Proceedings of the ASME 2020
International Design Engineering Technical Conferences
& Computers and Information in Engineering Conference
IDETC/CIE 2020
August 16-19, 2020, St. Louis, MO, USA

DETC2020-19932

Multi-Context Generation in Virtual Reality Environments using Deep Reinforcement Learning

James Cunningham[1], Christian Lopez[2], Omar Ashour[3], Conrad S. Tucker[1,4]

[1]Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA

[2]Computer Science & Mechanical Engineering, Lafayette College, Easton, PA

[3]Industrial Engineering, The Pennsylvania State University, Erie, PA

[4]Machine Learning, Carnegie Mellon University, Pittsburgh, PA

ABSTRACT

In this work, a Deep Reinforcement Learning (RL) approach is proposed for Procedural Content Generation (PCG) that seeks to automate the generation of multiple related virtual reality (VR) environments for enhanced personalized learning. This allows for the user to be exposed to multiple virtual scenarios that demonstrate a consistent theme, which is especially valuable in an educational context. RL approaches to PCG offer the advantage of not requiring training data, as opposed to other PCG approaches that employ supervised learning approaches. This work advances the state of the art in RL-based PCG by demonstrating the ability to generate a diversity of contexts in order to teach the same underlying concept. A case study is presented that demonstrates the feasibility of the proposed RL-based PCG method using examples of probability distributions in both manufacturing facility and grocery store virtual environments. The method demonstrated in this paper has the potential to enable the automatic generation of a variety of virtual environments that are connected by a common concept or theme.

1. INTRODUCTION

The National Academy of Engineering (NAE) Grand Challenges of (i) advance personalized learning and (ii) enhance virtual reality, have the potential to transform the manner in which STEM education is taught and STEM contexts are experienced [1], [2]. However, a fundamental bottleneck that needs to be overcome is the creation of content in order to personalize the learning experience. A common misconception in today's society is that all VR experiences are created equal. However, in addition to the costs of the devices come the labor costs associated with actual content creation for the VR experiences. While the costs of VR headsets have seen a steady price decrease over the years, the cost of labor for VR content creation has not. For example, the consumer version of the Oculus Rift that was launched in mid-2016 had an original selling price of $599 [3]. Less than two years later, an updated version of the Oculus Rift with higher specs had a selling price of $399 [4]. Unfortunately, there is still a significant amount of manual labor needed to create VR experiences, especially as the complexity and fidelity of the VR experience increases.

Procedural Content Generation (PCG) has been proposed as an approach to overcome the costs and scalability challenges of content creation in VR [5]–[9]. Using this approach, content is generated algorithmically, which has the potential to adapt to users' needs. While PCG methods have shown promise, one of the fundamental challenges is the generation of diverse contexts that meet the needs of a diverse student base. This is of particular importance in the engineering education domain wherein a given concept (e.g., probability distributions), could be introduced in one course but have multiple causal implications in the knowledge learned in subsequent courses and ultimately, the workforce.

The proposed method advances state of the art in PCG by enabling the generation of multiple contexts that have the same underlying STEM conceptual content being delivered. For example, given the STEM concept of probability distributions, the algorithm learns how to represent a VR experience that teaches probability distributions in one contextual domain (e.g., a manufacturing facility layout), and demonstrates the same concept of probability distributions in a different contextual domain (e.g., a grocery store) having different environmental variables and constraints. The fundamental challenge in solving this problem is the diversity of variables and constraints that exist across different contexts that an automated system must learn in order to demonstrate the generalizability of context representations.

2. LITERATURE REVIEW

2.1 Virtual Reality in Education

Virtual Reality (VR) is a technology that has been used in a wide range of applications outside entertainment, such as education, training, and manufacturing [10], [11]. VR technology has shown positive impacts on the education and learning process [12], [13]. There are many advantages to using VR in education [13], [14]. The ability of users to interact with virtual objects in real-time, and the feeling of presence that creates a "first-person" experience offered by VR [15], [16], make it a better learning tool in some dimensions, when compared to traditional learning environments [17], [18]. Researchers define presence as "*the subjective experience of being in one place or environment, even when one is physically situated in another*" [19, p. 225]. Thus, VR provides the opportunity to implement experiential learning, which has the potential to improve students' motivation and engagement [20]. In addition to improving students' engagement, studies have reported that VR use has resulted in a better learning performance on a variety of learning activities [21], [22].

The recent advancements in VR hardware technology and the steady decrease in the cost of the VR devices, has made the equipment more economically accessible to users. However, creating meaningful VR experiences is still a major barrier due to the manual labor needed to create virtual environments. The problem becomes more pronounced as the complexity and fidelity requirements increase. While VR systems offer many advantages, including the potential of improving users' engagement, there are many factors that impact their user experience (UX) [23]. UX can be influenced by the characteristics of the virtual environment [21], [24]. In addition, studies have shown that *novelty effects* exist with VR applications and is one of the reasons why many VR applications initially have positive effects that are not sustained over time [25], [26].

Given the value of VR and the current state of how VR content is created, there is a need for an approach that is capable of generating multiple contexts of a VR environment that have the same underlying pedagogical content. This work extends the previous work of the authors [7], [9] by generalizing a Deep RL-based PCG method for multiple-contextual immersive VR learning environments. Each environment is built to deliver the same underlying pedagogical concept. The method has the potential to reduce the barriers of creating VR content and could potentially help solve the issues associated with novelty effects, which may lead to better user engagement over time. Motivation and engagement can directly impact the overall learning process [27]–[29]. Hence, this approach will also facilitate the development of personalized and adaptive learning applications to improve the learning process.

2.2 Procedural Content Generation

For decades, Procedural Content Generation (PCG) has been used by the gaming industry to automatically create content [30]–[32]. PCG employs different algorithms and methods to generate digital content and has taken advantage of recent advancements in Artificial Intelligence (AI) and Machine Learning (ML) [33], [34]. For example, Mario AI (marioai.org) is a framework that integrates AI algorithms into PCG methods to generate new environments for Super Mario Bros [35]. Algorithms such as Deep Convolutional Generative Adversarial Networks [36], Deep Reinforcement Learning [37], and recently, Generative Playing Networks [38] have been used to automatically generate game content.

Despite the fact that PCG methods are extensively used for gaming purposes [33], the use of PCG methods is rarely used in the education and training fields [39], [40]. For example, researchers developed a firefighting training application that takes advantage of PCG methods [41]. The application is able to generate new firefighting scenarios such as different buildings that are partly collapsed to provide training for the required skill level. Another example involves a learning application to teach fractional arithmetic [42]. The application utilizes a PCG method that implements a constrained-focused generator design to create multiple levels within the application. Similar to this application, a PCG method and gamification were used to improve students' engagement in learning math [43].

Player Modeling and a metaheuristic-search PCG approach were combined in a serious game application [44]. The PCG method was based on a Neural Network (NN) and used to predict the distribution fairness of the players. This application clearly shows the value of using PCG methods in guiding learners in achieving specific learning objectives. A recent study suggested a PCG framework based on a genetic algorithm approach [45]. The approach can be used for educational game applications in which the generation process can be controlled based on the desired learning objectives and preferences. Another study used a data-driven PCG method based on Genetic and Support Vector Machine algorithms in a language learning application [39]. The approach was more effective in generating content that matches the individual's performance target, compared to a heuristic-based approach.

The abovementioned studies demonstrate the applicability and benefits of using PCG in educational and training applications. Researchers have started integrating ML and data-driven approaches into PCG methods. However, current Supervised ML PCG methods need some datasets to train their models [39], [44], [45]. These training datasets need to be generated *a priori*. Nonetheless, the generation process of this content could involve significant time and resources [45]–[47].

In light of these limitations, the authors of this work introduced an RL-based PCG method to generate new 3D virtual environments [7], [9]. The results of the case study indicate that the method was capable of generating new virtual manufacturing environments with different locations and orientations of several virtual objects. While the previous work supports the capability of using RL algorithms to generate new virtual environments, it was not shown that the PCG method was capable of generating a diversity of contexts. Hence, this paper extends the authors' previous work by demonstrating the ability of the proposed Deep RL-based PCG method to generate a diversity of contexts in order to teach the same underlying pedagogical concept. The case study presents examples of statistical distributions being generated by the proposed method in both manufacturing facility and grocery store virtual environments.

2.3 Reinforcement Learning

Reinforcement Learning (RL) can be defined as a Markov Decision Processes in which the RL agent is connected to a simulation environment through multiple sensory inputs. Since it is connected to a simulation environment, RL methods do not require a training dataset, compared to traditional supervised ML algorithms [48]. The simulation environments offer a way to generate and test complex situations and tasks. The RL agent aims to choose the actions that maximize the long-run reward. In the process, the RL agents "*learn*" the preferred action policy using a process that resembles trial and error through simulation [49]. RL methods reduce the challenges of solving learning control problems when compared to traditional supervised ML algorithms and dynamic programming optimization methods [50]. RL methods do not require additional training when there are changes in the environments. This is because RL methods focus on generating an action policy that can adapt to changes in the problem space, which is not the case for most optimization methods. Recently, researchers have also trained agents using Deep RL methods to perform tasks as complex as tasks performed by humans (e.g., puzzles, Atari games, the Chinese game of Go) [51]–[53].

In the context of educational systems, RL has been used to personalized narrative-centered applications [54]. Similarly, multi-armed bandit computational formalism, resembling the Deep RL framework, as well as Long-Short Term Memory Networks approaches, were proposed as a method to generate new training scenarios for the Army [55]. In the context of video games, Deep RL has also been used for automatically generating new game levels and digital content [37], [56], [57].

TABLE 1: SUMMARY OF EXISTING WORKS

Reference	*Meta-Heuristics*	*Supervised ML*	*RL*	*Learning Context*	*VR*
[30]–[32], [34], [36]–[38], [57], [58]		X			
[41] [42] [43] [44] [45]	X			**single**	
[39]		X		**single**	
[7], [9]*			X	**single**	X
This work			X	**multiple**	X

* Author's previous work

The trend of integrating ML algorithms into PCG methods is on the rise. Table 1 shows a summary of related studies focusing on the use of PCG methods. Many studies have started to apply PCG methods in learning applications, while many existing studies have focused on using PCG methods in the gaming field. Nonetheless, studies related to learning applications have focused on using meta-heuristics. Moreover, the focus was not on automatically generating content for VR learning applications.

To reap the advantages of both PCG and RL methods, this work extends the authors' previous work on using a PCG method based on a Deep RL approach to automatically generate new environments for VR learning applications. The work presented in this paper seeks to add robustness to the prior work presented in [7], [9]. The previous work focused on generating multiple virtual environments for a simulated manufacturing system to teach the same underlying concepts. This work focuses on generating multiple contexts not only for a manufacturing system but also for a service system (i.e., grocery store). The different environments create a diverse set of variables and constraints. The authors hypothesize that the presented approach will be used to potentially improve and maintain students' engagement and motivation over time by providing new immersive VR environments that offer experiential learning experiences and hence potentially reducing the novelty effect. Engaging experiential learning experiences through immerse environments have positive impacts on the learning process [20], [27]–[29].

3. METHOD

This work introduces a Deep RL approach to PCG that is able to generate virtual environments of multiple contexts that are connected by an overarching theme. The method is able to dynamically accept input from the user that dictates what context the agent should generate and certain parameters of the environment for that context, and build an environment which has been validated via simulation to satisfy these parameters.

Figures 1 shows a user interacting with the virtual environment and Figure 2 outlines the proposed approach.

Figure 1: USER INTERACTING WITH THE MAMUFACTURING VIRTUAL ENVIRONMENT

The agent's decision is framed as a one-step Markov Decision Process (MDP) because the parameters that influence the agent's decisions (i.e., user provided parameters) are not dependent on any of the agent's previous decisions. Another term for a one-step MDP in the RL literature is a Contextual Multi-Armed Bandit (CMAB). The Multi-Armed Bandit problem in RL is inspired by the problem of deciding the best slot machine (or one-armed bandit) to play, assuming that different machines or "arms" have different payoffs, but that the agent must learn to estimate these payoffs through trial and error. A CMAB adds the complexity that the payoffs change for each arm over time, and the agent observes some side information that is helpful for determining what the arm's payoff at that time will be. Thus, the agent must learn the best arm to pull (or *action* to choose), conditioned on the information (or *state*) it observes at the time. Or equivalently, the agent must learn a *policy,* that maps its observed state to an action. This can be stated mathematically as:

$$\boldsymbol{a} = \boldsymbol{\pi}(\boldsymbol{s}) \tag{1}$$

where

- $\boldsymbol{a}$ is an action belonging to action space **A**
- $\boldsymbol{s}$ is the observed state belonging to state space **S**
- π is the policy function.

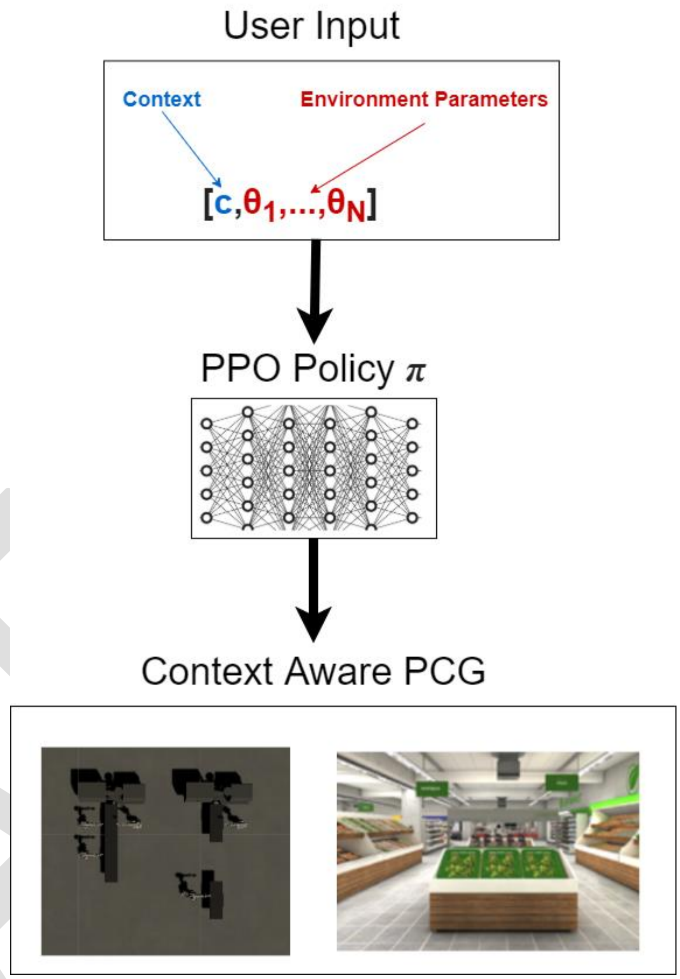

Figure 2: FLOW CHART OF PROPOSED METHOD

In the PCG context, the action corresponds to content generation decisions and the state corresponds to user provided parameters. This work improves upon the authors' previous PCG RL approach by adding robustness to multiple contexts by embedding context into the state space. Specifically, the state space in this work has the general form

$$[\boldsymbol{c}, \boldsymbol{\theta_1}, \ldots, \boldsymbol{\theta_N}] \tag{2}$$

where

- c is the context of the generated environment
- $\boldsymbol{\theta}$ are parameters that represent features of the generated environment that the user controls and are related to an overarching concept

By continuously exposing the agent to multiple contexts and user-provided parameters during training, the agent is able to generate environments that take into account user-input to demonstrate a common concept in multiple contexts.

An RL agent's goal is to find an optimal policy, π^*, that always chooses the action that maximizes the expected reward.

Let $\mu_{a|s}$ denote the mean reward when action, $\boldsymbol{a}$, is selected after observing state, $\boldsymbol{s}$. The optimal policy is defined by:

$$\boldsymbol{\pi}^*(\boldsymbol{s}) = \max_{\boldsymbol{a}} \boldsymbol{\mu}_{\boldsymbol{a}|\boldsymbol{s}} \qquad (3)$$

Of the many RL algorithms that are able to solve for π*, the Proximal Policy Optimization (PPO) algorithm is chosen for this work. PPO is a Deep Learning policy gradient approach to solving for π*, which means that it represents the policy function as a Neural Network (NN) and takes advantage of the fact that NNs are fully differentiable to directly take the derivative of the policy function with respect to a loss function based on an estimate of the advantage function of the policy (a quantity which is closely tied to the expected value of the policy's reward, readers are referred to [59] for more details). PPO is a Trust Region Policy Optimization method, which means that it limits the amount that a policy can deviate from the previous iteration's policy. This leads to stable training updates which are critical to solving an RL problem. Moreover, PPO provides a favorable tradeoff between sample complexity, simplicity, and wall-time [59].

Given that the goal of the agent is to generate an environment for a given context that dynamically takes into account user-defined parameters, the agent should be rewarded for how well it reflects these parameters in the environment. Through interactions with a simulated user input that randomly requests environments of different contexts and constraints, the agent is able to learn to generate environments across each of the simulated contexts that accommodate the user parameters to demonstrate an overarching concept.

4. EXPERIMENTS

The goal of the experiments in this work is to demonstrate the ability of the proposed method to procedurally generate virtual environments in multiple contexts that demonstrate a particular STEM concept. In this work, the concept of probability distributions was chosen to demonstrate the approach. The model is trained to generate environment layouts in two contexts: a manufacturing facility and a grocery store. The two contexts are different in many aspects such as the relationships between the objects in each context. The differences between these contexts will help demonstrate the robustness of the proposed approach. Sections 4.1 and 4.2 detail the specifics of each virtual environment context, and the ways in which the agent is capable of manipulating these environments.

The agent receives a total of 4-state space parameters from the user in the form $[c, \theta_1, \theta_2, \theta_3]$, where c is the context parameter, and θ_1, θ_2, and θ_3 are context-dependent probability parameters. In this work, the context parameter is binary and represents the manufacturing facility when 0 and the grocery store when 1. In both contexts, the agent must generate an action vector of length 3, where each element contains integers on the interval [0, 2], which also have context-specific meanings that are described in the following subsections.

All experiments were conducted on an Intel® Core™ i7-4770K 3.50 GHz CPU and 16 GB RAM. A total of 75,000 training iterations were performed on 32 parallel agents, for a total of 2.4 million virtual environment generations. The total training time was 20.3 minutes. Table 2 shows the mean and standard deviation of the reward of the virtual environments generated for each context of both the untrained (t=0) agent and trained (t=100,000) agent for 3200 generations with randomized user parameters.

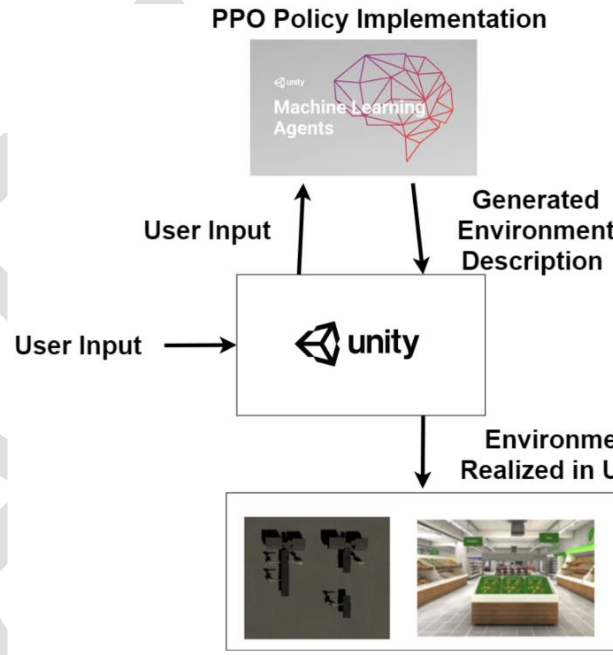

Figure 3: SOFTWARE ARCHITECTURE FOR PROPOSED METHOD

Figure 3 shows the software architecture used to implement the proposed method in these experiments. The Unity real-time engine was chosen to implement the 3D environments, in conjunction with the Unity ML-Agents SDK to implement the Deep RL-based PCG approach using PPO [60]. The user interacts directly with the application hosted in Unity, which passes this data to the ML-Agents PPO implementation. ML-Agents determines the environment to be generated and passes this information back to Unity, which it uses to realize the appropriate final 3D environment.

4.1 Manufacturing Facility

Figure 4 shows a virtual manufacturing line that produces power drills, where an injection molding press produces the plastic housing components. The housing components then cool down on a conveyor belt and are picked up by a robotic arm and placed in a tote. As shown in Figure 4, there are three discrete positions that the agent must make decisions about what to place. The elements of the action vector for this context represent:

0. Nothing
1. A conveyor belt and a robotic arm
2. A conveyor belt and two robotic arms

Figure 4 shows two possible configurations that the agent could generate for this environment. Note that in the rightmost configuration, the robotic arm that is located at the bottom does not contribute towards picking up drill housings, because it is not connected by a conveyor belt.

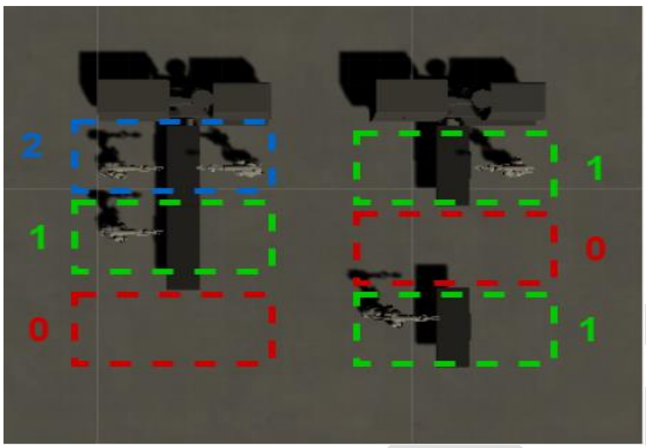

Figure 4: ILLUSTRATION OF MANUFACTURING ENVIRONMENT AND LAYOUT ACTIONS

The goal of this virtual environment is to demonstrate the concept of a Poisson distribution, which describes the probability of k occurrences of an event occurring in a fixed interval of time or space. It has the probability mass function $\lambda^k e^{\lambda}/k!$. For the manufacturing context, k represents the number of drill housings produced by the injection molding press per second. The state parameter Θ_1 corresponds to λ in the above probability mass function and corresponds the average amount of drill housings produced per second. Θ_2 controls the speed of the conveyor belt, such that Θ_2 is the amount of time taken for a drill housing to travel along the length of the conveyor belt in seconds. Finally, Θ_3 specifies the amount of time it takes for a robotic arm to pick up a drill housing, place it in the tote, and be ready to pick up another drill housing, also in seconds.

Thus, the agent must choose the number of robotic arms and conveyor belts needed to not allow any drill housings to land on the floor. This will be impacted by each of the three parameters specified by the user, as the first two will impact the throughput of drill housings from the injection molding machine, and the final parameter impacts the rate at which robotic arms can pick up the drill housings. If the rate is too low compared to the throughput, then drill housings will land on the floor. However, the agent is also penalized for each robotic arm it places, so it is incentivized to solve the problem with the least amount of resources possible.

Given N (10 in our experiments) drill housings are generated by the injection molding press, M total robotic arms are placed by the agent, and F drill housings dropped to the floor, the reward function for the context is:

$$\boldsymbol{R = N - M - F} \tag{4}$$

This task allows a student to dynamically visualize how a Poisson distribution with various parameters interact in a virtual environment that mimics a realistic Industrial Engineering application.

4.2 Grocery Store

Figure 5 shows the 3D virtual grocery store environment, where N customers will have a shopping list that contains a subset of items from an overall M possible items available in the grocery store. In these experiments, $N = 50$ and $M = 3$. For example, if the three possible items are pineapples, oranges, and bananas, each customer will have a subset of these three items on their shopping list. In these experiments, the 3 possible items that can be placed are pineapples, oranges, and bananas. Analogous to the manufacturing setting, there are 3 discrete positions (shelves) shown in Figure 5. For elements of the action vector, the agent selects one of the following:

0. Pineapples
1. Oranges
2. Bananas

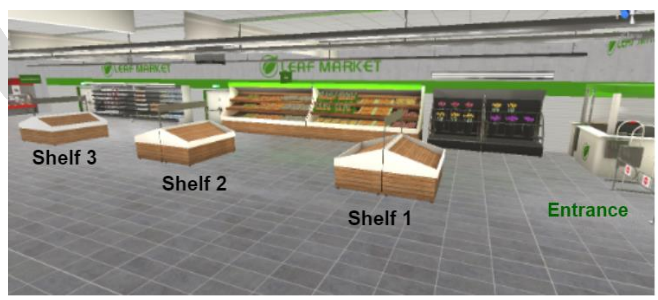

Figure 5: ILLUSTRATION OF GROCERY STORE ENVIRONMENT CONTEXT

The goal of this environment is to demonstrate the Bernoulli distribution, which takes the value 1 with probability p, and 0 with probability $(1 - p)$. The user-specified parameters Θ_1-Θ_3 correspond to the parameter p for the probability that each of the three grocery items will appear on a given customer's shopping list. The agent has control over which of the M items appears on each of the M shelves in the store. The shelves may be at varying distances from the entrance, and a single item may appear on more than one shelf.

The virtual customers will navigate the store to each item on their list and add them to their basket before checking out. If a customer did not find an item on their list on any shelf (e.g., because the agent left some item out by placing duplicates of other items), then a penalty is applied for each item where this occurred.

In addition to ensuring that all items are placed, the task of the agent is to minimize the average amount of time spent shopping by all customers, by deciding which items should appear on the closer shelves, and which items should appear on the more distant shelves. This reward function can be mathematically formulated as:

$$\boldsymbol{R} = \sum_{i=0}^{M} \boldsymbol{g}_i - \sum_{j=0}^{N} \boldsymbol{t}_j \qquad (5)$$

where

- g_i is 1 if the i-th item appears on any shelf and -1 otherwise
- t_j is the time spent shopping by the j-th customer.

Each action in the M-length action vector corresponds to placing an item in that position. The item placement actions will be informed by a state that contains the Bernoulli parameters for each item. For the fruit items example, if the Bernoulli parameters vector was 0.25, 0.5, and 0.75, respectively, then each customer will have pineapples on their list with probability of 0.25, oranges with probability of 0.5, and bananas with a probability of 0.75. In order to reduce the average amount of time spent in the store for each customer, the agent would want to place the bananas as close as possible to the entrance, and pineapples as far as possible from the entrance.

5. RESULTS AND DISCUSSION

Figure 6 shows the learning curve of the agent over the training iterations. The curve shows the average across all 32 parallel generations for that training iteration, and thus contains a mix of generations of both contexts. For this reason, the absolute value of the reward is less important than the fact that it monotonically increases to a saturation point where the agent cannot further improve the policy.

TABLE 2: REWARD OF TRAINED AND UNTRAINED AGENTS FOR EACH CONTEXT

	Mfg. Untrained	Mfg. Trained	Grocery Untrained	Grocery Trained
Mean Reward	2.78	7.13	1.09	2.19
STD Reward	3.93	2.01	0.986	0.256

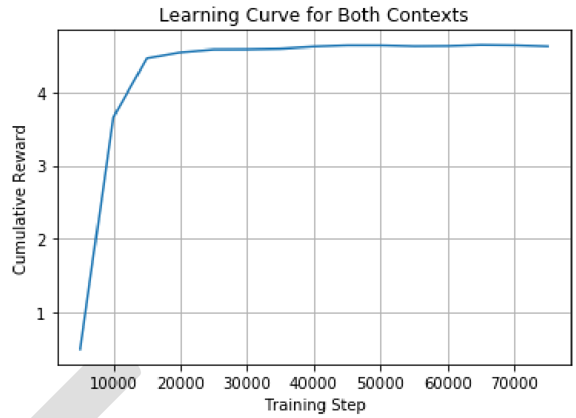

Figure 6: LEARNING CURVE OF REWARD

The results show that the agent is able to learn to maximize its reward function in multiple contexts across a diverse range of user input. In order to validate whether the agent achieved the original goal of generating environments that optimally exploit the user parameters, some test cases with easily verified solutions are visualized. Figure 7 shows the generated manufacturing environment for the user parameters of [3, 4, 2] and [1, 2, 2] on the left and right, respectively. In both cases, no drill housings touched the ground, but in the latter case, less robotic arms were utilized due to the smaller Poisson parameter values for the conveyor belt and the machine. This led to a lesser throughput of drill housings and thus the agent placed fewer robotic arms to optimize its reward function. Furthermore, there are no "stranded" conveyor belts such as the configuration on the right of Figure 4, where the robotic arm at the end does not contribute to the task. This suggests that the agent is able to generate the proper manufacturing environment given the user input, and the reward function aligns with the design goal of the agent.

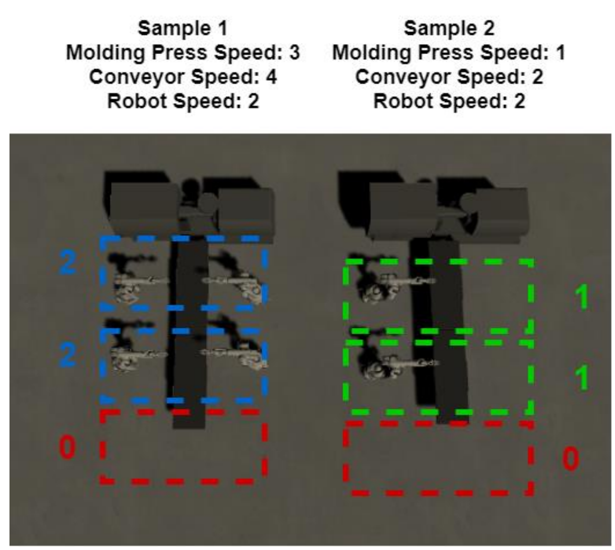

Figure 7: MANUFACTURING CONTEXT SAMPLE GENERATIONS. SPEED IS IN PARTS/SEC

Figure 8 shows sample grocery store generations for parameters of [0.16, 0.53, 0.41] and [0.78 ,0.53, 0.80]. In both cases, the items were placed progressively further away from the entrance in the order of decreasing Bernoulli parameter, and that all items were represented on the shelves. These samples suggest that the agent indeed learned to generate the appropriate environment based on the user input.

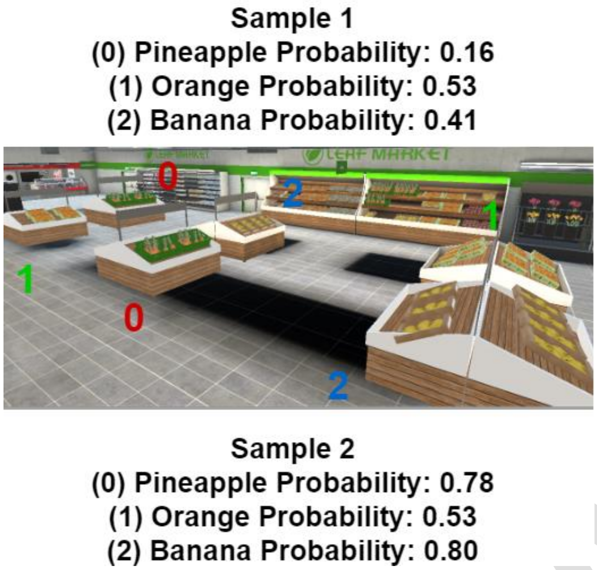

Figure 8:GROCERY STORE CONTEXT SAMPLE GENERATIONS

6. CONCLUSION

Virtual Reality (VR) technology has advanced quickly in recent years. While VR offers many advantages that has led to improvements in students' learning, motivation, and engagement, many studies have pointed out the impact of novelty effect in reducing the benefits of using VR in the classroom. The novelty effect might diminish individuals' motivation and engagement over time if they keep interacting with the same virtual environment. Moreover, while advances in VR hardware technology have been on the rise, the extensive resources needed to create VR content is still a barrier that hinders the wide use of VR in STEM education. PCG approaches to content generation not only reduce the resources required for product development and design, but increasingly offer the ability to customize content for an individual's needs. The application of ML in PCG advances the degree to which PCG can be customized over heuristic-based algorithms. However, there exists the drawback of requiring large datasets of example virtual environments to effectively generate new environments. In contrast, RL approaches do not require a dataset for training but instead, *learn* to generate valid environments through interaction with a simulation environment. One way to increase personalization of PCG methods is to allow the user to select from multiple contexts of environments to be generated.

In order to achieve this goal of personalization, this work presents a Deep RL-based PCG algorithm for the generation of virtual environments that include simulation environments in multiple contexts that are connected by a common theme in the training process. The resulting PCG algorithm is able to robustly generate environments multiple contexts. This furthers the degree to which PCG can personalize content.

The results of this work show that the proposed Deep RL-based PCG approach is able to take users' inputs which specify the parameters of probability distributions, and generate environments that demonstrate the underlying concept in both a manufacturing facility environment as well as a grocery store environment. However, this work addresses a simple version of the problem, where the context is expressed by a single parameter and the user input is limited to only three parameters. A future direction to expand this work would be to allow for less parameterized representation of contexts (e.g., using an image to represent the context) to allow for better generalizability.

Additionally, the impact of the proposed approach to increase personalization should be studied in a real classroom environment to assess its impact on motivation and learning outcomes. This work lays the foundation for these future extensions that may impact the level of personalization possible using PCG in educational settings.

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.

REFERENCES

[1] M. Miller, J. P. De Clerck, S. A. Sorby, L. M. Roberts, W. J. Endres, and K. D. Hale, "Meeting the NAE grand challenge: Personalized learning for engineering students through instruction on metacognition and motivation strategies," *ASEE Annu. Conf. Expo. Conf. Proc.*, 2013.

[2] J. A. Bennett and C. P. Saunders, "A Virtual Tour of the Cell: Impact of Virtual Reality on Student Learning and Engagement in the STEM Classroom †," *J. Microbiol. Biol. Educ.*, vol. 20, no. 2, pp. 2–4, 2019, doi: 10.1128/jmbe.v20i2.1658.

[3] J. P. Monge;, G. Lopez;, and L. A. Guerrero, "Advances in Human Factors and Ergonomics in Healthcare," vol. 482, pp. 309–315, 2017, doi: 10.1007/978-3-319-41652-6.

[4] Oculus, "Oculus Rift," 2018. .

[5] S. Azad, C. Saldanha, C. H. Gan, and M. O. Riedl, "Mixed reality meets procedural content generation in video games," *AAAI Work. - Tech. Rep.*, vol. WS-16-21-, pp. 22–26, 2016.

[6] J. P. A. Campos and R. Rieder, "Procedural content generation using artificial intelligence for unique virtual reality game experiences," *Proc. - 2019 21st Symp. Virtual Augment. Reality, SVR 2019*, pp. 147–

151, 2019, doi: 10.1109/SVR.2019.00037.

[7] C. Lopez, O. Ashour, and C. Tucker, "Reinforcement Learning Content Generation for Virtual Reality Applications," in *Proceedings of the ASME 2019 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE)*, 2019.

[8] J. W. Park and S. H. Oh, "A Study on Creation and Usability of Real Time City Generator via Procedural Content Generation: - Focus on virtual reality contents for senior," *2019 Int. Symp. Multimed. Commun. Technol. ISMAC 2019*, pp. 0–3, 2019, doi: 10.1109/ISMAC.2019.8836162.

[9] C. E. Lopez, J. Cunningham, O. Ashour, and C. S. Tucker, "Deep Reinforcement Learning for Procedural Content Generation of 3D Virtual Environments (accepted manuscript)," *ASME J. Comput. Inf. Sci. Eng.*, 2020, doi: 10.1115/1.4046293.

[10] D. A. Guttentag, "Virtual reality: Applications and implications for tourism," *Tour. Manag.*, 2010, doi: 10.1016/j.tourman.2009.07.003.

[11] 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, doi: 10.1177/1063293X14568814.

[12] L. Jensen and F. Konradsen, "A review of the use of virtual reality head-mounted displays in education and training," *Educ. Inf. Technol.*, vol. 23, no. 4, pp. 1515–1529, 2018, doi: 10.1007/s10639-017-9676-0.

[13] 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.

[14] 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, doi: 10.12753/2066-026X-15-020.

[15] 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, pp. 121-137., 2005, doi: 10.1080/10494820500173714.

[16] 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.

[17] O. Çalişkan, "Virtual field trips in education of earth and environmental sciences," in *Procedia - Social and Behavioral Sciences*, 2011, pp. 3239–3243, doi: 10.1016/j.sbspro.2011.04.278.

[18] 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, pp. 255-261., 2015, doi: 10.1109/TE.2015.2393842.

[19] B. G. Witmer and M. J. Singer, "Measuring presence in virtual environments: A presence questionnaire," *Presence Teleoperators Virtual Environ.*, 1998, doi: 10.1162/105474698565686.

[20] W. Winn, M. Windschitl, R. Fruland, and Y. Lee, "When Does Immersion in a Virtual Environment Help Students Construct Understanding ?," in *Challenge*, 2002, doi: 10.1016/j.vaccine.2008.02.054.

[21] W. S. Alhalabi, "Virtual reality systems enhance students??? achievements in engineering education," *Behav. Inf. Technol.*, 2016, doi: 10.1080/0144929X.2016.1212931.

[22] 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, doi: 10.1007/978-3-319-41769-1_4.

[23] T. A. Mikropoulos and A. Natsis, "Educational virtual environments: A ten-year review of empirical research (1999-2009)," *Comput. Educ.*, vol. 56, no. 3, pp. 769–780, 2011, doi: 10.1016/j.compedu.2010.10.020.

[24] L. M. Alves Fernandes *et al.*, "Exploring educational immersive videogames: an empirical study with a 3D multimodal interaction prototype," *Behav. Inf. Technol.*, 2016, doi: 10.1080/0144929X.2016.1232754.

[25] 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.

[26] 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, no. 1, pp. 1–11, 2017, doi: 10.1016/j.edurev.2016.11.002.

[27] 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, doi: 10.1080/00461528309529266.

[28] 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.

[29] 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, doi: 10.1002/sce.20344.
[30] A. Summerville and M. Mateas, "Super mario as a string: Platformer level generation via lstms," *arXiv Prepr.*, vol. arXiv:1603, 2016.
[31] 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.
[32] 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, doi: 10.1109/TCIAIG.2017.2740210.
[33] G. N. Yannakakis, "Game AI revisited," in *Proceedings of the 9th conference on Computing Frontiers - CF '12*, 2012, doi: 10.1145/2212908.2212954.
[34] 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.
[35] B. Horn, S. Dahlskog, N. Shaker, G. Smith, and J. Togelius, "A Comparative Evaluation of Procedural Level Generators in the Mario AI Framework," in *Foundations of Digital Games 2014*, 2014, pp. 1–8.
[36] V. Volz, S. M. Lucas, J. Schrum, A. Smith, J. Liu, and S. Risi, "Evolving Mario levels in the latent space of a deep convolutional generative adversarial network," in *GECCO 2018 - Proceedings of the 2018 Genetic and Evolutionary Computation Conference*, 2018, pp. 221–228, doi: 10.1145/3205455.3205517.
[37] N. Justesen, R. R. Torrado, P. Bontrager, A. Khalifa, J. Togelius, and S. Risi, "Illuminating generalization in deep reinforcement learning through procedural level generation," in *preprint arXiv*, 2018, p. arXiv :1806.10729.
[38] P. Bontrager and J. Togelius, "Fully Differentiable Procedural Content Generation through Generative Playing Networks.," *Prepr. arXiv*, 2020.
[39] 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, doi: 10.1111/jcal.12280.
[40] 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.
[41] 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, doi: 10.1145/1536513.1536538.
[42] 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, doi: 10.1145/2282338.2282370.
[43] 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.
[44] 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, doi: 10.1109/VS-GAMES.2011.39.
[45] 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, doi: 10.1177/0735633117706909.
[46] 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, doi: 10.1016/j.cag.2010.11.011.
[47] 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, doi: 10.1109/TCIAIG.2011.2159716.
[48] 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, doi: 10.1109/MSP.2017.2743240.
[49] K. LP, L. ML, and M. AW, "Reinforcement learning : a survey," *Int J Artif Intell Res*, vol. 4, pp. 237–285, 1996.
[50] 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, doi: 10.1016/j.ins.2013.08.037.
[51] V. Mnih *et al.*, "Playing atari with deep reinforcement learning," *arXiv Prepr.*, vol. arXiv:1312, 2013.
[52] D. Silver *et al.*, "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, p. 354, 2017, doi: 10.1038/nature24270.
[53] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, 2015, doi: 10.1038/nature14236.
[54] P. Wang, J. Rowe, W. Min, B. Mott, and J. Lester, "Interactive narrative personalization with deep reinforcement learning," in *IJCAI International Joint Conference on Artificial Intelligence*, 2017.
[55] J. Rowe, A. Smith, R. Pokorny, B. Mott, and J. Lester,

"Toward Automated Scenario Generation with Deep Reinforcement Learning in GIFT.," in *Proceedings of the Sixth Annual GIFT Users Symposium*, 2018, p. 65.
[56] R. R. Torrado, P. Bontrager, J. Togelius, J. Liu, and D. Perez-Liebana, "Deep reinforcement learning for general video game ai," in *IEEE Conference on Computational Intelligence and Games (CIG)*, 2018, pp. 1–8.
[57] N. Justesen, P. Bontrager, J. Togelius, and S. Risi, "Deep Learning for Video Game Playing," *IEEE Trans. Games*, 2019, doi: 10.1109/tg.2019.2896986.
[58] N. Shaker *et al.*, "The 2010 mario ai championship: Level generation track," *IEEE Trans. Comput. Intell. AI Games*, 2011, doi: 10.1109/TCIAIG.2011.2166267.
[59] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," pp. 1–12, 2017.
[60] A. Juliani *et al.*, "Unity: A General Platform for Intelligent Agents," pp. 1–28, 2018.