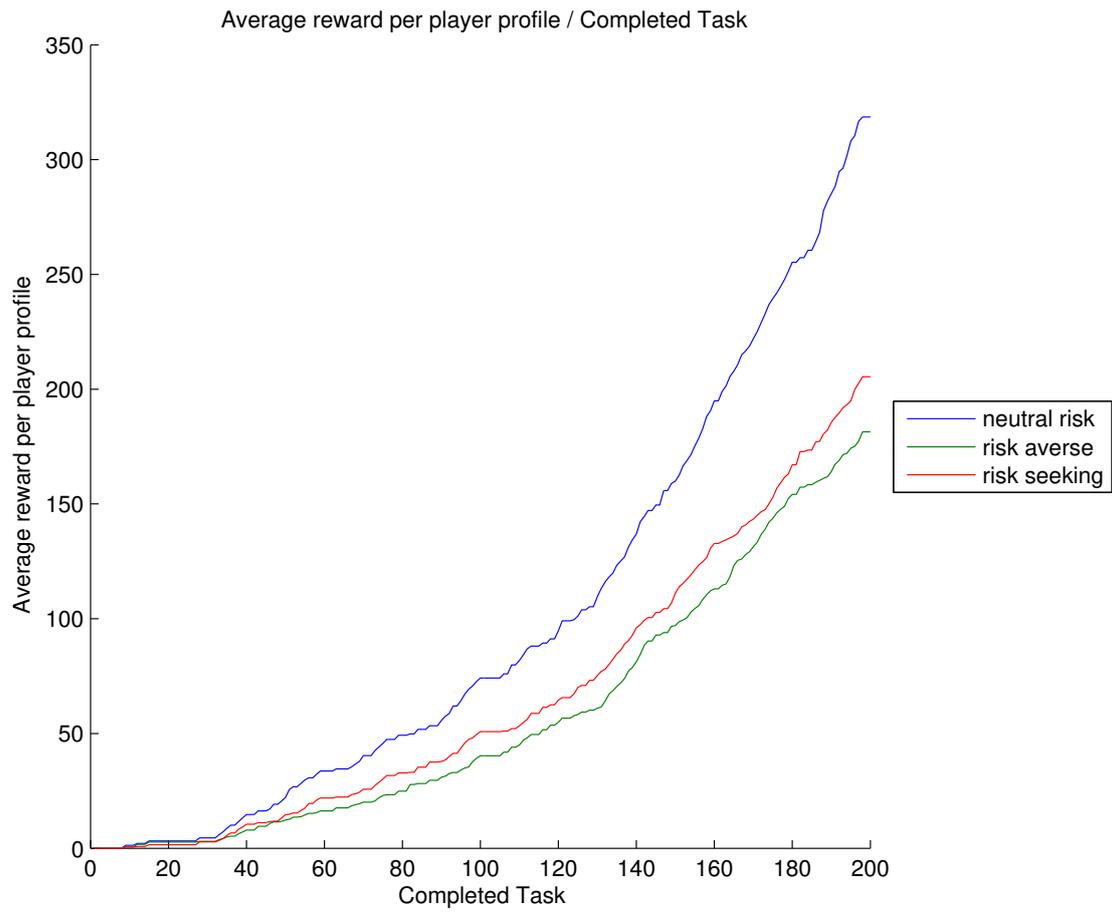


Figure 5.14: The metric of total reward for game version 1.

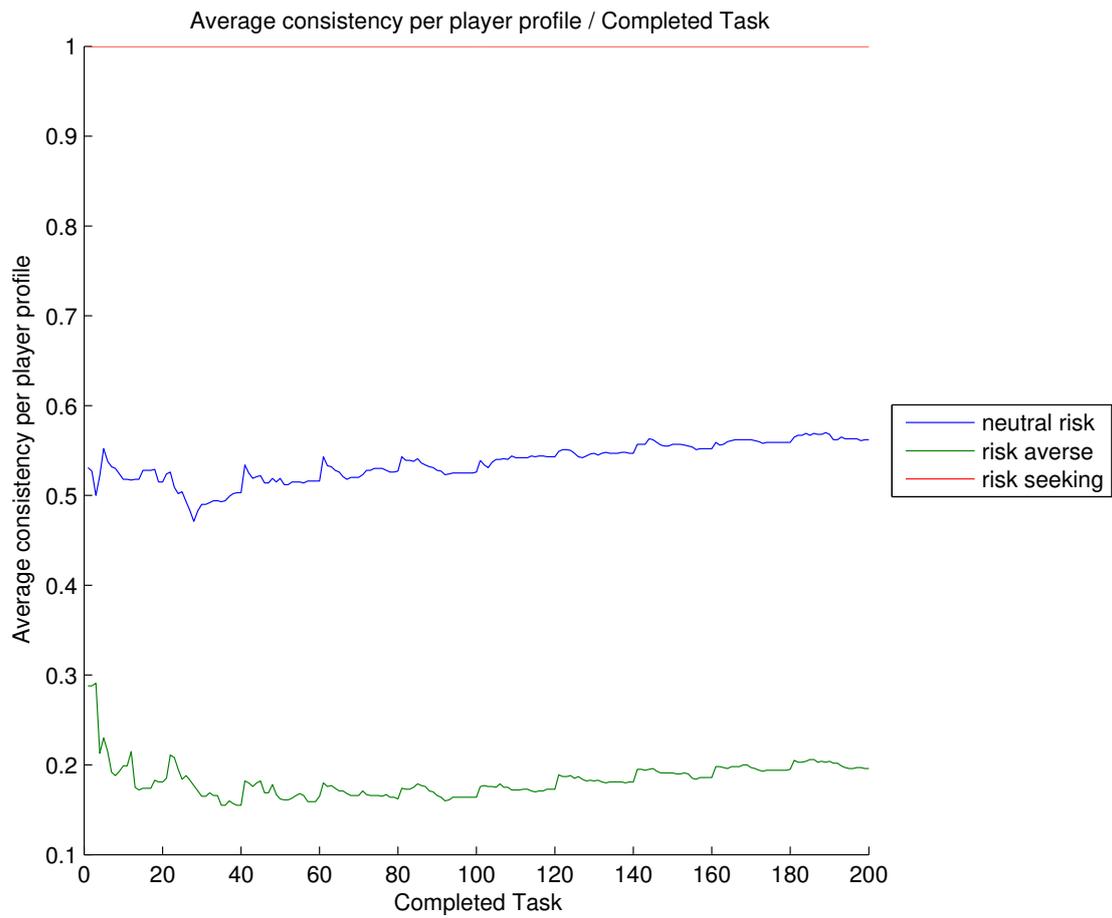


Figure 5.15: The metric of consistency for game version 2 (Note that the curve for risk seeking players is a straight line of value 1).

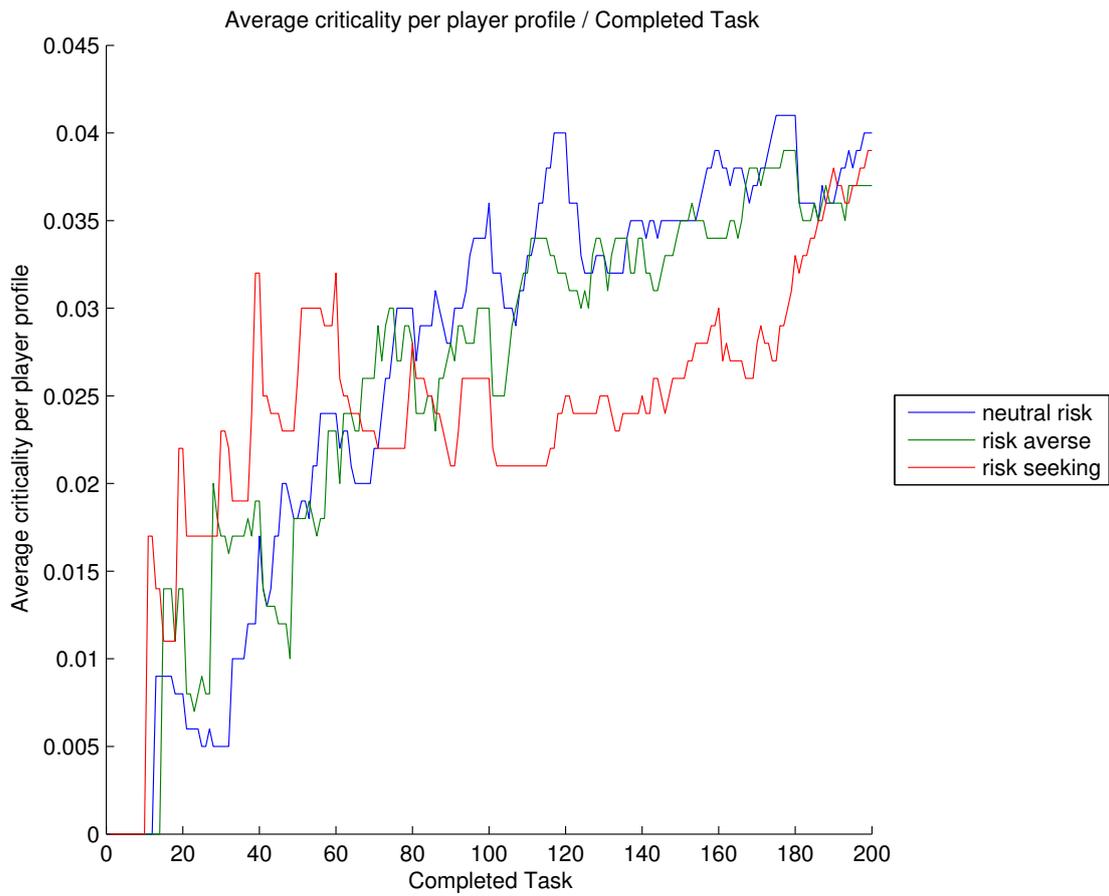


Figure 5.16: The metric of criticality for game version 2.

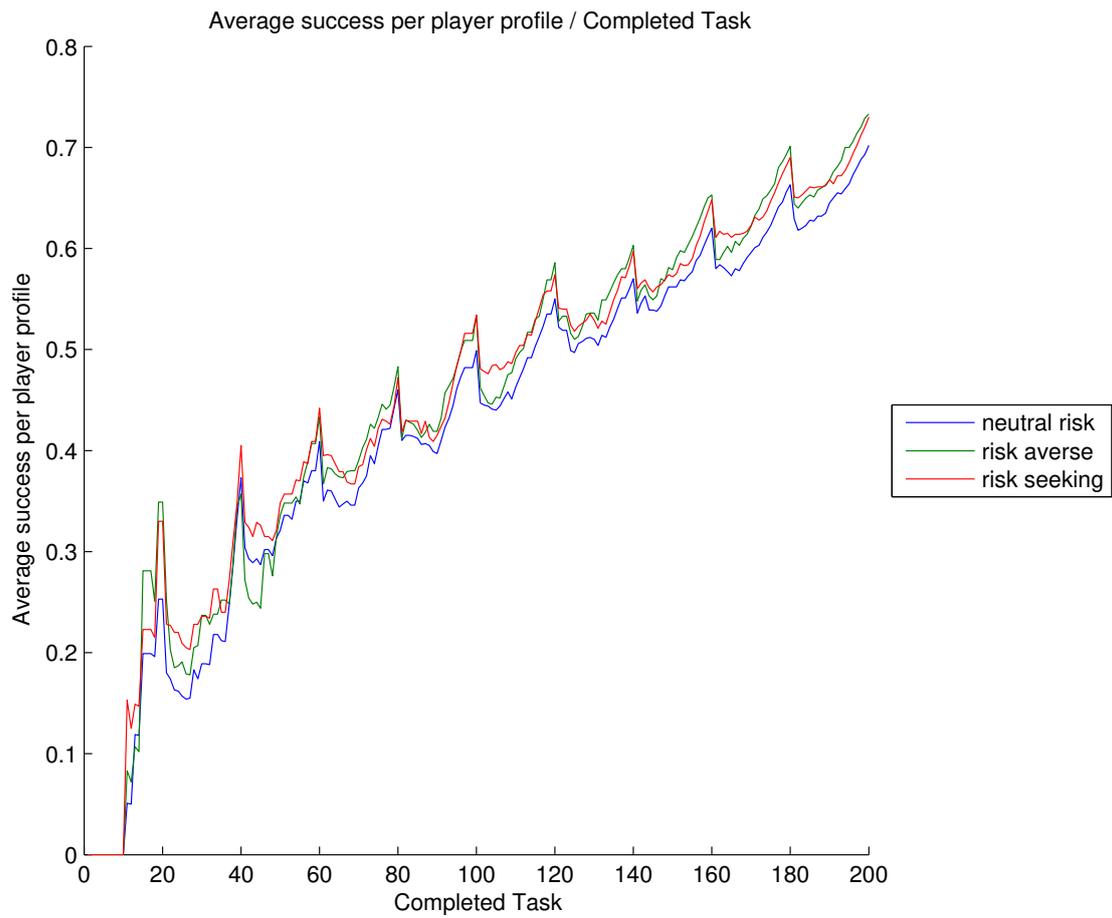


Figure 5.17: The metric of success rate for game version 2.

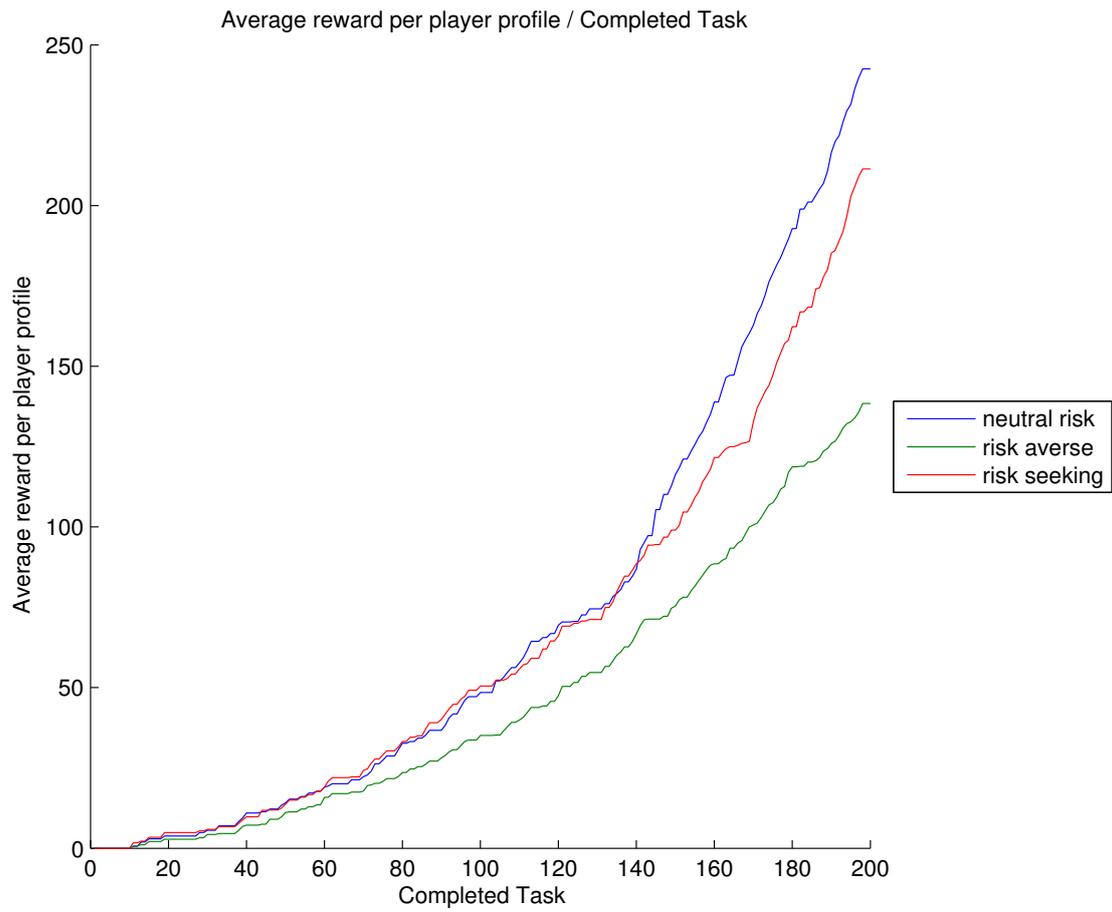


Figure 5.18: The metric of total reward for game version 2.

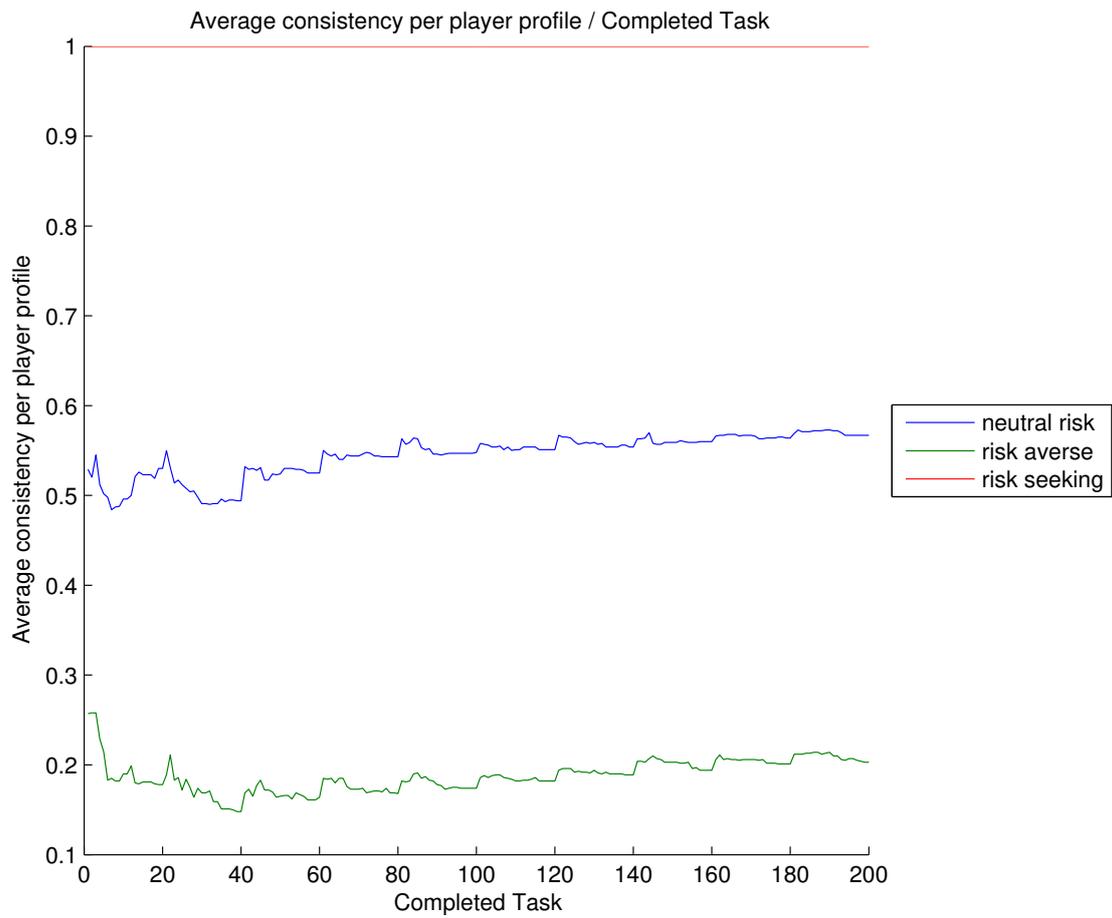


Figure 5.19: The metric of consistency for game version 3 (Note that the curve for risk seeking players is a straight line of value 1).

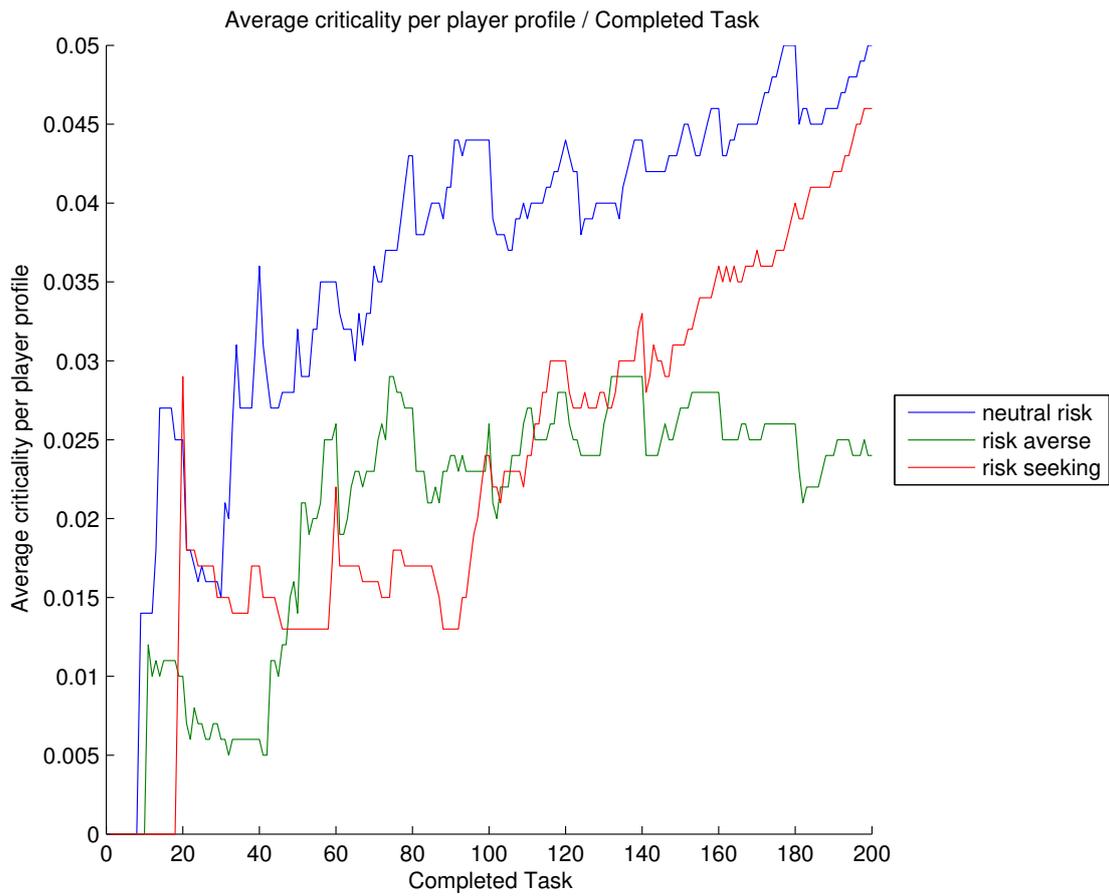


Figure 5.20: The metric of criticality for game version 3.

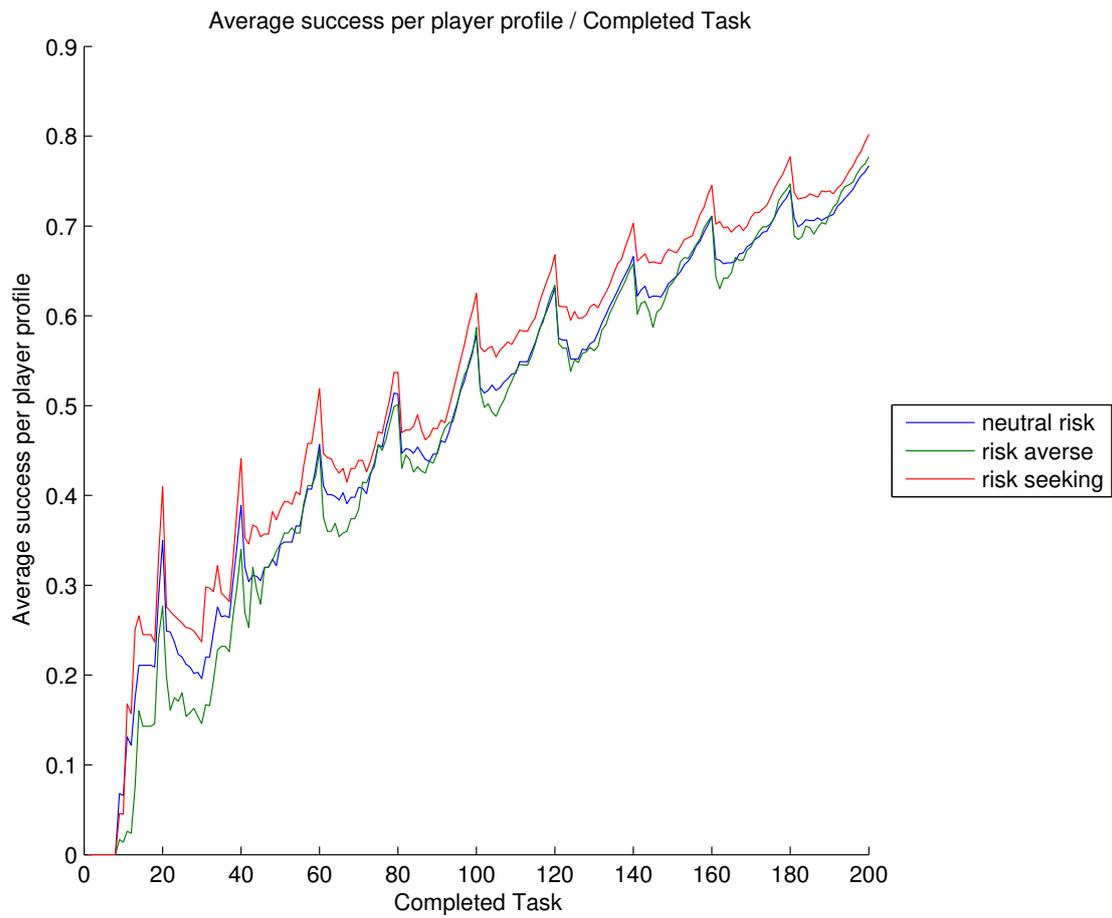


Figure 5.21: The metric of success rate for game version 3.

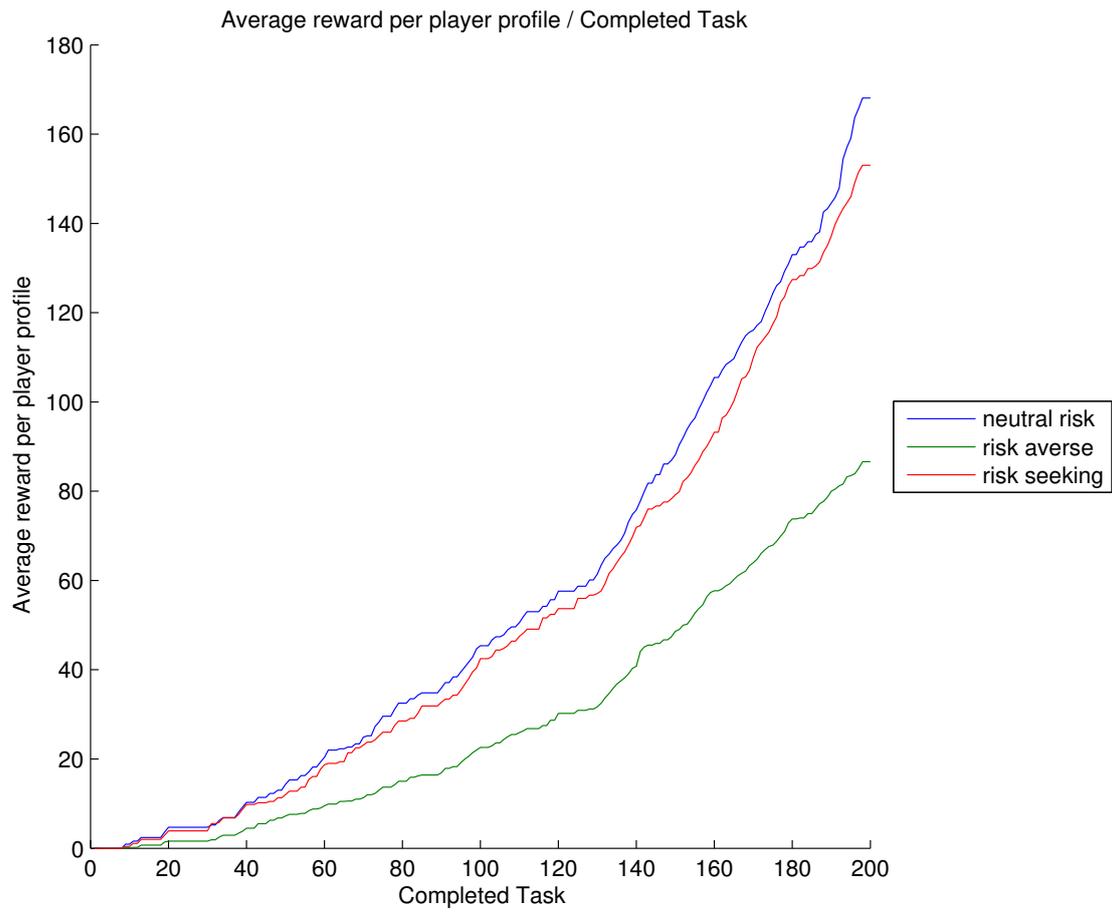


Figure 5.22: The metric of total reward for game version 3.

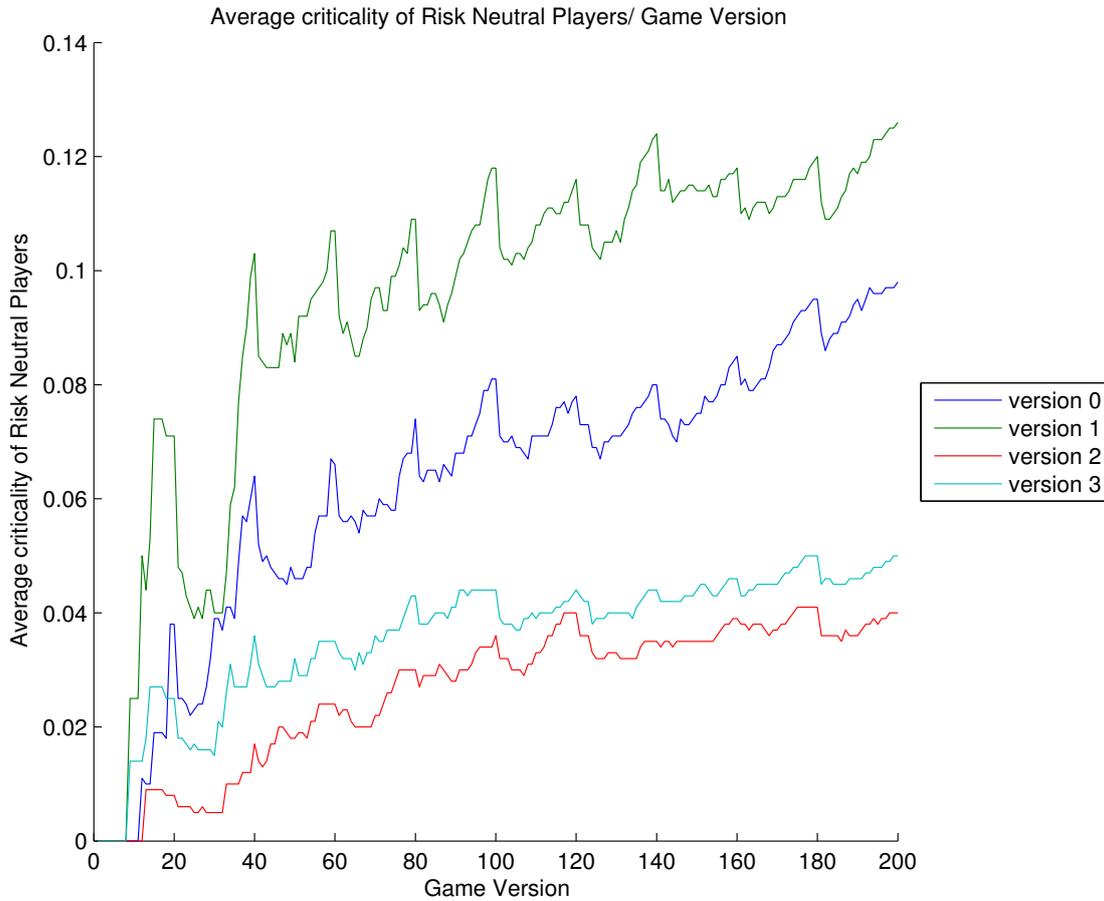


Figure 5.23: Average criticality for risk neutral players / game version

imminent pay-off during coalition formation, version 1 has consistently better curves for all types of risk profiles, that is they have higher success rate, more successful task and as a result the average total reward the players accumulate is of higher values. The above result is in line with the fact that our model is partially a hedonic game, and thus players do not act in order to maximize a numeric pay-off but in order to satisfy their hedonic preferences. All the players began the game with the default empathy beliefs over the other players, and therefore the same hedonic preferences over the coalitions. That is, at the beginning of the game each player is indifferent ( $e_i^j = 0.5$ ) towards any other player. However, we remind the reader that a task is successful with probability *coalition's empathy*; thus in this series of experiments the first successful tasks came earlier, and therefore the players started to be more friendly towards others earlier, achieving higher 'empathy', and more successful tasks overall. Different series of experiments and different probability initializations might lead to different results.



Figure 5.24: Average criticality for risk averse players / game version

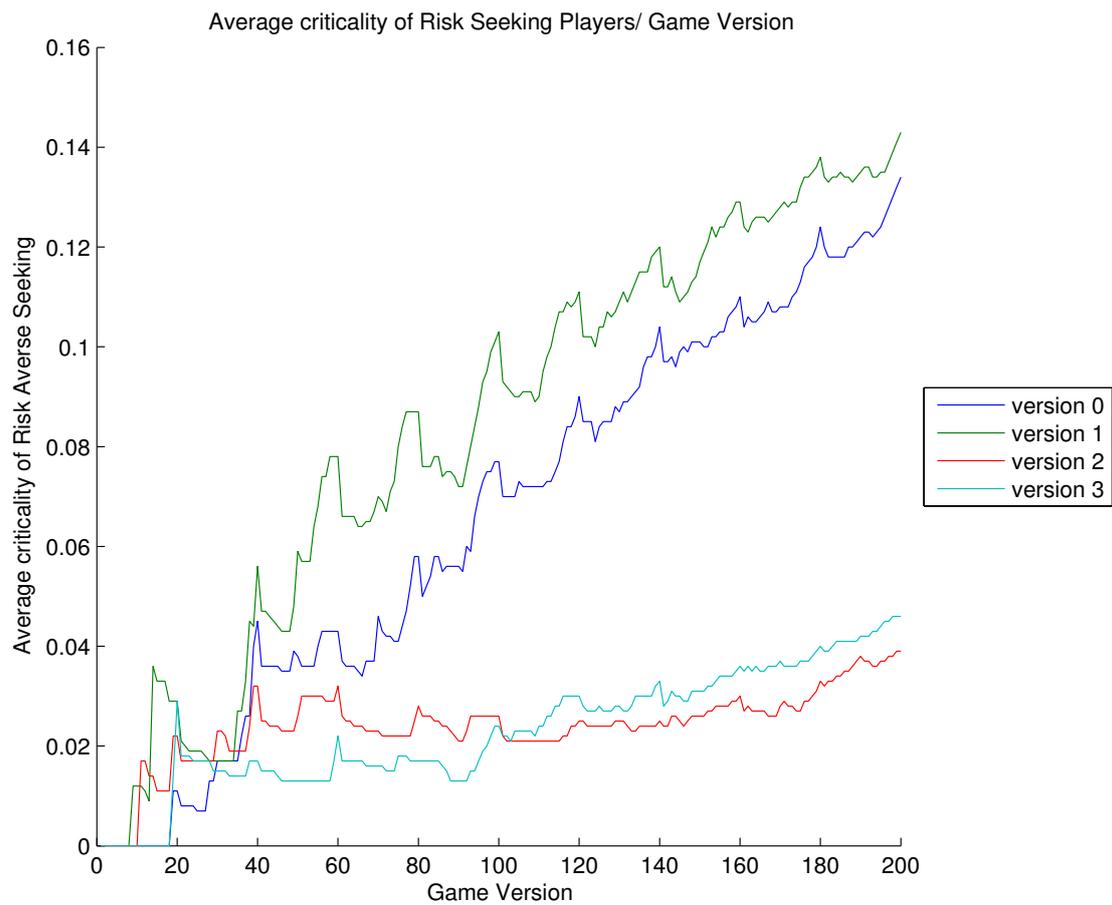


Figure 5.25: Average criticality for risk seeking players / game version

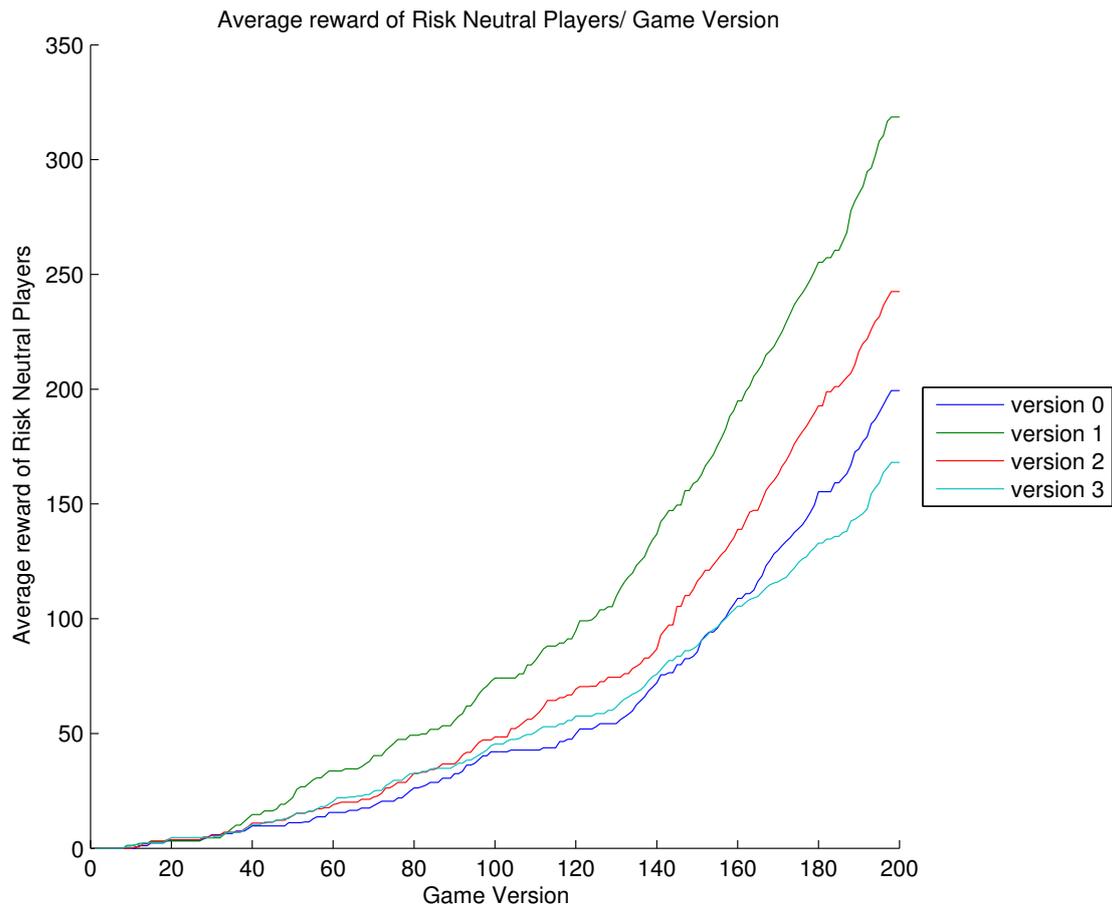


Figure 5.26: Average total reward for risk neutral players / game version

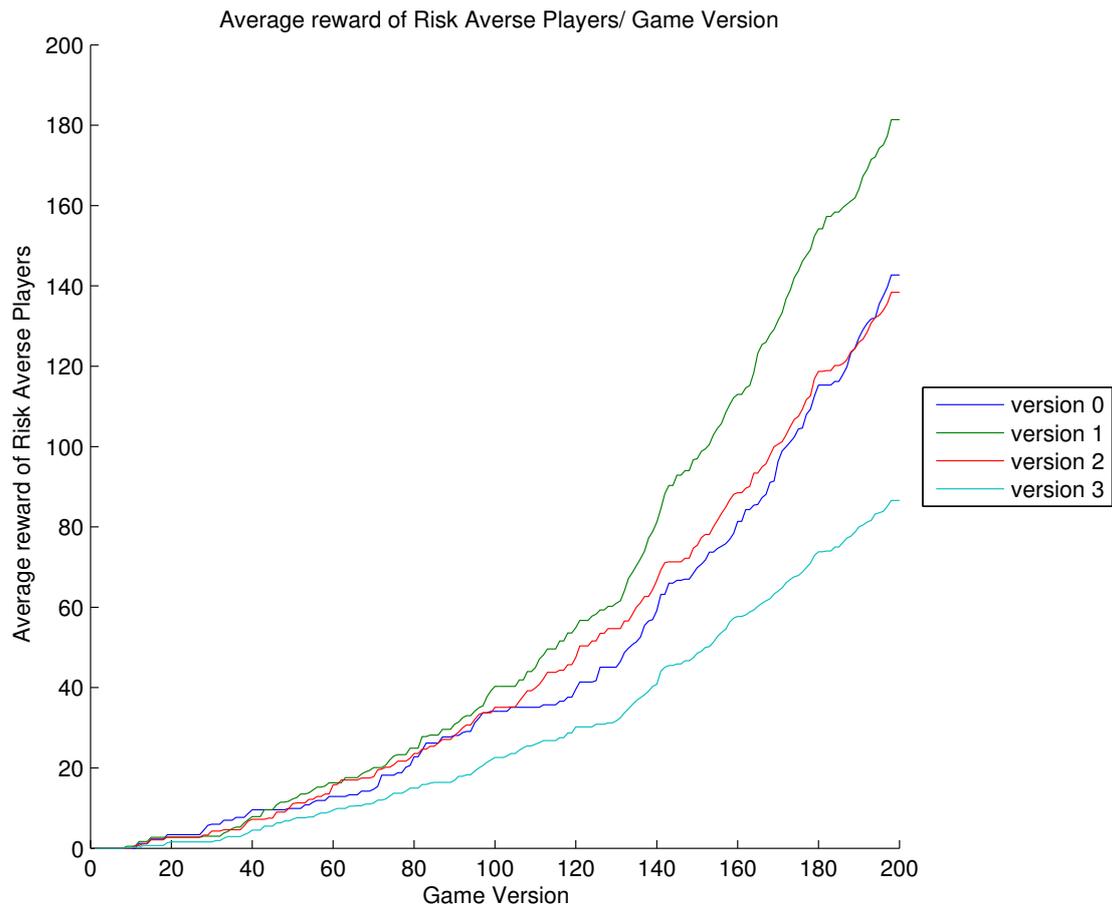


Figure 5.27: Average total reward for risk averse players / game version

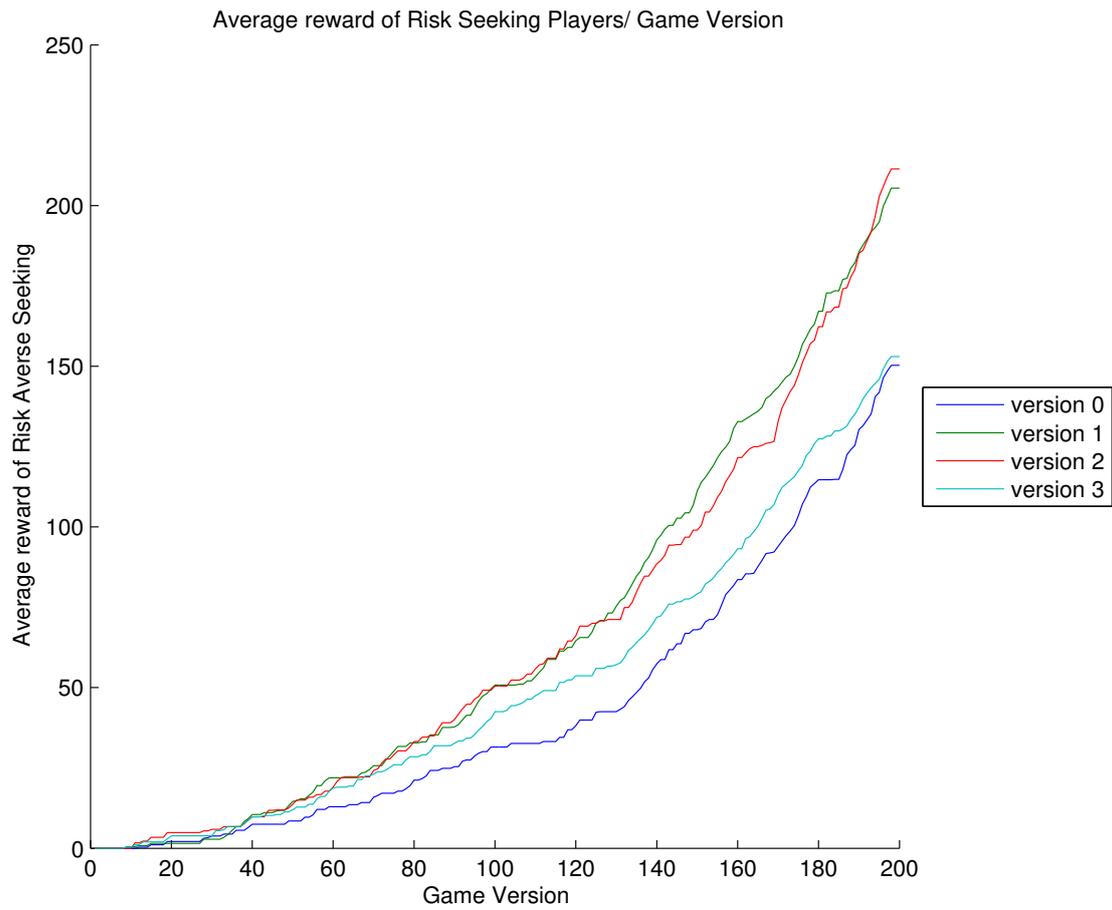


Figure 5.28: Average total reward for risk seekings players / game version

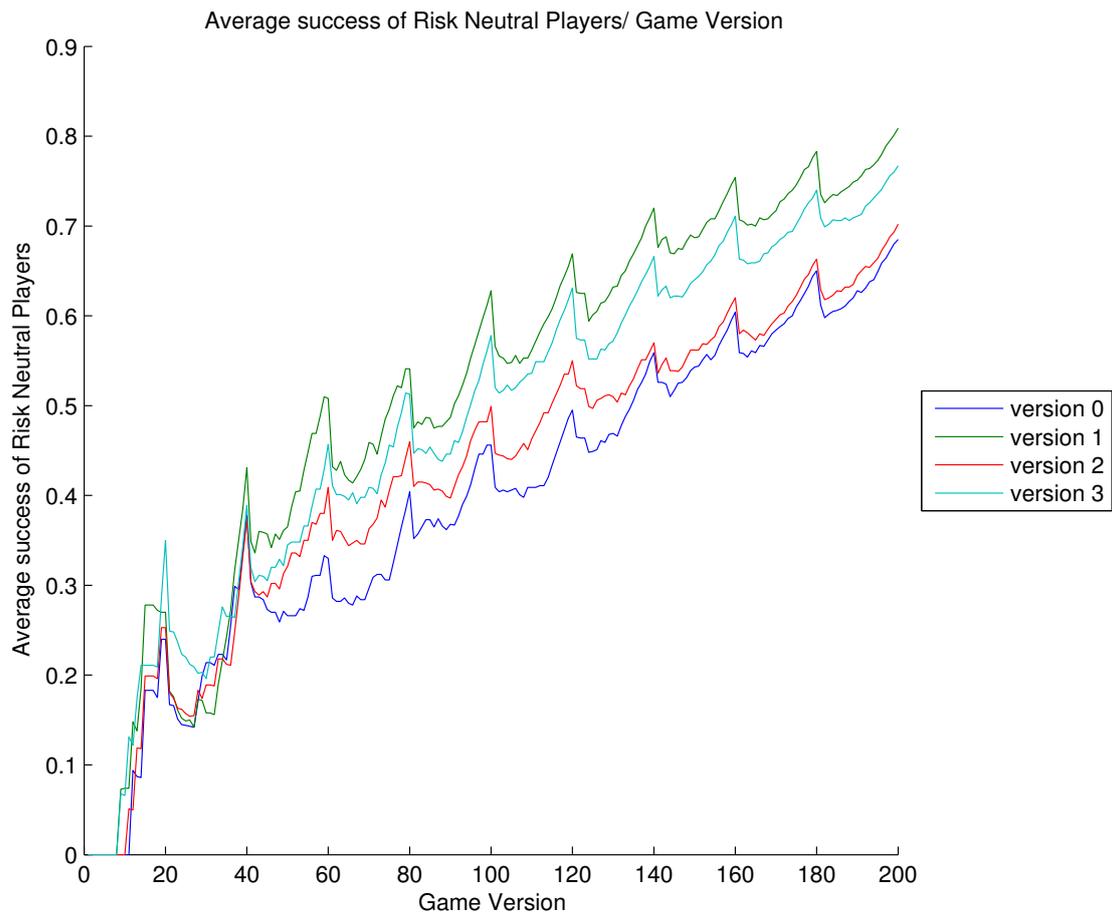


Figure 5.29: Average success rate risk neutral players / game version

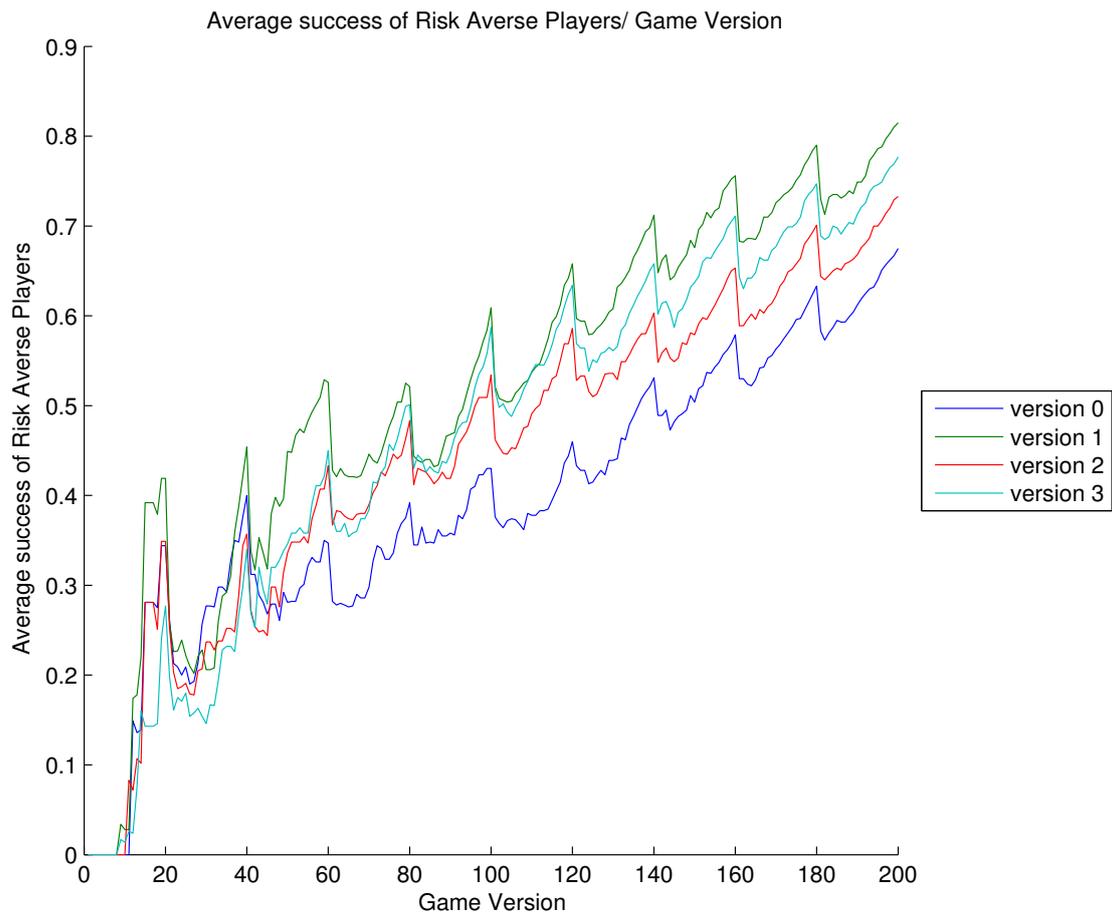


Figure 5.30: Average success rate risk averse players / game version

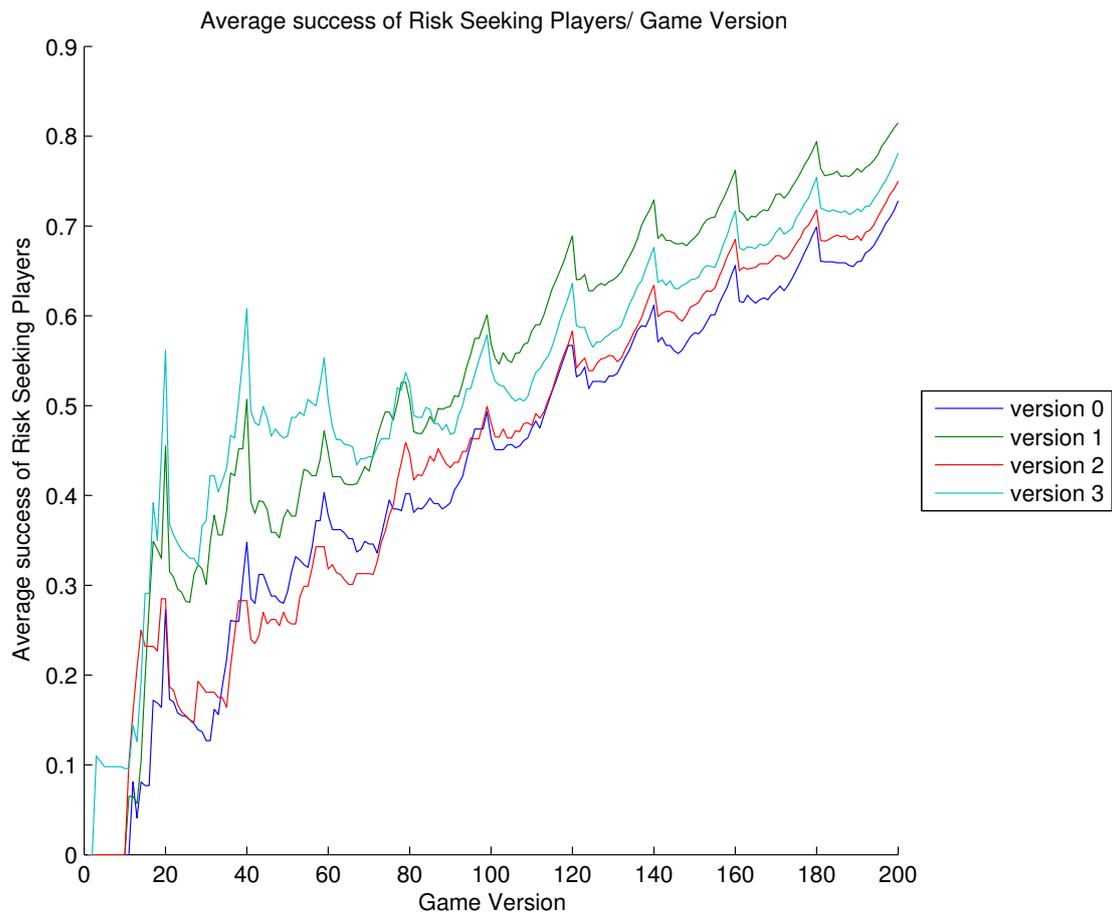


Figure 5.31: Average success rate risk seeking players / game version

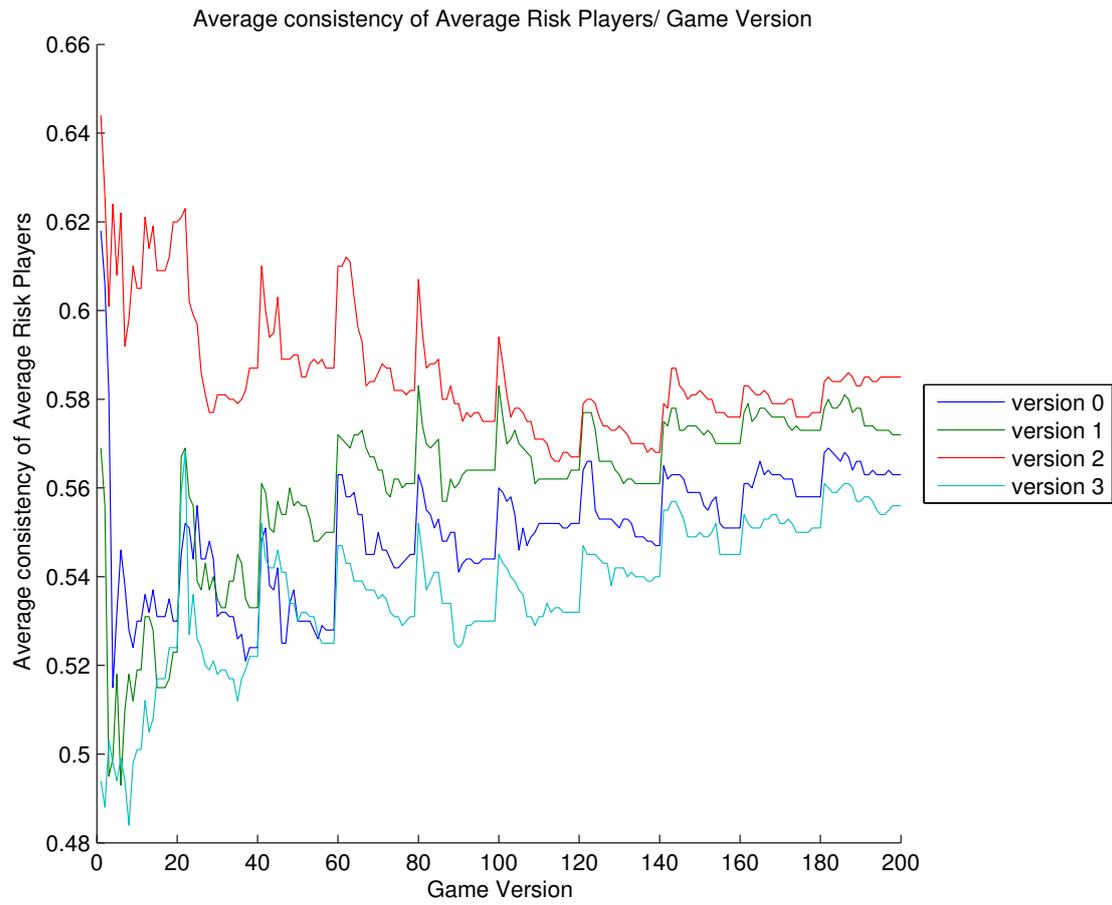


Figure 5.32: Average consistency neutral risk players / game version

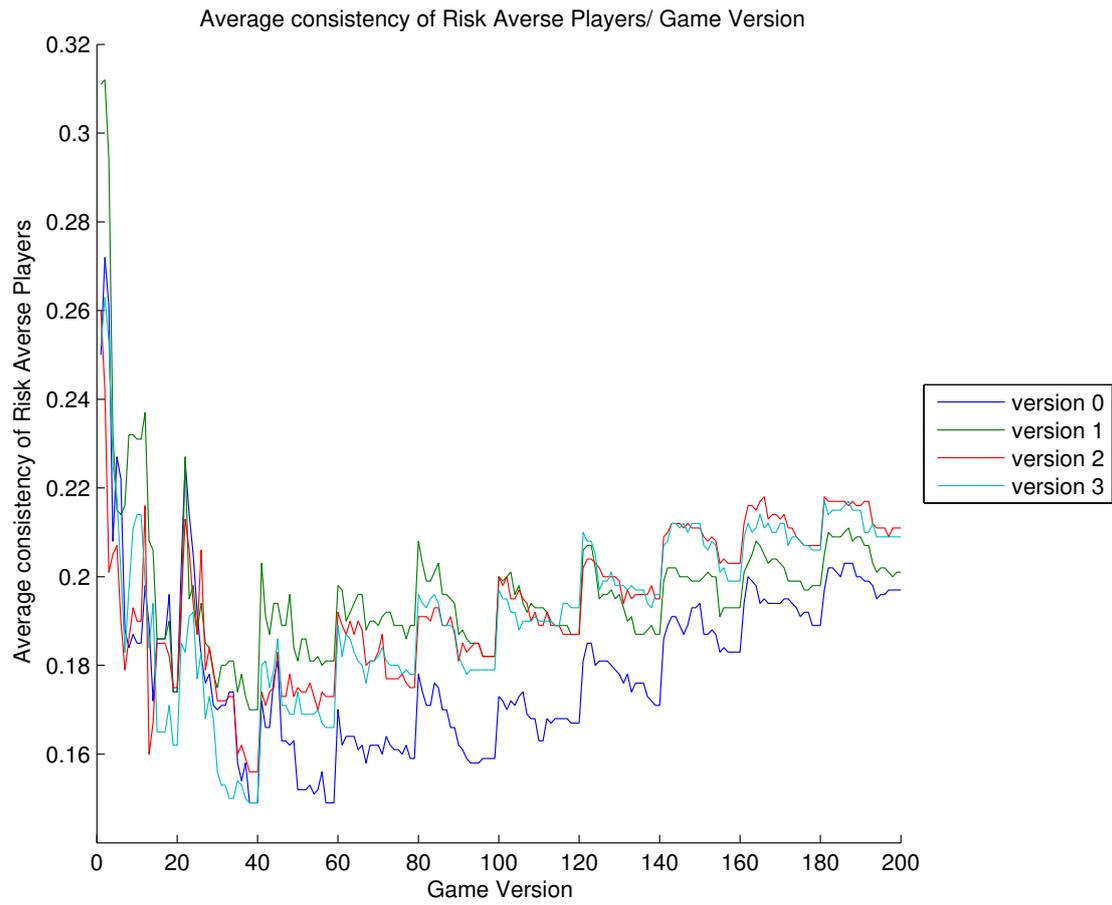


Figure 5.33: Average consistency risk averse players / game version

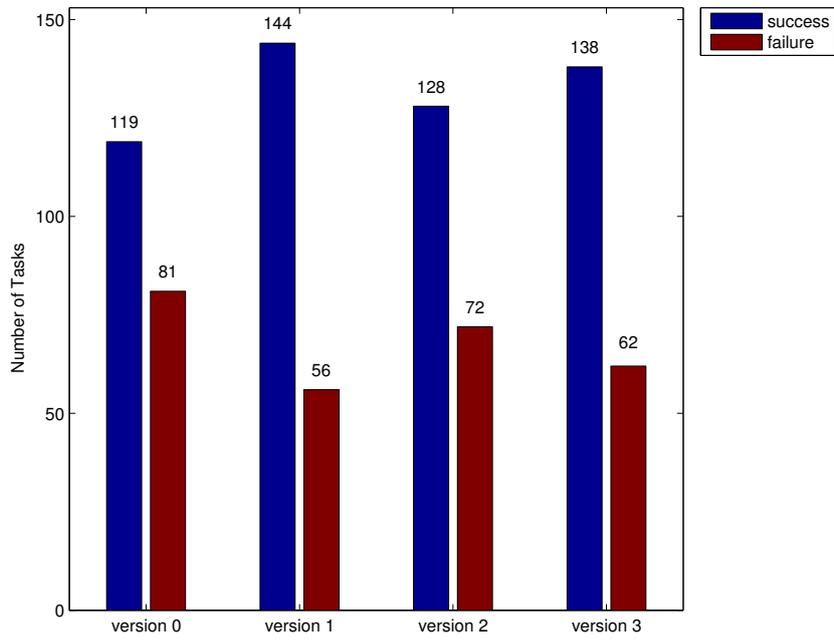


Figure 5.34: Number of successful and failed tasks per version

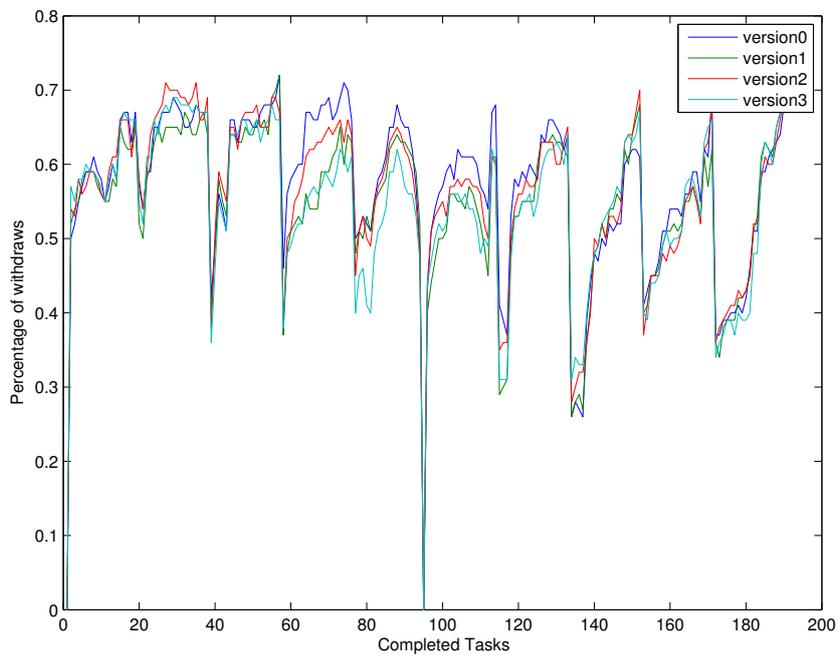


Figure 5.35: Percentage of withdrawals per version as tasks are completed

## 5.5 Conclusion

To conclude, the above results present some very interesting properties. First of all, the behavior of some agents influence the behavior of the others. That is, risk seeking players since they take part into the tasks more easily, start to adjust their empathy beliefs over their co-players earlier, and either lure or deter risk averse and risk neutral players to participate in strongly cohesive or non cohesive, respectively, coalitions.

The other interesting property is that, as we expected, the agents' imminent pay-offs and the coalition's utility does not affect the formation of coalitions. The players make their decisions depending only on their empathy beliefs regarding the coalitions, and not on their potential "financial" gain.

## Chapter 6

# Interface

We designed a simple web interface for our game. We chose to work with the programming language Python and to use Django, one of the most widespread python web frameworks. In the following two subsections, we will present the tools we used : the Django framework, with which we designed entirely the out web interface, and the Celery task queue, which we used to run some periodical function at the background (the user does not ‘wait’ until the function is completed). Then we proceed to show some views of our web interface.

### 6.1 Django Framework

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. With Django, developers can take web applications from concept to launch in a matter of hours. Django was designed to make common web-development tasks fast and easy. Django takes care of much of the hassle of web development, so developers can focus on writing their web application without needing to reinvent the wheel. It’s free and open source [16].

Django provides automated creation of projects and applications. It creates and organizes all the necessary files using the default settings, and the developer can begin to design the application without having to worry about low-level settings. Django follows the *model-view-template (MVT)* architecture pattern:<sup>1</sup> [25]

- **Model** is the data access layer. This layer contains anything and everything about the data: how to access it, how to validate it, which behaviors it has, and the relationships between the data.
- **Template** is the presentation layer. This layer contains presentation-related decisions: how something should be displayed on a Web page or other type of document.

---

<sup>1</sup>MVT is a very close to MVC architecture pattern.

- **View** is the business logic layer. This layer contains the logic that accesses the model and defers to the appropriate template(s). You can think of it as the bridge between models and templates.

The above three layers are the ones mainly concerning a developer, since with these three layers alone one can construct a descent database driven web application.

Django emphasizes reusability and ‘pluggability’ of components, rapid development, and the principle of “don’t repeat yourself”. That is, there is a plethora of Django packages, which are existing reusable apps we could incorporate in our project. Django itself is also just a Python package. This means that we can take existing Python packages or Django apps, and compose them into our own web project. We only need to write the parts that make our project unique.

Another basic component of Django is the middleware it provides. Middleware is a framework of hooks into Django’s request/response processing. It’s a light, low-level “plugin” system for globally altering Django’s input or output. Each middleware component is responsible for doing some specific function. For example, Django provides middleware for page caching (*UpdateCacheMiddleware*), session support (*SessionMiddleware*), user authentication (*AuthenticationMiddleware*), etc.

Django is suitable for database drive web applications. By default, Django configuration uses SQLite which is included in Python. However, one can use her preferred database management system by installing/including it in the project settings. In our web interface, we used the default database.

## 6.2 Distributed Task Queue Celery

Celery is a simple, flexible, and reliable distributed system to process vast amounts of messages, while providing operations with the tools required to maintain such a system. It is focused on real-time operation, but supports scheduling as well. The execution units, called tasks, are executed concurrently on a single or more worker servers using multiprocessing. Tasks can execute asynchronously (in the background) or synchronously (wait until ready). Celery is open source and licensed under the BSD License [8].

In our web interface, we used the Celery distributed task queue in order to periodically check the database and evaluate the tasks as success or failure when their time window elapses. We also use it to bring new tasks to the game in regular intervals.

## 6.3 Basic Views

We now present some screen-shots from our web interface. In Figure 6.1 we see the registration form the player has to fill the very first time he/she enters into the game.

Once a player is registered and logged in, he/she is redirected to home page, showed in Figure 6.2. In the home page, the player is informed of his/her available

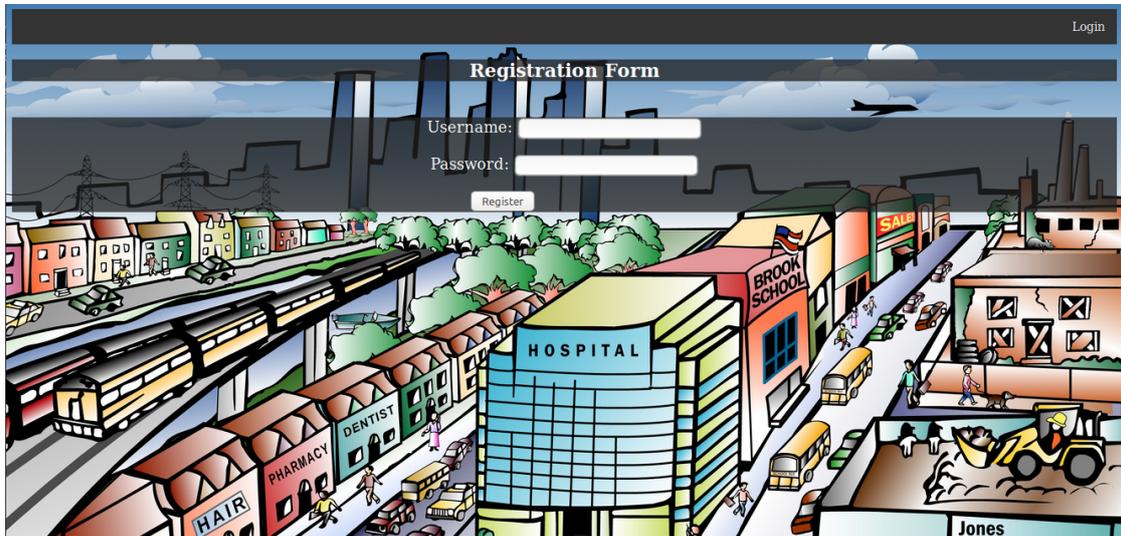


Figure 6.1: Registration Form

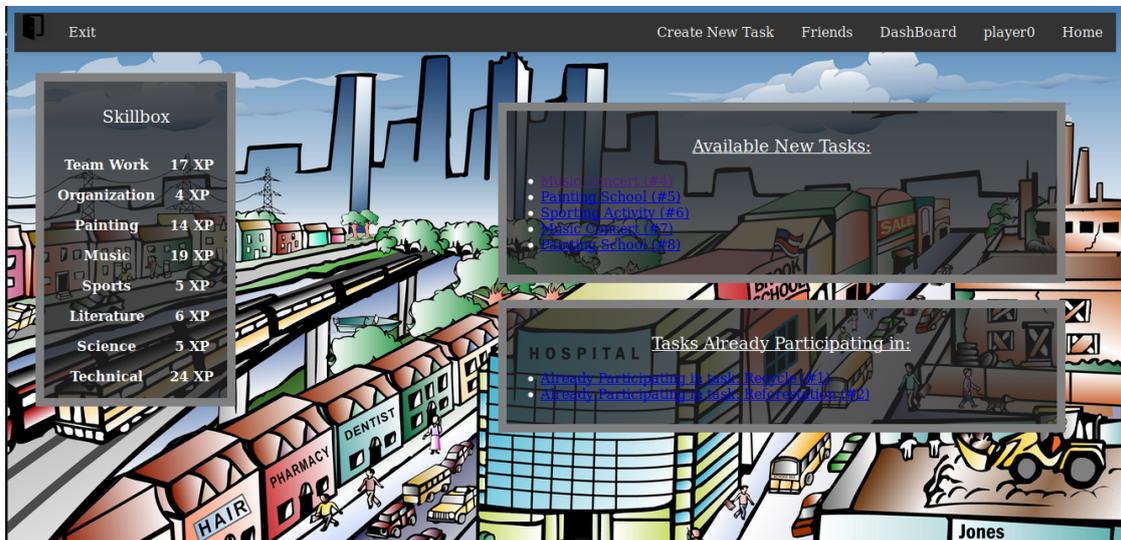


Figure 6.2: Home Page

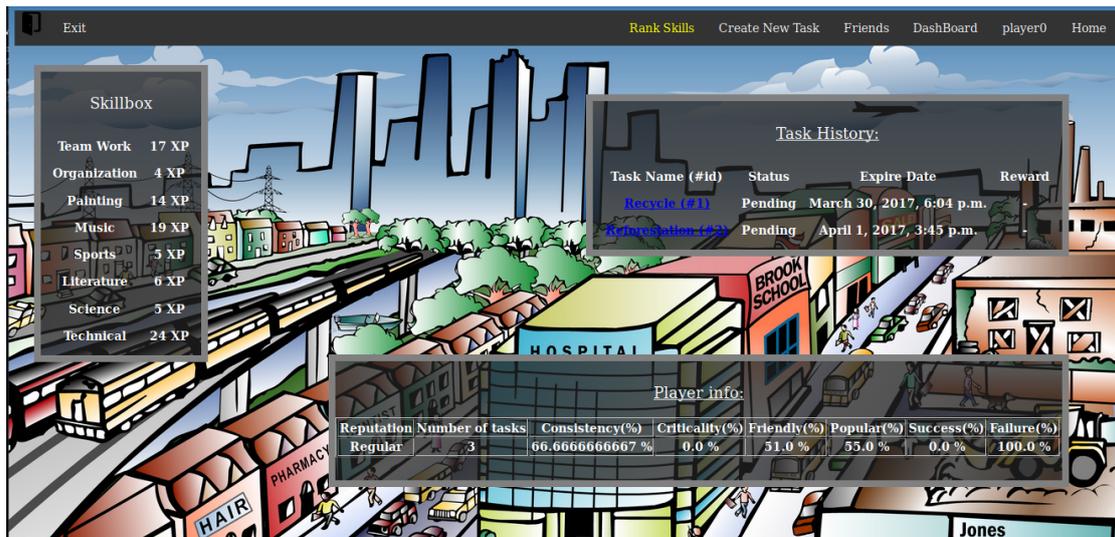


Figure 6.3: Profile Page

skill experience levels (4.1.5), the active tasks that he/she can join, and the pending tasks he/she is already participating in. Through the menu bar at the top of the page, can navigate to the rest of the pages. In Figure 6.3 is illustrated the profile page, where the player can see his/her statistics (4.1.10). The player also sees the ten most recent tasks he/she participated in, that is the tasks name along with its primary key, its status ('Pending', 'Success', or 'Failure'), the expiration date (time window 4.1.3), and the total reward gained if the task is successful. Looking at the menu bar at the profile page we notice a new option, that of *Rank SKills*. Through that option in the menu, the player can change his/her preferences over his/her skills (4.1.8). In Figure 6.4 we see the Rank Skill page.

In Figure 6.5, the player can see all other players with their statistics and their reputation. Either through the menu bar and the option *Friends*, or through the Dashboard page, the player can access the other players, and then change his/her empathy beliefs towards them. The page in Figure 6.6, presents the player's 'friends', that is, the other players he/she has already liked or disliked. By default all players have 50% empathy towards the other players; if a player has not changed any of his default empathy beliefs, then no players will be shown in his/her page. Figure 6.7 illustrates the page where the player can change his/her empathy beliefs towards a specific other player.

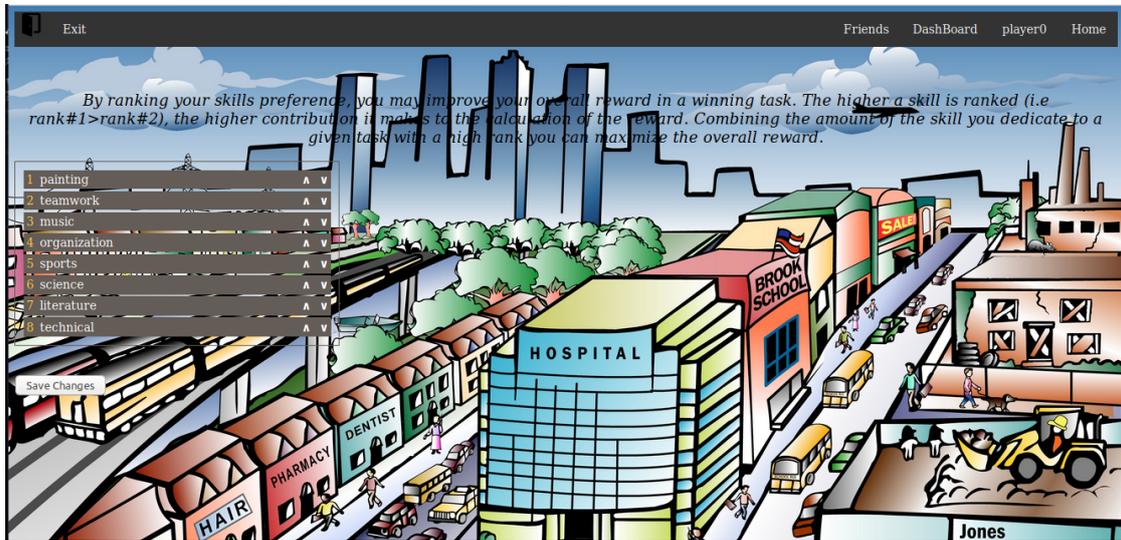


Figure 6.4: Rank Skill Page

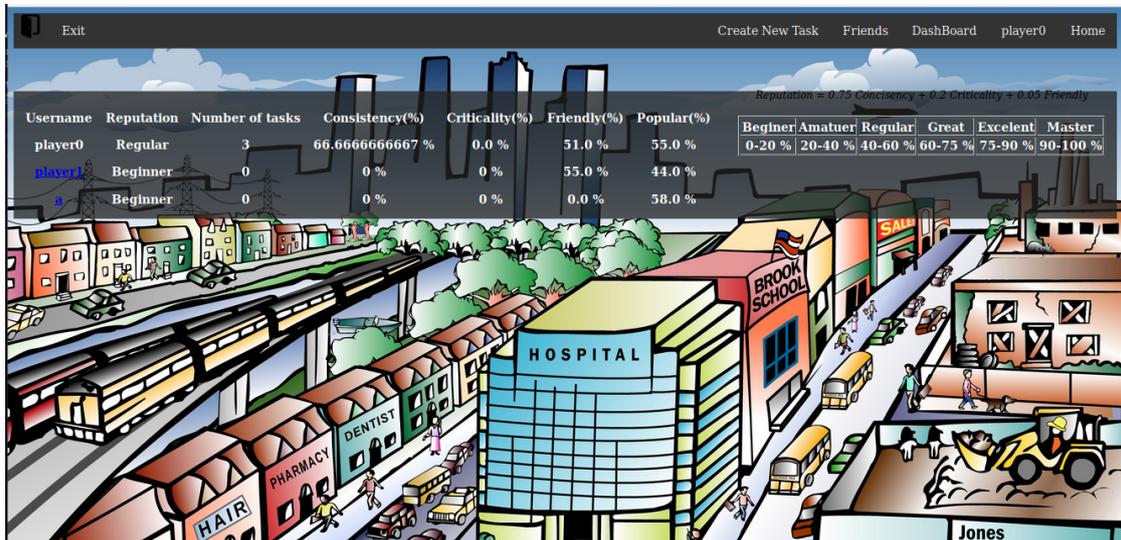


Figure 6.5: Dashboard Page

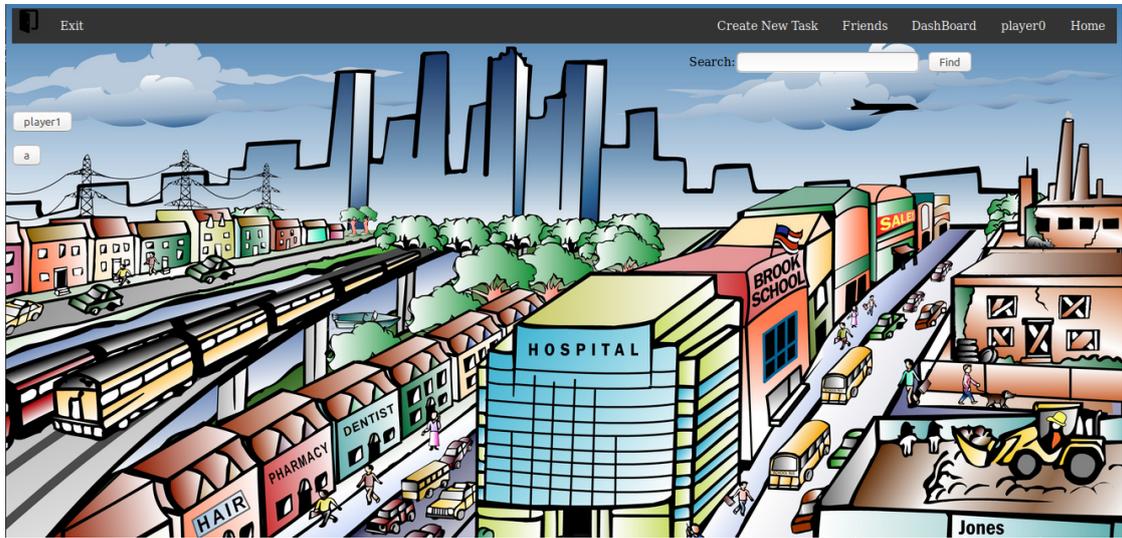


Figure 6.6: Friends Page

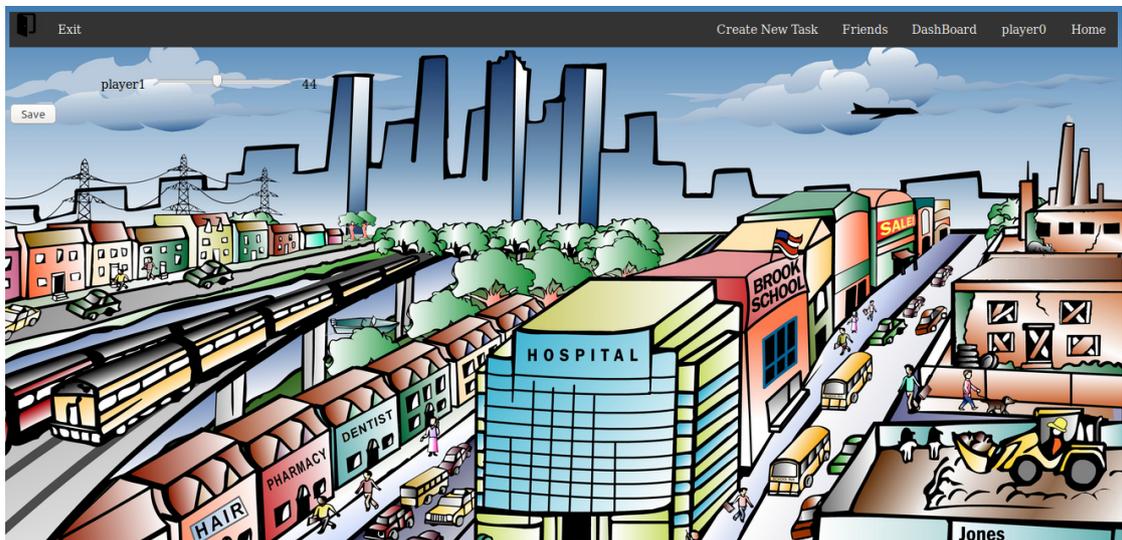


Figure 6.7: Update Friend Page



Figure 6.8: Organize New Task Page

A player has the option to organize his/her own task. In Figure 6.8 we see the page where the player can create a new task; that is, choose the task he/she wants, and set the experience level for each required skill (the required skills are predetermined, see Table 4.1). We remind the reader that we set an upper bound for the numeric requirements, see section 4.1.2.

In Figure 6.9 is shown the page in where the player can join a new active task. The player sets the amount of experience levels of each skills, he/she wants to contribute to that task. Moreover, there is further information about the task, such as its skills requirements, the current empathy factor within the coalition, and the empathy factor if the player joins the task. Figure 6.10 shows the page of a pending task. In this page, the player can see informations about the task: the expiration date, the skill requirements, the participants of the task, and the coalition's empathy. Furthermore, the player in this page has the option to withdraw from the task.

Exit Create New Task Friends DashBoard player0 Home

## Painting School

Task expires on : April 2, 2017, 8:26 p.m.

Skillbox	
Team Work	17 XP
Organization	4 XP
Painting	14 XP
Music	19 XP
Sports	5 XP
Literature	6 XP
Science	5 XP
Technical	24 XP

Requirements	
Skill Name	Amount Required
Team Work	7
Organization	5
Painting	11

Participants	
Participant	Reputation

Liking within coalition

Current	If join
0.0 %	0.0 %

Team work:  acceptable values in [0,17]  
 Organization:  acceptable values in [0,4]  
 Painting:  acceptable values in [0,14]

Figure 6.9: Task to Join Page

Exit Create New Task Friends DashBoard player0 Home

## Reforestation

Task expires on : April 1, 2017, 3:45 p.m.

Skillbox	
Team Work	17 XP
Organization	4 XP
Painting	14 XP
Music	19 XP
Sports	5 XP
Literature	6 XP
Science	5 XP
Technical	24 XP

Requirements	
Skill Name	Amount Required
Team Work	45
Organization	25
Science	39

Participants	
Participant	Reputation
player1	Beginner
player0	Regular

Liking within coalition : 49.5 %

Figure 6.10: Active Task Page

## Chapter 7

# Conclusions

### 7.1 Summary

In this thesis we studied an important part of game theory, that of cooperative games, and particularly the  $k$ -weighted voting games (a transferable utility class of games) and the hedonic games (a non-transferable utility class of games). We presented a novel hybrid class of games which combines the above two disjoint classes of cooperative games, the class of **hedonic  $k$ -weighted voting games**. For the first time in literature, we proposed a model that forms coalitions according to hedonic games, while it implements a  $k$ -weighted voting game. In more details, each agent has hedonic preferences over coalitions, same as in hedonic games, along with a voting weight for each component game, same as in  $k$ -weighted voting games. Thus in our proposed model, agents form coalitions that satisfy their hedonic preferences, without considering the utility of the coalitions or aiming to maximize their imminent pay-off.

Our intention was to use our proposed model to design *a serious game that will promote civic actions at local level*. We designed the serious game application described in Chapter 4. Through our game, we aim to introduce and promote notions such as collaboration, volunteerism, public good, and civic actions among young people. We endeavor to provide an entertaining way to encourage teenagers to approach concepts such as collective action, social conscience, and environmental consciousness. Furthermore, we implemented a fully functional front/end user interface, presented in Chapter 6. We also ran a series of experimental simulations to see the game in progress. The simulations allowed us to study four different versions of our game, and different risk profiles. An interesting observation resulting from the experiments is that certain group of agents can induce or deter other players to join some coalition or not.

## 7.2 Future Work

Our work provides ground for extensions, since it is novel class of cooperative games. Now we will present some future work we consider.

### 7.2.1 Enhance Definition

We intend to **enhance the formal definition** of our proposed model, the hedonic k-weighted voting game (Chapter 3). During our game application design, we introduced the empathy factor within a coalition as a measure of the coalition's social cohesion. We used the empathy factor as a probability of a coalition be successful in a given hedonic k-weighted voting game. As future work, including the empathy factor in the formal definition raises theoretical interest.

### 7.2.2 Repeated Hedonic k-Weighted Voting Game

A second intriguing topic for further research is to study **the progress of hedonic k-weighted voting games in the repeated form**, and specifically how players tend to update their empathy beliefs is interesting future work. As we mentioned, the way we used to update our agents' beliefs during the simulations is very naive. Thus, studying the players' tendency toward their preferences updating, and specifying the conditions under which players update their beliefs is appealing.

### 7.2.3 Stability on Hedonic k-Weighted Voting Game

As we already mentioned in section 3.1.1, in our work we are not concerned with concepts of stability beyond those of pure hedonic games. However, Hedonic k-Weighted Voting Games provide rich framework for study **stability**, since they combine preference relations with pay-offs actually related with utility. Thus, studying the stability by focusing on the relationship that links the preference relations and the utility is appealing future work.

### 7.2.4 Interface Integration

Finally, an extension we have in mind is that of **completely integrating the system's back-end to our front-end**. Even though the interface we developed is functional and can be used for testing, there is a need of integrating the system to it in order to be used as a pilot game.

# Bibliography

- [1] Asiz H .and Savani R. (2016). “Hedonic Games”. In: *Handbook of Computational Social Choice*, pp. 356-376.
- [2] Asiz H. and Brandl F. (2012). “Existance of Stability in Hedonic Coalition Formation Games”. In: *Proc of the 11th International Conference on Autonomous Agents and Multiagent Systems*.
- [3] Bachrach Y. and Rosenschein J. (2008). “Coalitional Skill Games”. In: *Proc of 7th International Conference on Autonomous Agents and Multiagent Systems*, pp. 1023-1030.
- [4] Bachrach Y., Markakis E., Procaccia A., Rosenschein J. and Saberi A. (2008). “Approximating Power Indices”. In *Proc of 7th International Conference on Autonomous Agents and Multiagent Systems*.
- [5] Bers M. U. (2010). “Let the Games Begin: Civic Playing on High-Tech Consoles”. In: *Review of General Psychology*, 14(2), 147–153. doi:10.1037/a0019490.
- [6] Bogomolnaia A. and Jackson M. (1998). “The Stability of Hedonic Coalition Structure”.
- [7] Burke J. W., McNeill M., Charles D., Morrow P., Crosbie J. and McDonough S. (2009). ”Serious games for upper limb rehabilitation following stroke”. In *Games and Virtual Worlds for Serious Applications, Conference in*, vol. 0, pp. 103–110.
- [8] Celery: Distributed Task Queue, <http://www.celeryproject.org>.
- [9] CivicAction : <http://www.civicaction.ca/>.
- [10] Chalkiadakis G. (1999). “An Agent-Based Architecture for the Conduction of Voting”. Msc Thesis, Department of Computer Science, University of Crete.
- [11] Chalkiadakis G. and Boutilier C. (2004). “Bayesian reinforcement learning for coalition formation under uncertainty”. In: *Proc. of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 1090–1097.
- [12] Chalkiadakis G., Elkind E. and Wooldridge M. (2012). “Computational Aspects of Cooperative Game Theory”, published by Morgan & Claypool.

- [13] Corti K. (2006). “Gamesbased Learning; a serious business application”. In: *PIXELearning Limited*.
- [14] Di Loreto I., Mora S. and Divitini M. (2009). “Collaborative serious games for crisis management: an overview”. In *IEEE 21st International WETICE*.
- [15] Dital Civics: <https://digitalcivics.io/>.
- [16] Django, The web framework for perfectionists with deadlines: Documentation, <https://docs.djangoproject.com/en/1.11>.
- [17] Elkind E., Chalkiadakis G. and Jennings N. (2008). “Coalition Structures in Weighted Voting Games”. In: *Proc. of the 2nd International Workshop on Computational Social Choice*.
- [18] Ferreira, N. (2002). “Serious Games”.
- [19] Graafland M., Schraagen J.M. and Scijven M.P (2012). “Systematic review of serious games for medical education and surgical skill training”. In : *Wiley Online Library (www.bjs.co.uk)*. DOI:10.1002/bjs.8819
- [20] Haferkamp N., Kraemer N. C., Linehan C. and Schembri M. (2011). “Training disaster communication by means of serious games in virtual environments”. In : *International Federation of Information*.
- [21] Jansiewicz D.R. (1973). “The New Alexandria Simulation: A Serious Game of State and Local Politics”.
- [22] McAlpine, K. B. (2017). “Shake and create: reappropriating video game technologies for the enactive learning of music”. In: *Ma M., Oikonomou A. (Eds.), Serious games and edutainment applications (Vol. 2, pp. 77-97)*. Cham, Switzerland: Springer International Publishing.
- [23] Koivula M. and Huttunen K. and Mustola M. and Lipponen S. and Laakso ML. (2017). “The Emotion Detectives Game: Supporting the Social-emotional Competence of Young Children”. In: *Ma M., Oikonomou A. (Eds) Serious Games and Edutainment Applications*. Springer, Cham
- [24] Mamakos M. (2017). “Overlapping Coalition Formation under Uncertainty”. Msc Thesis, School of EECE, Technical University of Crete.
- [25] Mastering Django : Core (The Django Book), <https://djangobook.com/the-django-book>.
- [26] Moro G. (2010). “Civic Action”. In *International Encyclopedia of Civil Society*, pp. 145-150.
- [27] Osborne J. M. (2004). “An introduction to game theory”, published by Oxford University Press India.

- [28] PlayGen, the UK's leading serious games and gamification development studio : <http://playgen.com/>.
- [29] Shoham Y. and Leyton-Brown K. (2009). "Multiagent Systems : Algorithmic, Game-Theoretic, and Logical Foundations".
- [30] Susi T., Johannesson M. and Backlund P. (2007). "Serious games: An overview". Sweden: University of Skövde. Technical Report HS- IKI -TR-07-001.
- [31] Van Eck R. (2006). "Digital Game-Based Learning: It's Not Just the Digital Natives Who Are Restless". In: *EDUCAUSE Review*, vol. 41, no. 2 (March/April 2006), pp. 16-30.
- [32] Vlackokyriakos V., Gordon E., Crivellaro C., Wright P., Le Dantex C. A. and Olivier P. (2016). "Digital Civics: Citizen Empowerment With and Through Technology". In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '16)*.
- [33] Vlackokyriakos V., Comber R., Ladha K., Taylor N., Dunphy P., McCorry P. and Olivier P. (2014). "PosterVote: Expanding the Action Repertoire for Local Political Activism". In *Conference Interactive Systems*.
- [34] Wooldridge M. (2002). "An introduction to multiagent systems", published by John Wiley & Sons.
- [35] Yang Ya-Ting C. (2012). "Building virtual cities, inspiring intelligent citizens: Digital games for developing students' problem solving and learning motivation". In *Computers & Education*, vol. 59 no. 2, pp. 365-377.

# Appendices

# Appendix A

## Voting Theory

Voting Theory is the mathematical theory that deals with the transformation of the individual views of members of democratic societies or groups into a single view that pertains to a particular subject and expresses the whole.

Voting is the process of collecting and counting the votes of the electorate. Electorate is the group of people who have the right to vote. The electorate, aka the voters, submit their personal opinion/view on the subject under discussion. To do so, they use ballots, a document where all different and possibly conflicting proposals are enlisted.

The outcome of a voting process depends on the electoral system in question. There is a plethora of voting systems to be implemented on various electoral situations. Even though there is a consensus of views on the desired features in a voting system, there is strong controversy as to which is most meaningful. The features of a voting system enlist it into one of the following categories [10]:

- **Majoritarian Systems.**

Systems in this category, are said to satisfy the criterion of majority. Winner of the vote shall be the one who holds the *absolute* or *relative* majority of the votes. In this category, are also included systems that satisfy the *Condorcet* criterion (winner of the vote is the candidate who can defeat each of his/her opponents in personal confrontation).

- **Proportional Representation Systems.**

In this category, there are many widely used systems for the election of representatives, which are based on proportional representation of voters in absolute or relative proportions.

- **Preferential/Positional Systems.**

These systems use preference ballots to rank voter options. The information recorded in the preferences lists is exploited to a different extent by the different electoral systems of preferences (others take advantage of all of them, others use them gradually and others do not).

- **Candidates' Elimination Systems.**

These systems declare a winner after successive candidates' exclusions, based on successive voting rounds, or voter-preferred voter options in a single round.

- **Utilitarian Systems.**

In these systems the voter does not simply categorize the candidates in order of preference, but also identifies them with an (arbitrary or predetermined scale) numerical weight that determines the degree of appreciation to the candidate (hence the expected benefit from him).

- **Weighted Systems.**

In these systems, the opinion of specific voters is assessed with a different weight than the others.