

A Battle of Wit: Applying Computational Humour to Game Design

Dormann Claire

► **To cite this version:**

Dormann Claire. A Battle of Wit: Applying Computational Humour to Game Design. 14th International Conference on Entertainment Computing (ICEC), Sep 2015, Trondheim, Norway. pp.72-85, 10.1007/978-3-319-24589-8_6 . hal-01758416

HAL Id: hal-01758416

<https://hal.inria.fr/hal-01758416>

Submitted on 4 Apr 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



A Battle of Wit: Applying Computational Humour to Game Design

Dormann Claire

University of Ottawa / Institute of Dementia, University of Salford
Allerton Bg, Salford M6 6PU, UK

cdormann@acm.org

Abstract. There is still a dearth of humour in computer games. To spur the use of humour in games and overcome some of the difficulties in producing humour, we advance that game design can benefit from research in computational humour. The focus of this paper is thus on verbal humour and humour design. Integrating computational humour in games could facilitate humour scripting and solve one of the oldest problems in game humour related to repetition. A humour bot could enhance gamers' experiences, by stimulating social bonding or supporting comic relief. We believe that the use of computational humour for game design would enhance players' laughter and designers' creativity. Last, as game design can benefit from advances in computational humour, so virtual agents can from game research.

Keywords: humour design, computer games, players, tools, computational humour, verbal humour, one-liner

1 Introduction

Games can challenge us, move us and affect us in many ways, including making us laugh. Humour enhances fun; it induces laughter and creates a highly pleasurable experience. The role of humour in games is manifold: humour can surprise or delight us, while it is an important component of interpersonal relations and can provide comic relief [1]. Despite a few notable exceptions such as *Portal 2* (2011) or *Borderland 2* (2012) for the more recent ones, many players and game reviewers still deplore the lack of humour in many computer games.

Indeed, crafting humour and stimulating mirth in games is not easy, and repetition and timing can be problematic. To make humour work, writers and designers need a good knowledge of humour mechanics, as well as a good dose of creativity. In an interview by Ashcraft [2], Ramis stated that writing humorous scripts for games is quite demanding, with all the many comic alternatives to produce. He added that "*it is like writing three hit movies, the scripts are impossibly long*". To spur the use of humour in games and overcome some of the difficulties in producing humour, we advance that games' design can benefit from research in computational humour.

This paper is centred on humour design and is situated in a long-standing investigation dedicated to stimulate the design of humorous games and comical gameplay. Furthermore, we seek to understand the players' experiences, the role of humour in game design and how it is embedded in games. Our investigation includes game players' interviews, the analysis of game scripts and walkthroughs, playtesting and the elaboration of humour patterns.

In this paper, we first review research in computational humour that can be applied to games. As computational humour relates to verbal humour, we focus on its use in games by outlining issues and game objects related to verbal humour. Then we discuss how and where games could benefit from the addition of computational humour: within the game itself as well as within the game space. We describe the foundation of a conceptual framework for the development of computational humour for games. Last, we highlight research issues related to the application of computational humour to games. We believe that as game design can benefit from advances in computational humour, so virtual agents can from game research. We hope to inspire a novel research agenda that integrates computational humour with game design and development.

2 Background in Computational Humour

Computational humour broadly includes the recognition or detection of humour and the automatic generation of humour. It spans a number of areas from the development of humour engines to the design of humour bots, humorous virtual agents and robots.

The automatic generation of humour relates to the development of verbal humour, as it is relatively easier to analyse and compute than other forms of humour. Binsted et al. [3] developed the Joke Analysis and Production Engine (JAPE), a punning riddles generator "*what do you call a murderer with fiber? A cereal killer*". Since those early days a number of systems have been produced that generate short forms of verbal humour such as simple idiom-based witticisms with the WISCRAIC system "*the friendly gardener had thyme for the woman!*" or, generating whimsical sentences with the HAHAcronym system – "*FBI: Fantastic Bureau of Intimidation*" [4]. To improve the quality of verbal humour, specific techniques and methods have been elaborated to find the best candidate for a joke. For example, Sjobergh and Araki [5] derived a measure of funniness based on the frequencies of words that occur in jokes.

Another strand of research in computational humour connects to systems aiming to identify, index or classify humour from a large corpus such as Internet collections of jokes [6, 7]. Humour bots such as those developed by Dybala et al. [8] and Augello et al. [9] recognise and mine jokes from the Web or from a lexical database such as WorldNet.

The last important area of research connects to the design of humour bots and humorous virtual agents, or Humoroid, as Dybala called them. Augello et al. [9] created a humoristic chat bot for Yahoo Messenger, which functioned as a kind of humour partner. The system retrieved jokes after asking users what kind they wanted to hear. Their bot could also recognise different forms of humour generated by users (i.e. based on antinomy, alliteration or adult slang) and then react accordingly by providing appropriate feedback. Dybala et al. [8] developed talking agents that used

humour during dialogues: their system selected a keyword from a user utterance and then outputted a short verbal humour statement. Moreover, Ptaszynski et al. [10] furthered their system to integrate an emotion analysis engine. Taking into consideration users' utterances and emotional reactions, the engine was able to detect when it was appropriate to tell a joke or not. As Stock and Strappavara [4] have pointed out, a humorous system should be able to “*recognize situations appropriate for humour, choose a suitable kind of humour for the situation, generate an appropriate humorous output, and, ... evaluate the feedback*”.

3 Verbal Humour in Game Design

To illustrate the benefit of computational humour for game design, we review briefly the use of verbal humour within games and outline issues with its production. Then we discuss humour in the game script, identifying short forms of verbal humour that are likely candidates for computational humour. More generally, we see two ways in which computational humour could be advantageous for humour design: as tools to help game writers to insert humour in games, and as tools to produce humour, or one-liners, automatically during gameplay.

3.1 Issues in Verbal Humour

Writing game scripts of games celebrated for their humour such as LucasArt games or games like *Portal 2* or *Borderland 2*, where humour plays a significant role in the game, is impressive. Bridgett [11] described different aspects of spoken (or written) game text as game story pieces (such as background “once upon a time”, cut-scenes, narrators, etc.), and in-game mission dialogues (describing quests' information or instructions on how to progress in a game). The most important part of game text is conveyed through dialogues with the avatar, main Non-Player Characters (NPCs), and a host of various minor NPCs. To that, we can add various game objects ranging from in-game items, to interface elements, etc.

Of course, writing humour for games is hard. As Bridgett [11] remarked, “*depending on how many characters there are, and the AI behaviors the characters have, there could be as many as 10,000 to 15,000 individual contextual reactions that will need to be written*”. According to its creator, in *Rat Race* [12] humour will be integrated into dialogue-driven mini-games, adding up to hundreds of possible alternatives that will be heard in function of a player's performance. Indeed, a key aspect of game text is the sheer numbers of variations that might have to be written for a game event, character or situation; more so for games which are text-intense, such as adventure games or role-play games. Writing and designing humour in a game requires a considerable investment. Not everyone has enough knowledge or creativity to write enjoyable comedy narratives, produce jokes or one-liners at will.

Besides the sheer number of comic dialogues and humour game elements to produce, the worst problems with humour are timing and especially repetition. Timing is important in jokes, between the beginning of the joke and the punch line, to create surprise and the unexpected effect. Game timing can be disrupted by players' control and choices. More importantly, many jokes are funny once, perhaps twice, but once

you have heard them ten times then they become quite annoying. In some games, like First Person Shooter (FPS) and battle sequences, some lines are repeated hundred times. Because of failure, some puns might be heard over and over. In an interview by Gonzales [13], Schafer summarised the problem quite well as, “*the hilarious boss monster who has four taunts he yells at you over and over, each of which is funny approximately one time, if you’re lucky*”.

3.2 Verbal Humour and Game Script

As we have seen, computational humour relates to the production of short forms of verbal humour. We subsequently refer to this as a “*Oneliner*” (for lack of a generic term) for any short forms of verbal humour including witticism, pun, putdown, teasing, banter, riddles, and so on. This is in contrast with the longer forms of verbal humour like comic narratives or dialogues.

We do not advocate nor believe that Oneliners are a panacea to insert verbal humour in games. First, humour for games should be carefully planned in function of humour mechanics, within the context of the game goal and storyline. Once a comic strategy is established, an overview of the game humour and its elements should be outlined, as a sort of humour wireframe for the game.

Second, not all verbal humours are delivered as Oneliners. Main characters and events often require a tighter control of the humour through a more complex comic script that advances the plot forward. In this situation, humour is then written as comic narratives rather than Oneliners. As many game writers have stated, writing every single character as a Oneliner firing machine is a recipe for disaster. It gets tiresome and weakens the player’s ability to care about each character, and thus disrupts the effectiveness of humour within the game. However, as some stand-up comedians who have mastered the art of Oneliners to perfection show, a judicious choice and use of Oneliners can be tremendously funny.

If we take *Portal 2* (2011) as an example, as it is typical of Weatley, a comic monologue is taking place just at the beginning of the game when Weatley is trying to rouse Chell. This example illustrates the longer form of verbal humour:

“Fine! No absolutely fine. It’s not like I don’t have, you know, ten thousand other test subjects begging me to help them escape. You know, it’s not like this place is about to EXPLODE...All right, look, okay, I’ll be honest. You’re the LAST test subject left. And if you DON’T help me, we’re both going to die...”

Nevertheless, through the course of the game, Weatley has many Oneliners (short feedback lines), which depend on the players’ actions and gameplay situations. Similarly, the cores, to put it simply, are rambling on different themes through Oneliners.

3.3 Oneliners and Games

Verbal humour permeates every aspect of computer games and we found examples for every feature, from names of objects and NPCs (e.g. Hemer Nesingway in *World of Warcraft*, 2008) to dialogues / monologues as we have seen, and to game documentation (e.g. parodies of Agony Aunt columns in *Maniac Mansion*, 1987). To

emphasise the usefulness that computational humour tools could have, we looked for and identified game text objects which have the Oneliner format. We searched for places in the game script which have the following characteristics: the lines are smaller and stand-alone. Moreover, these lines need more alternatives and / or occurred frequently during gameplay. We found a number of those that pertain to different script elements, from character dialogues to missions and game objects. No doubt there are many others, and their uses depend on the humour strategy, the genre of games, or players.

Barks are small parts of dialogues, a few words that characters blurt out. Barks are used to give players' feedback while making the game world alive and feel like a responsive entity. Although barks are found in many contexts, common occurrences are during battle sequences. Players can hear a hundred of them like "*I will kill you*", "*Argh*", "*I am dying*", etc., over and over. Thus, barks can become quite annoying. As Hamilton stated [14], the *No One Lives Forever* series created the most memorable stealth segments. *Borderland 2* is also infamous for barks said in combat, e.g. for Nomads' "*Midget on the loose!*" and "*I miss my midget*" (on death).

Taunting and insult fights between characters are all prime candidates for computational tools. The crude gallows humour that some FPS players seem to like could be generated at various times before confronting an enemy. Taunting can also give rise to insult fights, used frequently in computer games based on boxing, wrestling, martial arts, and so on. However, the most famous comical insult fight is found in the *Monkey Island* series (1980–2000): Insult: "*Soon you'll be wearing my sword like a shish kebab!*", Comeback: "*First you better stop waiving it like a feather-duster*".

Comic banter or comic exchange between two characters is an excellent source of fun; it is an easy way to avoid any problems with timing, as the characters' lines follow each other. While banter between main characters could be more difficult to generate automatically, such is not the case for secondary characters. As noted by Grönroos [15] in *Dragon Age* (2009), banter in that game is triggered at random when a player explores the game world without being intrusive.

Similarly, a host of secondary characters can be used to insert humour for comic relief or comic fun. Comic relief is often provided to relieve stress and lighten the mood after a tense combat mission or grinding to level up [16]. To accommodate different players' experiences, a character in a party as the class clown is often designed as humorous, delivering a high number of Oneliners. Thus, depending on whether a game is designed as comical or uses a high number of comic characters, a huge number of Oneliners would then be needed. In *World of Warcraft* (2008) players are given the opportunity to tell predefined jokes, which vary based on the avatar's race and gender. Thus, using computational humour tool there would provide an infinite variation of these jokes so they stay fresh and still surprise the player each time, making the game more playful.

Game objects are also an important source of humour for which computational tools could be used. In comic adventure games like *Day of the Tentacle* (1983), all the objects that the players manipulate are given comic feedback. For example, the object "*Chattering teeth*" as the line "*Jumpy little sucker*". Some players delight in finding and reading all these Oneliners. A prototypical comic object used to insert humour without intruding on the main script is the radio (e.g. *Grand Theft Auto series*).

Although radios can provide comic narratives, they also use a high number of Oneliners. Similarly, Oneliners can be inserted through a newspaper's headlines, a joke calendar, etc., and be varied in function of the game segments. Only the writer / designer's imagination is the limit of what might be possible.

4 Game Space and Humour Bot

As we are just beginning to explore the application of computational humour to games, we also want to highlight another example where such tools could be useful: in the game space through the addition of a humour bot in multiplayer games to enhance players' experiences.

A number of multiplayer games have an inbuilt voice and / or text chat. Most often, players can communicate privately on a one-to-one mode with a group of preselected players (e.g. member of the guild), or with everyone. Nijholt [17] proposed the use of a humour bot in the chat channel to enhance fun in computer games. Although fun should perhaps emerge primarily from the gameplay, using a humour bot has intriguing possibilities. Indeed, humour seems to be an important element of game chat and of players' experiences. A study of a game chat identified the use of humour as wordplay and amusing references to pop culture, but players also made fun of each other [18]. Similarly, Duchenault [19] found that players commonly engage in small talk and share game tips as well as humour. Players exercise their comic wit, for example, by improvising in-character dialogues. For some players, this is an essential aspect of their gameplay; the game will be fun as much as the players are fun. For those players, negative and aggressive players can ruin their experiences of the game.

Humour and laughter have a variety of possible socio-emotional functions and uses: laughing with, laughing together, and laughing at or against [20]. Humour has an important function for maintaining group relationships and to enhance players' moods to make the gameplay more enjoyable. Laughter and humour are indicative of moments of shared fun. Moreover, humour is a source of comfort and support between guild members.

Thus, a humour bot could use positive and enhancing humour to induce joking behaviour and support social bonding. This could be more important at the beginning of the game session, during idle time or, when visiting social places such as a tavern in games. The humour bot could also provide comic relief after a raid or a defeat and encourage players to level up. However, the most important and complex function of the bot might be to try to curb aggressive behaviour through humour. Players vent frustration or some aggravate others through trash-talking and swearing, as well as verbally attacking them. While some players and game communities tolerate this, such is not the case for all. Thus, a humour bot could criticise lightly this kind of behaviour, remonstrating with players in a self-mocking fashion, channelling aggression through humour and providing jokes to release tension.

Besides chatting in a group, players can also communicate through private chat. Thus, another use for a humour bot would be as a game companion or helper. It can be quite confusing and a bit daunting to learn a new gameplay and join another community of play. While some players are helpful, others are prone to taking

advantage of newbies, mocking and tricking them. A number of helper agents have been developed like Sergeant Blackwell, a witty avatar that answer questions in a virtual military application [21] or, one that provides feedback on players' chess moves [22]. Thus, a humorous game companion could provide assistance to novice players through the private channel, offering in-game information, as well as enhancing players' moods through Oneliners; thus, motivating players to advance in the game by being playful rather than didactic.

5 Towards a Computational Humour Framework for Games

We have discussed ways in which computational humour could be useful and used in games. To further bring the two fields together, we present elements for a basic conceptual framework for games. We use agents or humour bots such as those developed by Dybala et al. [8] and Augello et al. [9] as a point of departure for the discussion.

5.1 Differences between Games, Bot and Agent Systems

There are important distinctions between games, agent systems and chat bots, not least that players do not usually hold a conversation directly with an NPC, as a user would converse with a virtual agent or bot. More importantly here, game worlds are far more complex than agent systems: game characters interact with each other and many objects in an ever-changing virtual world. Thus, in turn, the use of verbal humour in games is richer and more complex than the one used by agent systems.

In most cases in computational humour, the focus is on the production and output of an Oneliner. Beyond creating or retrieving the best Oneliner from a set, there is not much discussion on the type of humour produced. If virtual agents do not have much use for aggressive humour, such is not the case for computer games. Taunting and insult fights use aggressive humour like a putdown. Typically within a game, NPC allies will deliver positive and sympathetic humour, while boss monsters and enemies will use negative humour. A very simple mapping between humour polarity (i.e. from negative to positive) and types of Oneliners could consist in using putdown and insult (and comeback) for negative humour, and pun or jokes for positive humour. Beyond the structure and form of humour, the content of the Oneliner such as with ethnic jokes can also determine if a joke is perceived negatively. Thus, we might want to rate some types of jokes as negative humour.

If we devise a very simple framework for a computational humour game engine based on the work of a humour bot, the computational humour frame for games would have two important features: the polarity of humour that should be used, and as with agents or bots, what the content of the joke should relate to (see Table 1).

Table 1. Computational humour

Agent	Games	Effect
<user utterance>	<game event = trigger Oneliner>	<i>Player talk to Tavern Master</i>
	<humour event polarity >	<i>Positive</i>
<extract keywords>	<humour event characteristic>	<i>Drunk patron</i>
<query computational humour engine >	<query game humour engine>	
<retrieve Oneliner>	<retrieve Oneliner>	
<output Oneliner >	<game event Oneliner output >	<i>Tavern Master: jokes</i>
	< Oneliner counter >	

Another distinction between humorous agents and games relates to keyword vs humour event characteristics. In the agent / humour bot, the Oneliner like a pun is produced according to the user utterance <keyword>, but such is not the case in computer games. Typically in games, Oneliners will depend on a number of factors, such as for NPCs: the types of characters (human vs dwarf), the function of characters (bartender vs soldier), setting in the game world (e.g. in a tavern or castle), or at the highest level, on the kind of game theme used (e.g. a middle age / fantasy, military, horror / zombie game, etc.). Thus, we might need to elaborate a knowledge base, a lexicon of game situations that connect humour events <tavern master> to the content of possible Oneliners to retrieve <joy of drinking>. Then the humour event characteristic would act as the keyword in agents and an appropriate Oneliner would be retrieved for the tavern master. We might also want to keep track of Oneliners that have been given to players, so as far as possible, they never hear the same Oneliner twice. Last, a review for the computational humour engine frame is given in Figure 1. As with the development of humour bots, the computational humour engine could use a very large database of Oneliners or, a mixed system that includes a humour engine that generates a Oneliner automatically when feasible and uses the database when it is not.

5.2 Using a Humour Model for Games

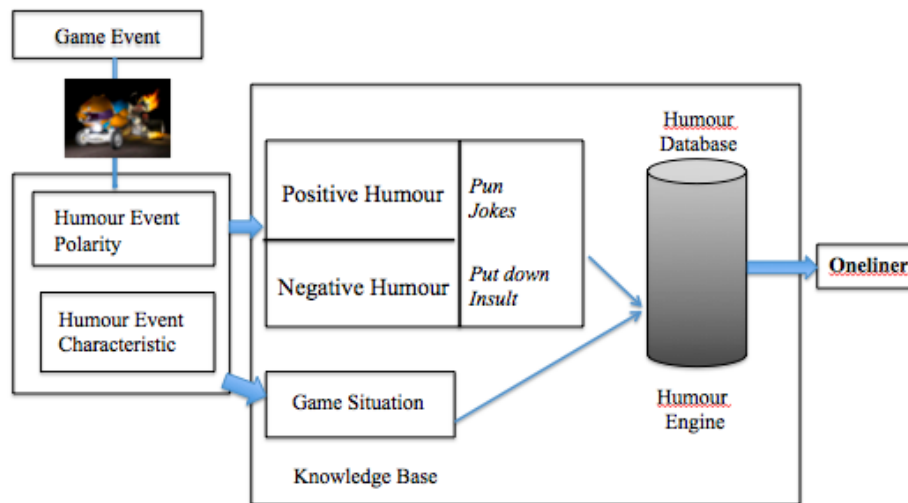
We might need to elaborate a humour model for game design that accounts for more variations in the use of humour than between positive and negative humour. We propose to adopt as a starting point a model based on humour styles associated with personality characteristics [23]. The four types of humour are defined as: affiliative humour (to enhance interpersonal relationships by amusing others), self-enhancing humour (related to mood and an optimistic view of life), self-defeating humour (to amuse others by saying funny things at one's expense) and aggressive humour (to put others down through sarcasm and ridicule).

Such a model could work quite well for game characters, the most frequent sources of Oneliners. However, to make the model more generic and better reflect some of the use of humour in games, we put forward some modifications. We rename the first category *Comic Humour*, for humour that makes people laugh out loud and to thus create a playful and pleasurable experience. In our experience, humour used in games

can be quite exaggerated, wacky, absurd or nonsensical, more so perhaps than what we might expect as affiliative humour. The second category becomes then *Enhancing Humour*, positive humour that is utilised to support relationships or to enhance mood and diminish tension. The third and fourth categories of *Self-Defeating Humour* and *Aggressive Humour* remain the same.

While there are many examples of aggressive humour in games, self-defeating humour is not as frequent but can occur with certain class of characters such as lovable loser or comic relief characters. Affiliative humour would be the type of humour used by the tavern master, while a goofy character such as Glottis of *Grim Fandago* (1998) could be expected to use comic humour. Such a model would accommodate for different functions of humour, and we can also modulate humour according to players' choices. For example, depending on players' answers (e.g. negative), the tavern master could turn sarcastic and use aggressive humour. When transferring a humour model from one discipline to another context, it is difficult to predict entirely how the model will relate to the new domain. Thus, a humour model such as the one we have discussed should be tested with different game scripts and scenarios, and thus refined.

Fig. 1. Computational humour engine frame



5.3 Chat Humour Bot

As humour bots used elsewhere, based on the conversational keywords, a humour bot for a multiplayer chat should be able to retrieve Oneliners that relate to the current conversation. Using a type of humour model such as the one we have discussed would insure that the appropriate humour is delivered at the right time. As before, we might

want in the first instance to establish a knowledge base of game chat situations, to decide when humour is appropriate or not (e.g. coordinating a raid) or what type of humour to initiate (e.g. self-defeating humour for swearing). Further developments could accommodate and integrate an emotion engine with the computational humour engine.

6 Discussion

To strengthen and stimulate the design and use of humour in games, we have discussed the use of computational humour tools. We have suggested two ways in which computational humour could be advantageous for humour design: as tools to help game writers and as tools to generate Oneliners during gameplay. Using a bootstrapping process more specifically attuned to games and automatic humour recognition techniques such as those described by Mihalcea and Strapparava [6], large databases of Oneliners could be developed for specific games, or game genres.

We have outlined elements for a computational humour framework for games. To develop a new research agenda in this domain there are many issues that need to be addressed. From a humour design perspective, to create a multiplayer humour chat bot we might want to study in more depth the use of humour within those games and elaborate the most appropriate knowledge base for the humour bot. Conceptual designs should be advanced and tested to evaluate the effect and players' reactions to the humour bot.

For game design, besides refining the humour model that we have discussed, we might consider developing a more formal classification for Oneliners. Inspired by humour theories and humour patterns, humour classification algorithms could then be derived to produce the best match between the game event— humour and the output— Oneliner. Such research should take into consideration and might contribute to overcoming the current limitations of computational humour recognition tools.

Game engines are becoming more efficient at tracking where players have been and what they have done. Thus, by more fully integrating data from the game engine tracking system with the computational humour engine could lead to the generation of humour that is more contextual and more reactive to the gameplay. Another element that we might introduce within a conceptual framework for computational humour in games is comic expressiveness. For NPCs, comic expressiveness would be tied up with the production of Oneliners and correspond to an NPC facial expression, body language and / or type of laughter.

We suggest that the addition of computational humour could lead to innovations in game design, within game mechanics, mini-games and ultimately perhaps the design of novel game concepts. As Root [13] stated rightly in an interview regarding some of the use of humour in games: *“Solving a puzzle, or defeating an opponent, when you are rewarded with a joke or, often as not, a bad pun, it just doesn't work. It's as if someone handed you a bag of phrases--a priest, ... a hot tub, ... and four fifths of Irish whisky--and then told you to assemble your own joke”*. But could we not just do that? Computational humour recognition tools are becoming more sophisticated; thus, a computational humour engine could also rate some types of Oneliners produced by players: the better the joke, the more points, the more goblins laugh until they die, etc.

Nakatsu and Tosa [24] have developed an Artificial Intelligence (AI) comedy system based on comic dialogue generation. Could a mini-game work like a comedy stand-up performance with an NPC as a humour partner? Games that use natural language interaction such as *Façade* (2005) or *Bot Colony* [25] could benefit the most from computational humour uses.

Another emerging trend of research related to computational humour is laughter. Laughter is very expressive and is associated with the definition of humour. As with humour, the value of laughter is greatly underestimated in game design. Computational systems have been developed that are able to detect and respond to human laughs in real time [26]. Fukushima et al. [27] proposed a system composed of a set of toy dolls that laugh when the user laughs. Such laughter was shown to enhance the user's enjoyment of cartoon animations. Thus, would canned laughter enhance game enjoyment and funniness as it does in films? Would that be more desirable with some specific games, such as social types of games on the Wii? Intriguing questions also relate to the game avatar: could your avatar, breaking the fourth wall, laugh with you during gameplay and what would be the effect? Should we instead do that with the avatar sidekick? Could an avatar become funnier if the player does initiate more jokes during gameplay, etc.?

To conclude, we want to outline the benefit of game research to computational humour and agent design. Virtual agents lack the vocabulary for full comical expressiveness that game characters have. Designing a comical agent demands very different models of expressions and body postures than the more naturalistic representations generally used in virtual agent design. Thus, a virtual humour agent could also benefit from a humour model and from game character design to exploit more fully their comic potential. All research in computational humour relates to verbal humour, except for the work of Thawonmas et al. [28] on pantomime, and David and Mateas [29] on the Road Runner and Coyote jokes. The study of non-verbal forms of humour in game design through game patterns might help to bridge the gap from a high level description of humour such as slapstick / incongruity to a low level description of agent parameters. Last, the game industry has overcome the film industry in terms of revenue. Thus, games could be an ideal test bed for the development of computational humour, stimulating and driving a new research agenda forward.

7 Conclusion

Humour can surprise us, making us roll around with laughter; it creates a joyful experience and makes us more human. As Crecente [30] stated about *Dead Rising 2* (2010), there are a lot of zombie games, shooters and sandbox games but there is only one that does comedy, action and zombie. It made the game truly unique. Indeed, as LucasArt games in their time, *Portal 2* (2011) would not be the same without GLaDOS and Weatley.

Thus, to enhance and stimulate the use of humour in games and make them more enjoyable, we explore how game design could benefit from computational humour.

Computational humour would enable a greater use of humour, as it would facilitate game scripting by reducing the most monotonous humour writing tasks. It would

solve one of the oldest problems in scripted verbal humour related to repetition: tired old jokes that are not funny anymore. Computational humour tools could also support the gameplay within multiplayer games, by making the player experience more engaging and enjoyable. However, the more interesting application of computational humour might be in supporting creativity and new possibilities for game design. Through our exploration of verbal humour and computational tools for game design, we hope to gain some insights to support humour design in games and with virtual agents.

Future research in computational humour and laughter for games would have to address a number of issues. These issues will range from humour design to player experiences, and the development of the computational humour engine architecture and its integration within the game engine. The field of AI and computer games, or games AI, is emerging and diversifying from research on NPC behaviour to player experience adaptation and procedural content [31]. Moreover, AI game views on NPC do not relate necessarily to making human-like NPCs but also look as we did, at other questions more typically anchored in computer game design, such as making the game more enjoyable. As far as we know, computational humour has not been applied to game design nor discussed within game AI. Thus, computational humour could become a specific strand of game AI to adapt and develop the computational humour engine for games, to make better comic NPCs, or within affective games to adapt the type and content of humour to a player's personality or mood. Let us engage in a battle of wit, and develop a new research agenda that bridges computational humour and game design, and give a new impetus to computational humour, game AI, as well as game design.

Acknowledgment. This work was supported by the Social Sciences and Humanities Research Council of Canada.

References

1. Dormann, C., Biddle, R.: A Review of Humor for Computer Games: Play, Laugh and More. *Simulation & Gaming* 40(6), 802–824 (2009)
2. Ashcraft, B.: Harold Ramis On Why it's Hard to Make Funny Video games <http://kotaku.com/5293133/harold-ramis-on-why-its-hard-to-make-funny-video-games> (2009)
3. Binsted, K., Ritchie, G.: Computational Rules for Punning Riddles. *International Journal of Humor Research* 10(1), 25–76 (1997)
4. Stock, O., Strapparava, C.: The Act of Creating Humorous Acronyms. *Applied Artificial Intelligence*, 19(2), 137–151 (2005)
5. Sjöbergh, J., Araki, K.A.: Very Modular Humor Enabled Chat-bot for Japanese. In: *Pacling 2009*, pp. 135–140, Sapporo, Japan, (2009)
6. Mihalcea, R., Strapparava, C.: Technologies that Make you Smile: Adding Humor to Text-based Applications. *IEEE Intelligent Systems*, 21(5), 33–39 (2006)
7. Raz, Y.: Automatic Humor Classification on Twitter. In: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics* pp. 66–70, ACL, Stroudsburg PA (2012)
8. Dybala, P., Ptaszynski, M., Rzepka, R., Araki, K.: Humorized Computational Intelligence towards User-Adapted Systems with a Sense of Humor. In: *Giacobini, M., Lecture Notes in*

- Computer Sciences, Applications of Evolutionary Computing vol. 5484, pp. 452-461. Springer, Berlin Heidelberg (2009).
9. Augello, A., Saccone, G., Gaglio, S., Giovanni, P.: Humorist Bot: Bringing Computational Humour in a Chat-bot System. In: Proceeding of the International Complex, Intelligent and Software Intensive Systems, pp. 703–708. IEEE Press, New York (2008)
 10. Ptaszynski, M., Araki, K., Dybala, P., Rzepka, R., Higuhi, S., Shi, W.: Towards Socialized Machines: Emotions and Sense of Humour in Conversational Agents. In: Zeeshan-Ul-Hassan Usmani (ed.), Web Intelligence and Intelligent Agents, <http://www.intechopen.com/books/web-intelligence-and-intelligent-agents/towards-socialized-machines-emotions-and-sense-of-humour-in-conversational-agents> (2010)
 11. Bridgett, R.A.: Holistic Approach to Game Dialogue Production, Gamasutra, http://www.gamasutra.com/view/feature/132566/a_holistic_approach_to_game_.php?print=1 (2015)
 12. Totilo, S.: ‘Rat Race’ Funnyman Hopes PlayStation 3’s First Sitcom Video Game Fills Comedy Void, <http://www.mtv.com/news/1575219/rat-race-funnyman-hopes-playstation-3s-first-sitcom-video-game-fills-comedy-void/> (2007)
 13. Gonzales, L.: A Brief History of Video Game Humor, <http://www.gamespot.com/features/6114407/p-2.html> (2004)
 14. Hamilton, K.: Why Video Game Characters Say Such Ridiculous Things, Kotaku <http://kotaku.com/5921878/why-video-game-characters-say-such-ridiculous-things> (2012)
 15. Grönroos, A.M.: Humour in Video Games: Play, Comedy, and Mischief. Master Thesis of Art, Aalto University, Finland (2013)
 16. Dormann, C., Boutet, M.: Incongruous Avatars and Hilarious Sidekicks: Design Patterns for Comical Game Characters. In Proceeding of the Digital Games Research Association, <http://www.digra.org/digital-library/> (2013)
 17. Nijholt, A.: Why and When ‘Laughing out Loud’ in Game Playing, <http://eprints.eemcs.utwente.nl/21848/> (2012)
 18. Wright, T., Boria, E., Breidenbach, P.: Creative Player Actions in FPS Online Video Games, Playing Counter-Strike. Game Studies, The International Journal of Computer Game Research, 2(2), <http://gamestudies.org/0202/> (2002)
 19. Ducheneaut, N., Yee, N., Nickell, E., Moore, R.: Alone Together? Exploring the Social Dynamics of Massively Multiplayer Online Games. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 407–416, ACM, New York (2006)
 20. Zijdeveld, A.: The Sociology of Humor and Laughter. *Current Sociology*, 3, 1-64 (1983)
 21. Rich, C., and Sidner, C.L.: Robots and Avatars as Hosts, Advisors, Companions, and Jesters. *AI Magazine* 30 (1), 29–41 (2009)
 22. Castellano, I., Leite, A., Pereira, C., Martinho, A., Paiva, C., McOwan, P.W.: It's All in the Game: Towards an Affect Sensitive and Context Aware Game Companion. In: Proceeding of the International Conference on Affective Computing and Intelligent Interaction. IEEE Press, New York (2009)
 23. Martin, R.A., Puhlik-Doris, P., Larsen, G., Gray, J., Weir, K.: Individual differences in uses of humor and their relation to psychological well-being: Development of the Humor Styles Questionnaire. *Journal of research in personality* 37(1), 48–75 (2003)
 24. Tosa, N., Nakatsu, R.: Interactive Comedy: Laughter as the Next Intelligence System. In International Symposium on Micromechatronics and Human Science. pp. 135–138. IEEE Press, New York (2002)
 25. Joseph, E.: Bot Colony – a Video Game Featuring Intelligent Language-Based Interaction with the Characters, https://www.botcolony.com/doc/BotColony_paper.pdf (2014)
 26. Urbain, J., Bevacqua, E., Dutoit, T., Moinet, A., Niewiadomski, R., Pelachaud, C., Picart, B., Tilmanne, J., Wagner, J.: AVLaughterCycle, An Audiovisual Laughing Machine. In: Camurri, A., Mancini, M., Gualtiero, V. (eds.) Proceedings of the 5th International

- Summer Workshop on Multimodal Interfaces, pp. 79–87. DIST University of Genova, Genova (2009)
27. Fukushima, S., Hashimoto, Y., Nozawa, T., Kajimoto, H.: Laugh Enhancer Using Laugh Track synchronized with the user's laugh motion. In: Proceeding of CHI conference on Human Factors in Computing Systems, Extended Abstracts pp. 3613-3618. ACM, New York (2010)
 28. Thawonmas, R., Hassaku, H., Tanaka, K.: Mimicry: Another Approach for Interactive Comedy. In Proceedings of Conference on Simulation and AI in computer games, (pp. 135-138). EUROSIS, Oostende (2003)
 29. David O., Mateas, M.: Beep! Beep! Boom!: Towards a Planning Model of Coyote and Road runner cartoons. In Proceedings of the 4th International Conference on Foundations of Digital Games pp. 145-152. ACM, New York (2009)
 30. Crecente, B: Kotaku, <http://kotaku.com/5524674/dead-rising-2s-dark-comedy-and-interactive-vomit> (2010)
 31. Yannakakis G. N.: Game AI Revisited. In Proceedings of ACM Computing Frontiers In Proceeding of the 9th Conference on Computing Frontiers pp. 285-292. ACM, New York (2012)