

## **The Symbiosis of AI and Computer Games: Two Examples of Games as a Bidirectional Interactive System**

Robin Baungarten  
Eurídice Cabañes  
<http://euridicecaban.es.tl/>

Usually, when we talk about computer games and artificial intelligence, we are talk about how the artificial intelligence is used in order to simulate intelligence in characters that appear in computer games. This kind of artificial intelligence is quite limited. Our proposal in this paper is to explore other possibilities in this interesting field, in which we will show two different proposals. The first one is using artificial intelligence in order to create adaptive computer games. The second one is to use computer games to train artificial intelligence programs. We present an example for each concept; a survey analysing the behaviour of players of Pac-Man, and an association game exploring the synesthetic relationship of images, sounds and text.

### **KEYWORDS**

Artificial Intelligence, computer games, adaptive games, human computation, computational creativity.

### **1. INTRODUCTION**

Usually, when we talk about video games and artificial intelligence, we talk about how the AI is used in order to simulate intelligence in characters that appear in the video games. This kind of AI, which produces the illusion of intelligence in the behaviour of non-player characters (NPCs) is called Game Artificial Intelligence and is quite limited. This means that it doesn't explore all the possibilities that the AI provides – not just because it allow hacks and cheats – but because, in many cases, the computer abilities must be toned down to give human players a sense of fairness, too.

We are convinced that the relationship between computer games and AI can be more fruitful. But in order to get this goal we need to change the paradigm, because if we limited ourselves to enhance the AI in the current paradigm, we would just end up with computer games with a high difficulty level that no one can win.

What does changing the paradigm mean?

So far, the interactivity that the computer games allow has been understood in a unidirectional way, which means taking just the information that the computer will provide to the player in consideration, while forgetting that, in this interaction, the player provides information to the computer too. The computer game is giving information to the person who is playing, and the person is also giving information to the machine in a bidirectional manner.

Shifting the paradigm in this context means a total understanding of the machine/human interaction which allows us to focus on the other direction in which the interaction occurs, shifting the focus from the computer as information provider to the player as information provider.

Which kind of information can be provided by the player?

Basically, the information that we can extract from the player can be divided in two groups, depending on the objective that we are trying to achieve.

Using AI to improve computer games: Here, we look at information about player behaviour. For instance, taking in consideration the frequency of pressed keys, performed actions and in-game events, we can attempt to infer their proficiency levels, preferences and create a player profile which can be used to classify and compare players (Baumgarten, 2010). This kind of information is useful to generate adaptive computer games.

Using computer games to improve AI: The second group of information is specific human knowledge, which is the target of Human Computation, and depends on the kind of computer game we develop: Starting from a certain desired information, we can develop a specific video game to extract it from the human players.

Both frameworks will be explored in detail in the following sections.

## **2. USING AI TO IMPROVE VIDEO GAMES**

### **2.1 Adaptive Games and Player Experience**

The gaming industry depends heavily on user engagement and immersion, as this is one of the main purposes of games. To cater for many different player preferences and playing styles, the game has to have either a very broad appeal to many users, avoiding too extreme difficulties and too varied content, or try to tailor the experience towards individual players or at least player groups. In general, these adaptive games require the developer to classify user behaviour, predict it online and finally optimise the software towards this behaviour.

Classification of user behaviour can be understood as the task of grouping users according to their interactions with the software (i.e., key-strokes, mouse movement, pressed buttons, etc.).

Prediction of behaviour into a given class can be defined as estimating the unknown value of an attribute - user behaviour - of a system under study given the values of other measured attributes. This is a predictive learning problem that can be automated and solved by a machine.

The process of optimising a program with respect to a given behaviour pattern requires a mapping of behaviour classes to program modifications that have a high probability of improving the fitness of the program. Again, this can be automated as a learning process during software development.

The optimisation process requires the evaluation of the fitness value for each iteration, which depends on the specifications of the developer (conceivable fitness functions include user

satisfaction, time to task fulfilment and number of user mistakes). In many cases, these values can only be retrieved through actual user testing, and as such the optimisation process could be applied during a user test phase.

From an academic point of view, computer games are an ideal research test-bed for adaptive content generation and predictive machine learning, because they typically provide a closed world with changeable rules and several degrees of freedom that allow for individual behaviour patterns to emerge.

One area of our current research interests is the automated adaptation of video games. As part of this research, we want to show that a fully automated framework for adapting video game content is feasible and create such a framework. This process can be divided into several sub-projects:

**Data collection:** A suitable platform for behaviour analysis has to be found and adapted. The chosen video game will be implemented / adapted for easy and efficient data collection and user polls.

**Sub-grouping and Clustering of game data:** The collected game data will be used to create groups of players. The measures which are most distinguishing can be used to assign attributes to the groups in order for game developers to understand the separation.

**Predictive machine learning:** Given the behaviour groups, incomplete data from a player can be used to predict the group in which the current player fits best. The earlier this distinction is possible, the sooner the software can be adapted to the specific group.

**Iterative adaptation learning:** Adapting software to a user is not straight forward. A priori it is not clear which modifications are satisfying for the user.

**Generalization of framework:** Several games will be used to show that the user groups are game-independent, i.e., if two players are classified into a group for one game, it is likely that they are classified in a group for another game.

The points outlined above are part of ongoing research. In section 2.3, we outline an experiment we conducted to investigate the first three points above: data collection in a game, the subsequent analysis, and first steps towards player behaviour prediction.

## **2.2 Adaptive Games Background**

The use of machine learning with respect to video games has been a growing field for the last years, and a substantial body of work exists (Galway, L. et al, 2008). We will highlight some works on adaptive games and give some examples of commercial games that use adaptive games.

### **2.2.1 Adaptive games in research**

G. Yannakakis and J. Hallam (2004) have conducted experiments on neuro-evolution mechanisms

applied to predator/prey multi-character computer games, and realised that near-optimal behaviours are less interesting for humans to play against than behaviours that emerge by following a generic measure of predator/prey computer games' interest which the authors introduce.

P. Spronck (2006) created a technique called dynamic that employs an adaptive rulebase for the dynamic generation of game AI during game sessions. He argues that online learning of game AI should meet four computational (speed, effectiveness, robustness and efficiency) and four functional requirements (clarity, variety, consistency and scalability).

### **2.2.1 Commercial adaptive games**

Adaption in commercial games is most often realised as a method of dynamically balancing the game difficulty. A prominent example of substantial alteration of game content is Valve's Left 4 Dead and its successor Left 4 Dead 2, which use a so called AI Director. It is used to procedurally generate a different level setup for the players each time the game is played. It tracks individual players performance and how well they work together as a group to pace the game, determining the amount of enemies that attack the player and the location of boss encounters based on information gathered. Besides pacing the Director also controls some video and audio elements of the game to set a mood for a major encounter or to draw the players attention to a certain area. In the second part of the franchise, the Director also alters the structure of the level by removing or adding obstacles, thus modifying the path the players have to take.

In the 1999 real-time strategy title Homeworld (Relic Entertainment), the number of ships that the AI begins with in each mission will be set according to how powerful the game estimates the player's fleet to be. Good players have larger fleets because they take fewer losses. In this way, a player who is successful over a number of missions will begin to be challenged more and more as the game progresses.

Many racing games, such as EA's Need for Speed series, Nintendo's Mario Kart, and Criterion's Burnout series adapt to player skill through a mechanism dubbed the "rubber band AI", where AI controlled opponents are sped up or slowed down depending on their distance to the player, with the effect that the racers can be perceived as tied together by a rubber band.

## **2.3 A Player Behaviour Classification Experiment**

We performed an experiment to see whether it is possible to measure and distinguish behaviour of a player playing a simple game with very limited inputs and a limited variety of possible actions. We used the arcade style action game Pac-Man to record a variety of game-related metrics in an online survey comprised of 245 players, each playing 5 sessions of Pac-Man. The resulting data is analysed using discretisation methods and linear discriminant analysis.

### **2.3.1 Pac-Man**

In Pac-Man, the player can navigate in 2 dimensions (up, down, left, right) to eat all pills scattered over the playing area. He has to avoid touching the four ghosts, which hunt him through the labyrinth. He can temporarily turn invulnerable to ghosts by eating one of the four power-pills in the level, and gets bonus points when eating ghosts during this phase. Eating all four ghosts under

the influence of a power-pill results in a significant amount of bonus points that are key to obtaining a good score. Occasionally a fruit appears somewhere in the level that can be eaten for bonus points as well.

### **2.3.2 Survey**

We conducted a survey where participants were asked to play 5 rounds of Pac-Man. The survey was conducted over the internet, using Facebook as a platform, which simplified user authentication and the collection of simple statistical metrics such as age groups and gender. The game uses a database to store all relevant game information, so that games can be fully reconstructed post-hoc. This was then used to identify suitable metrics which are extracted and used in the classification process. While this information depends on the game used, it is desirable to keep the information sufficiently general, as the classification process is aimed to be game independent.

### **2.3.3 Data Analysis**

Discriminant analysis is used in statistics and machine learning to characterise or separate classes of objects based on a set of measurable features and class information of these objects. Linear discriminant analysis (LDA) utilises a linear combination of these features to (attempt to) separate the groups of objects. In other words, it maximizes the ratio of between-class variance to the within-class variance in any particular data set, thereby guaranteeing maximal separability.

In our experiment, LDA was used to extract information about player behaviour. The weights of the features in the first dimensions of the LDA solution indicate the most important features that reflect the behaviour of a player and how it differs from other players. A positive side-effect of this method is that unimportant features are eliminated automatically.

We use a multi-class variant of LDA because we treat each set of 5 sessions of a player as one class, which leads to a large number of classes. The rationale of using this approach is that if the sessions of a player are grouped together and the features recorded in the game are used to separate players as much as possible, these distinguishing features become apparent and we have found a representation of the players that reflects their playing style and skill. Furthermore, players with similar behaviour and playing style will have a small distance in this LDA space, effectively grouping similar playing styles together. In this sense we can use LDA as a classifier for player behaviour; not necessarily predicting the correct class (i.e., player) for a given object (playing session), but the local group of classes will give an indication of the playing style this player is exhibiting.

Of course, the quality of this indication of playing style hinges on the quality of the linear discriminant analysis. If the measured features are unsuitable for distinguishing player behaviour, or the game does not provide for enough expressivity, a good class separation will not be possible, thus creating a space that does not represent player behaviour and skill well.

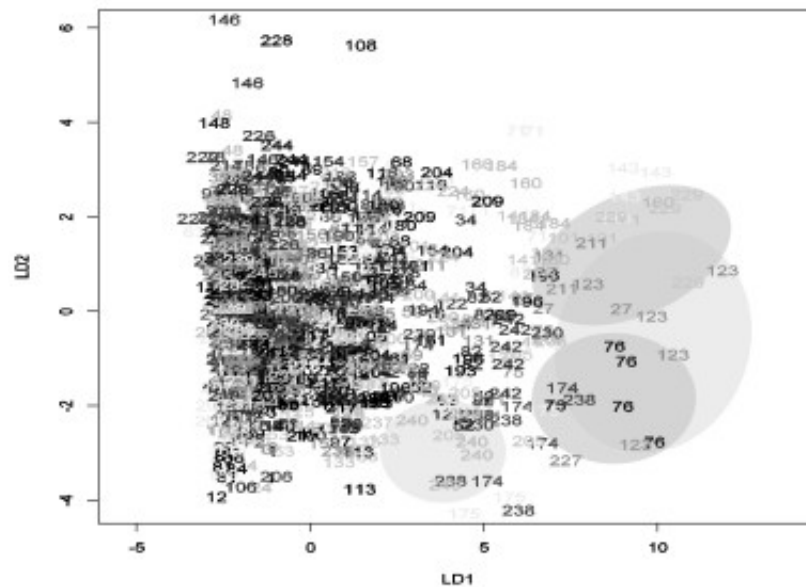


Figure 1. Plot of the first two Linear Discriminant Analysis dimensions of the Pac-Man survey. Each number represents a player, appearing five times as five games were recorded. Some players (240, 76, 229, 123) have been highlighted with circles.

### 2.3.4 Results

Figure 1 shows the first two LDA dimensions of the Pac-Man dataset. Each number represents a player, and will appear five times, its position marking the characteristics of one session of that player. The figure suggests that the sessions of the players share similarities and the chosen features can be used to classify player behaviour. In the graph we can see groups of the five sessions from the same player clustered together, which indicates a similarity between these sessions (a player behaviour trait) has been identified. The features used in our dataset are sufficient to reveal differences between players and similarities between games of the same player and we can therefore interpret the LDA space as an approximation to a space where the distance between sessions is determined by player behaviour and style.

Further analysis of LDA dimensions has been done by Baumgarten (2010).

## 2.4 Conclusions of the Experiment

The analysis of the survey data indicates successful detection of player behaviour in game data recorded in a relatively simple computer game. We have shown that linear discriminant analysis is a useful tool to process the data, reduce significant dimensions and identify important features, and will continue to use it as a first step in our player behaviour analysis. Of course, more analysis will be required to further quantify the correctness and usefulness of this approach and compare it to other methods such as principal component analysis and bayesian network classification.

In our data, we were able to determine that the most important gameplay features returned by our analysis are also what defines a game in general: physical interaction with the device, and secondly

the main goals of the game. We discovered that players can easily be aligned along the dimensions of risk averse over risk accepting to risk seeking. In particular, in Pac-Man that meant that players tried to avoid ghosts, or had an efficient strategy to deal with them, or risked too much and lost lives early.

### **3. USING COMPUTER GAMES TO IMPROVE AI**

We have shown in the last section how AI can be implemented in computer games in order to improve them, but now we will be going in the opposite direction: How computer games can contribute to improve Artificial Intelligence, and more concretely, Computational Creativity.

We argue that computer games can be used to improve AI systems, harnessing the human potential for the machine training process. First of all because computer games can transform a boring or disgusting task in an interesting and fun activity. Indeed, computer games can be an excellent way to provide a lot of different information that is more difficult to conceptualise for machines than humans (about feelings, aesthetics, etc).

Using computer games, we can transform a wide number of players into trainers of our Artificial Intelligence programs, accelerating the velocity of training processes exponentially. In this paper we propose a concrete way to harness human skills such as classification of abstract artistic images and musical pieces in terms of aesthetic value and evoked feeling to train our computational creativity system. In order to achieve this goal we will present our program as a multi-player game.

Computational Creativity is a challenging field, one reason being that all the artefacts created by machines will be judged by humans following an human criterion. Because of this, the training is a fundamental part of the process: Training processes need an human as a information provider or as judge, during a long period before the program can really be creative by itself.

In this paper we propose the use of computer games to harness the human potential to make the machine training process a task easier and - why not - more fun.

#### **3.1 Why use Computer Games?**

##### **3.1.1 Computer games are fun**

Huizinga (2007) defines the human as homo ludens, he argues that one of the things that defines us is our game capacity and our love for games.

It can be difficult for us to be focused in a task for a long time, but we can easily spend hours playing games without a decline in attention. In the same way, our willingness to play is more enthusiastic than our willingness to do other tasks. This feature of games can be exploited to achieve that more people perform certain tasks.

Obviously, this idea isn't new, and has been used to raise the interesting of boring or mundane tasks. Volkswagen, for example, created The Fun Theory (<http://www.thefuntheory.com>), which are

games to improve the behaviour of people towards themselves, the environment or others. We can find a lot of different examples, one of them is the bottle bank arcade machine, a simple game that rewards you with points if you put a bottle inside the recycled container in the correct moment, which is used to motivate people to recycle.

### **3.1.2 Social interaction**

One of the components that makes a game attractive is social interaction. Games are by themselves a strong motivation to do something, but the multi-player aspect of games, which allows interaction and socialization with other players makes the game even more attractive. This is because players are not only motivated by the challenges of the game but by its social interaction aspects, too.

Another component, strongly linked with social interaction, is social competition and a sense of achievement. Players often feel accomplishment when they pass a level or a mission, when they obtain a good score, and, of course, when they are the first in the ranking. Rankings create social competition and bring players to try to pass other players in the ranking in order to obtain social recognition.

If we consider all these factors and implement them in a computer game, we can increase the chances that many people play our game for a long time. Or, in our case, get a lot of trainers for our computational creativity system.

### **3.1.3 Computer games as information providers**

We saw that the games can transform a boring task into an interesting task, but this is not all they allow us to do. Computer games can be an excellent way to provide information about human behaviour, feelings, aesthetics, and so on.

There are a number of activities in which humans are more skilled than computers. For instance, humans are better in describing images in a textual way and, of course, in recognizing aesthetic values and feelings.

If we consider humanity as an extremely advanced and large-scale distributed processing unit, it is easy to see that we can use human processing power in order to solve a lot of problems in Computational Creativity field.

This is what we call Human Computation: Humans, working in parallel with computers in order to resolve problems (von Ahn, 2006).

Many programs based on this idea exist, for instance:

**Galaxy Zoo** is an astronomic application for classifying galaxies according to their shapes. If all, or at least most people agree on one galaxy classification, the system learns the classification of the galaxy. If the classification votes are ambiguous, experts can focus on this concrete galaxy, which might prove more interesting.

**The ESP game** is a textual indexation and description of pictures program: it is a two-player game where you are randomly partnered with another person. The goal of the game consists in finding the

same words to describe one image. If both persons write the same word the image-word pair is retained by the system.

**Peekaboom** is a game to locate objects in images that can be used to train artificial vision systems. It is a two player game where one player chooses one object in the picture and clicks it, the other player obtains just the part of the picture that the player clicked and writes the name of the presented object. (von Ahn et al, 2006)

FunSAT is a visual puzzle game. It transforms solving Electronic Design Automation instances into puzzles and presents them to human players who attempt to solve them. (DeOrio and Bertacco, 2009)

### **3.2 Concrete Proposal**

Our concrete project proposes to leverage humans skills such as classification of abstract artistic images and musical pieces in aesthetic terms to train our computational creativity system.

During our research in the computational creativity framework we have investigated creativity on a neuronal level, finding an interesting relationship between synesthesia and creativity (Cabañes, 2009 a,b). We consider that synesthesia can be a good model for computational creativity, first, because if creativity consists of establishing new relationships between old ideas or works, by using synesthesia we can extend these possible relationships. This allows them to be established in different perceptual and conceptual worlds, and thus are not completely random because they follow a pattern and can be modalized. And, secondly, if we understand that part of the social validation of an artistic work is related to the feelings that this work produces in the viewer, using synesthesia we can find the relationships between colours, notes, rhythm, forms, etc. and feelings, and for this reason, generate computational creativity systems able to generate specific feelings.

It is because of this connection of creativity and synesthesia that we started to develop a synesthetic computational creativity program. However, in order to simplify the program we restrict it to the most common and the most useful kind of synesthesia: the sound-vision synesthetic. In order to do this we need establish the relationship between images and sounds extracted from The Painting Fool ([www.thepaintingfool.com](http://www.thepaintingfool.com)) and The Freesound Project ([www.freesound.org](http://www.freesound.org)) respectively.

All the used images are abstract and the sounds do not have lyrics because the aesthetic feelings that one picture or sound can generate in a person are strongly related with two very different kinds of information. One is the feeling that the combination of colours, intensity, light, forms, etc. in images, and the combination of tones, notes and rhythm in songs transmit to the recipient. The other is the representative, or meaning of the medium. For instance we can transmit the feeling of violence in an abstract image using red and black colors, very sharp angles and abrupt changes in image, or, in a representative image by drawing a war, or someone killing another person with a knife, etc. The first kind is more instinctive, and is related to the primitive relationship between senses and feelings. The second one requires a superior level of abstraction and knowledge about the objects represented in the image.

For a computer program it is easier to recognize and generate images in the first way, and this is one of our arguments to choose abstract images. Additionally, we are more interested in the primitive an

instinctive connections between senses and aesthetic feelings, and sometimes the second kind of information can disturb or interfere in the apprehension of the first kind of information.

### **3.2.1 Game Description**

The game we are creating for our research is based on the agreement of two players when choosing fitting attributes for a combination of images and sounds shown to them.

When the players start the game, random images and sounds (from our previous selection) are presented and they can choose a random word from a list of words originating from the aesthetic world (also previously selected). If both players select the same words to describe the image or the sound, the image-sound-word combination is stored by the system.

The dynamic that we use is similar to the ESP game, with some changes. The aesthetic description of paintings and sounds is more difficult than the textual description of images, which means that if we give total freedom to describe the artistic works (by typing free text), a match in a description will be very rare. For this reason we preselected the words that can be used by the players. The problem that we find here is that if we impose restrictions, the program can be limited, but if we don't give restrictions the game can be impossible to pass. For this we consider that we can provide a set of initial words and allow the players to include new words (they can suggest a word and if more than a certain amount of players suggests the same word, it will be included in the initial list of words). In the first few levels we can start with simple and few words (for example: sad, happy, harmonic, disharmonious, relaxed... pairs of antonyms), and the followings levels the number of words increases, the most common word selected to describe a concrete image or sound are suppressed and word suggested by the players are included.

### **3.2.2 Conclusions of the proposal**

With this game we simulate the aesthetic classification of the selected sounds and paintings and find the matches between them (which sounds or paintings generate the same aesthetical feelings, or similarly, which can be described with the same words). The final goal of this game is to train our future synesthetic computational creativity program, that, after the training process, should be able to:

Generate sounds with a painting as input

Generate paintings with a sound as an input

Generate sounds and paintings with a feeling or aesthetic concept as an input.

The proposal is currently being developed, and for this reason we have not results yet, but based on the results that human computation has had in other areas of AI (as shown in the background section), we are confident that our game will produce useful results. The guiding principle is the same: using video games to train computer programs to develop activities in which humans are more skilled than computers. Additionally, the dynamic of our game is quite similar to the dynamics used in some of the examples shown in the background.

## **4. Conclusion**

Just a little change in our perception about the computer games can mean a big paradigm change. When we understand computer game interaction in all their meaning, we can see that the interaction between human and machine which computer games allow is bidirectional, and we can focus not just in what computer games can show to the player, but in what the machine can learn from the player, too.

The two concrete examples in adaptive games and human computation fields that we have described here are just to illustrate the possibilities that we discovered when analysing and reversing the flow of information between players and the machine. However, our goal in this paper is not just to show concrete examples that we are working on, or our results in our respective fields, but to show how understanding computer games as a bidirectional interactive system, and focusing on information that we can extract from the player implies a paradigm change that leads to a wide world of new possibilities in the computer games and AI field.

## **REFERENCES**

Baumgarten, R., 2010. Towards Automatic Player Behaviour Characterisation using Multiclass LDA, In the Proceedings of AISB'10 Symposium: AI & Games.

Cabañes, E., 2009. Synaesthesia and Creativity: A research at neurological level. The Copenhagen Neuroaesthetics Conferenca. University of Copenhagen. Available online at <http://euridicecaban.es.tl/Publicaciones.htm>.

Cabañes, E., 2009. Sinestesia y Creatividad Artificial. III Congreso sobre Sinestesia, Ciencia y Arte. Granada. Available online at <http://euridicecaban.es.tl/Publicaciones.htm>.

DeOrio, A. and Bertacco, V., 2009. Human Computing for EDA. DAC 2009. San Francisco, California, USA.

Galway, L. et al, 2008. Machine learning in digital games: a survey. Artificial Intelligence Review 2008, Springer.

Huizinga, J., 2007. Homo ludens. Alianza, Madrid, Spain.

Spronck, P. et al, 2006. Adaptive game AI with dynamic scripting. Machine Learning, Vol. 63, No. 3.

Von Ahn, L., 2006. Games With A Purpose. IEEE Computer Magazine, June 2006, pp 96-98.

Von Ahn, L., Liu R., and Blum, M., 2006. Peekaboom: A Game for Locating Objects in Images. ACM Conference on Human Factors in Computing Systems, pp 55-64.

Yannakakis, G. and Hallam, J., 2004. Evolving opponents for interesting interactive computer games. From Animals to Animats, Citeseer.