Reinforcement Learning to Generate 3D Shapes: Towards a Spatial Visualization VR Application*

Charles Silverman[1], Liam Stewart [2], and Christian Lopez[1,3]
[1]Computer Science Department
[2]Electrical and Computer Engineering Department
[3]Mechanical Engineering Department
Lafayette College,Easton, PA 18042
lopezbec@lafayette.edu

Abstract

The objective of this work is to introduce a Reinforcement Learning (RL) approach to automatically generate 3D shapes of different complexities. This is with the goal to help tailor the spatial visualization task of a Virtual Reality (VR) application designed to help develop students' spatial skills. Spatial visualization skills are important skills needed and frequently used in the STEM fields. While VR has been used to help develop these skills, most of the existing applications do not necessarily tailor their content to students' skills level. Automatically generating 3D shapes can help VR applications tailor spatial visualization tasks to the skills level of students. The results of this work indicate that an RL agent is capable of creating an action policy that can generate 3D shapes with complexities similar to a given desired complexity provided. However, the results also show that the task of automatically generating 3D shapes that meet a given complexity is not trivial given the issues of sparsity in the reward space. Nevertheless, this work lays the foundation to leverage RL to automatically generate 3D shapes for VR applications designed to help develop students' spatial visualization skills.

1 Introduction

An individual's capacity to mentally manipulate and understand 3D shapes is known as their spatial or visuo-spatial ability [26]. Rotating and generating cross-sections of a shape are usual metrics for evaluating one's spatial ability or skill [18]. In STEM programs, both GPA and capacity to conduct self-monitored learning have been shown to relate to a student's spatial skills [33, 28]. Further reinforcing the importance of spatial skills, several fields, including Chemical Engineering, Civil Engineering, and Computer Science, have found a correlation between students' spatial skills and academic success [34, 18, 33]. Unfortunately, many STEM students lack these critical skills at the beginning of their studies. Moreover, standard instruction and class coursework at the introductory level might not be enough to thoroughly develop these skills [32].

Spatial skills could be learned through the manipulation of 3D objects [28]. For some, this is as simple as playing with building blocks as a child, whereas for others, it may need to be learned in an academic setting with repetitive exercises. Thanks to recent advancements and the penetration of Virtual Reality (VR) technology, educators are leveraging VR as a pedagogical tool [24, 16]. Moreover, it has been found that students could perform better within VR settings when compared to other traditional settings [20, 19]. The greater the immersion of the VR application, the greater the performance increase of the students [19]. Thanks to its unique characteristics that facilitate "first-person" experiences and allow users to interact with 3D virtual objects, VR is suited to teach and help develop spatial visualization skills. For example, some researchers have already started exploring this and have shown promising results [8]. However, researchers have also shown that students can have different levels of expertise which can directly impact their state of flow in a given application [25]. The Flow theory of motivation indicates that students would best execute the tasks whose difficulty aligns with their skills level. Consequently, any educational VR applications (e.g., ones designed to help develop spatial visualizations skills) that do not provide any tailored content or only enable users to interact with a limited set of content, might motivate users just for a short period of time (e.g., "novelty-effects").

Furthermore, the cost of VR technology has steadily decreased, thus increasing the economic accessibility of the devices for different socio-economic groups[21]. However, the resources needed to generate new VR content are still high. The existing challenges for generating new VR content might prevent educators to tailor their VR content to their students. Moreover, novelty effects could set in and lower students' motivation as they grow accustomed to the VR environments they are learning from [14]. Thus, new content might be required to counteract potential novelty effects.

The gaming industry has already leveraged methods to generate new content automatically. These Procedural Content Generation (PCG) methods have been used since the '80s, and are able to not only help reduce the resources needed to create content but also help with users' long-term engagement and motivation [9]. VR applications could greatly benefit from methods that can help generate content automatically. Specifically, VR applications designed to help develop spatial visualization skills could benefit from methods to automatically generate new 3D shapes. These 3D shapes could also be tailored based on students' skill levels, which could help improve their state of flow. One potential solution to achieve this would be to leverage PCG methods to generate new content [6, 17, 15], like 3D shapes of different complexities that students can interact with while developing their spatial visualization skills.

Based on the importance of spatial visualization skills and the advantages of VR to help develop these skills, the authors of this work have introduced a VR application designed to help develop spatial visualization skills in which students can use their hands to interact with 3D shapes and perform spatial visualization tasks [35]. This work extends previous efforts by introducing a PCG method based on a Reinforcement Learning approach to automatically generate 3D shapes of different complexities that students can interact with while developing their spatial visualization skills.

2 Literature Review

2.1 Virtual Reality for Education

As VR technology has grown more available, they have been introduced to educational mediums[27]. During the first uses of VR in education in the previous decade, close to half of them pertained to engineering[22]. Although the term immersive and not been formalized in this context at the time, most of those pioneering studies compared immersive vs non-immersive platforms [27]. The standardized definition of immersive VR has two components: the capacity for the user to interact with the virtual environment, and the incorporation of a head-mounted display [19, 38]. Striving for immersion is important because it is directly correlated to the benefits of using VR; specifically, a user's motivation, long-term retention, and enjoyment of the material [19]. Regardless of these benefits, there are still limitations to the currently available VR educational applications. For example, the novelty effect causes the benefits of VR to decrease in magnitude as users adjust to the new learning environment. Moreover, the cost of generating new VR content has not decreased along with the cost of headsets, which limits the development of VR educational applications [30]. Both limitations might be surpassed with help of Machine Learning methods capable of automatically generating new content for VR applications.

2.2 Development of Spatial Skills

In most cases, spatial skills are learned via the interactions an individual has with objects during their youth [37, 7]. Nonetheless, previous studies have proven that an individual can improve their spatial skill beyond what they developed as a child [8]. The main way that one improves their spatial skill is through the completion of spatial tasks, like the ones present in the Mental Rotation Test or the Purdue Spatial Visualization test [4]. STEM professionals are more likely to have more developed spatial skills than the population in general, mainly because they develop spatial skills when working with spatial tasks like those in STEM fields [2]. Moreover, in STEM programs, both GPA and capacity to conduct self-monitored learning have been shown to relate to a student's spatial skills [33, 28]. This is one of the reasons why it is important to ensure and help students entering STEM programs develop their spatial visualization skills. Knowing that spatial skills can be taught to students via the repetition of spatial tasks, researchers have leveraged VR to help develop students' spatial visualization skills

2.3 VR for Spatial Skills

Using VR as a platform to teach spatial skills is not a novel idea, in fact, several studies have already detailed its effect [38]. The result of these studies indicates that VR is an efficient platform for teaching spatial skills [8]. In a study that compared the improvement of students that used immersive VR and those that used traditional mediums, the students that used VR had greater improvement in their spatial skills [23]. The findings of this work indicate that learning in VR gives the students greater spatial perception than learning on a 2D platform (e.g., a desktop computer) [23].

One of the metrics frequently used to assess and develop spatial visualization skills are the Mental Rotation Test and the Purdue Spatial Visualization test. The application introduced in a previous work by the authors [35], also leverages these tests to help develop students' spatial skills by practicing spatial tasks. For example, Fig.1, from [35] figure 3, shows an example of the mental rotation tasks users are able to perform while using the desktop version of the application. The application introduced in [35], also leveraged hand tracking as a more natural user interface that should increase the user's immersion when compared to using controllers.

While VR has already been shown to help develop spatial skills by allowing students to practice spatial tasks in a 3D virtual environment, it has been also shown that students can have different skill levels that can directly impact their state of flow. The Flow theory of motivation suggests that individuals would be more engaged to execute tasks whose difficulty aligns with their skills level [25].

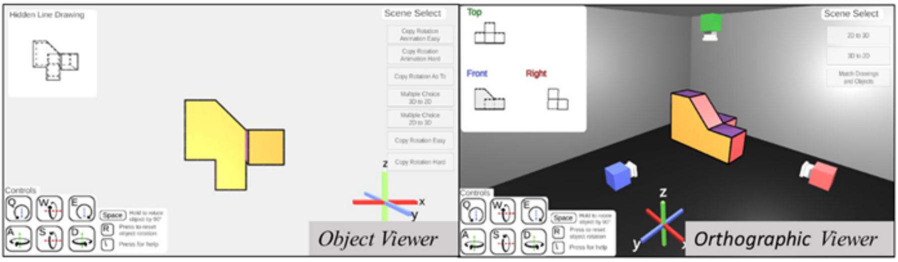

Figure 1: Spatial Visualization Tasks Examples from [17]

Unfortunately, most of the existing VR applications designed to help develop spatial visualization skills only enable users to interact with a limited set of 3D shapes that are not necessarily tailored to their skill level. Nevertheless, previous studies have shown promising results of using Procedural Content Generation and Reinforcement Learning methods to generate new content for VR applications [6, 17, 15].

2.4 Reinforcement Learning

Methods to automatically generate new content, known as Procedural Content Generation (PCG), have been used extensively by the gaming industry [29, 10], and most recently in educational applications as well [11]. In addition, researchers have started leveraging Machine Learning to automatically generate new content. However, methods based on Supervised Machine Learning approaches, require collecting or generating data a priori to train their models[11, 10].

Unlike other forms of Machine Learning, Reinforcement Learning (RL) does not need an established training dataset per se [15]. It instead leverages a simulation environment to generate an action policy to effectively address complex situations [15]. An RL agent uses the current state of a simulation environment and a reward function it tries to maximize. The agent implements a series of “trial-and-error” runs of the simulation environment to generate an action policy that would maximize its long-term reward [1]. Reinforcement Learning agents also have the benefit of generating an action policy that enables them to act in simulated environments with different states without the need for additional training. Studies have shown that Reinforcement Learning agents can effectively generate action policies to perform a task in complicated tasks, such as puzzles, retro video games, and board games like Go [22, 13, 31].

Table 1. shows some of the existing work done to help develop students’ spatial visualization skills, as well as the use of PCG and RL in educational

applications. While several works have explored the use of VR to help develop spatial visualization skills, most of the proposed solutions have several limitations that arise from the challenges of using a finite and limited set of 3D shapes users can interact with. Hence, in this work, the authors introduce a PCG method based on a Reinforcement Learning approach, that would be capable of automatically generating new 3D shapes of a given complexity. This would also enable to tailor of 3D shapes based on students' skill levels in order to improve their state of flow and motivation while interacting with a VR application designed to develop their spatial visualization skills.

Table 1: Summary of existing works

Reference	Solution for SVS	VR for SVS	PCG	PCG with RL
[33, 28, 7]	X			
[8, 5, 3]	X	X		
[10, 11, 12, 37]			X	
[6, 17, 15]*			X	X
This work	X	X	X	X

* *Author's previous work, Spatial Visualization Skills=SVS*

3 Method

This work introduces a Procedural Content Generation (PCG) method based on a Reinforcement Learning (RL) approach that automatically generates 3D shapes of a given desired complexity. The RL agent uses a 3D shape generator environment to create and test different 3D shapes. A metric that measures the symmetry of the shape is used as a complexity metric since previous studies have shown that 3D shapes that are symmetrical are perceived as less complex than shapes that are not symmetrical [36]. The reward function used to train the RL agent considers the difference between the desired complexity and the complexity of the shape generated, as well as other factors to help mitigate the sparsity problems. The RL agent and training process are explained in more detail next.

3.1 3D Shapes Generation

To algorithmically generate 3D shapes, like those present in the Purdue Spatial Visualization test [4], a system of wedges and voxels was implemented. Figure 3 shows a representation of this system. There are a total of 8 voxels arranged in a cube configuration. This means, that the biggest 3D shape that can be generated would have 2 voxels of length on any axis. Moreover, each voxel is composed of 12 different wedges that can be enabled or disabled to generate different 3D shapes. These 12 wedges come from having a wedge (i.e., cube cut in diagonal) in each of the three axes that can be rotated 90 degrees at least four times (i.e., 3*4=12). To have a complete voxel, at least two complementary wedges need to be enabled. A video of the 3D shape generation can be seen here `https://youtu.be/Z9n2zUuqk-E`. This process can generate shapes without curvatures, which are the most predominant shapes in the Purdue Spatial Visualization Test (i.e., 70% of the shapes do not have any curvature).

Based on previous literature, a complexity metric was developed to measure the symmetric shapes generated [38]. Figure 2, shows a visualization of the complexity metric. This metric aims to measure complexity based on the symmetry of 3D shapes. For each shape, a moving cutting plane in each of the x, y, and z axes was used to scan the shape. This sweep allows to potentially capture complexity arising from internal parts of the 3D shape (e.g., if a shape has a whole) and to account for the difference in the shape along an axis (i.e., the front is different than the back). For each of the cutting planes, the voxel area used to generate the 3D shape was identified. This is to help the metric to be size and translation invariant (i.e., a shape should have the same complexity no matter its size or location). Subsequently, the pixel level Euclidean difference between areas of the shape that were divided by each of the 4 potential lines of symmetry was calculated. The difference was then normalized to have values between 0-1.

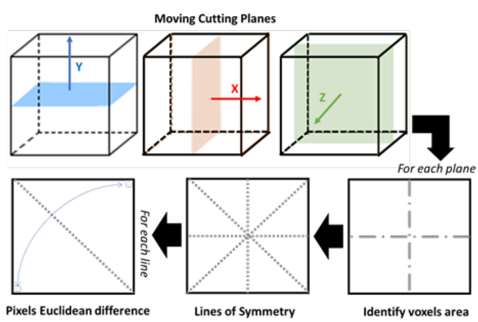

Figure 2: Visualization of Complexity Calculation

A 3D shape with a complexity of 0 would indicate a shape than in all its axes, any cutting plane taken, would have a total of 4 lines of symmetry (i.e., symmetrical horizontally, vertically, and both diagonals). A 3D shape that would have a complexity of 0 would be a cube. Figure 3 shows an example representation of the complexity metric for a cube and a wedge. As shown in Fig. 5, any cutting plane of the cube would have all 4 lines of symmetry, while the wedge would only have 1 (i.e., gray dotted lines). Therefore, the complexity value of the wedge is greater than the cube.

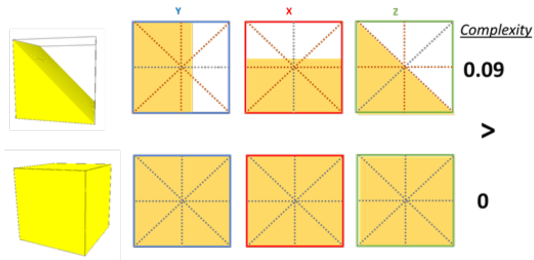

Figure 3: Complexity Calculation Examples

3.2 RL Agent Reward Function

Since there are a total of 8 voxels each with12 wedges that can be enabled or disabled to create a 3D shape, the size of the possible combinations is equal to 2 to the power of 12 by 8, or $7.9*10^{28}$ (2^{96}) . The possible wedges combinations grow rapidly as more voxels are added. This presents a valuable opportunity to use RL to generate shapes since the number of potential combinations to generate different shapes grows rapidly with the number of voxels used, which makes it intractable to test all possible combinations.

The method introduced in this work uses an RL agent that implements a Neural Network that takes as input a vector $[S, C]$, where S is an 8 by 12 matrix containing binary variables $S_{(i,j)}$, $i \in \{1..8\}$ and $j \in \{1..12\}$ that indicates if a given wedge j of voxel i is enabled or disabled. C is the desired complexity of the 3D shape the agent needs to generate and can range between values of 0 and 1. The network outputs an 8 by 12 matrix of probabilities X, where $X_{(i,j)}$, $i \in \{1..8\}$ and $j \in \{1..12\}$ represent the probability if a given wedge j of a voxel i should be enabled or disabled. This probability vector is then used to update the vector S using a threshold of 0.5. This configuration creates an environment with a multi-discrete action space, which allows the agent to make a shape at each step and make changes to the network weights after each step, based on the reward function.

Equation (1) show the reward function that takes as input the current shape and the desired complexity $[S, C]$, and returns a reward based on how close the complexity of the shape generated is to the desired complexity. Small penalties are detracted from the overall reward for each wedge used, and any additional wedges of a voxel used beyond the complementary wedges. This is to help mitigate the sparsity problem since having any additional wedges enabled in a voxel after two complementary wedges are enabled would not generate a different shape. This is because once two complementary wedges on a voxel are enabled a cube is the only shape that can be generated using that voxel. Moreover, if an empty shape is returned (i.e., no wedges enable), it receives a penalty to deter it from returning no wedges enabled (i.e., no shapes).

$$R(S,C) = \begin{cases} 0 \text{ if } \sum_{i=1}^{8}\sum_{j=1}^{12} S_{i,j} = M \\ 1 - |C - c(S)| - \beta(\frac{\sum_{j=1}^{12} S_{i,j}}{100}) - \theta(\gamma_i) \text{ otherwise} \end{cases} \tag{1}$$

Where:

- S is the matrix containing the shape being passed in.
- $S_{i,j}$ is a binary variable that equals 1 if the wedge j of voxel i is enabled, $i \in \{1..8\}$ and $j \in \{1..12\}$. passed in.
- C is the desired complexity.
- $c(S)$is the complexity of the given shape S
- γ_i counts the number of wedges enable in voxel i beyond any pair of complementary wedges.
- M is a negative number that serves as a penalty.
- β and θ are positive values that serves as weights

During each training epoch, the agent starts by randomly generating a desired complexity C between 0 and 1 from a discrete uniform distribution, and feeds that into the Neural Network with a S matrix that has no wedges enabled (i.e., no shape). The matrix S is "flattened" to a vector of length 96 before passing as input to the network. The agent creates a probability vector for the wedges and generates a new shape as a result, then feeds it back into the Neural Network with C.

4 Results and Discussion

The RL agent was trained on a Windows computer with an Intel© Core™ i7-9750H 2.6 GHz CPU and 32 GB of RAM. The RL agent and Neural Network were implemented in Python using the Keras library.

Some initial hyperparameter tunning was performed in which the number of layers and number of neurons per layer of the RL agent neural network, the β and θ weights of the reward functions, and the optimization algorithms learning rate were explored. From these initial results, an RL agent with a fully dense Neural Network with 10 hidden layers, each having 30 neurons was implemented. The hidden layers used a rectified linear unit activation function while the output layer used a SoftMax activation function. After each training epoch, the loss function for the neural network is calculated from the weights of the network, and gradients are collected until the network is updated. The β and θ weights of the reward functions are set to 0.6. Each epoch the agent will try to find a shape that maximizes the reward (i.e., max value of 1) up to 10 times before the desired complexity changes. The weights of the network are updated after every 10 training episodes, using an Adam optimization algorithm with a learning rate of 0.001.

Figure 4, shows the reward of the RL agent over a range of 3,000 training episodes. On the first 200th training iterations, the RL agent achieved an average reward of -2.56 (Min=-3.650, Max= -1.04, SD=0.54). However, in the last 200th iterations, the RL agent achieved an average reward of 0.67 (Min=0.37, Max= 0.93, SD=0.15). This indicates that the RL agent generated an action policy that significantly improves the reward functions since, on average, the differences in the rewards in the first and last 200th iterations are statistically different ($p-value$<0.001). Similarly, the variance is statistically different ($p-value$<0.001) indicating that the training process allowed the RL agent to generate an action policy that is more consistent at generating 3D shapes that maximize the reward (e.g., with complexity similar to the desired complexity).

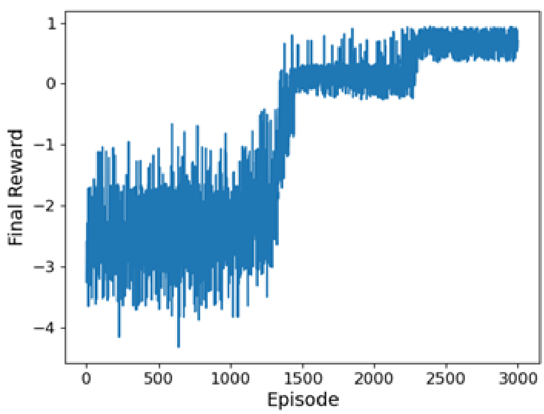

Figure 4: Reward vs Training Episodes

Nevertheless, it is important to highlight that even at the end of the training process the action policy is not able to generate a 3D shape with the same complexity as the desired complexity in 100% of the cases. This could be attributed to the RL agent still not finding the optimal policy and requiring more training or hyperparametric tuning, and because the training environment is randomly picking the desired complexity between 0 and 1 from a discrete uniform distribution with two decimal points of precision. Hence, the training environment might ask the RL agent to generate a shape with a complexity of 0.89 and the latter one with a complexity of 0.9. However, it might not be feasible with the current 3D shape generator systems to create two 3D shapes one with a complexity of 0.89 and another with a complexity of 0.9.

While these results showcase the challenges that raise from the sparse reward space and the current training environment, they also indicate that the RL agent managed to create an action policy that is capable of generating different shapes that have a complexity similar to the given desired complexity. Hence, this work lays the foundation for using RL agents to automatically generate new 3D shapes of the desired complexity. This could potentially be used in VR applications designed to help develop spatial visualization skills.

5 Conclusion and Future Work

Spatial visualization skills are of great importance in STEM fields. While VR has been shown to help develop these skills, most of the existing applications do not necessarily tailor their content to students' skills level. Automatically generating 3D shapes with Procedural Content Generation (PCG) methods, could help VR applications tailor spatial visualization tasks to the skills level of students, as well as generate a wider range of tasks.

This work introduces a PCG method based on a Reinforcement Learning (RL) approach to automatically generate 3D shapes of different complexities. The results indicate that an RL-agent is capable of creating an action policy that can generate 3D shapes with complexities similar to a given desired complexity provided. This could potentially help a VR application design to help develop students' spatial visualization and automatically generated different 3D shapes for the spatial visualization tasks that are in line with students' skill levels. However, the results also show that the task of automatically generating 3D shapes that meet a given complexity is not trivial. Moreover, they show that there are a lot of areas for improvement.

One area that could help improve generalizability and the training of the RL agent, would be the 3D shape generation system. It can be improved by increasing the number of voxels it uses to generate shapes, which will allow a wider range of shapes to be created. In addition, it should be updated so that 3D shapes with curvature are feasible to generate. Lastly, the generation system should be improved to reduce the sparsity of the solution space. With regards to the complexity metric, the authors are already exploring how the complexity metric can be updated to not only help with the sparsity issues but also to better capture individuals' perceived complexity of shapes used in spatial visualization tasks.

Acknowledgement

This research was partially funded by the EXCEL scholar program at Lafayette College. Any opinions, findings, or conclusions found in this paper are those of the authors and do not necessarily reflect the views of the sponsors.

References

[1] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38, 2017.

[2] Kinnari Atit, David I Miller, Nora S Newcombe, and David H Uttal. Teachers' spatial skills across disciplines and education levels: Exploring nationally representative data. *Archives of Scientific Psychology*, 6(1):130, 2018.

[3] Patrick Carlson, Anicia Peters, Stephen B Gilbert, Judy M Vance, and Andy Luse. Virtual training: Learning transfer of assembly tasks. *IEEE transactions on visualization and computer graphics*, 21(6):770–782, 2015.

[4] Lynn A Cooper. Mental rotation of random two-dimensional shapes. *Cognitive psychology*, 7(1):20–43, 1975.

[5] Matthew Coxon, Nathan Kelly, and Sarah Page. Individual differences in virtual reality: Are spatial presence and spatial ability linked? *Virtual Reality*, 20(4):203–212, 2016.

[6] James Cunningham, Christian Lopez, Omar Ashour, and Conrad S Tucker. Multi-context generation in virtual reality environments using deep reinforcement learning. In *International Design Eng. Technical Conf. and Comp. and Information in Eng. Conf.*, volume 83983, page V009T09A072. American Society of Mechanical Engineers, 2020.

[7] Zachary S Gold, James Elicker, Ashleigh M Kellerman, Sharon Christ, Aura A Mishra, and Nina Howe. Engineering play, mathematics, and spatial skills in children with and without disabilities. *Early Education and Development*, 32(1):49–65, 2021.

[8] Tibor Guzsvinecz, Éva Orbán-Mihálykó, Erika Perge, and Cecilia Sik-Lányi. Analyzing the spatial skills of university students with a virtual reality application using a desktop display and the gear vr. *Acta Polytechnica Hungarica*, 17(2):35–56, 2020.

[9] Mark Hendrikx, Sebastiaan Meijer, Joeri Van Der Velden, and Alexandru Iosup. Procedural content generation for games: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 9(1):1–22, 2013.

[10] Danial Hooshyar, Moslem Yousefi, and Heuiseok Lim. A procedural content generation-based framework for educational games: Toward a tailored data-driven game for developing early english reading skills. *Journal of Educational Computing Research*, 56(2):293–310, 2018.

[11] Danial Hooshyar, Moslem Yousefi, Minhong Wang, and Heuiseok Lim. A data-driven procedural-content-generation approach for educational games. *Journal of Comp. Assisted Learning*, 34(6):731–739, 2018.

[12] Britton Horn, Steve Dahlskog, Noor Shaker, Gillian Smith, and Julian Togelius. A comparative evaluation of procedural level generators in the mario ai framework. In *Foundations of Digital Games 2014, Ft. Lauderdale, Florida, USA (2014)*, pages 1–8. Society for the Advancement of the Science of Digital Games, 2014.

[13] Ionel-Alexandru Hosu and Traian Rebedea. Playing atari games with deep reinforcement learning and human checkpoint replay. *arXiv preprint arXiv:1607.05077*, 2016.

[14] Wen Huang. *Investigating the novelty effect in virtual reality on stem learning*. PhD thesis, Arizona State University, 2020.

[15] Christian E Lopez, Omar Ashour, and Conrad S Tucker. Reinforcement learning content generation for virtual reality applications. In *International Design Eng. Technical Conf. and Comp. and Information in Eng. Conf.*, volume 59179, page V001T02A009. American Society of Mechanical Engineers, 2019.

[16] Christian E Lopez, Omar M Ashour, and Conrad Tucker. An introduction to the click approach: Leveraging virtual reality to integrate the industrial engineering curriculum. In *ASEE annual Conf. & exposition*, 2019.

[17] Christian E López, James Cunningham, Omar Ashour, and Conrad S Tucker. Deep reinforcement learning for procedural content generation of 3d virtual environments. *Journal of Computing and Information Science in Eng.*, 20(5), 2020.

[18] Yukiko Maeda, So Yoon Yoon, Gyenam Kim-Kang, and PK Imbrie. Psychometric properties of the revised psvt: R for measuring first year engineering students' spatial ability. *International Journal of Eng. Education*, 29(3):763–776, 2013.

[19] Guido Makransky, Stefan Borre-Gude, and Richard E Mayer. Motivational and cognitive benefits of training in immersive virtual reality based on multiple assessments. *Journal of Comp. Assisted Learning*, 35(6):691–707, 2019.

[20] Guido Makransky and Lau Lilleholt. A structural equation modeling investigation of the emotional value of immersive virtual reality in education. *Educational Technology Research and Development*, 66(5):1141–1164, 2018.

[21] Tassos A Mikropoulos and Antonis Natsis. Educational virtual environments: A ten-year review of empirical research (1999–2009). *Computers & Education*, 56(3):769–780, 2011.

[22] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[23] Rafael Molina-Carmona, María Luisa Pertegal-Felices, Antonio Jimeno-Morenilla, and Higinio Mora-Mora. Virtual reality learning activities for multimedia students to enhance spatial ability. *Sustainability*, 10(4):1074, 2018.

[24] Takudzwa Mujuru and Christian Lopez. Creating virtual reality teaching modules for low-cost headsets. In *International Design Eng. Technical Conf. and Comp. and Information in Eng. Conf.*, volume 85376, page V002T02A083. American Society of Mechanical Engineers, 2021.

[25] Beatrice Ottiger, Erwin Van Wegen, Katja Keller, Tobias Nef, Thomas Nyffeler, Gert Kwakkel, and Tim Vanbellingen. Getting into a “flow” state: a systematic review of flow experience in neurological diseases. *Journal of neuroengineering and rehabilitation*, 18(1):1–21, 2021.

[26] James W Pellegrino, David L Alderton, and Valerie J Shute. Understanding spatial ability. *Educational psychologist*, 19(4):239–253, 1984.

[27] Jaziar Radianti, Tim A Majchrzak, Jennifer Fromm, and Isabell Wohlgenannt. A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda. *Computers & Education*, 147:103778, 2020.

[28] Cristina Roca-González, Jorge Martín Gutiérrez, Melchor García-Dominguez, and María del Carmen Mato Carrodeguas. Virtual technologies to develop visual-spatial ability in engineering students. *Eurasia Journal of Mathematics, Science and Technology Education*, 2017.

[29] Luiz Rodrigues, Robson Parmezan Bonidia, and Jacques Duílio Brancher. A math educacional computer game using procedural content generation. In *Brazilian Symposium on Comp. in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 28, page 756, 2017.

[30] Vicente Román-Ibáñez, Francisco A Pujol-López, Higinio Mora-Mora, Maria Luisa Pertegal-Felices, and Antonio Jimeno-Morenilla. A low-cost immersive virtual reality system for teaching robotic manipulators programming. *Sustainability*, 10(4):1102, 2018.

[31] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

[32] Sheryl Sorby. A new and improved course for developing spatial visualization skills. In *2001 Annual Conf.*, pages 6–66, 2001.

[33] Sheryl Sorby, Norma Veurink, and Scott Streiner. Does spatial skills instruction improve stem outcomes? the answer is 'yes'. *Learning and Individual Differences*, 67:209–222, 2018.

[34] Sheryl A Sorby, Gavin Duffy, Norman Loney, Lance Perez, et al. Spatial skills and their correlation with engineering problem-solving. In *29th Australasian Association for Eng. Education Conf.*, volume 2018, page 10, 2018.

[35] Liam Stewart and Christian Lopez. Developing spatial visualization skills with virtual reality and hand tracking. In *International Conf. on HCI*, pages 390–398. Springer, 2021.

[36] Sreenivas R Sukumar, David L Page, Andreas F Koschan, and Mongi A Abidi. Towards understanding what makes 3d objects appear simple or complex. In *2008 IEEE Comp. Society Conf. on Comp. Vision and Pattern Recognition Workshops*, pages 1–8. IEEE, 2008.

[37] Brian N Verdine, Roberta Michnick Golinkoff, Kathy Hirsh-Pasek, and Nora S Newcombe. I. spatial skills, their development, and their links to mathematics. *Monographs of the society for research in child development*, 82(1):7–30, 2017.

[38] Diego Vergara, Manuel Pablo Rubio, and Miguel Lorenzo. On the design of virtual reality learning environments in engineering. *Multimodal technologies and interaction*, 1(2):11, 2017.