

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]. Fullerton *et al* [3, p. 150] state there are two kinds of prototyping in games: Physical and Software prototypes. Since this book was published back in 2004, the accessibility of software tools 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 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. To discover whether mixed-initiative tools really have a place in level design the selection of the prediction method must be optimal. When researching prediction methods scoping considerations have been taken into account. Some prediction methods have been

the centre of studies with far more resources than this project. Prediction methods referenced are not cutting edge in the field. The definition of mixed-initiative used in this paper will also be defined in Section II-A. The category of mixed-initiative tool to be used defined as by [5] definition, this definition is used by other researchers [6]. This literature review will also look at mixed-initiative editors already published using their results to refine the design of interface.

A. Mixed-initiative

The term mixed-initiative was first introduced by Jaime R [7]. 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 have been described as a backbone tool for designers [6].

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 [8], these uses of CAD are ever increasing as publishers seek to lower costs of production [9], [10]. 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 [11]. 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 I the shorter time prototyping the better. This is where an IE tool may prove to be more helpful than CAD algorithm. Even with parameters available the core of the creative process will be on the

computer. Within an IE environment 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. Yanakakis *et al* [12] claim that CAD examples like PCG limit the designers intentions as they follow their own algorithms which ideally should have a set of designer centric parameters as their only form of control [13].

B. Mixed-Initiative Level Designers

A user interface should be intuitive to use and not require any additional help systems [14]. Liapis *et al* [15] aim to achieve this by creating a design tool that allows users to create levels using a low resolution graphical interface. Figure 1 shows their interface used to design levels for a strategy game. The designer can place tiles on the map which will colour in the given tile with the placed tiles colour. During a placement the tool will test for the maps playability, checking to see if placing the current tile will break any of the games constraints. The Sentient sketchbook also provides alternative viewing modes, examples of these alternate viewing models can be seen in Figure 2. There is no mention in the study how often these tools were used, it can be argued that the virtualisation while nice, just added needlessly to skill required to use this tool effectively.

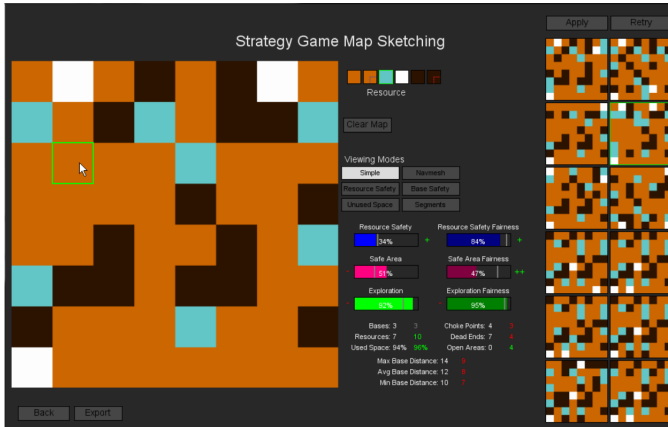


Fig. 1. A. Liapis *et al* Sentient Sketchbook during a design session [15].

Horvitz *et al* [16] propose 12 critical factors to take into consideration when making a mixed-initiative user interface. While Horvitz *et al* [16] focuses on an mixed-initiative assistant for Microsoft Outlook (emailing software) it can be argued that some of these factors are relevant for level design. 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 by emailing assistant is the sorting of a users emails into different categories. Within in the context of [15] they satisfy [16] first critical factor by allowing the computer to automate some of the map design services like checking for broken game constraints. Liapis *et al* [15] also allow their algorithm take on a creative role all be it based



Fig. 2. A. Liapis *et al* Sentient Sketchbook Different viewing modes [15].

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 [17]. 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 [16] is that an MI tool must consider minimizing the costs of poor guesses 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. Some authors find value in these missed guesses and even seek to find novelty search spaces [15]. Other authors [6], [12], [18] claim these kind of mistakes can foster creativity and alternate suggestions that do not aim to predict the user can be beneficial. On the right hand side of Figure 1 is where the guessing algorithms results are shown. Clicking retry will quickly remove the selected map and create a new one.

C. Prediction Methods

Predictive texting increases the average message length users send to each other [19] as well as the speed the words are written [20]. 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. However, the principle with working with predictive texting is different to that of game design.

The researchers of [21] 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, [22] used the history of the humans social media page to predict the users interests. Perhaps if the authors of [21] 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.

III. PROPOSAL

The experiment proposed in this studies involves...

REFERENCES

- [1] A. H. Blackwell and E. Manar, "Prototype." accessed 12 Nov. 2018. [Online]. Available: http://link.galegroup.com/apps/doc/ENKQZQ347975681/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] A. Alvarez, S. Dahlkog, 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.
- [7] J. R. Carbonell, "Mixed-initiative man-computer instructional dialogues." BOLT BERANEK AND NEWMAN INC CAMBRIDGE MASS, Tech. Rep., 1970.
- [8] 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.
- [9] L. Doherty, M. Whitney, J. Shakya, M. Jordanov, P. Loughheed, 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.
- [10] 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.
- [11] D. Karavolos, A. Bouwer, and R. Bidarra, "Mixed-initiative design of game levels: Integrating mission and space into level generation." in *FDG*, 2015.
- [12] G. N. Yannakakis, A. Liapis, and C. Alexopoulos, "Mixed-initiative co-creativity." in *FDG*, 2014.
- [13] 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.
- [14] R. Oppermann, "User-interface design," in *Handbook on information technologies for education and training*. Springer, 2002, pp. 233–248.
- [15] A. Liapis, G. N. Yannakakis, and J. Togelius, "Sentient sketchbook: Computer-aided game level authoring," in *FDG*, 2013, pp. 213–220.
- [16] 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.
- [17] A. Jordanous, "Defining creativity: Finding keywords for creativity using corpus linguistics techniques." in *ICCC*, 2010, pp. 278–287.
- [18] A. Liapis, G. N. Yannakakis, C. Alexopoulos, and P. Lopes, "Can computers foster human users' creativity? theory and praxis of mixed-initiative co-creativity."
- [19] 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.
- [20] M. D. Dunlop and A. Crossan, "Predictive text entry methods for mobile phones," *Personal Technologies*, vol. 4, no. 2-3, pp. 134–143, 2000.
- [21] 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.
- [22] 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.

APPENDIX A FIRST APPENDIX

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