AI-guided game design

Matthew Bedder, Andrei Iacob, Athanasios Kokkinakis, Mihail Morosan

I. Introduction

Improving the player experience has been the main goal of game designers since the beginning of the computer games industry. A plethora of techniques have been used into try to achieve this, of which some approaches have proven to be very effective, and others less so.

In this study we propose two variations on a game designed to improve player experience, and use both AI agents and testing with human players to try to demonstrate that the game variants improve the player experience. For the AI agents, a number of metrics are recorded from AI versus AI gameplay, and some simple analytics used to see if the intended impact is achieved. We then use human players to try to confirm these results, and to try to demonstrate the usefulness of using analytics over AI gameplay to measuring the impact to changes on video games.

In the first instance, intelligent agents were used to play the game and a number of game metrics and analytics were collected in order to quantify their experience. In the second part of the project, human players were asked to participate in an experiment, playing three different versions of the game, one being the classical battle, and the other two versions containing incremental modifications and increased complexity. The human players were asked to fill in a questionnaire while playing through each version of the game, in order to understand their gaming experience and their responses to the changes in the mechanics of the game brought by the modified versions.

II. Existing work

A. Player experiences of games

Recently there has been much work into using automated methods for optimising game designs. Work by Isaksen et al.[1][2] has looked into using survival analysis to optimising parameters defining gravity and jump force in the game "Flappy Bird". Using their technique, a simple, partially stochastic AI agent is made to play the game for a parameter set a number of times, with the distribution of the scores achieved by the agent being recorded. These score distributions can then be recorded for a number of different sets of parameter values, and these distributions used to select sets of parameters that provide gameplay fitting some criteria, with potential criteria including the difficulty of the game, and the "uniqueness" of the gameplay provided by the parameter set. This reduces the need for the game designer to explore the space of the parameter set manually, and suggests that the results of automated playtesting correlate with human players experience with the game.

As well as using automated playtesting to test parameter values, automated playtesting can also be used to assess different sets of game mechanics. Work by Nelson et al. looked into using AI players to evaluate mechanics of procedurally generated games[3]. In this approach it was found that the automatic generation of criteria about the playability of the game were effective when combined with a genetic programming approach for automatically generating game rule sets. In many of these approaches, the criteria used for assessing the quality of games have been relatively complex in order to try to capture the subtleties of player experience, including metrics on whether game states that appear advantageous often result in the player winning[3] or whether the player the ability to significantly impact the score they will achieve[4].

B. AI-based game design

The reasons behind why people enjoy particular videogames but are uninterested in others has remained elusive, despite numerous attempts to quantify them[5][6]. A generally agreed accepted principle is that of similarly skilled opponents, whether they are human or A.I. generated, who are challenging enough for the player to keep him focused in the game while being realistically beatable so that the player does not experience frustration or even anxiety[7].

Human performance in relatively simple videogames that rely on reaction times has been the Learning Strategies Program, originally funded by the Defense Advanced Research Projects Agency[8]. This program gave birth to the Space Fortress game, a game that largely resembles Asteroids[9]. Subsequent examinations of this game showed that Attention, Working Memory and even Fluid Intelligence all highly correlated with game performance[10]. All the aforementioned variables are highly mediated by a multitude individual differences such as gender, nationality, age, socioeconomic status, handedness, working memory and even language[11][12][13][14][15].Thus the selection and development of the appropriate A.I. agents, fitted to players individual differences, is imperative for maximising user experience.

III. Chosen domain

For the purpose of this study, we generated three variations on a Battle Asteroids game. In each of these variations, players are made to control a ship and compete against an AI-controlled player. The player is able to control the rotation of the ship, apply thrust, and fire a finite-number of missiles. If the player travels off the side of the screen, they re-appear on the opposite side of the map. A number of pickups are distributed around the map, and any player that collects the pickup scores a number of points. In all of the variations, the player is also able to score points by shooting the opposing ship. The game ends after a set amount of time, with the ship with the most points being classed as the winner.

The first variation upon the base game was the inclusion of simple asteroids. These asteroids are distributed randomly as the start of the game, and are destroyed when they are shot or a ship collides with them. Players are given no explicit reward for shooting asteroids, and have their score penalised for colliding with the asteroids. The second variation used also included asteroids, but split rather than get destroyed when they collide with missiles or ships.

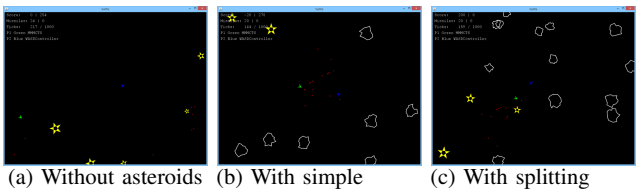
The intention of these variants was twofold; firstly, we hoped that the inclusion of asteroids would force the players to move around the map more, and secondly we hoped that by introducing additional complexity to the game it would be easier to distinguish between good and bad players.

IV. Analytics over AI agents

A. Motivation

In order to try to confirm that the game variants generated had the intended effects on gameplay, we initially generated AI agents to

Fig. 1: Screenshots from the three game modes.



(a) Without asteroids (b) With simple asteroids (c) With splitting asteroids

play the games. We would then log the score, velocity, and missile count of each of the agents over the course of multiple runs, and use these results to try to try to quantify the effects of the variations in the game.

B. Methodology

For the purpose of testing the different variations of the game, 4 agents were used:

- EmptyController, an agent that performs no actions
- RotateAndShoot, an agent that continually turns in a single directly while shooting
- MMMCTS, a version of Monte Carlo Tree Search using hand-crafted heuristics that is given 40ms to select an action
- MMMCTS2, which is identical to MCTS but with a shorter 20ms budget

Agents of different complexity were used in order to estimate the difference between the results of skilled versus unskilled players. For this, we consider EmptyController to represent the weakest of players who do not react to what is on the screen. By comparison, we would expect the RotateAndShoot agent to represent a slightly more skilled player who attempts to shoot the enemy ship, but has little strategy or skill. Finally, we would employ the MCTS-based agents to represent skilled players that would do whatever they could to win the game, always planning ahead a short amount into the future. Between the two, we would expect the MCTS agent with a larger computational budget to perform better, as the additional time given to make decisions would allow the agent to build a larger tree of known states, and therefore could plan further into the future.

The MCTS agents used in this assessment were partially based of the winner of the Physical Travelling Salesman Competition[16]. The exact implementation for this problem included macro-actions of length 3[1].

These agents were made to compete in a round-robin tournament for all three game variations over 30 trials. During the tournament, the score, remaining missile count, and velocity of each agent was logged, with the results analysed at the end. Comparisons between like agents (e.g. MMMCTS versus MMMCTS) were not computed in order to save time. Plots of the score and missile count averaged over all runs for each agent can be seen in Figures 5, 6, and 7.

C. Results

Looking at the performance of the agents, the results are remarkably similar to our expectations. The EmptyController agent always performed the worst, managing the occassional win against agents that were unlucky in hitting many asteroids, and therefore ending with a negative score. The RotateAndShoot agent performed slightly

[1]In order to stop the MCTS agents from quickly expending the limited missile count, any macro-actions that would have included firing for each of the three primitive actions instead only fires for the first, with the thrust and turning values being unchanged.

Fig. 2: AI results over the game without

	Empty Controller	Rotate and Shoot	MCTS (20ns)	MCTS (40ms)
Empty Controller	-	0%	3%	0%
Rotate and Shoot	1000%	-	3%	3%
MCTS (20ns)	97%	97%	-	30%
MCTS (40ns)	100%	97%	70%	-

Fig. 3: AI results over the game with simple asteroids

	Empty Controller	Rotate and Shoot	MCTS (20ms)	MCTS (40ms)
Empty Controller	-	0%	0%	0%
Rotate and Shoot	100%	-	20%	6%
MCTS (20ms)	100%	80%	-	57%
MCTS (40ms)	100%	93%	44%	-

better, more often than not beating the EmptyController agent, but still losing to the MCTS agents. The MCTS agents were dominant compared to the non-MCTS agents, and the one given a larger computational budget ended up usually outperforming the one with the tighter budget.

Looking at the graphs of score-versus-time for each game mode, we can see that the first game variant (including simple asteroids) resulted in the simpler agents ending with a slightly negative score, with this effect being significantly stronger in the second game variant (featuring splitting asteroids). This is inkeeping with what we expected, although the effect is slightly smaller than we had hoped.

Comparison of the average velocity for the MCTS agents for each of the game variations suggests that the two variants including asteroids resulted in the agents moving at a higher average speed. This is as we had expected, and could suggest that in this respect the inclusion of asteroids had the desired impact.

V. Human Playtesting

A. Methodology

32 participants were recruited through opportunity sampling (21 male and 11 female; 29 right handed, 2 left handed and 1 ambidextrous; ages ranging between 15 to 61 with $\mu = 38.66$ and $\sigma = \pm 13.55$). All participants provided written consent prior to the experiment and were properly debriefed about the purposes of the

Fig. 4: AI results over the game with splitting asteroids

	Empty Controller	Rotate and Shoot	MCTS (20ms)	MCTS (40ms)
Empty Controller	-	3%	0%	0%
Rotate and Shoot	97%	-	3%	0%
MCTS (20ms)	100%	97%	-	30%
MCTS (40ms)	100%	100%	70%	-

Fig. 5: Average metrics for AI players over the game without asteroids

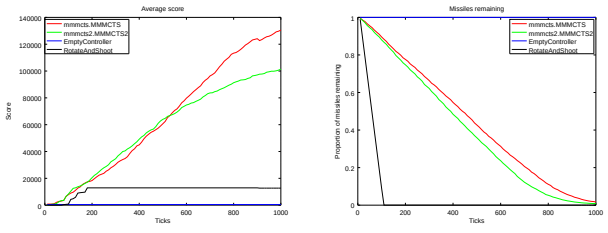


Fig. 6: Average metrics for AI players over the game with simple asteroids

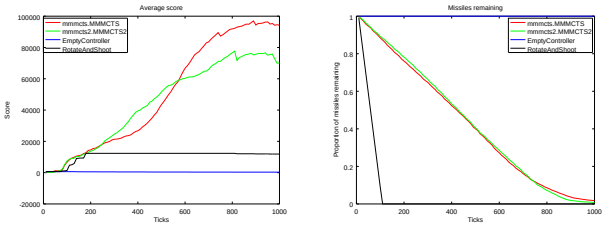
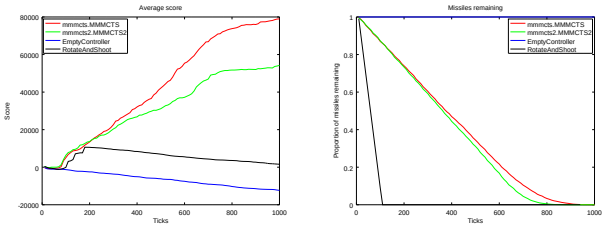


Fig. 7: Average metrics for AI players over the game with splitting asteroids



experiment afterwards. A questionnaire was created to record the participants' responses. No sensitive information was asked, and the data recorded was fully anonymised.

Participants played four trials of each version of Space Battle.

After each version of the game was played participants were asked to answer questions regarding that specific version. For counterbalancing we created 6 subconditions (ABC, ACB, BAC, BCA, CAB, CBA) in order to account for possible order-fatigue effects. These procedures were created to reflect best practices in research design.

B. Results

In order to decide whether parametric or non-parametric tests should be performed, tests of normality were performed. Both the Kolmogorov-Smirnov and Shapiro-Wilk tests suggest that our data significantly deviates from the normal distribution ($p < 0.001$), meaning that non-parametric tests should be used.

Using Friedman's Test were were unable to find any significant results comparing the variations introduced into the game and the player's enjoyment of it ($\chi^2(2) = 3.06$, $p \gg 0.05$).

VI. Future Work

The researchers acknowledge the limited generalizability of our results due to the small number of participants. Future studies should focus on the more detailed and precise measurement of participants game enjoyment which was beyond the scope of this paper.

This study does show that automated AI methods and analytics can be very helpful to assessing, at least at a broad level, how good, or bad, a game mode can be. By immediately discarding, without requiring human input, all the bad variations of a game, designers can focus on tweaking the ones automated testing deem as good. This is far from a completely tapped out research area, as digitizing the human factor is not a solved task.

VII. References

[1] A. Isaksen, D. Gopstein, and A. Nealen, "Exploring game space using survival analysis," *Foundations of Digital Games*, 2015.

[2] A. Isaksen, D. Gopstein, J. Togelius, and A. Nealen, "Discovering unique game variants paper type: Full paper," *Computational Creativity and Games Workshop*, 2015.

[3] M. J. Nelson, J. Togelius, C. B. Browne, and M. Cook, "Rules and mechanics," *Procedural Content Generation in Games*, 2015.

[4] M. Cook and S. Colton, "Multi-faceted evolution of simple arcade games.," in *CIG*, 2011, pp. 289–296.

[5] G. N. Yannakakis, H. H. Lund, and J. Hallam, "Modeling children's entertainment in the playware playground," in *Computational Intelligence and Games, 2006 IEEE Symposium on*, IEEE, 2006, pp. 134–141.

[6] K. Procci, A. R. Singer, K. R. Levy, and C. Bowers, "Measuring the flow experience of gamers: An evaluation of the dfs-2," *Computers in Human Behavior*, vol. 28, no. 6, pp. 2306–2312, 2012.

[7] J. Ibáñez and C. Delgado-Mata, "Adaptive two-player videogames," *Expert Systems with Applications*, vol. 38, no. 8, pp. 9157–9163, 2011.

[8] E. Donchin, M. Fabiani, and A Sanders, "The learning strategies program: An examination of the strategies in skill acquisition," *Acta Psychologica, 71*, 1989.

[9] W. R. Boot, "Video games as tools to achieve insight into cognitive processes," *Frontiers in psychology*, vol. 6, 2015.

[10] P. Rabbitt, N. Banerji, and A. Szymanski, "Space fortress as an iq test? predictions of learning and of practised performance in a complex interactive video-game," *Acta Psychologica*, vol. 71, no. 1, pp. 243–257, 1989.

[11] K. B. Lyle, D. P. McCabe, and H. L. Roediger III, "Handedness is related to memory via hemispheric interaction: Evidence from paired associate recall and source memory tasks.," *Neuropsychology*, vol. 22, no. 4, p. 523, 2008.

[12] V. Cazzato, D. Basso, S. Cutini, and P. Bisiacchi, "Gender differences in visuospatial planning: An eye movements study," *Behavioural brain research*, vol. 206, no. 2, pp. 177–183, 2010.

[13] J. W. Stigler, S.-Y. Lee, and H. W. Stevenson, "Digit memory in chinese and english: Evidence for a temporally limited store," *Cognition*, vol. 23, no. 1, pp. 1–20, 1986.

[14] D. I. Templer and J. S. Stephens, "The relationship between iq and climatic variables in african and eurasian countries," *Intelligence*, vol. 46, pp. 169–178, 2014.

[15] J. Greiner, M. A. Schoenfeld, and J. Liepert, "Assessment of mental chronometry (mc) in healthy subjects," *Archives of gerontology and geriatrics*, vol. 58, no. 2, pp. 226–230, 2014.

[16] E. Powley, D. Whitehouse, and P. Cowling, "Monte carlo tree search with macro-actions and heuristic route planning for the physical travelling salesman problem," in *Computational Intelligence and Games (CIG), 2012 IEEE Conference on*, 2012, pp. 234–241. DOI: 10.1109/CIG.2012.6374161.