AUTOMATED FLOORPLAN OPTIMISATION FOR BATHROOM

How can automated bathroom layout designs be accurate when assessed against regulations and client's requirements?

K. IZUKA,
University of New South Wales, Sydney, Japan
izukakoki@gmail.com

Abstract. The automation is attractive as it has generated significant benefits in a range of fields. Floor plan layout automation allows architects to generate and test multiple possible options. If the automation is successfully implemented to an architectural project, it has a great potential to improve the time efficiency, reliability, and total project cost. (Anderson et al. 2018) As the automation is still relatively new to the field, the current style of design process does not include the automated process. With the use of 3D modelling software 'Rhino 6' and the visual scripting program 'Grasshopper' as the base, an automated layout designing system has been generated following the basic bathroom regulations set by the NSW government in Australia. The generated algorithms in the system can be categorized into two sections. One is specialized in dividing overall bathroom area into smaller segments for components and this algorithm was based on the "a multi-objective evolutionary algorithm" by Darcy Chia and Lyndon While. (2014) The other algorithm dedicated to allocating bathroom items to the divided segments. It may be reasonably challenging to implement a fully automated process to the practical works. However, the system could be used to generate possible iterations of office building bathrooms for architects to explore and improve the design which may support them to be more efficient in projects.

Keywords. Automation; Layout Design; Office Building; Bathroom; Binary tree; Optimization.

1. Introduction: (Research context and motivations)

The world is rapidly evolving. The several processes in a range of fields got automated with the use of robots, artificial intelligence, and creative systems and software. As the automation has a great potential to improve and change the work environment, the interest in the automation has been growing massively. If the automation is properly implemented to a project, it is likely to reduce the total cost of project and improve the reliability, productivity, and efficiency of workflow. (Anderson et al.) There are several fields or businesses that implemented the automation to a process and achieved higher project efficiency as the result.

This also can apply to the field of architecture. The architectural projects commonly take a few months to complete from the start to end. The construction stage often takes the longest time. However, even the designing/planning stages take 6-10 weeks to complete. Especially in a project where a requested building is enormous such as office buildings, malls, schools, and universities, it generally takes a longer period for each phase of the project. Therefore, if there is any system that improves the efficiency and productivity of projects and workflow, it would be beneficial for both architects and clients.

To shorten the total time frame, one way is to implement the automation on the design phase of architectural projects. Although the automation of layout designing has been challenged in several ways since the early 1960's, (Liggett 2000, p. 197) there has been no universally used system in the architectural field as of now. There are various factors such as the building regulations and client's requests which creates a difficult environment for the automation to be implemented on architectural projects. However, the bathroom is one of the core components of buildings. There is almost no building that does not have a bathroom. On the office buildings, the locations of bathrooms are likely to be on the same spots throughout the floors to simplify the piping for water. In addition, if there is no extra special request set by clients, the layouts for bathrooms tend to be similar and simple as well. Hence, it sometimes is repetitive work for architects to design bathrooms for an office building. The automated systems' ability can be utilized more if the work is simple and specific. Thus, this would be a desirable opportunity to implement the automation to a part of the project.

This paper will explore the reliability and effectiveness of the automated design process on layouts of bathrooms in practical situations. The research was divided into three sections. The first section dedicates on examining the regulations of office buildings set by NSW in Australia. The second section explains the system that divides the overall bathroom area into segments for components. Lastly, the third section demonstrates the component allocation algorithm for each divided segment in the bathroom. The result of this research is discussed accordingly with the bathroom regulations of NSW and precision.

2. Research Aims

This research aims to explore the potential to implement the automated designing process to the architectural projects. Considering the project's duration is set to maximum ten weeks, these are the main aims of the research:

1. Develop a script that generates practically acceptable office building bathroom layouts on Grasshopper.

2. Optimize the script so that it can accept different regulations and item requirements.

3. Research Question(s)

To implement an automated layout designing system for practical projects, it requires the high accuracy that follows the regulations set by the government and client's requests. Thus, based on the issues outlined in the introduction and the derived aims, the question the research this project investigates is:

- How can automated bathroom layout designs be accurate when assessed against regulations and client's requirements?

.

4. Methodology

It is once claimed by Azhar et al. (2009) that construction engineering and management (CEM) requires a research approach that concurrently combines the objectives of both existing and standard research by contributing toward solution of realistic problems and creation of new theoretical knowledge. The existing research methodologies are sometimes excessively focusing on existing and conceptual issues over what is really needed in the real industry. Thus, the outcomes of research may be limited as the result.

O'Brien (1998) has defined the action research that the action research is "Learning by doing"; a group of researchers identifies a problem, try something to solve it, analyze how successful their efforts were, and if not satisfied, reattempt. The last step of this workflow has the evaluation of results that may lead into reattempting of resolving steps. Therefore, this action research methodology works in a spiral until it reaches the goal. This should improve the quality of the final outcome of the research in theory.

By applying the right action research methodology, the outcomes of the research should be highly effective and more reliable for solving practical situations. The workflow of action research by O'Brien would be generally effective to all practical research projects. The "problem" is simply the research question of this research, 'How can automated bathroom layout designs be accurate when assessed against regulations and client's requirements?'. The desired research outcome will be investigated throughout various attempts of Grasshopper scripting until a practically acceptable outcome is achieved. The research workflow will be split into sections which are categorized by the functionality of algorithms. Each attempt in sections will be analyzed and criticized carefully with the bathroom regulations set by NSW government. This process will help to understand the current issues deeper which should open more possibilities to further improve the quality of research. By taking "Learning by Doing" attempt, it may be possible to find a method that the person could never find without attempting some trials on the existing problem. This methodology meets the two essential requirements for the research approach that CEM needs; concurrently combines the objectives of both existing and standard research by contributing toward solution of realistic problems and creation of new theoretical knowledge.

5. Background Research/Literature review

By automating a component in a design process, it brings the potential of increased cost performance in projects, reliability, and efficiency by systemizing repetitive works. (Anderson et al. 2018) The automation of layout designing has been tackled in various methods since the early 1960's. (Liggett 2000, p. 197) From then on, the architects, computational designers and developers have been attempting to produce an evolutional design method to solve layout problems. However, a universally applicable method has not been produced for computer-aided layout design. (Schneider et al. 2011)

Maciej Nisztuk and Pawel B. Myszkowski from The Wroclaw University of Science and Technology have conducted research on the usability of contemporary tools for the computational design of architectural objects. In their research, they assessed the computational designing tools with various criteria. From the research, they concluded that the computational tools are not fully utilized in a creative aspect; the existing tools usually lack an intuitive interface and they are limited in functionalities such as ignoring legal regulations and design rules. (Myszkowski & Nisztuk 2017)

Several precedents for automated layout design method can be briefly categorized into two, search and exploration. 'Search is a process for locating values of variables in a defined state space while exploration is a process for producing state space.' (Anderson et al. 2018) A group of researchers, Carl Anderson, Carlo Bailey, Andrew Heumann and Daniel Davis have worked with an architectural firm WeWork to produce the automated layout design system that focuses on the desk arrangement in commercial buildings. They approached this project with the application of search algorithms. It used data of desk arrangement from several pre-existing real projects for studying past patterns to create the algorithms. This project focused on one specific criterion, efficiency. They measured the efficiency by comparing computer-generated designs with human-generated designs. The initial result was lacking accuracy. However, after easing the regulations, the result of the project turned out noticeably positive; the designs generated by set algorithms achieved 83% of success rate of the cases at the task of arranging the maximum possible desks in a provided area while following the regulations. (Anderson et al. 2018)

Anderson et al. (2018) claimed that by reducing the restrictions on layout designing algorithms, the achieved outcome got improved. However, a group of researchers from the Stanford University once claimed that designing residential building layouts is more challenging than designing schools, hospitals and office buildings as objectives are less precisely defined and harder to operationalize. (Merrell et al. 2010) This creates a conflict and thus, leading to a question, 'What is the optimal amount of regulations/rules to create an effective automated layout design?'.

While the previous project example worked with a very specific item in layout, the project undertaken by Darcy Chia and Lyndon While worked with multiple items, the layout of rooms. This project is dedicated for the game environment; however, it can be applied to the architectural aspect for various building types from a small apartment to an office floor. Their algorithm is called 'a multi-objective evolutionary algorithm' (MOEA), which generates multiple floor plans with a flexible input system via hard and soft constraints that lets a user to specify required rooms and relationship between each room such as the amount of specific rooms required, maximum aspect ratio, and minimum and maximum length, width and area. The algorithm also contains nine fitness criteria to assess the quality of design compared to the specification. (Chia & While 2014) Their base algorithm can be represented as a binary tree, where the origin of the tree represents the dimensions of the total area available and each node after the origin represents subdivisions which are either an assigned room or available space after the split of space. This methodology is very flexible and easy to optimize in several natural ways.

Similar to the project undertaken by Darcy and Lyndon, the research project done by Fernado Marson and Soraia Raupp Musse uses the Treemaps methodology to generate floor plans. (Marson & Musse 2010) The basic algorithms for generating rooms are extremely similar to the previous project. However, the process to generate corridors is different in these two projects. The existence of the corridors is valued in the projects as it is required to connect generated rooms. Therefore, in both projects, the rooms which are not linked to the public space (e.g. living room) are identified in order to create access to all rooms. Darcy and Lyndon generated the corridor in the stage of generating subdivisions while Fernado and Soraia generated corridors after generating rooms so that all rooms will be connected to public spaces or corridors. (Chia & While 2014) The result of these two turned out fairly similar and realistic. However, it could be observed that the methodology taken by Fernado and Soraia have more unutilized spaces left.

The limitations of algorithms come from several sources and they often cannot be described with a specific theory or mathematical formulation; it is challenging to find consistency among limitations. (Cao et al. 1990, p. 213) Each various approach in above examples have strengths and limitations in the field of automated layout designing. The approach that uses past works may be somewhat limited in creativity as it may be an imitation of pre-existing works. However, it should be efficient in creating realistic layouts. The projects that utilize binary tree algorithms were limited in shape of rooms. The generated rooms have very simplified shapes, which also limits the creativity of the work. Yet, the majority of commercial buildings have rectangular or polygonal shapes for the rooms so it will be still effective and useful. By limiting the coverage of algorithms that generate the layout system, the outcome should be functional and effective from a realistic perspective.

The binary tree-like algorithm used in the project of MOEA can be applied to this research project. The algorithm could be the base system for dividing the overall bathroom area into components.

6. Case Study

6.1. RESEARCH

To start designing the automated layout designing system, it requires a research on item requirements and regulations of the basic office building bathroom layouts. The regulations of office building bathrooms differ in various countries and states. Therefore, this project will focus on the regulations set by NSW government for the purpose to create a practice environment.

From the research, the basic items in the bathroom were listed. These include the toilets, toilet cubicles, washbasins, and doors. The urinals are the extra items for the male bathrooms. The listed items are going to be embedded in the generated layout designing system. In order to create the base of algorithms, it requires the standard dimensions of items to divide the bathroom into segments.

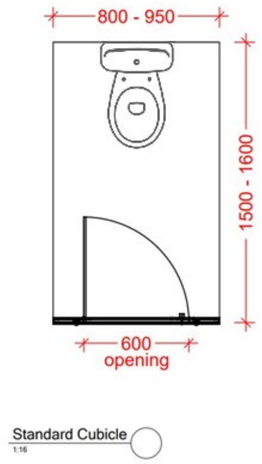

Figure 1. Standard cubicle dimensions

The dimensions in Figure 1. are going to be used in the later script to decide on the dimensions of the door, cubicle depth and width.

6.2. DIVISION OF SPACE INTO SEGMENTS

The first stage of the automated layout designing system development focused on dividing spaces for the components of the bathroom. In the research phase, 5 components of the bathroom were identified for male's bathroom (4 for female's bathroom). However, as the toilets are inside the toilet cubicles, they will be counted as a single component for this phase.

This script works under a premise that the outline of the bathroom is provided and predefined. Therefore, the area of the bathroom will be divided into 4 segments for components and the area for path. As a result, the area of the bathroom will be divided into 5 segments. The 5 segments are: door area, washbasin area, cubicle area, urinals area, and path. If the bathroom is for females, the extra space for urinals will be merged to the path.

The space division algorithm works in a sequence of dividing spaces for components. The sequence follows the order of 'door area → wash basin area → cubicle area → urinal area → path area'. This algorithm was inspired from the project by Darcy Chia and Lyndon While. The binary tree-like algorithm used in the project is modified to suit the bathroom layouts.

The first step is to find the favorable location for the entrance door. The location of the entrance door is decided by testing whether the sides of bathroom overlap with the outlines of building or rooms that cannot be connected to the entrance of bathroom. If a side of bathroom overlaps with the building outline, people are physically unable to walk into the bathroom from the door on this side. Thus, the door will be located on a side that does not overlap with the specified factors. With the given dimension of door, the door area will be created in a square shape. The dimension of this area is 1.1 times longer than the width of the door. The overall bathroom area will be subtracted with the door area. This subtraction process will take a place whenever a component gets assigned to a space in the bathroom.

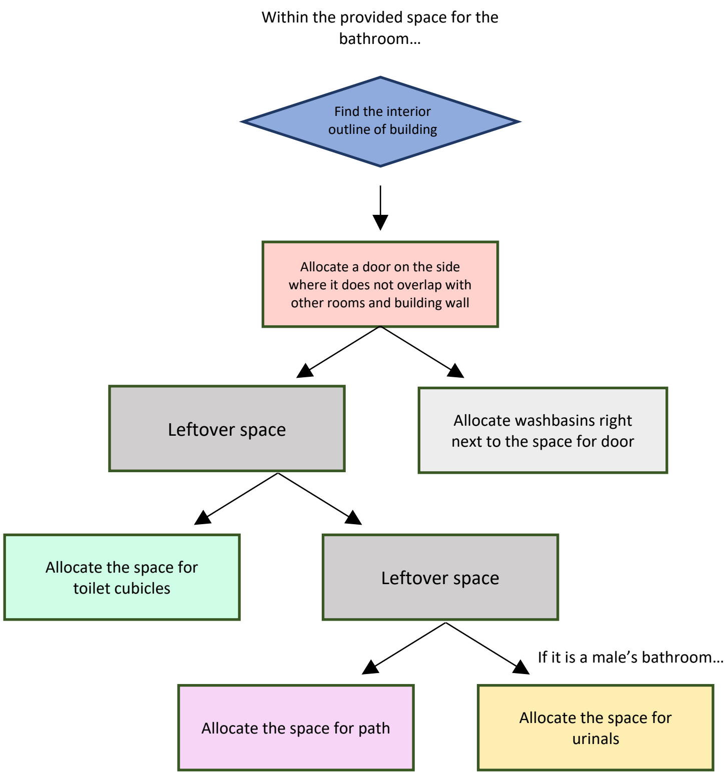

Figure 2. Bathroom space division algorithm.

The next step was to identify the space for wash basins. The area of wash basin is located right next to the door area. This area can be defined with the width of predefined door area. A line that divides the overall area into 2 segments is drawn from an edge of the door line. On one side of segment, there are door and wash basin areas. Since there is the door area defined within the segment already, the remaining area is for wash basins as shown in Figure 3.

Figure 3. Bathroom space after 2nd division.

On the other side of the segment, there are toilet cubicles, path, and urinals' areas. This segment's division is conditional. If this is the male's bathroom it is going to be divided into 3 segments as there needs to be a space for urinals. However, if the bathroom is for females, it will simply be spaces for cubicles and path in front of the cubicles. For this segment, the space for toilet cubicles is defined first. There is another algorithm involved here. To find the best side within the remaining segment, it checks all the sides of the remaining segment and returns a side with the highest capacity. In this way, if the bathroom only requires less than the highest capacity, it can reduce and adjust the number of toilets to suit the requirement of the building. The length of the standard cubicle was defined at the beginning of the system. This length is used as the factor to decide on the position of the line that divides the remaining space into two segments. The length of the cubicle gets subtracted from the length of the overlapping side to get the position of middle line.

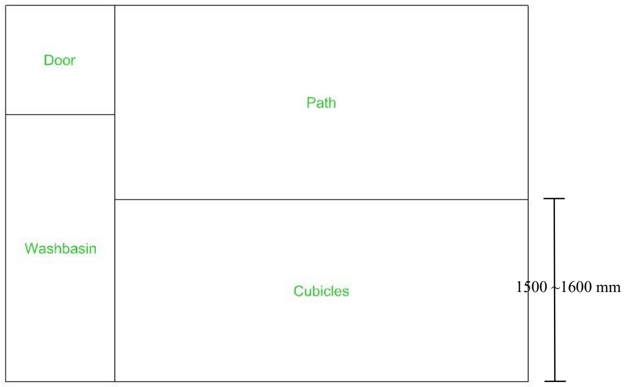

Figure 4. Bathroom space after all divisions.

At this point of the script, all spaces of the bathroom are filled with the spaces for components. (if it is a female bathroom) The urinal space is not necessary to be defined as it is technically placed in the path area. In further phases, the components will get allocated to the provided segments.

6.3. ITEM ALLOCATIONS

In this phase of the script, the components of the bathroom figured out in the research phase are going to be allocated to the defined spaces. The space division phase took a binary tree-like algorithm. In this phase, the base algorithm is designed with a combination of 'if loops' using simple python coding. The items of bathroom except the entrance door, the script starts with calculating which sides have the highest capacity of items within the provided segments. After the calculation, it sorts the list of sides from the highest to lowest capacity. The script starts checking if the sides have no issues for items to be placed adjacently. If it has an issue such as unreasonable location for toilets in cubicles, it eliminates the side from the list. This if loop runs from highest to lowest capacity and continues until it finds a favorable side for the item.

After finding an ideal side in the segments, the script now divides the side by the number of items needed. If a bathroom is defined to have 3 toilets, it will divide the side by 3 to create 3 cubicles. In this case, the divided side will have three new individual lines. Then, the toilets are placed and aligned to the center point on these individual lines.

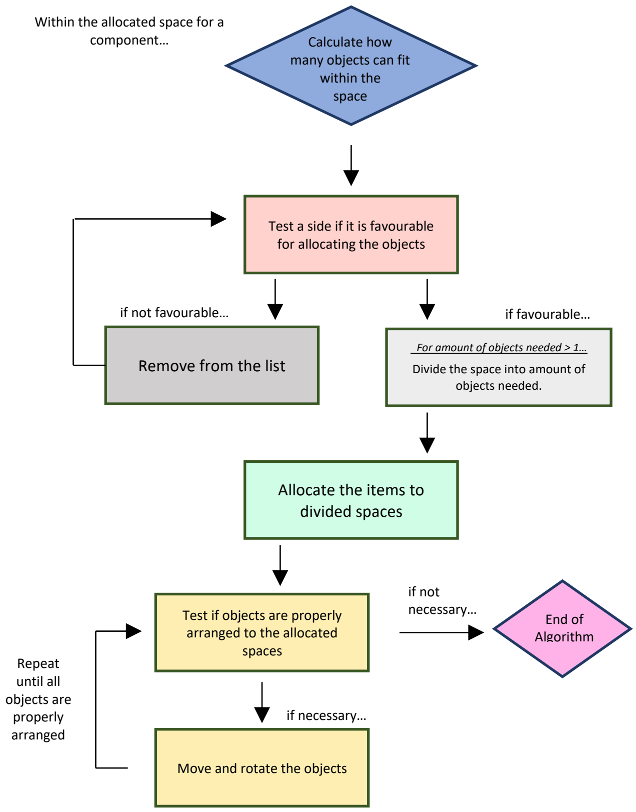

Figure 5. Item allocation algorithm.

After placing the items into the segments, another algorithm, which dedicates on the rotational allocation runs to fix the orientation of items. This is still a component of 'Item Allocation Algorithm', however, this section of script may be not necessary to run, depending on the orientation of placed items. First, it checks the placed items whether they locate fully inside the allocated segments. This is checked by taking the four center points of items as shown in Figure 6. The four points are

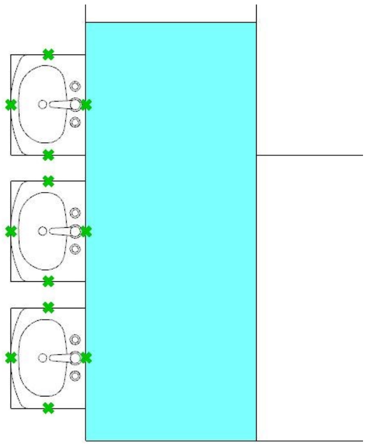

Figure 6. Extracting 4 points from items.

After extracting 4 points, the script checks how many points are located inside the provided area. In Figure 6, there are three points from each washbasin which are outside of the provided space. If there are no points outside the area, the script simply will not run. In the case where there are 3 points outside as in Figure 6, the python script will return the value of 180, which rotates the items by 180 degrees. If there are 2 points outside, it will return the value of 90, which rotates the items by 90 degrees. This script will run the maximum of three times until all items are rotated inside. When all items are rotated inside the segments, this is the end of the script.

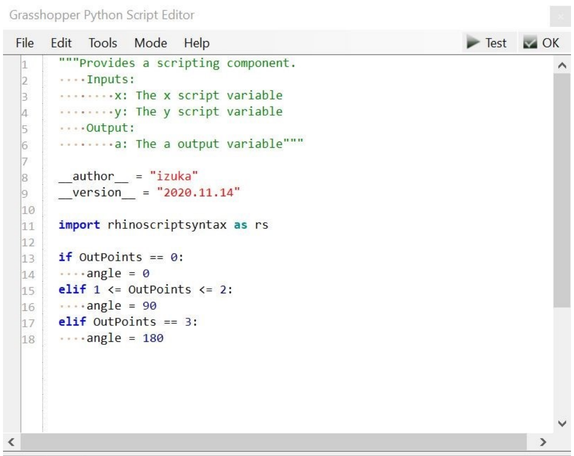

Figure 7. Python code for rotation angle.

6.4. GENERATED LAYOUTS

By using the script written earlier, the layouts could be outputted. Figure 8 shows the horizontal layout of male and female bathrooms which were outputted with the script. In male's bathroom, the urinals are created on the path area and there is an extra wall created for the water pipes to be placed inside. These outputted layouts should follow NSW's bathroom regulations. However, in the female's bathroom, it could be observed that the space of path has a noticeably huge area. This may be inefficient in terms of space utilization. Figure 9 shows the generated vertical layout. The area of space was increased, therefore, the number of items improved as the result. Once again, it could be observed that the path has an unnecessary large space. This part of the script would be a valuable factor to consider for future projects.

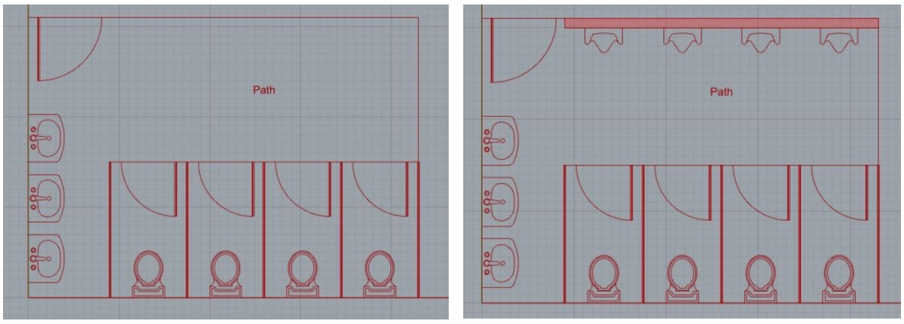

Figure 8. Generated horizontal layouts (Female and Male).

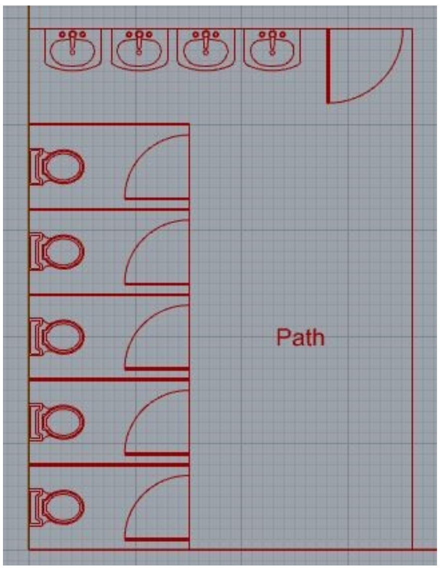

Figure 9. Generated vertical layouts.

7. Discussion (evaluation and significance)

This research has explored the potential of automation in floor plan layouts. The created script of an automated layout designing system has covered the aims. In the research aims, it is mentioned to develop a script that generates practically acceptable office building bathroom layouts on Grasshopper. The script can generate both male and female bathrooms. The outputted bathroom layouts with this script had a decent appearance and outcome, in the case if the pre-defined bathroom outline does not have complex shapes. However, it is somewhat extraordinary for the bathroom to have a complicated room shape. Therefore, there is a potential to implement this script to practical works. Another aim was to optimize the script so that it can accept different regulations and item requirements. The script allows users to switch the toilets, urinals, and wash basins models. In addition, they can also change the amount of these items to include in the bathroom. Although it requires some manual changes to the input values of the script, it is still flexible to change the regulations.

In the research question, it is questioned in what ways automated bathroom layout designs can be accurate when assessing against regulations and client's requirements. The last optimized script returned the layout that did not have obvious errors. By having precise numbers that are based on the regulations inputted into the script, the accuracy of generated layouts could be controlled to certain extent. Yet, if the types of components increase in the bathroom and the number of regulations increase as a result, it may lose the flexibility of scripts and there is a chance of losing some accuracy due to over-limiting.

In addition to this, some areas of generated layouts could be improved. In the generated layouts, when the predefined room sizes were massive, there are some inefficiently empty spaces which may be a little too large for just the path in the bathroom. Thus, the utilization of unused spaces could be improved. This now raises another question, "How can space optimization of automated bathroom layout designs be increased?". In addition, as this script did not include ambulant bathrooms, this would be another factor to improve. In the future works, supporting other building types would be beneficial as well. If the script works in various building types, it improves the versatility and potential, which should result in increased the needs of the automated script.

8. Conclusion

In this project, the automated bathroom layout designing system, which particularly focuses on the office building bathrooms has been produced. The system is capable of producing standard office building bathroom layouts that can be applied to the most of room sizes in a building. In the research question, it is being asked how automation of layouts can be accurate while maintaining the regulations and requirements set by clients. The generated system accepts a wide range of regulations and requirements and generated layouts possess acceptable appearance. However, the script is unreliable in the aspect of flexibility of layouts. The generated layouts tend to be repetitive which makes the layouts not creative.

The algorithms which were used to generate the system may help the future projects to consider the path which they can take to generate an effective system. The outcome of this project may provide a sufficient knowledge of algorithms that supports to automate a layout designing for future projects. As this project specifically focused on a type of building and room, "the office building's bathrooms", the future projects can take two directions. One potential direction is to increase the building types that the script can support. The building types such as residential units, apartments, schools, and restaurants have different regulations and item requirements. By supporting these building types, this will improve the script's versatility which may result in creating a more functional script. Another way is to support other building room types. As the produced script can support only one type of a building, it may not improve the efficiency of workflow noticeably. If the script can produce other room layouts, this should improve the efficiency of the design workflow furthermore. Developing computational tools to automate layout designing will contribute to improve the efficiency of projects. By reducing the time spent on information searching and repetitive tasks, architects and designers can return their focus to what they do best: design.

Acknowledgements

I would like to thank Nicole Gardener, Hank Haeusler and Daniel Yu for guiding and supporting my research project throughout the course of UNSW CoDe. I also would like to thank all of my colleagues for providing me with fantastic ideas to consider.

References

Anderson, C, Bailey, C, Heumann, A & Davis, D 2018, 'Augmented space planning: Using procedural generation to automate desk layouts', *International Journal of Architectural Computing*, vol. 16, no. 2, pp. 164-177, DOI: 10.1177/1478077118778586.

Cao, X, He, Z & Pan, Y 1990, 'Automated design of house-floor layout with distributed planning', *Computer-Aided Design*, vol. 22, no. 4, pp 213-222, DOI:10.1016/0010-4485(90)90050-M.

Chia, D & While, L 2014, ‘Automated Design of Architectural Layouts Using a Multi-Objective Evolutionary Algorithm’, *Asia-Pacific Conference on Simulated Evolution and Learning*, DOI: 10.1007/978-3-319-13563-2_64.

Duracube. 2020. *A Guide To Toilet Cubicle Dimensions In Australia*. [online] Available at: <https://duracube.com.au/a-guide-to-toilet-cubicle-dimensions-in-australia/> [Accessed 20 November 2020].

Liggett, R 2000, ‘Automated facilities layout: past, present and future’, *Automation in Construction*, vol. 9, no. 2, pp. 197-215, DOI: 10.1016/S0926-5805(99)00005-9.

Marson, F & Musse, S 2010, ‘Automatic Real-Time Generation of Floor Plans Based on Squarified Treemaps Algorithm’, *International Journal of Computer Games Technology*, vol. 2010, no. 2, DOI: 10.1155/2010/624817.

Merrell, P, Schkufza, E & Koltun V 2010, ‘Computer-Generated Residential Building Layouts’, *ACM Transactions on Graphics*, vol. 29, no. 6, DOI: 10.1145/1866158.1866 203.

Myszkowski, P & Nisztuk, M 2017, ‘Usability of contemporary tools for the computational design of architectural objects: Review, features evaluation and reflection’, *International Journal of Architectural Computing*, vol. 16, no. 1, DOI: 10.1177/1478077117738919.

Schneider, S, Fischer, J & Koenig, R 2011, ‘Rethinking Automated Layout Design: Developing a Creative Evolutionary Design Method for the Layout Problems in Architecture and Urban Design’, *Design Computing and Cognition ‘10*, pp. 367-386, DOI: 10.1007/978-94-007-0510-4_20.