Can an Unknowing Participant distinguish between a Multi-Agent system and Human Designed Interiors?

Thomas O'Leary

Abstract—This Proposal is still WIP

Attempt to see if a rooms' interior designed by a Multi-Agent system can be perceived as human designed. Comparing to see if a participant is able to distinguish between the two.

I. INTRODUCTION

Urban open world games such as Grand Theft Auto V [1], The Division [2] and Batman: Arkham Knight [3] have such large built-up areas for players to venture in. However, only a few handpicked buildings in these large cities are accessible and have modelled interiors leaving the others to be blocked off for decorative purposes. This could be resolved by modelling and designing each room in these cities, but this would become incredibly impractical. Other issues with this can lean towards rendering and the storage of such heavily dense areas.

Procedural Generation

Procedural Generation (PCG) refers to automatically creating content using algorithms [4]. PCG has many applications in video games, some notorious examples being the world/cave generation in Minecraft [5], the texture [6] and world generation [7] in Spore [8] and the procedural texture and music generation in .kkrieger [9].

Using PCG, this largely time-consuming task of designing room interiors can be automated. And can possibly help maintain a player's immersion within the game - an issue with this however is that PCG tool's can be seen as boring and repetitive [10, Chapter 2].

Through my literature review, I have found many implementations and techniques of Procedural Interior Generation (PCIG) but only a few of these implementations are compared with Human Designed interiors to check its authenticity. My research is aimed towards testing a PCIG tool against human designed rooms in a user study. More specifically a Multi-Agent system that creates a rooms' interior.

II. LITERATURE REVIEW

My literature review consists of two parts that I believe to be important to my research question. It first describes different implementations of Procedural Interior Generation (PCIG) then explores ways in which Artificial Intelligence (AI) is compared to Humans.

A. Implementations of Procedural Interior Generation

Although PCG has a lot to show and offer in game development, the use of Procedural Interior Generation (PCIG) in games however is scarcely come by.

A game that does use PCIG is Catlateral Damage [11], a small indie game developed by Manekoware where you play as a cat on a destructive rampage in its own house. In 2017, Chris Chung (the developer behind the game) wrote a case study about the level design in his game [10, Chapter 6]. When developing Catlateral Damage, Chung was undecided on how to design the levels and ultimately went for PCIG [10, Chapter 6]. Before the interior decoration can take place, a Squarified Treemap algorithm is used to generate the room layouts and floor plans within the level [12]. Each room generated from this algorithm has an associated data file, containing information such as furniture available and maximum type of each furniture. The furniture objects that can be placed, have physics components attached to allow them to be accurately placed within the level - for this, a Rectangle Packing algorithm [13] is used to place these objects within allocated surface areas on the floor and on top other furniture objects. Concluding the case study, Chung states that most players could not notice that the levels were procedurally generated - although this is a promising statement, Chung has not shown any evidence to back this claim.

On 29th April 2021, Sony Interactive Entertainment published patent US20210121781, titled "AI-Generated Internal Environments Based On External Geometry" [14]. The patents' description goes onto explain a Machine Learning (ML) tool that takes in data from the external structure of a virtual building and generates an interior environment just from this data. Although this is just a patent for an ML tool, this could be the start of PCIG being used more commonly in the Games industry.

Despite there not being many implementations of PCIG in games, there are however a handful of published papers that have used their own techniques to emulate room interiors. I will explain the key elements of these implementations and ==something something conclusion based on each one==.

1) Multi-Agent System: In 2009, *T. Germer, et al.* [15] sought out to procedurally arrange a rooms' furniture in real-time. The aim for this system was to allow the quick generation of a rooms' interior while a player is walking around a building - a live demonstration of this can be viewed on YouTube [16].

This system involves a Multi-Agent (M-A) based solution where each furniture object, in a given room, is seen as an individual agent that seeks a suitable parent furniture object. These agents have custom semantic descriptions to allow them to create different object layouts. Each agent has 3 states:

1) Search
   - All agents start in this state, they begin by searching for possible parent objects - if a parent is found that suits its semantics, the agents' state changes to *Arrange*, if a parent can't be found at all the agent is deleted.

2) Arrange
   - In the *Arrange* state, the agent attempts to position itself with the parent accordingly. Whilst doing so, the Separating Axis Thereom [17] is used to check for collisions - if no collisions are found the agents' state changes to *Rest*. If a collision does occur however, it must attempt to re-position itself with the parent.

3) Rest
   - In the *Rest* state, potential child agents are now able to seek this object as a parent. If the parent moves, the resting agent will move along with it - however if this move results in a collision, its parent is lost and the agents state is changed back to *Search*.

Although a large proportion of the rooms furnishing is handled by the agents themselves, a big drawback is that the system requires a lot of user input before this can happen. Every room must have clear user defined data and every object type must have manually defined semantic descriptions. If starting from scratch this process can take a long time, but each object type only requires a singular semantic description. Some objects with matching behaviors can also share semantics - this creates a lot of flexibility when designing the agents as they are completely autonomous of one another [15]. Another issue with this implementation is that the system is never evaluated based upon how realistic or natural the furniture arrangements are based upon human designs.

2) Rule-Based Layout: A Rule-Based layout approach was proposed in 2009, users would be able to specify what objects can be placed within a layout - these would represent an instance of a class and contain certain rules on how it should be placed [18].

Rules can be defined in multiple ways - they can be associated specifically with an object class or defined in the layout planner. An example of defining a rule with an object class, as told by the authors, is by setting a rule for a sofa to always face an instance of a TV. The layout planner is responsible for sending objects to the solver. The planner can have custom rules to allow it to be applicable to different room layouts (living room, factory floor, waiting room). It also has a backtracking rule that is only triggered if an object of interest is not placeable. If this is the case, the planner would backtrack to place previous objects in different positions to allow this object to be placed.

The solver is given an object from this planner and the current layout. With this, it finds all possible locations for the new object - these locations are based upon the rules of this object and the rules already set in place for existing objects in the layout. The possible locations then take specific parameters into account, such as the amount of clearance an object requires or if an object's area is off limits (for example its bounding box) [18]. A Minkowski Sum [19] is carried out containing these inaccessible areas and removed from the list of possible locations. With this completed, the object is then given a list of all possible locations in the layout it can be placed.
Something good. Something bad.

3) Constraints: P. Henderson, et al. [20] presented a data driven system that learns from the SUNCG [21] database to generate furniture layouts. This database contains over 45000 apartment layouts that are designed by humans, from this database - 2500 models are categorised into 170 furniture object classes. Their system is presented in such a way that it can be left to be fully automated [20], but does allow flexibility with the user allowing them to change constraints within the layout.
These constraints include:

- Room size, shape & type
- Exclusion of Object classes
- Furniture clearance
- Locations of specified furniture
- Locations of doors & windows

A user study was carried out in their paper, where 1400 pairs of layouts are presented to 8 non-experts in an image format [20]. These participants are asked to identify the layout that has a more realistic/natural setting. In each pair one layout is from their system, the other being human designed - the order in which the images are shown is randomised per pair. In this study, both constrained and unconstrained layouts are put to the test. For unconstrained layouts, they were presented in 2 different styles; 1st person and an overhead view. Layouts that were presented in 1st person, were seen to be slightly preferred over the human designs and those presented overhead were seen to be indistinguishable from human designs. Constrained layouts were only presented in an overhead format, but two sets of constraints were used:

   i. Fixed room size & fixed placement of a singular object
   ii. Fixed room size & fixed door/window locations

With the constrained layouts of (i), the model layouts were seen to be indistinguishable from those of human design. Whereas the layouts of (ii), human designs were preferred over the models.
Something good. Something user study. Something bad.

4) Statistical Relationships: In 2011, a PCIG system using statistical relationships was proposed [22] [23]. This system would take in existing relationships from previous working examples (spatial, hierarchical and pairwise relationships) and pass these relationships into a cost function. The spatial relationship represents the objects distance and orientation

to its nearest wall. The hierarchical relationship represents a child/parent relationship between objects - for example a candle (child) placed on a table (parent). The pairwise relationship represents the interaction between different pairs of objects (TV and a sofa), these pairs are subject to each other's distance and orientation. The cost function is used to quantify the realism or functionality of the state of the furniture arrangement. The higher the cost of an object, the higher the priority it takes. There are 5 stages to the cost function - each stage has an individual weighting and adds to the overall cost of the objects' arrangement.

- Accessibility
  - For an object to be "functional", it must be accessible. Each object has a defined *Accessible Space* as well as *Bounding Box*. If another object enters an *Accessible Space*, the cost increases.
- Visibility
  - Certain objects must be viewed from a specific direction - these objects are given a *Viewing Frustum* (some objects include TV's and paintings). Similarly to the Accessibility cost function, whenever another object obstructs a *Viewing Frustum*, the frustums values are passed in and the cost increases.
- Pathways
  - Within the furniture arrangements, pathways are created using *Cubic Bézier curves*. These curves are represented as rectangles in the 3D space. The cost function is similar to that of Accessibility & Visibility, yet in this case the control points of the *Bézier curve* is used as the positional value and applied to the rectangles used to represent the pathway in the arrangement.
- Prior Spatial Relationships
  - The prior Spatial relationship (distance and orientation of the nearest wall) is subtracted from the objects current Spatial relationship.
- Pairwise Relationships
  - Similar to the Prior cost function, the pairwise cost function is defined to subtract the distance and orientation of the paired objects.

To create the arrangement with the cost function, a mixture of Simulated Annealing and the Metropolis-Hastings (M-H) algorithm is used. Simulated Annealing originates from the physical process Annealing used to heat objects to remove defects and slowly bring the object back down into a low-energy state [24] - in this system, it is used for the placement of the furniture. At first the objects are "*heated up*" to allow for more freedom whilst they are arranging until they "*cool down*", with each temperature decrement the cost function is called to evaluate the current iteration of the arrangement. With every iteration (or "*temperature decrement*"), the M-H algorithm [25] is used to compare the previous and the proposed new arrangement.

To see if the use of the cost function did produce furniture arrangements with a realistic/functional state, the system was put to the test in a perceptual study against human designed interiors. 25 volunteers (14 of which stated that they did not have any expertise in interior design) were used in this study and were unaware of its true purpose. Each participant viewed a total of 35 pairs, in each pair containing a synthesized and a human designed arrangement. The participants were told to select the furniture arrangement that they would prefer. The synthesized arrangement would only be considered to have been the victor if the human designs were not shown as the "clear" winners within the results. Of the 35 pairs shown to the volunteers, only 13 synthesized arrangements were seen to be the preferred choice.
==Something good. Something bad.==

B. Artificial Intelligence Compared to Humans

In 1950, a mathematician named Alan Turing proposed the question; "*Can Machines Think?*" [26]. As it is a large question, Turing replaces the question with the idea of a problem, called the "*Imitation Game*" (or now modernly referred to as the *Turing Test* (TT)). This *game* involves 2 participants and a judge. The judge is hidden away from the participants and is informed to identify, through questioning, which participant is a man and which is a woman. In order for this *game* to help answer Turing's question, one participant is replaced with a machine. If the judge chooses the machine at the end of the questioning, it is considered that it can "*think*".

In 2010, a new variation of the TT was proposed by Margaret Boden. This variation considers the TT oriented around artistic creativity, rather than the imitation of intelligence [27]. Art that is considered for this TT must be completely independent of human involvement, it is also stated that computer *assisted* art does not count (the use of software such as Photoshop [28]). For an art program to pass this variation it must:

1) Be indistinguishable from human produced artwork And/Or
2) Be seen having similar aesthetic value to human produced artwork

This TT however does set concern for the concept of creativity as it is very much of personal opinion [27], [29]. Although creativity is thought as indefinable, work that is considered creative can be seen as both appropriate and novel [30].
==paper that compares pcg to art== [31]

III. RESEARCH QUESTION

Through my literature review, there is evidence to show that PCIG should be compared to human designs more frequently to test its authenticity. With this I propose the following research question; **Can an Unknowing Participant distinguish between a Multi-Agent system and Human Designed Interiors?**

A. Hypotheses

1) When unaware, the M-A System is picked more often than Human designed interiors by participants
2) When notified, the participant is not able to distinguish between human and M-A system interiors

IV. Artefact

A Multi-Agent (M-A) system will be created that generates the interior of a room at run time. This system is given a pre-defined room size to generate in and will also have access to pre-made furniture assets. It will be influenced by the work from *T. Germer, et al.* [15] and will be implemented into the Unity game engine (Version 2020.3.12) [32]. ==Something something== See Fig. 1

A. How will I ensure Quality

Unit tests & Acceptance Tests?

B. Why will this answer the Research Question

The M-A system itself will not contribute towards answering the research question directly but will contribute towards the research study that will follow.

V. Research Methodology

A. Experimental Design

A/B test. Paper uses something similar [20].

B. Limitations

Time, resources

C. Sampling Plan

Sample size, sampling method

D. Data management plan

Managing, collecting, & storing data

E. Data Analysis

See Appendix B

F. Ethical Considerations

As the research study requires human participants this creates a medium ethics risk according to the Falmouth University Ethics Board. To facilitate this risk, a Falmouth University ethics form has been completed and will be signed off by the project Supervisor in consultation with the Head of Subject. The artefact itself is of low risk/concern as it will not be used for militarization and participants are able to opt out at any point during their participation. Due to the nature of this study, no personal information about the participants involved will be collected - the European Union's General Data Protection Regulation (GDPR) [33] does not need to be followed. However, to protect the participants rights, the Nuremberg Code will be followed to keep and ensure this research study is ethically sound [34]. All participants will be handed a Participant Information Sheet that details the key information they must know before the study, a consent form is also supplied to ensure they have agreed to participate. Participants are still able to withdraw at any point until submitting data.

COVID-19: At the current state of the pandemic and following the latest Government Guidelines in England [35], participants will be requested to wear a face covering during their participation. All surfaces will be sanitised between usages.

References

[1] Rockstar North, "Grand Theft Auto V," [PC Game] Rockstar Games, Steam, 2013.

[2] Massive Entertainment, "The Division," [PC Game] Ubisoft, Steam, 2016.

[3] Rocksteady Studios, "Batman: Arkham Knight," [PC Game] Warner Bros. Interactive Entertainment, Steam, 2015.

[4] J. Togelius, E. Kastbjerg, D. Schedl, and G. N. Yannakakis, "What is procedural content generation? mario on the borderline," in *Proceedings of the 2nd International Workshop on Procedural Content Generation in Games*, ser. PCGames '11. New York, NY, USA: Association for Computing Machinery, 2011. [Online]. Available: https://doi.org/10.1145/2000919.2000922

[5] Mojang, "Minecraft," [PC], 2011.

[6] D. g. DeBry, H. Goffin, C. Hecker, O. Quigley, S. Shodhan, and A. Willmott, "Player-driven procedural texturing," in *ACM SIGGRAPH 2007 Sketches*, ser. SIGGRAPH '07. New York, NY, USA: Association for Computing Machinery, 2007, p. 81–es. [Online]. Available: https://doi.org/10.1145/1278780.1278878

[7] K. Compton, J. Grieve, E. Goldman, O. Quigley, C. Stratton, E. Todd, and A. Willmott, "Creating spherical worlds," in *ACM SIGGRAPH 2007 Sketches*, ser. SIGGRAPH '07. New York, NY, USA: Association for Computing Machinery, 2007, p. 82–es. [Online]. Available: https://doi.org/10.1145/1278780.1278879

[8] Maxis, "Spore," [PC Game] Electronic Arts, Steam, 2008.

[9] Farbrausch, ".kkreiger," [PC Game] Farbrausch, 2004.

[10] T. Short and T. Adams, *Procedural Generation in Game Design*. CRC Press, 2017.

[11] C. Chung, "Catlateral Damage," [PC Game] Manekoware, 2015.

[12] F. Marson and S. Musse, "Automatic real-time generation of floor plans based on squarified treemaps algorithm," *International Journal of Computer Games Technology*, vol. 2010, 09 2010.

[13] E. Huang and R. E. Korf, "Optimal Rectangle Packing: An Absolute Placement Approach," *Journal of Artificial Intelligence Research*, vol. 46, pp. 47–87, 2013.

[14] W. Benedetto, Sony Interactive Entertainment Inc., "AI-Generated Internal Environments Based On External Geometry," Patent Number: US20210121781, April 29 2021. [Online]. Available: https://patentscope.wipo.int/search/en/detail.jsf?docId=US322909393&_fid=WO2021081415

[15] T. Germer and M. Schwarz, "Procedural Arrangement of Furniture for Real-Time Walkthroughs," *Computer Graphics Forum*, vol. 28, no. 8, pp. 2068 – 2078, 2009.

[16] T. Germer, "Procedural Arrangement of Furniture for Real-Time Walkthroughs," YouTube, June 2009. [Online]. Available: https://youtu.be/xwVUknGeycQ

[17] S. Gottschalk, M. Lin, and D. Manocha, "OBBTree: A Hierarchical Structure for Rapid Interference Detection," *Computer Graphics*, vol. 30, 10 1997.

[18] T. Tutenel, R. Bidarra, R. Smelik, and K. J. de Kraker, "Rule-based layout solving and its application to procedural interior generation," in *Proceedings of the CASA Workshop on 3D Advanced Media in Gaming and Simulation (3AMIGAS)*, 01 2009.

[19] "CGAL 5.3 - 2D Minkowski Sums." [Online]. Available: https://doc.cgal.org/latest/Minkowski_sum_2/index.html

[20] P. Henderson and K. Subr and V. Ferrari, "Automatic Generation of Constrained Furniture Layouts," *Computer Vision and Pattern Recognition*, 2017.

[21] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic Scene Completion from a Single Depth Image," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 190–198.

[22] L.-F. Yu, S.-K. Yeung, C.-K. Tang, D. Terzopoulos, T. F. Chan, and S. J. Osher, "Make it home: Automatic optimization of furniture arrangement," in *ACM SIGGRAPH 2011 Papers*, ser. SIGGRAPH '11. New York, NY, USA: Association for Computing Machinery, 2011. [Online]. Available: https://doi.org/10.1145/1964921.1964981

[23] HKUST VGD, "SIGGRAPH 2011 - Make it Home: Automatic Optimization of Furniture Arrangement," YouTube, April 2011. [Online]. Available: https://youtu.be/vlDoSv6uDKQ

[24] A. Zhigljavsky and A. Zilinskas, *Stochastic Global Optimization*, P M. Pardalos and Ding-Zhu Du, Ed. New York Springer, 2008.

[25] S. Chib and E. Greenberg, "Understanding the Metropolis-Hastings Algorithm," *The American Statistician*, vol. 49, no. 4, pp. 327–335, 1995.

[26] A. M. Turing, "Computing Machinery and Intelligence," *Mind*, vol. 59, pp. 433–460, 1950.

[27] M. A. Boden, "The Turing Test and Artistic Creativity," in *Turing Test*, M. Bishop, Ed. Bradford: Emerald Group Publishing Limited, 2010, ch. 1, pp. 409–413.

[28] Adobe Inc., "Adobe Photoshop," 2021. [Online]. Available: https://www.adobe.com/uk/products/photoshop.html

[29] A. Pease, S. Colton, and J. Charnley, "The Turing Test and Computational Creativity," in *Contributed talks proceedings of the Turing Centenary Conference*, 2012. [Online]. Available: http://ccg.doc.gold.ac.uk/wp-content/uploads/2016/11/pease_tc12.pdf

[30] N. Barrantes-Vidal, "Creativity & Madness Revisited from Current Psychological Perspectives," *Journal of Consciousness Studies*, vol. 11, no. 3, pp. 58–78, 2004.

[31] O. Korn, M. Blatz, A. Rees, J. Schaal, V. Schwind, and D. Görlich, "Procedural Content Generation for Game Props? A Study on the Effects on User Experience," *Computers in Entertainment*, vol. 15, no. 2, pp. 1–15, April 2017. [Online]. Available: https://doi.org/10.1145/2974026

[32] Unity Technologies, "Unity," 2021. [Online]. Available: https://unity3d.com/unity/whats-new/2020.3.12

[33] "General Data Protection Regulation (GDPR)," 2016. [Online]. Available: https://gdpr-info.eu/

[34] "The Nuremberg Code," *Law, Medicine and Health Care*, vol. 19, no. 3-4, p. 266–266, 1991.

[35] Cabinet Office, "Coronavirus: How to stay safe and help prevent the spread," Dec 2021. [Online]. Available: https://www.gov.uk/guidance/covid-19-coronavirus-restrictions-what-you-can-and-cannot-do

APPENDIX A

GENERAL

List of unit tests & testing plan

APPENDIX B

DATA ANALYSIS

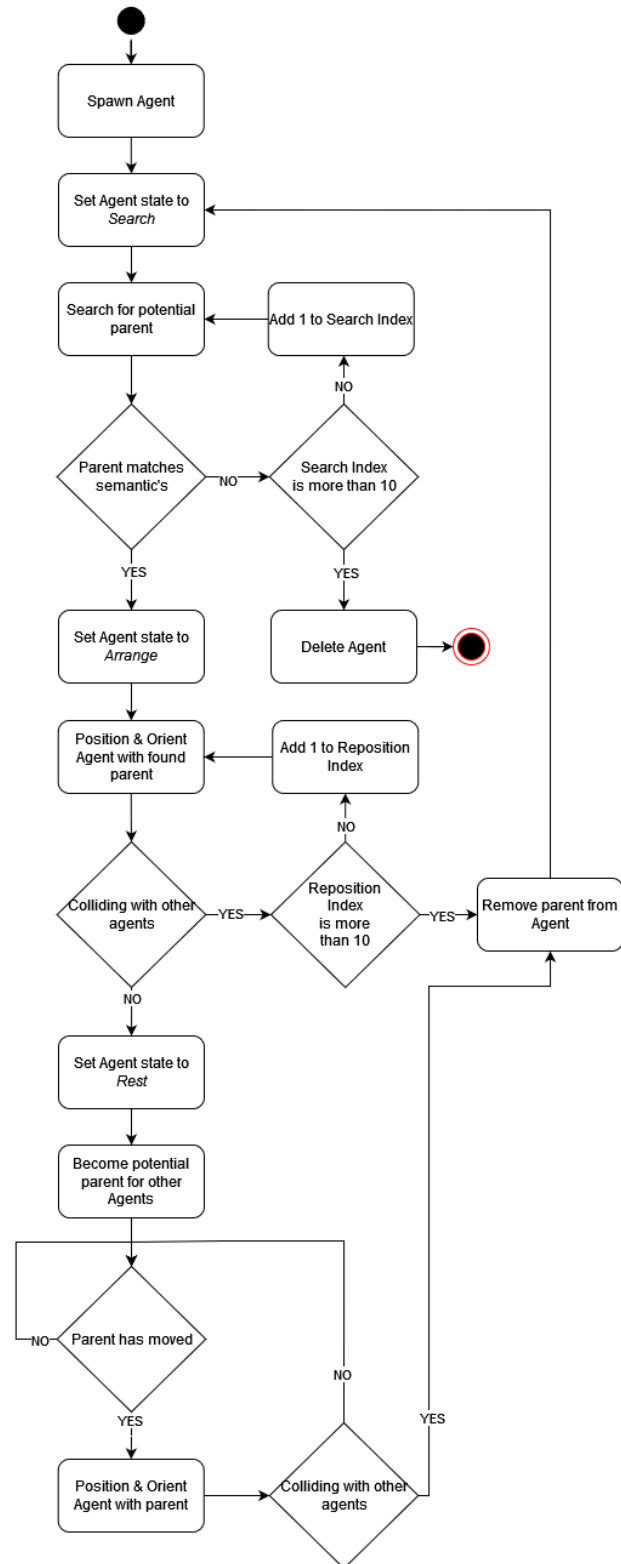Data analysis code, supporting screenshots

APPENDIX C

SOFTWARE ARCHITECTURE

Fig. 1: Agent Behaviour represented in an Activity Diagram