Proceedings of the ASME 2019 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2019
August 18-21, 2019 Anaheim, CA, USA

DETC2019-97711

REINFORCEMENT LEARNING CONTENT GENERATION FOR VIRTUAL REALITY APPLICATIONS

Christian E. Lopez[1]
[1]Department of Industrial and Manufacturing Engineering
The Pennsylvania State University,
University Park, PA 16802
Email: cql5441@psu.edu

Omar Ashour
Department of Industrial Engineering
The Pennsylvania State University,
Erie, PA 16563
Email: oma110@psu.edu

Conrad S. Tucker[1,2]
[2]School of Engineering Design, Technology and Professional Programs
The Pennsylvania State University
University Park, PA 16802
Email: ctucker4@psu.edu

ABSTRACT

This work presents a Procedural Content Generation (PCG) method based on a Neural Network Reinforcement Learning (RL) approach that generates new environments for Virtual Reality (VR) learning applications. The primary objective of PCG methods is to algorithmically generate new content (e.g., environments, levels) in order to improve user experience. Researchers have started exploring the integration of Machine Learning (ML) algorithms into their PCG methods. These ML approaches help explore the design space and generate new content more efficiently. The capability to provide users with new content has great potential for learning applications. However, these ML algorithms require large datasets to train their generative models. In contrast, RL based methods do not require any training data to be collected *a priori* since they take advantage of simulation to train their models. Moreover, even though VR has become an emerging technology to engage users, there have been few studies that explore PCG for learning purposes and fewer in the context of VR. Considering these limitations, this work presents a method that generates new VR environments by training an RL in a simulation platform. This PCG method has the potential to maintain users' engagement over time by presenting them with new environments in VR learning applications.

Keywords: Procedural Content Generation, Reinforcement Learning, Virtual Reality, Education, Learning.

1. INTRODUCTION

The objective of Procedural Content Generation (PCG) methods is to generate content automatically. PCG has been used by the gaming industry since the '80s [1], mainly to generate new game levels by manipulating game design elements (e.g., terrains, maps, objects locations, environment). There exist several advantages to automatically generate new content for the development and design of new applications [2]. For example, PCG methods can help reduce the resources needed to create new content. Moreover, content that is automatically generated can be tailored to an individual's preference in order to maximize the user experience. Finally, PCG methods can help designers explore the design space, and potentially help co-create more creative content [3]–[5].

In recent years, immersive Virtual Reality (VR) has become an emerging technology with great potential to improve learning [6], [7]. Immersive VR systems allow users to experience being physically present in simulated environments (e.g., virtual worlds) [8]. This capability offers a significant potential to make learning more engaging to individuals by facilitating experiential learning [9]. Furthermore, several studies have shown that VR can improve individuals' engagement and performance [10], [11]. Unfortunately, researchers caution that some of the positive effects of VR might be due to *novelty effects* (i.e., individuals' performance improves with the use of new technology due to the initial interest in the technology, and not due to actual learning gains) [12], [13]. Due to novelty effects, individuals' interest can diminish over time if they continue interacting with the same application; hence, the value of generating new VR environments for them to interact.

The use of PCG methods have the potential to improve an individual's experience and the capability of a learning application to engage them (e.g., replay value) [5], [14]. However, there has been a limited number of studies that have explored the use of PCG for learning purposes [15], [16]. Moreover, most of the recent PCG methods implement Machine Learning (ML) algorithms to generate new content, which requires some training dataset. In contrast, Reinforcement Learning (RL) based methods implement sensory input (e.g., pixels acquire from images in a video game) to generate efficient representations of complex situations and tasks through simulation [17]. Hence, there is no need to capture training data *a priori*, which can help reduce cost [3]–[5]. In addition, studies have shown that artificial agents based on RL algorithms are capable of accomplishing a diverse array of challenging tasks at human level performance (e.g., puzzles, logic games, strategy games) [18].

In light of the advantages that PCG has for learning applications and the potential of RL algorithms, this work presents a PCG method based on an RL approach that generates new environments for immersive VR learning applications. Figure 1 shows an outline of this method. In this method, a Neural Network RL agent generates new VR environments that are validated via a simulation platform. The RL agent generated the new environments according to individuals' preferences for the location of a subset of virtual objects. Once a new environment is generated, it can be used for learning purposes, and the user can interact with it in the immersive VR learning application. The ability to generate new environments for VR applications has the potential to improve users' learning gains.

2. LITERATURE REVIEW

2.1 Virtual Reality and Learning

Virtual Reality (VR) is an emerging technology that has shown great potential for improving the learning process [6], [7]. Researchers have used VR on a wide range of applications (e.g., education, training, manufacturing) [19], [20]. For learning applications, VR offers many advantages [7], [21]. Particularly, the capability of individuals to interact with virtual objects in real-time, as in traditional environments, makes VR an effective learning tool [22], [23]. VR applications provide a sense of presence and create a "*first person*" experience [24], [25]. Presence is defined as "*the subjective experience of being in one place or environment, even when one is physically situated in another*" [26, p. 225]. Hence, VR offers great potential to make the learning process more engaging for individuals by facilitating experiential learning [9]. Furthermore, several studies have shown that VR can improve individuals' engagement and performance on a variety of learning activities [10], [11], as well as enhance understanding and reduce misconceptions [27].

Thanks to recent advancements in technology, VR systems are becoming more common and economically accessible [6]. For example, the recently released standalone VR headset Oculus Go™ costs only $199.00 (www.oculus.com/go/). Low-cost systems make the use of VR more accessible and affordable for learning applications. Particularly, researchers are developing new methods to leverage immersive VR in order to advance engineering education [28], [29]. However, while immersive VR systems are becoming more economically

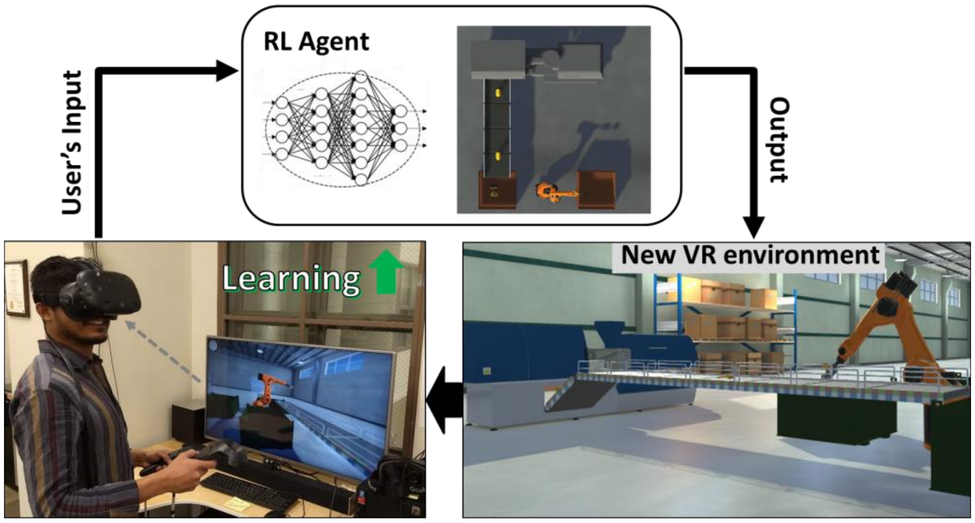

FIGURE 1. OUTLINE OF RL PCG METHOD

accessible and have shown potential to improve individuals' engagement, several factors can affect their user experience (UX) [27]. Studies have shown that factors such as the characteristics of the virtual environment or system use can impact UX [10], [30]. Moreover, studies indicate that some of the positive effects of VR might be due to *novelty effects* [12], [13]. Because of these effects, individuals' interest can diminish over time if they continue interacting with the same application; hence, the importance of generating new environments for VR learning applications. Considering these limitations, this work presents a PCG method for immersive VR learning applications. The method can generate new environments, which can be used by users of VR learning applications. The capability of generating new environments has the potential to mitigate possible novelty effects and maintain users engaged over time. Making sure individuals are motivated and engaged with a learning system has direct implications on improving their learning performance [31]–[33].

2.2 Procedural Content Generation

Procedural Content Generation (PCG) can be defined as the field that studies the development of algorithms and methods capable of generating content automatically. The gaming industry has used PCG for decades [1] and recently has started developing new methods founded on Artificial Intelligence (AI) algorithms [34], [35]. One of the most well-known examples of integrating ML and AI into PCG methods to generate new environments is for the game Super Mario Bros (www.marioai.org) [35]–[37].

While PCG methods are commonly used within the gaming community [34], there has been a limited number of studies that have explored the use of PCG methods for learning purposes [15], [16]. For example, Hullett and Mateas [14] present an application capable of generating new scenarios for a firefighting training application. The application was able to generate different scenarios of buildings partly collapsed based on the desired skills the users wanted to train on. Smith et al. [38] implement a PCG method for creating levels in a learning application aimed to teach students about fractional arithmetic. The method implements a constraint-focused generator design approach. Similarly, a learning application that implemented PCG and gamification to engage students in solving math problems is introduce in [39]. This PCG method was founded on template based and constructive algorithms.

In the context of conflict resolution, a serious game application that combined a Player Modeling and a metaheuristic-search PCG approach is introduced in [40]. This PCG method was driven by a Neural Network used to predict the distribution fairness of the players. The results of this study support the value of PCG to guide the learning of individuals toward targeted objectives. Most recently, Hooshyar [5], proposes a PCG framework for educational game applications based on a Genetic algorithm approach. The framework allows designers to control the generation process given various learning objectives and preferences. In a different study, [15] presents a data-driven PCG approach based on a Genetic and a Support Vector Machine algorithms. They implemented their method in a language learning application and compared the method against a heuristic based approach. Their results indicate that their data-driven approach was more effective at generating contents that matched the performance target of individuals compared to the heuristic approach.

The previous studies show how PCG methods can be implemented in learning applications and their potential benefits. These studies also show that researchers are starting to use ML and data-driven approaches (e.g., Neural Network, Support Vector Machines, Genetic algorithms) to train their PCG models. They train their models on datasets from existing content or datasets containing users' data, which has to be generated or collected *a priori* [5], [15], [40]. The process of generating new content can require significant time and resources [3]–[5]. In recent years, researches have started exploring how realistic, synthetic data can be automatically generated [41], [42]. However, while studies have shown that these approaches can generate synthetic datasets that cannot be accurately distinguished from human generated ones [43]–[45], they still require some initial datasets to train their models.

The objective of PCG methods to generate new environments given certain criteria is analogous to the Facility Layout Planning (FLP) optimization problem. The objective of FLP algorithms is to identify the optimal arrangement of equipment or facilities in accordance with some criteria and given certain constraints [46]. FLP problems are a NP-complete problem, which means that "*the computational time required to find an optimal solution increases exponentially with the problem size*" [47, p. 25]. This is one of the reasons why researchers have proposed multiple meta-heuristics algorithms to solve the FLP problem, such as Simulated Annealing and Genetic algorithms [46]. However, one of the limitations of optimization approaches is that a given optimal solution might not continue to be optimal under a different problem configuration. For example, if an additional constraint is added (e.g., now machine *Z* must be in the coordinates *x* and *y*), the algorithm needs to be run again to find the optimal or near-optimal solution.

2.3 Reinforcement Learning

While traditional supervised ML algorithms require the use of a training dataset, RL methods do not require a training dataset *a priori* since they take advantage of simulation environments to generate efficient representations of complex situations and tasks [17]. RL can be described as a Markov Decision Processes, were the RL agent connect to a simulation environment via different sensory inputs. The objective of the agent is to develop a model that selects the actions that maximize its long-run reward. In other words, the agents learn the desired action policy by a process of trial and error via simulation [48]. RL methods are suitable for solving learning control problems, which are challenging for traditional supervised ML algorithms and dynamic programming optimization methods [49]. Since RL agents focus on generating an action policy, they can adapt to changes in the environment without the need for additional training (e.g., now machine *Z* must be the coordinates in *x* and *y*), which is not the case for most optimization methods. Researchers have used RL methods to train agents capable of mastering complex tasks at human level performance (e.g., puzzles, Atari games, the Chinese game of Go)[18], [50], [51].

TABLE 1. SUMMARY OF EXISTING WORKS

Reference	*Meta-Heuristics*	*Supervised ML*	*RL*	*Learning Context*	*VR*
[1], [35]–[37], [52]		X			
[14] [38] [39] [40] [5]	X			X	
[15]		X		X	
This work			X	X	X

Table 1 shows a summary of existing work related to PCG methods. As it can be shown from this table, researchers have started integrating ML algorithms into their PCG methods. Moreover, while PCG methods are frequently used in gaming applications, researchers are starting to explore the use of PCG methods for learning purposes. However, most of the studies on learning applications implement meta-heuristics, and they rarely focus on automatically generating content for VR learning applications. In light of the advantages of PCG methods and the potential of RL algorithms, this work presents a PCG method based on an RL approach that generates new environments for VR learning applications. The approach validates the new virtual environments via a simulation platform; hence, it does not require any training data to be collected *a priori*. The PCG method presented has the potential to maintain users' engagement over time and mitigate possible novelty effects by presenting them with new environments in immersive VR applications that promote experiential learning. Promoting experiential learning with immerse environments while ensuring that individuals are engaged with the learning system has direct implications on improving their learning performance [9], [31]–[33].

3. METHOD

In this work, a PCG method based on an RL approach is introduced. The method is capable of dynamically generating new environments for immersive VR learning applications by implementing an RL agent that validates the content via a simulation platform. Figure 2 shows the framework of the RL algorithm implemented and examples of 2D aerial views of the simulation platform used to validate the environments generated. The RL model can be described as a Markov Decision Processes, were the agents connect to the simulation environment on a given time *t* via the sensory inputs of state ($\mathbf{S}_t$) and action ($\mathbf{A}_t$) (see Fig. 2). In each training epoch *t,* the agent takes as input the current state: $\mathbf{S}_t$ and chooses an action to be executed: $\mathbf{A}_t$. This produces a state transition in which the environment reacts to the action executed and provides a reward signal (i.e., reinforcement): R_t. The sensory inputs of the state and action can be in a vector form, containing information about the state of the environment and information regarding the action the agent is taking, respectively.

For example, in the task of learning how to play the game of chess, a chess game simulator can be used to train an RL agent. In every training epoch *t*, $\mathbf{S}_t$ will contain information about the location of all the game pieces in the 8x8 grid, while $\mathbf{A}_t$ will contain information about the locations where all the black game pieces should be moved (i.e., assuming the RL agent is playing with the black game pieces, this indicates the next moves). In the case, the intention is to train a RL agent to generate new manufacturing layouts for learning purposes (as in Fig.1 & 2), $\mathbf{S}_t$ will contain information about the current state of the layout (e.g., locations of a subset of equipment), while $\mathbf{A}_t$ will contain information about the locations of the reminder equipment, as shown in Fig. 2.

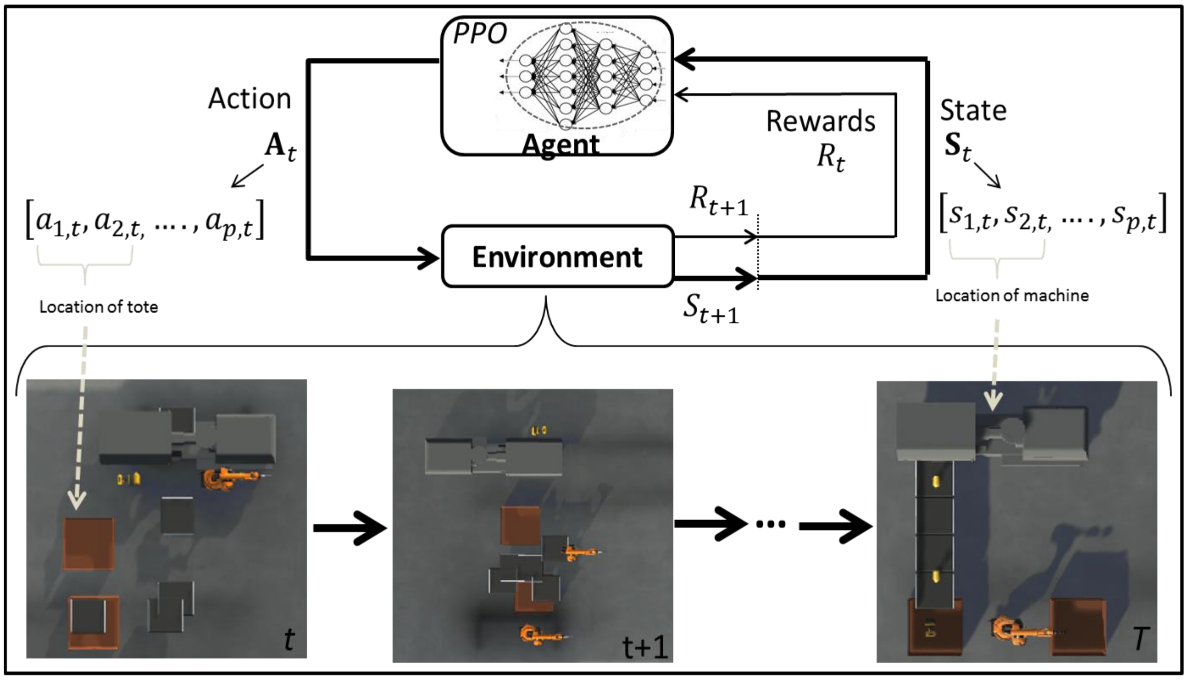

FIGURE 2. OUTLINE OF RL METHOD

The goal of the agent is to develop a model that selects the actions that maximize the long-run reward signal, which takes the form of a scalar value. Going back to the example of the chess game, the agent could be rewarded every time that it manages to checkmate the king of the opponent and penalized every time its king is checkmated. The function could also reward the agent based on the number of opponent's pieces removed from the game and penalize the agent based on the number of its own pieces removed. The elements of the reward function will depend on the underline behavior that the designers expected the RL agent to model (i.e., learn).

For the PCG method to generated VR environments that provide a sense of presence and create a "*first person*" experience to the users, the environments need to be realistic. Therefore, the RL agent needs to be rewarded for generating new environments that are functional and not just a random placement of virtual objects. This can be achieved by designing a reward function that incentivizes the generation of functional environments and penalizes nonfunctional ones (e.g., makes parts as in the manufacturing layout example of Fig. 1 & 2). Through its interactions with the simulated environment, the RL agent is trained (i.e., learns) to model an action policy that will maximize its reward function. Once the RL agent is trained, it will be able to generate new VR environments given an initial state provided by the user (i.e., user input, see Fig.1). In the example shown in Fig.2, this could be the initial location of the injection molding machine. Hence, the RL agent will place the reminder equipment in a way that will create a functional manufacturing layout.

In this work, the Proximal Policy Optimization (PPO) algorithm was employed to train the RL agent [53]. PPO is a Neural Network policy gradient ascent-based algorithm capable of moving between sampling data through interactions with the simulation environment. The algorithm can update its gradient policy more than once per simulation iteration. Schulman et al.'s [53] study reveals that the PPO algorithm outperformed other policy gradient algorithms, and provided a more favorable tradeoff between sample complexity, simplicity, and wall-time. PPO is based on the Trust Region Policy Optimization (TRPO) algorithm introduced by Schulman et al.'s work [54]. For details about the PPO algorithm implementation, readers are referred to [53].

The PCG method presented will generate new VR environments to be used for learning purposes. The capability to generate new environments has the potential to facilitate experiential learning in a wide range of learning contexts. As in the example of generating manufacturing layouts shown in Fig. 2, the method presented can help generate new layouts that could potentially be used to teach a range of engineering concepts, from the basics of stochastic processes (e.g., Poisson distribution) to the analysis of complex manufacturing system (e.g., Little's Law).

4. CASE STUDY

For this case study, the authors used a VR learning application designed to teach Industrial Engineering (IE) concepts (i.e., Poisson distribution, Little's Law, Queuing Theory) with the use of a simulated manufacturing system. Specifically, a manufacturing system that produces power drills was simulated, as shown in Fig.1 & 2. The objective of this VR application is to provide a tool with a common theme that educators could use to teach IE concepts and integrate course knowledge into their curriculum. A power drill manufacturing line was selected since previous studies that aim to integrate IE course knowledge have implemented similar power tools [55]. The virtual environment simulates the initial steps of the process to manufacture a power drill. In these steps, the plastic housing of the drill is manufactured.

Figure 3 shows, from a user's point of view, a functional layout for this manufacturing system. In this layout, first, an injection molding press produces the plastic housing components. Then, they are cooled down with the use of a conveyor belt. Finally, the plastic housings are placed in a tote with the use of a robotic arm in order to be transported to the assembly line. In the VR module, students can interact with the parts and machinery of the virtual systems.

For this application, the agent is rewarded based on the efficiency and functionality of the layout generated to produce goods (e.g., the rightmost image on Fig. 2 has a high reward, while the two other images have a low reward). Specifically, the reward function used in this case study can be mathematically expressed as follows:

$$R = \beta_1(Tote) - \beta_2(Floor) - \beta_3(\sigma_\lambda) + \beta_4(Flow) \quad (1)$$

For,

$$Tote = \sum_{p=1}^{P} \phi_p \quad (2)$$

$$Floor = \sum_{p=1}^{P} \varphi_p \quad (3)$$

$$\sigma_\lambda = \sqrt{\frac{\sum_{e=1}^{E}(\lambda_e - \bar{\lambda})^2}{E}} \quad (4)$$

$$Flow = \sum_{p=1}^{P} \sum_{e=1}^{E} \Delta_{p,e} \quad (5)$$

$$\beta_i > 0 \quad \forall \ i = \{1,2,3,4\} \quad (6)$$

Where,

- ϕ_p is a binary variable that indicates if a given part *p* was correctly placed in a tote $\phi_p = 1$ or not $\phi_p = 0$, for $p \in \{\mathbf{P}\}$
- φ_p is a binary variable that indicates if a given part *p* fall to the floor $\varphi_p = 1$ or not $\varphi_p = 0$, for $p \in \{\mathbf{P}\}$
- λ_e is a parameter that describes the behavior distribution of equipment *e*, for $e \in \{\mathbf{E}\}$
- $\Delta_{p,e}$ is a binary variable that indicates if a given part *p* interacted with a given equipment *e*, for $p \in \{\mathbf{P}\}$ and $e \in \{\mathbf{E}\}$

The reward function shown in Eq. (1), will be maximized when all the parts *p* are placed in a tote and no parts fall on the floor, following Eq. (2) and (3), when the standard deviation of the parameters that describe the behavior distribution of the equipment set $\{\mathbf{E}\}$ is minimized, following Eq. (4), and when all the parts interact with all the equipment following the manufacturing process, as shown in Eq. (5). This reward function was designed to reinforce the generation of functional manufacturing layouts that have a constant flow of parts to the tote. This reward function will be computed for every simulation epochs *t* (R_t), as shown in Fig. 2. In addition, in every simulation epoch *t*, the RL agent will be able to control the placement (x_e, y_e), orientation (θ_e), and the parameters that describes the behavior distribution (λ_e) of

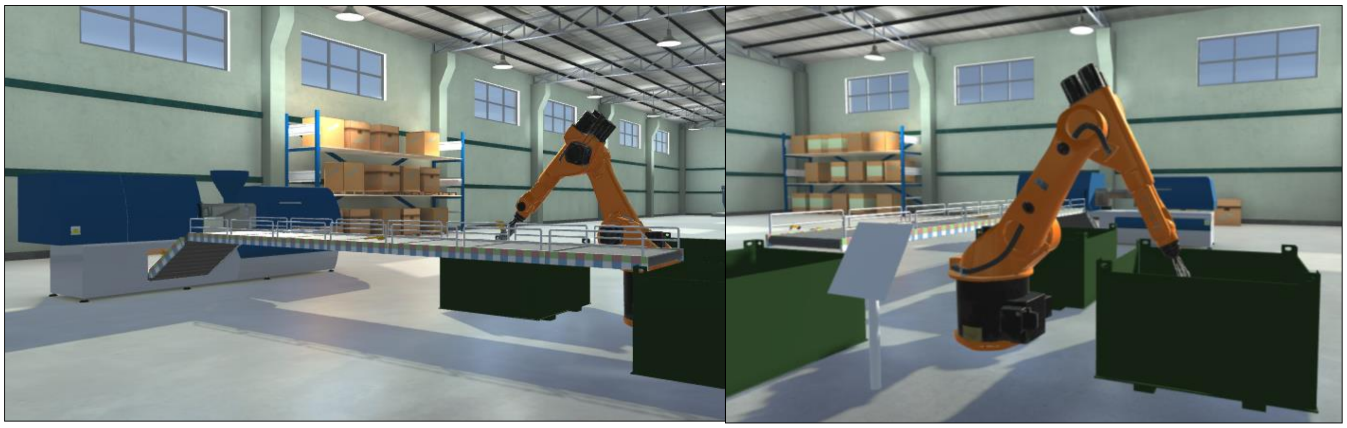
FIGURE 3. USER'S VIEW OF A FUNTIONAL LINE LAYOUT

the equipment set {**E**}. The environment will provide the agent with the state information about the placement (x_u, y_u) of the equipment placed by the user {**U**}. The set of equipment {**U**} will allow users (e.g., students, teachers) to customize the VR environment. In the case the user does not want to customize the environment, the equipment set {**U**} can be placed randomly, to generate a new environment.

For this application, users can select the location of the injection molding machine, **U** = {Injection molding machine}. In the other hand, the RL agent will manipulate one conveyor belt, one tote, and one robot arm. This means that the set **E** will contain three different equipment (i.e., virtual objects). The $\lambda_{conveyor}$ parameter will control the speed of the conveyor, while λ_{robot}, will control the rate at which the robot can handle parts. The tote is used for storing, and its parameter λ_{tote} will control the rate at which it can store parts.

Finally, in this work, the game engine Unity [56] is used as the simulation platform to train the RL agent. Because of its fidelity, physics simulation capabilities, accessibility, and community support, Unity is widely used by developers and designers of learning game applications [57], [58], as well as by researchers [59], [60]. Furthermore, Unity recently launched its ML-Agents Toolkit [61], which provides several algorithms and functionalities for the development and design of RL applications [62]. For each simulation epoch t, a total of 10 parts were simulated (i.e., p = {1-10}). This number of parts was selected to reduce the complexity of the simulation while allowing the simulation platform to generate the state transition in which the environment reacts to the action executed and provides a reward signal. However, this number can be increased, and the relative difference between the rewards score of layouts would not change. That is, a layout that allows all the parts to fall on the floor will always have a worse reward than one that places all the parts on the tote, no matter how many parts are simulated.

5. RESULTS AND DISCUSSION

The RL agent was trained using an ASUS Q551LB Intel® Core™ i7-5500U 2.40 GHz CPU and 8 GB RAM. A total of 75,000 training iterations (t = 75000) were used to train an RL agent in the simulated environment shown in Fig. 2. Figure 4 shows the evolution of the RL agent's reward score given the training epoch t. The y-axis shows the bounds of the reward function (i.e., [-10, 20]). While the plot shows that the RL agent did not achieve the maximum reward score on the given simulation iterations, it shows that the agents' rewards score was significantly and strongly correlated with the simulation iterations (ρ = 0.98, *p-value* < 0.001). This indicates that the agent managed to train a model that describes an action policy that maximized the long-run rewards function use in this case study. Due to computational limitations, the RL agent was not trained for a longer period of time. Just training with t = 75,000 took a total of 338.52 minutes (5.66 hours) on the equipment described above. However, once a model is trained it takes less than a second for the agent to generate a new layout given the machine location.

In addition, in this training process the location of the injection molding machine (i.e., environment state :{ x_u, y_u}) were randomly updated every 25,000 simulation iterations (dotted line in Fig. 4). This was done to help on the modeling of the RL agent's action policy and facilitate its generalization (i.e., trained to generate new environments without being constrained on the location of the machine). As it can be seen in Fig. 4, after the environment state was updated at t = 25000 and t = 50000, the reward function was not significantly affected. This indicates that the RL agent was able to model an action policy that was generalizable (i.e., transfer learning between state environments [63]). These results indicate that the RL agent managed to model an action policy capable of generating new VR environments using the location of the injection molding machine as an input.

This finding shows promising results for using PCG methods based on RL algorithm to generate new environments for immersive VR learning applications. For example, Fig. 5 shows how the environment can be integrated into the VR application used in this case study, to teach individuals about IE concepts. However, the PCG

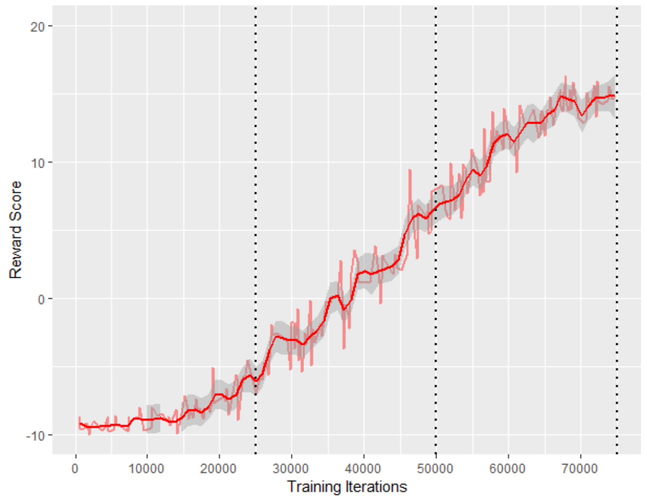

FIGURE 4. RL AGENT REWARD vs. TRANING

FIGURE 5. STUDENTS INTERACTING WITH THE VR APPLICATION

method presented can be applied to other VR learning applications, since it is not constrained to only the application used in this case study. Other VR application can benefit from the PCG method presented and using RL agent to generate new environments. Designers can implement this method in their VR applications by creating a rewards function based on the environment they would like the RL agent to generate.

6. CONCLUSIONS

VR has become an emerging technology that can facilitate experiential learning and improve individuals' learning gains. Unfortunately, researchers caution that some of the positive effects of VR might be due to novelty effects. Because of these effects, individuals' motivation and engagement can diminish over time if they continue interacting with the same environments. Implementing PCG methods allows designers to generate new environments that can help improve the user experience and learning. The most recent PCG methods develop integrate ML algorithms, which allow designers to explore the design space and generate new content more efficiently. However, these algorithms require large datasets to train their generative models. In contrast, RL methods do not require any training data to be collected *a priori* since they take advantage of simulation. Moreover, while PCG offers many advantages to sustain users' engagement and motivation over time, there have been few studies that explore the use of PCG for learning purposes and fewer in the context of VR applications.

In light of these limitations, this work presents a PCG method based on an RL approach that generates new environments for VR learning applications. This method trains a model by implementing an RL agent that validates new environments via a simulation platform; hence, it does not require any training data to be collected *a priori*. This PCG approach has the potential to sustain individuals' engagement in VR learning applications and mitigate possible novelty effects by presenting them with new virtual environments. Promoting experiential learning with immersive VR environments while ensuring that users are engaged with the learning system has direct implications on improving their learning performance. In this work, the method presented is implemented in a VR learning application designed to teach IE concepts with the use of a simulated manufacturing system. The preliminary results indicate that the RL agent was able to model (i.e., learn) a policy that allows it to automatically generate new and functional VR environments that can be used for learning purposes.

While this work presents a novel PCG method based on an RL approach, there still exist a lot of areas for improvement. First, the method should be tested with multiple VR learning applications that differ from the one used in this work in their degree of complexity and context of the application. Moreover, while the RL approach does not require the collection of any data *a priori* since it takes advantage of simulation to train its model, the reward function, which impacts the action policy the RL agent models, can be challenging to design under certain conditions. One of the biggest areas for improvement is the training procedure of the RL agent used in the case study. Future work should take advantage of the GPU and multi-agent parallelization capabilities of the Unity ML-toolkit. This will drastically reduce the time required to train the RL agent in the method presented. Moreover, future work should explore how the implementation of this PCG method can impact the motivation and learning of users that interact with VR learning applications, as shown in Fig 5.

ACKNOWLEDGMENTS

This research is funded by the National Science Foundation NSF DUE #183446. Any opinions, findings, or conclusions found in this paper are those of the authors and do not necessarily reflect the views of the sponsors. The authors would also like to thank the hard work of Bradley Nulsen, Gerard Pugliese Jr., Adith Rai, and Matthew Rodgers on developing and implementing the application used in this work.

REFERENCES

[1] A. Summerville and M. Mateas, "Super mario as a string: Platformer level generation via lstms," *arXiv Prepr.*, vol. arXiv:1603, 2016.

[2] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne, "Search-based procedural content generation: A taxonomy and survey," in *IEEE Transactions on Computational Intelligence and AI in Games*, 2011.

[3] R. Bidarra, K. J. de Kraker, R. M. Smelik, and T. Tutenel, "Integrating semantics and procedural generation: key enabling factors for declarative modeling of virtual worlds," in *FOCUS K3D Conference on Semantic 3D Media and Content*, 2010.

[4] G. Smith, J. Whitehead, and M. Mateas, "Tanagra: Reactive planning and constraint solving for mixed-initiative level design," in *IEEE Transactions on Computational Intelligence and AI in Games*, 2011.

[5] D. Hooshyar, M. Yousefi, and H. Lim, "A Procedural Content Generation-Based Framework for Educational Games: Toward a Tailored Data-Driven Game for Developing Early English Reading Skills," *J. Educ. Comput. Res.*, vol. 56, no. 2, pp. 293–310, 2018.

[6] L. Jensen and F. Konradsen, "A review of the use of virtual reality head-mounted displays in education and training," *Educ. Inf. Technol.*, 2018.

[7] S. Kavanagh, A. Luxton-Reilly, B. Wuensche, and B. Plimmer, "A systematic review of Virtual Reality in education," *Themes Sci. Technol. Educ.*, vol. 10, no. 2, pp. 85–119, 2017.

[8] Z. Merchant, E. T. Goetz, L. Cifuentes, W. Keeney-Kennicutt, and T. J. Davis, "Effectiveness of virtual reality-based instruction on students' learning outcomes in K-12 and higher education: A meta-analysis," *Comput. Educ.*, vol. 70, pp. 29–40, 2014.

[9] W. Winn, M. Windschitl, R. Fruland, and Y. Lee, "When Does Immersion in a Virtual Environment Help Students Construct Understanding ?," in *Challenge*, 2002.

[10] W. S. Alhalabi, "Virtual reality systems enhance students??? achievements in engineering education," *Behav. Inf. Technol.*, 2016.

[11] D. Janßen, C. Tummel, A. Richert, and I. Isenhardt, "Towards measuring user experience, activation and task performance in immersive virtual learning environments for students," in *Communications in Computer and Information Science*, 2016.

[12] G. Tsaramirsis, S. M. Buhari, K. O. Al-Shammari, S. Ghazi, M. S. Nazmudeen, and K. Tsaramirsis, "Towards simulation of the classroom learning experience: Virtual reality approach," in *Proceedings of the 10th INDIACom; 2016 3rd International Conference on Computing for Sustainable Global Development, INDIACom 2016*, 2016, pp. 1343–1346.

[13] M. Akçayır and G. Akçayır, "Advantages and challenges associated with augmented reality for education: A systematic review of the literature," *Educ. Res. Rev.*, vol. 20, pp. 1–11, 2017.

[14] K. Hullett and M. Mateas, "Scenario generation for emergency rescue training games," in *Proceedings of the 4th International Conference on Foundations of Digital Games - FDG '09*, 2009.

[15] D. Hooshyar, M. Yousefi, M. Wang, and H. Lim, "A data-driven procedural-content-generation approach for educational games," *J. Comput. Assist. Learn.*, vol. 34, no. 6, pp. 731–739, 2018.

[16] D. Hooshyar, M. Yousefi, and H. Lim, "A systematic review of data-driven approaches in player modeling of educational games," *Artificial Intelligence Review*, pp. 1–27, 2017.

[17] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 26–38, 2017.

[18] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, 2015.

[19] D. A. Guttentag, "Virtual reality: Applications and implications for tourism," *Tour. Manag.*, 2010.

[20] S. Choi, K. Jung, and S. Do Noh, "Virtual reality applications in manufacturing industries: Past research, present findings, and future directions," *Concurr. Eng. Res. Appl.*, 2015.

[21] L. Freina and M. Ott, "A literature review on immersive virtual reality in education: state of the art and perspectives," in *Conference proceedings of »eLearning and Software for Education« (eLSE)*, 2015, pp. 133–141.

[22] O. Çalişkan, "Virtual field trips in education of earth and environmental sciences," in *Procedia - Social and Behavioral Sciences*, 2011, pp. 3239–3243.

[23] P. Barata, M. Filho, and M. Nunes, "Consolidating learning in power systems: Virtual reality applied to the study of the operation of electric power transformers," *IEEE Trans. Educ.*, vol. 58, no. 4, p. 255–261., 2015.

[24] M. D. Dickey, "Brave New (Interactive) Worlds: A review of the design affordances and constraints of two 3D virtual worlds as interactive learning

environments," *Interact. Learn. Environ.*, vol. 13, no. 1–2, p. 121–137., 2005.

[25] L. Dawley and C. Dede, "Situated learning in virtual worlds and immersive simulations," in *Handbook of Research on Educational Communications and Technology: Fourth Edition*, M. Spector, M. D. Merrill, J. Elen, and M. J. Bishop, Eds. New York: Springer, 2014, pp. 723–734.

[26] B. G. Witmer and M. J. Singer, "Measuring presence in virtual environments: A presence questionnaire," *Presence Teleoperators Virtual Environ.*, 1998.

[27] T. A. Mikropoulos and A. Natsis, "Educational virtual environments: A ten-year review of empirical research (1999-2009)," *Computers and Education*. 2011.

[28] L. Stuchlíková, A. Kósa, P. Benko, and P. Juhász, "Virtual reality vs. reality in engineering education," in *15th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, 2017.

[29] C. Lopez, O. Ashour, and C. Tucker, "An introduction to CLICK: Leveraging Virtual Reality to Integrate the Industrial Engineering Curriculum," in *ASEE Annual Conference & Exposition*, 2019, pp. 1–12.

[30] L. M. Alves Fernandes, G. Cruz Matos, D. Azevedo, R. Rodrigues Nunes, H. Paredes, L. Morgado, L. F. Barbosa, P. Martins, B. Fonseca, P. Cristóvão, F. de Carvalho, and B. Cardoso, "Exploring educational immersive videogames: an empirical study with a 3D multimodal interaction prototype," *Behav. Inf. Technol.*, 2016.

[31] L. Corno and E. B. Mandinach, "The Role Of Cognitive Engagement in Classroom Learning and Motivation," *Educ. Psychol.*, vol. 18, no. 2, pp. 88–108, 1983.

[32] T. Chao, J. Chen, J. R. Star, and C. Dede, "Using digital resources for motivation and engagement in learning mathematics: Reflections from teachers and students," *Digit. Exp. Math. Educ.*, vol. 2, no. 3, pp. 253–277, 2016.

[33] K. J. Pugh, L. Linnenbrink-Garcia, K. L. K. Koskey, V. C. Stewart, and C. Manzey, "Motivation, learning, and transformative experience: A study of deep engagement in science," *Sci. Educ.*, vol. 94, no. 1, pp. 1–38, 2010.

[34] G. N. Yannakakis, "Game AI revisited," in *Proceedings of the 9th conference on Computing Frontiers - CF '12*, 2012.

[35] A. Summerville, M. Behrooz, M. Mateas, and A. Jhala, "The learning of zelda: Datadriven learning of level topology," in *Proceedings of the FDG workshop on Procedural Content Generation in Games.*, 2015.

[36] P. Shi and K. Chen, "Learning Constructive Primitives for Real-time Dynamic Difficulty Adjustment in Super Mario Bros," *IEEE Trans. Comput. Intell. AI Games*, vol. 10, no. 2, pp. 155–169, 2018.

[37] M. Guzdial, N. Sturtevant, and B. Li, "Deep Static and Dynamic Level Analysis : A Study on Infinite Mario," in *AIIDE Workshop AAAI Technical Report WS-16-22*, 2016, pp. 31–38.

[38] A. M. Smith, E. Andersen, M. Mateas, and Z. Popović, "A case study of expressively constrainable level design automation tools for a puzzle game," in *Proceedings of the International Conference on the Foundations of Digital Games - FDG '12*, 2012.

[39] L. Rodrigues, R. P. Bonidia, and J. D. Brancher, "A math educacional computer game using procedural content generation," in *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, 2017, p. 756.

[40] C. Grappiolo, Y. G. Cheong, J. Togelius, R. Khaled, and G. N. Yannakakis, "Towards player adaptivity in a serious game for conflict resolution," in *Proceedings - 2011 3rd International Conferenceon Games and Virtual Worlds for Serious Applications, VS-Games 2011*, 2011.

[41] M. L. Dering and C. S. Tucker, "Generative adversarial networks for increasing the veracity of big data," in *Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017*, 2018, pp. 2595–2602.

[42] C. Beecks, M. S. Uysal, and T. Seidl, "Gradient-based signatures for big multimedia data," in *IEEE International Conference on Big Data*, 2015, pp. 2834–2835.

[43] C. E. Lopez, S. R. Miller, and C. S. Tucker., "Exploring Biases Between Human and Machine Generated Designs," *J. Mech. Des.*, vol. 2, no. 141, p. 021104, 2019.

[44] C. Lopez, S. R. Miller, and C. S. Tucker, "Human validation of computer vs human generated design sketches," in *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2018, pp. DETC2018-85698.

[45] Y. Chen, S. Tu, Y. Yi, and L. Xu, "Sketch-pix2seq: a Model to Generate Sketches of Multiple Categories," *arXiv Prepr. arXiv1709.04121.*, 2017.

[46] H. Hosseini-Nasab, S. Fereidouni, S. M. T. Fatemi Ghomi, and M. B. Fakhrzad, "Classification of facility layout problems: a review study," *Int. J. Adv. Manuf. Technol.*, vol. 94, pp. 957–977, 2018.

[47] P. M. Pardalos, D. Z. Du, and R. L. Graham,

Handbook of Combinatorial Optimization. 2013.

[48] K. LP, L. ML, and M. AW, "Reinforcement learning : a survey," *Int J Artif Intell Res*, vol. 4, pp. 237–285, 1996.

[49] X. Xu, L. Zuo, and Z. Huang, "Reinforcement learning algorithms with function approximation: Recent advances and applications," *Inf. Sci. (Ny).*, vol. 261, pp. 1–31, 2014.

[50] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv Prepr.*, vol. arXiv:1312, 2013.

[51] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. Van Den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, p. 354, 2017.

[52] N. Shaker, J. Togelius, G. N. Yannakakis, B. Weber, T. Shimizu, T. Hashiyama, N. Sorenson, P. Pasquier, P. Mawhorter, G. Takahashi, G. Smith, and R. Baumgarten, "The 2010 mario ai championship: Level generation track," *IEEE Trans. Comput. Intell. AI Games*, 2011.

[53] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv Prepr.*, vol. arXiv, p. 1707.06347, 2017.

[54] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel, "Trust Region Policy Optimization," *arXiv.org*, p. 1502.05477, 2017.

[55] J. Terpenny, H. Harmonosky, A. Lehtihet, V. Prabhu, A. Freivalds, E. Joshi, and J. Ventura, "Product-based Learning: Bundling Goods and Services for an Integrated Context-rich Industrial Engineering Curriculum," in *Annual Conference of the American Society for Engineering Education (ASEE)*, 2018.

[56] W. G. & C. Pope, "Unity_Game_Engine," *UNITY GAME ENGINE Overv.*, 2011.

[57] P. Petridis, I. Dunwell, S. De Freitas, and D. Panzoli, "An engine selection methodology for high fidelity serious games," in *2nd International Conference on Games and Virtual Worlds for Serious Applications, VS-GAMES 2010*, 2010.

[58] A. Alsubaie, M. Alaithan, M. Boubaid, and N. Zaman, "Making learning fun: Educational concepts & logics through game," in *International Conference on Advanced Communication Technology, ICACT*, 2018.

[59] J. Cunningham and C. S. Tucker, "A Validation Neural Network (VNN) metamodel for predicting the performance of deep generative designs," in *Proc. ASME Int. Des. Eng. Tech. Conf. Comput. Inf. Eng. Conf.*, 2018, p. VT86299.

[60] C. E. Lopez, S. R. Miller, and C. S. Tucker, "Exploring Biases Between Human and Machine Generated Designs," *J. Mech. Des.*, vol. 141, no. 2, pp. 1–10, 2018.

[61] A. Juliani, V.-P. Berges, E. Vckay, Y. Gao, H. Henry, M. Mattar, and D. Lange, "Unity: A General Platform for Intelligent Agents," *arXiv Prepr.*, vol. arXiv:1809, 2018.

[62] Y. Burda, H. Edwards, D. Pathak, A. Storkey, T. Darrell, and A. A. Efros., "Large-scale study of curiosity-driven learning," *arXiv Prepr.*, vol. arXiv:, p. 1808.04355, 2018.

[63] Taylor, M. E., and P. Stone, "Transfer learning for reinforcement learning domains: A survey," *J. Mach. Learn. Res.*, vol. 10, pp. 1633–1685, 2009.