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

Tristan Barlow-Griffin

***Abstract*—This paper builds upon a feature request by users in Alvarez *et al* [1] study into a mixed-initiative level design tool.**

I. Introduction

THIS research project will look whether a prototyping tool that predicts user requirements will increase the size and speed of levels designed. A prototype is the initial design of an object [2]. The prototyping phase of a project is used to quickly test a certain aspect of a products' design so the designer can identify and clear up any problems [3].Fullerton *et all* [4, 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 tools that helps designers prototype has increased. Fullerton *et al* [4, p. 164] also describes a level editor as a good way to prototype levels. The free game engine called *Unreal Engine 4* (UE4) has a level editor built into it. Within this editor, the designers can create basic geometry scaling them to fit their needs. In addition, designers can add custom meshes and programmable objects.

This paper will build upon a normal level editor by adding a Mixed-Initiative(MI) component. This MI component will predict the users' requirements. The MI tool seeks to reduce the time taken to design a level. The prototyping phase will test the design of a product. 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 product. 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 [5, p.489]. Arguably, this will 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 there were limited cases of prediction methods to be found. The method of prediction must be accurate for the hypothesis to be tested accurately. The range of complexity of prediction methods is large, for scoping reasons, this paper will focus on simple prediction methods. Most prediction methods have been the centre of studies with far more resources than this project as such. The definition of mixed-initiative used in this paper will be defined in Section III. In Section III mixed-Initiative tools are group into two broad categories using Liapis *et al* [6] definition. Defining these groups makes it easier to distinguish between the most common type of mixed-initiative tool. In Section IV there is an evaluation of the current state of mixed-initiative level designers being used. This paper uses knowledge in this section to identify features that if implemented could help to speed up the prototyping phase.

III. Mixed-initiative

The term mixed-initiative was first introduced by Jaime R [7]. It describes a process whereby a computer and a human designer work together to achieve a goal. Other definitions of mixed-initiative build on the idea of human-computer co-creativity. This paper will use the first definition of mixed-initiative presented. As it is easier to define and the definition of creativity is a very complex matter in itself.

There are two broad categories that MI tools can be grouped into, Interactive evolution and Computer-aided design [6].

- *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. In CAD, the human designer will evaluate the work and use these evaluations to move towards a desirable product space.

The first mixed-initiative tool created helped students to learn the English language. The uses of mixed-initiative tools have greatly expanded since 1970 and have been described as a backbone tool for designers [1]. The application of mixed-initiative systems in more complex environments have given mixed results [8]. Barnes *et al* [8] found that in most cases, systems that divide decision making between a human and an intelligent agent were generally more effective than if decisions were just dependent on the one. This can be seen with predictive texting, only choosing the words suggested to you by your phone can result in unexpected and hilarious results [9]. Barnes *et al* [8] found that the humans were better at making abstract decisions and inferring the significance of an object or event. Kantosalo *et al* [10] focus their MI tool on user-centred design, this means their AI agent played a back-seat role. Kantosalo *et al* [10] propose future work where the agent and the designer have an equal role in the system, they do not make any conjecture on the anticipated results.

The field of procedural content generation has advanced significantly [11], the uses of PCG are ever increasing as publishers seek to lower costs of production [12], [13]. PCG and CAD are closely related fields, the difference between CAD and PCG is in the evaluation period. If a designer were to use a PCG algorithm to generate a level and then look through the produced results evaluating and picking maps this would be considered CAD [6]. For it not to be considered CAD, the PCG algorithm would have to perform the evaluation itself. For example, this might include checking if the map is completable or is of a certain size.

One may choose to use PCG in games as it will increase the quantity and variation of levels produced ensuring replayability [14]. PCG algorithms can also be shipped with the games they are made for, this allows for an inexhaustible source of new maps [15]. Doing so will extend the games life-span giving the players an amount of content that would otherwise be impossible. Although there is no guarantee that this content will be interesting or unique. However, Using CAD tools to generate maps there will always be a human element evaluating the maps. With this input the human designer may discard maps that are similar to existing maps, thus providing quality assurance not present in a PCG algorithm.

Prototyping a level should be a fast iterative process [16], the method used to prototype should allow for instant feedback on the design. This feedback needs to have a channel back to the design so that amendments can be made easily. Map generation algorithms, even in under ideal circumstance will only provide the designer with a set of parameters to change [17]. Small variations in the given parameters can produce large changes in the maps design [18]. This will make it difficult to make small changes when given feedback. As a result, reduce the effectiveness of a CAD approach when trying to prototype levels.

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. Ideally, the computer will add value by 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* [19] claim that CAD examples like PCG limit the designers' intentions as they follow their own algorithms. Doran *et al* [17] inadvertently corroborate Yanakaki*et al* [19] view, as they say, a PCG algorithm should ideally, have a set of designer-centric parameters as their only form of control. This limited control over the creation of levels may, in some situations, be enough. During the prototyping phase, when the requirements needed from a map are changing the fastest, the level of control provided through CAD would not be enough.

IV. Mixed-Initiative Level Designers

A user interface should be intuitive to use and not require any additional helping systems [20]. Liapis *et al* [21] aim to achieve this by creating a design tool that allows users to create levels using a low-resolution graphical interface. Figure 1 shows the 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 tile placement, the tool will test for the playability of the map, 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, Galtiz [22, p. 752] warns against too many graphics on the screen.

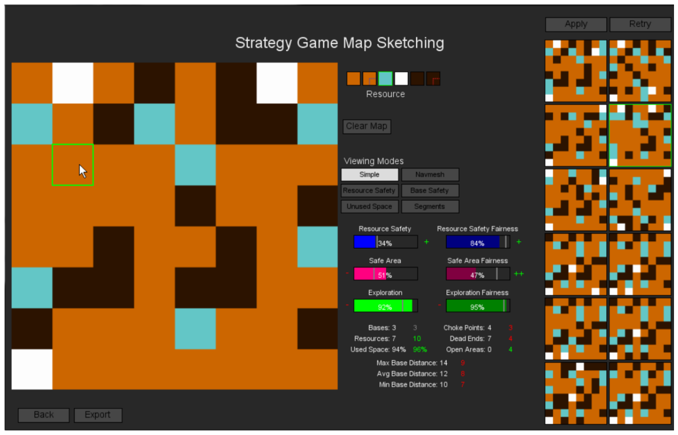


Fig. 1. Liapis *et al* Sentient Sketchbook during a design session [21].

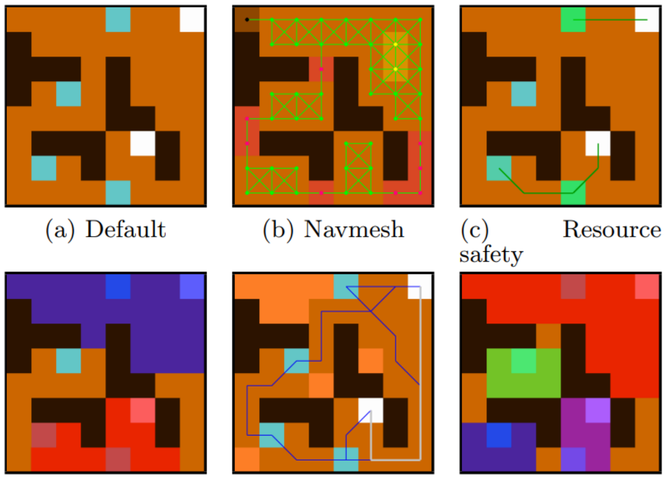
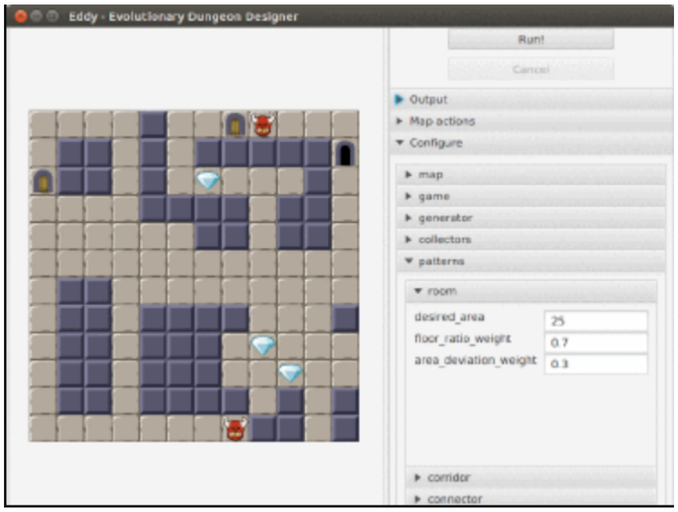


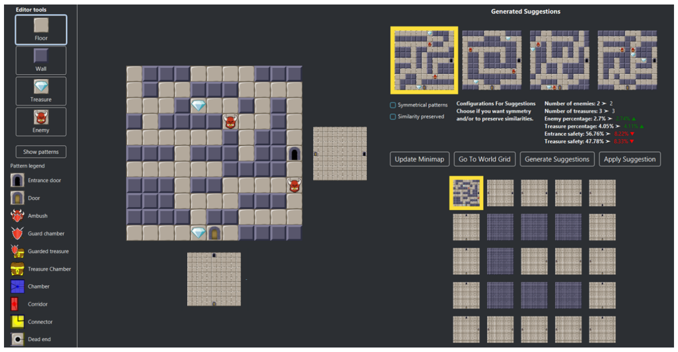
Fig. 2. Liapis *et al* Sentient Sketchbook Different viewing modes [21].

Baldwin *et al* [23] have also implemented a mixed-initiative dungeon designer called the evolutionary dungeon designer or (EDD) for short. EDD is closer to a CAD tool than the IE tool Sentient Sketchbook [21]. While the approaches may be different see Figure 3, both MI tools allow for large customisation of the levels generated see Figure 3. Baldwin *et* *al* [23] take a different approach to Liapis *et al* [21]. Baldwin *et al* [23] core concept is to identify design patterns within the level design. These design patterns consist of multiple tiles that constitute common patterns found in games. Alvarez *et al* [1] builds upon the EDD suggest in [23] by adding a IE element to it. It could be argued that the new version of the EDD has an improved interface with lots of its excess drop-down menus gone see Figure 4. Beyond the ascetic differences, Alvarez *et al* [1] dungeon designer integrates key aspects of the Sentient Sketchpad [21]. The second edition of the EDD allows the user to design their own levels, it will then offer suggestions based up the map the user created, this can be seen the top right corner of figure 4. The results from Alvarez *et al* [1] experiment focused on whether their tool fostered creativity in the participants using it. Included with the results is a table of requested features made by the participants of the study. One key element highlighted is that the EDD should do a " bit more automated assistance when doing manual designs, which can reduce clicking around the program" [1, Table 2]. Another significant feature request, was for the dungeon designer to take into account the pattern of the entire dungeon. Using the map of the entire dungeon to generate new rooms.



desired_area	25
floor_ratio_weight	0.7
area_deviation_weight	0.3

Fig. 3. Baldwin *et al* Evolutionary Dungeon Designer user interface [23].



Configurations For Suggestions
Choose if you want symmetry and/or to preserve similarities.

Number of enemies: 2 ➢ 2
Number of treasures: 3 ➢ 3
Enemy percentage: 2.7% ➢ 2.74% ▲
Treasure percentage: 4.05% ➢ 4.11% ▲
Entrance safety: 56.76% ➢ 8.22% ▼
Treasure safety: 47.78% ➢ 8.33% ▼

Fig. 4. Alvarez *et al* Evolutionary Dungeon Designer with modifications user interface [1].

Horvitz *et al* [24] propose 12 critical factors to take into consideration when making a mixed-initiative user interface. While Horvitz *et al* [24] focuses on a 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 an MI tool needs to add significant value through the automation of services. An Examples of a service automated by emailing assistant is the sorting of a user's emails into different categories. Within in the context of [21] they satisfy [24] first critical factor by allowing the computer to automate some of the map design services like checking for broken game constraints. Liapis *et al* [21] also allows their algorithm to take 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 [25]. Alvarez *et al* [1] found their MI tool is better at providing controllability than expressivity, when the user imposes their vision, as it is hard for a computer to capture the designers' vision. It can be argued that an MI tool could not consistently add value if it cannot capture the designers' vision. Smith *et al* [16] believe that human designers strengths lay in creativity and their ability to evaluate good content and the computer lacks this ability.

Another factor raised by Horvitz [24] is that a tool must consider minimizing the costs of poor guesses about the users' goals. Even with an extensive history of the users' requirments, a novel use of the tool might be required. It is important for a system to recognise when something novel is happening and for it not to attempt predictions. Some authors find value in these missed guesses and even seek to find novelty search spaces [21]. Other authors [1], [19], [26] 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 the results of the guessing algorithm are shown. Clicking retry will remove the current maps and create new ones for the designer to evaluate.

None of the above examples [1], [21], [23] satisfy Barnes *et al* [8] statement that a mixed-initiative systems UI must provide transparency to the reasoning behind the agents actions. In all cases above [1], [21], [23] , when generating new suggestions the reasoning behind each suggestion was not given to the designer. The designer is presented with the statstics of the current map generated(density, number of resources ETC) but it is not clear from the interface how these statistics are being used. When automating a system transparency is required for a human to trust the automation process [27].

V. Prediction Methods

Predictive texting increases the average message length users send to each other [28] as well as the speed the words are written [29]. 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. In this section, this paper will look into different methods of predicting human requirements, this paper will also discuss the pros and cons of each technique.

The researchers of [30] 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. In addition, they found that when trying to predict the next human input, trying to identify patterns in a history of events was not successful. Instead, using just the last event yielded much better results, hence their title ”less is more”. Instead of using current human inputs, [31] used the history of the users' social media page to predict the users' interests. Perhaps if the authors of [30] 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.

A. Markov Chains

Markov Chains is a theory similar to the most successful method found by Bhatia*et al* [31]. A Markov chain is a special kind of process that works under the assumption that the state at time *t+1* depends only on the current state. To put it another way, the state at time *t+1* is exclusively dependent on the state at time *t*. This means State *t+1* is not dependent on the history of the states leading up to it [32]. While this technique may be useful for predicting anomalies in systems [32]–[34] where the states are heavily dependent on the latter state happening. It is hard to see how during a creative process where the next state is dependent on the vision of the design state *t+1* can be predicted just from knowing state *t*. The introduction of a Markov model makes this method more viable. A Markov model can be used to describe the probabilistic relationship between the previous states in a Markov Chain [35]. Higher order Markov chains relax the rule of the next state being only dependent on the current state by allowing the network structure to look *n* number of states back from the current state [36]. Snodgrass *et al* [37]used Markov models as a way to model level data .Using these Markov models Snodgrass *et al* [37] applied different sampling techniques, they found using their higher order Markov chains were more effective than using just Markov chains. This conflicts with Chipalkatty *et al* [30] findings of ”less is more”, Snodgrass *et al* [37] found that using a history of states to predict the next state was more effective to achieve their goals.

B. N

N GRAM

Wave Function collapse

Learning to Generate Video Game Maps Using Markov Models

https://github.com/mxgmn/WaveFunctionCollapse

Francois pachet continuator

VI. Methodology

The research question proposed in this project is: How Will a Mixed-Initiative Level Designer that Predict User Requirements Affect the Size and Speed of the Levels Created? A discussion of the hypothesis drawn from this question can be view in Section VI-B

A. Creating levels

The experiment proposed in this paper need at least two dependent variables. Firstly the speed of which a level was created. To measure this, the participants will be restricted in the size of the level they are allowed to create see Table I settings 3 and settings 4. Before this first part of the experiment the users will be explained the rules. There will be a minimum number of tiles required before a level is deemed ”complete”. The designer must place at least one start, one end and one other kind of tile. At least three props will be placed anywhere inside any of the placed rooms. After doing this the level complete button will be available. The size of the grid presented to the user will be *24 x 24*.

TABLE I
Editor Settings

Editor Settings	**Predictive Placements Enables**	**Infinite Level Enabled**
Settings 1	X	X
Settings 2		X
Settings 3	X	
Settings 4		

B. Hypothesis

When creating a level designer the aim is to always increase the ease at which levels could normally be created. For the scope of this paper, we will look not look at how the editor created performs, but how the mixed-initiative tool performs.

References

[1] 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.

[2] 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

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

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

[5] 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.

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

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

[8] M. J. Barnes, J. Y. Chen, and F. Jentsch, “Designing for mixed-initiative interactions between human and autonomous systems in complex environments,” in *Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1386–1390.

[9] S. Ochs, “Just for fun: Show us your best quicktype poetry,” 2014, accessed 12 Nov. 2018. [Online]. Available: https://www.macworld.com/article/2692885/just-for-fun-show-us-your-best-quicktype-poetry.html

[10] A. Kantosalo, J. M. Toivanen, P. Xiao, and H. Toivonen, “From isolation to involvement: Adapting machine creativity software to support human-computer co-creation.” in *ICCC*, 2014, pp. 1–7.

[11] 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.

[12] 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.

[13] 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.

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

[15] 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.

[16] G. Smith, J. Whitehead, and M. Mateas, "Tanagra: Reactive planning and constraint solving for mixed-initiative level design," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, no. 3, pp. 201–215, 2011.

[17] 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.

[18] J. Regier and R. Gresko, "Random asset generation in diablo 3," *Invited Talk, UC Santa Cruz*, 2009.

[19] G. N. Yannakakis, A. Liapis, and C. Alexopoulos, "Mixed-initiative co-creativity." in *FDG*, 2014.

[20] R. Oppermann, "User-interface design," in *Handbook on information technologies for education and training*. Springer, 2002, pp. 233–248.

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

[22] W. O. Galitz, *The essential guide to user interface design: an introduction to GUI design principles and techniques*. John Wiley & Sons, 2007.

[23] 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.

[24] 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.

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

[26] A. Liapis, G. N. Yannakakis, C. Alexopoulos, and P. Lopes, "Can computers foster human users' creativity? theory and praxis of mixed-initiative co-creativity."

[27] J. D. Lee and K. A. See, "Trust in automation: Designing for appropriate reliance," *Human factors*, vol. 46, no. 1, pp. 50–80, 2004.

[28] 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.

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

[30] 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.

[31] 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.

[32] N. Ye *et al.*, "A markov chain model of temporal behavior for anomaly detection," in *Proceedings of the 2000 IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop*, vol. 166. West Point, NY, 2000, p. 169.

[33] W.-H. Ju and Y. Vardi, "A hybrid high-order markov chain model for computer intrusion detection," *Journal of Computational and Graphical Statistics*, vol. 10, no. 2, pp. 277–295, 2001.

[34] R. Gwadera, M. Atallah, and W. Szpankowski, "Markov models for identification of significant episodes," in *Proceedings of the 2005 SIAM International Conference on Data Mining*. SIAM, 2005, pp. 404–414.

[35] A. Markov, "Extension of the limit theorems of probability theory to a sum of variables connected in a chain," 1971.

[36] W.-K. Ching, M. K. Ng, and E. S. Fung, "Higher-order multivariate markov chains and their applications," *Linear Algebra and its Applications*, vol. 428, no. 2-3, pp. 492–507, 2008.

[37] S. Snodgrass and S. Ontanon, "Learning to generate video game maps using markov models," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 9, no. 4, pp. 410–422, 2017.

Appendix A
First appendix

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