How Will the Introduction of a Mixed-Initiative Component that Predict User Requirements Affect the Size and Speed of the Levels Created?

Tristan Barlow-Griffin

***Abstract*—The abstract goes here.**

I. Introduction

THIS research project will look whether a prototyping tool that predict user requirements will increase the size and speed of levels designed. A prototype is the initial design of an object [1]. The prototyping phase of a project is used to quickly test certain aspect of a products' design so the designer can identify and clear up any problems[2]. In [3, p. 150] the author claims there are two kinds of prototyping in games: Physical and Software prototypes. Since book was published back in 2004, the accessibility of software tool to help prototyping has increased. The author of [3, p. 164] also describes level editors as a good way to prototype levels. *Unreal Engine 4* (UE4) implement their own version of a level editor. Within this editor the designers can create basic geometry scaling them to fit their needs as well as addition custom meshes and programmable objects.

This paper looks to build upon a normal level editor by adding a Mixed-Initiative component that will predict the users requirements. The component will aim to reduce the time it takes to produce a level prototype. As discussed above the prototyping phase is meant to test a design, the less time and resources required to produce an artefact that can demonstrate the proposed design the better. Beyond the benefit of saving time, the less time a designer puts into a particular design the less attached to the design they become. When collaborating in a group, differing opinions can cause different constraints to be set on the design of a level. While a given design may satisfy the original designers set constraints, the prototype may have to be discarded as it did not meet the other requirements set by the team. Identifying and discarding concepts early in development can save a lot of time and energy [4, p.489] and arguable may reduce the negative impacts to interpersonal relations that idea dismissal may have.

II. Related Work

The focus on this paper is how the mixed-initiative tool will interact with the user. The aim is to discover if the tool will supplement the designer in such a way that they will increase the their normal output of levels. As a result the main focus of this literature review will be on prediction methods. For the research into prediction methods the scope went beyond just game design as their were limited cases of prediction methods to be found. The definition of mixed-initiative used in this paper will also be outlined with the category of mixed-initiative tool to be used defined as by [5] definition. This literature review will also look at mixed-initiative editors already published using their results to refine the design of interface. In Section II-A there is a breakdown of the broad categories of mixed-initiative.

A. Mixed-initiative

The term mixed-initiative was first introduced by Jaime R [6]. It describes a process where by a computer and a human designer work together to achieve a goal. The first instance of mixed-initiative was a tool to help students learn the English language. The uses of mixed-initiative tools has greatly expanded since 1970 and has come a long way. The two broad categories that MI tools can be grouped into are: Interactive evolution and Computer-aided design [5].

- *Interactive evolution*(IE) is where the designer has the idea and the computer helps them realise it. The computers role is to evaluate the humans design, presenting alternative solutions if any constraints are broken.
- *Computer-aided design* (CAD) is where the computer generates the content, but does not evaluate the quality of the produced work. Instead, a human designer will evaluate the work and use the evaluations to move towards a more desirable product space.

The field of procedural content generation has advanced significantly [**?**], these uses of CAD are ever increasing as publishers seek to lower costs of production[**?**], [**?**]. As the field grows in popularity the research into it grows also. Using CAD may increase the quantity and variation of levels produced ensuring replayability [18]. These algorithms are the topics of large research papers, with in depth analysis to how re-playable they make a game. From reading within the field it can be argued that all of the instances of CAD require a programmer of modest ability to implement the generation algorithms. As a result, when prototyping levels, generations algorithms are not usually employed unless a programmer can be spared. As discussed above in Section **??** the shorter time prototyping the better. This is where an IE tool may prove to be more helpful than CAD algorithm. Doran *et al* [21] describes a procedural content generation algorithm should ideally have a set of designer centric parameters. Even with parameters available the core of the creative process will be on the computer.

Unlike CAD, in IE, the core of the creative process relies on the human designer. As the main creative driver the human has the most input, with the computer ideally providing supplementary support. It can be argued as the constraints are determined more by the human the size of the possible output space will be larger. Allowing for more flexibility of the MI tool.

In [7] the author proposed 12 critical factors to take into consideration when making a mixed-initiative user interface. Since the papers publication in 1999 the power of machines have improved considerably, meaning some of the computation costs described in the publication are less of a problem. The authors' list of critical factors focus on an mixed-initiative assistant for Microsoft Outlook (emailing software). The first factor that is listed is that a MI tool needs to add significant value through the automation of services. An Examples of a services automated given by [7] is the sorting of a users emails into different categories. The authors of [9] have created a design tool that allows users to create levels using a low resolution graphical interface. Within in the context of [9] they satisfy [7] first critical factor by allowing the computer to automate the map design service by generating alternate maps based on user created ones. In this example, the computer takes on a creative role all be it based on an original human designed map. Within this project the focus will not be on the creative aspect as the definition of creativity is hard to for a computer to understand [8]. It can be argued that an MI tool could not consistently add value as it can not understand the designers creative vision.

Another factor raised by [7] is that an MI tool must consider uncertainty about the users goals. Even with an extensive history of user goals, novel goals might be implemented during this time the system will benefit from the understanding it cannot predict what the user is trying to accomplish. Once more [9] propose a way around that through a novelty search algorithm. Where by the MI tool seeks new search spaces as opposed to existing established areas.

Within this project, the MI tool will fall under the category of interactive evolution as it will work during the design phase suggesting alternative as supplementing a designers creation process.

B. Prediction Methods

The researchers of [10] tested alternate methods for predicting human input so as to abstract the low-level movements of the robots the humans were controlling. They built on the idea that humans are good for high-level abstract tasks, but an AI agent was much better at performing low-level repetitive control tasks. They also found that when trying to predict the input the human would do next, trying to identify patterns in a history of inputs was far less successful at predicting the humans intention than just using the last input given by the human. Instead of using current human inputs, [11] used the history of the humans social media page to predict the users interests. Perhaps if the authors of [10] had looked less at the input history of the human and instead focused on grouping inputs together to create larger actions. Similar, to how modern day phones often predict entire sentences rather than just single words.

Predictive texting increases the average message length users send to each other [12] as well as the speed the words are written [13]. The same theory may apply to game design. If patterns to a users game design are established. an AI system may be able to assist in design. This tool may increase the size of the levels the designers may produce and reduce the production time.

III. Proposal

The experiment proposed in this studies involves...

IV. References

A. Mixed-initiative interaction[14]

Link : See slack ED

B. Fostering creativity in the mixed-initiative evolutionary dungeon designer[15]

Link

C. Mixed-initiative procedural generation of dungeons using game design patterns[16]

Link

D. Cellular automata for real-time generation of infinite cave levels[17]

Link

E. Mixed-initiative design of game levels: Integrating mission and space into level generation.[18]

Link

F. Experience-driven procedural content generation[19]

Link

G. Evaluating collaborative filtering recommender systems[20]

Link

H. Controlled Procedural Terrain Generation Using Software Agents[21]

Link

I. Applications of Intelligent Agents[22]

Link

REFERENCES

[1] A. H. Blackwell and E. Manar, "Prototype." accessed 12 Nov. 2018. [Online]. Available: http://link.galegroup.com/apps/doc/ENKDZQ347975681/SCIC?u=dclib_main&sid=SCIC&xid=0c8f739d

[2] R. Budde, K. Kautz, K. Kuhlenkamp, and H. Züllighoven, "Prototyping," in *Prototyping*. Springer, 1992, pp. 33–46.

[3] T. Fullerton, C. Swain, and S. Hoffman, *Game design workshop: Designing, prototyping, & playtesting games*. CRC Press, 2004.

[4] J. Stempfle and P. Badke-Schaub, "Thinking in design teams-an analysis of team communication," *Design Studies*, vol. 20, no. 5, pp. 439–452, 1999.

[5] A. Liapis, G. Smith, and N. Shaker, "Mixed-initiative content creation," in *Procedural content generation in games*. Springer, 2016, pp. 195–214.

[6] J. R. Carbonell, "Mixed-initiative man-computer instructional dialogues." BOLT BERANEK AND NEWMAN INC CAMBRIDGE MASS, Tech. Rep., 1970.

[7] R. Van der Linden, R. Lopes, and R. Bidarra, "Designing procedurally generated levels," in *Proceedings of the the second workshop on Artificial Intelligence in the Game Design Process*, 2013.

[8] L. Doherty, M. Whitney, J. Shakya, M. Jordanov, P. Lougheed, D. Brokenshire, S. Rao, S. Menon, and V. Kumar, "Mixed-initiative in computer games: Algorithmic content creation in open-ended worlds," in *AAAI Fall Symposium on Mixed-Initiative Problem-Solving Assistants*, 2005, pp. 46–56.

[9] J. M. Font, R. Izquierdo, D. Manrique, and J. Togelius, "Constrained level generation through grammar-based evolutionary algorithms," in *European Conference on the Applications of Evolutionary Computation*. Springer, 2016, pp. 558–573.

[10] D. Karavolos, A. Bouwer, and R. Bidarra, "Mixed-initiative design of game levels: Integrating mission and space into level generation." in *FDG*, 2015.

[11] J. Doran and I. Parberry, "Controlled procedural terrain generation using software agents," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 2, no. 2, pp. 111–119, 2010.

[12] E. Horvitz, "Principles of mixed-initiative user interfaces," in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM, 1999, pp. 159–166.

[13] A. Liapis, G. N. Yannakakis, and J. Togelius, "Sentient sketchbook: Computer-aided game level authoring." in *FDG*, 2013, pp. 213–220.

[14] A. Jordanous, "Defining creativity: Finding keywords for creativity using corpus linguistics techniques." in *ICCC*, 2010, pp. 278–287.

[15] R. Chipalkatty, G. Droge, and M. B. Egerstedt, "Less is more: Mixed-initiative model-predictive control with human inputs," *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 695–703, 2013.

[16] V. Bhatia and V. Hasija, "Targeted advertising using behavioural data and social data mining," in *Ubiquitous and Future Networks (ICUFN), 2016 Eighth International Conference on*. IEEE, 2016, pp. 937–942.

[17] R. Ling, "The length of text messages and the use of predictive texting: Who uses it and how much do they have to say," *Association of Internet Researchers, Chicago, IL*, 2005.

[18] M. D. Dunlop and A. Crossan, "Predictive text entry methods for mobile phones," *Personal Technologies*, vol. 4, no. 2-3, pp. 134–143, 2000.

[19] J. Allen, C. I. Guinn, and E. Horvtz, "Mixed-initiative interaction," *IEEE Intelligent Systems and their Applications*, vol. 14, no. 5, pp. 14–23, 1999.

[20] A. Alvarez, S. Dahlskog, J. Font, J. Holmberg, C. Nolasco, and A. Österman, "Fostering creativity in the mixed-initiative evolutionary dungeon designer," in *Proceedings of the 13th International Conference on the Foundations of Digital Games*. ACM, 2018, p. 50.

[21] A. Baldwin, S. Dahlskog, J. M. Font, and J. Holmberg, "Mixed-initiative procedural generation of dungeons using game design patterns," in *Computational Intelligence and Games (CIG), 2017 IEEE Conference on*. IEEE, 2017, pp. 25–32.

[22] L. Johnson, G. N. Yannakakis, and J. Togelius, "Cellular automata for real-time generation of infinite cave levels," in *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*. ACM, 2010, p. 10.

[23] G. N. Yannakakis and J. Togelius, "Experience-driven procedural content generation," *IEEE Transactions on Affective Computing*, vol. 2, no. 3, pp. 147–161, 2011.

[24] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 5–53, 2004.

[25] N. R. Jennings and M. Wooldridge, "Applications of intelligent agents," in *Agent technology*. Springer, 1998, pp. 3–28.

APPENDIX A
FIRST APPENDIX

Appendices are optional. Delete or comment out this part if you do not need them.