Quantum Cryptography Exercise Schedules with Concept Dependencies

Abhishek Parakh
University of Nebraska, Omaha
Omaha, USA
aparakh@unomaha.edu

Vidya Bommanapally
University of Nebraska, Omaha
Omaha, USA
vbommanapally@unomaha.edu

Parvathi Chundi
University of Nebraska, Omaha
Omaha, USA
pchundi@unomaha.edu

Mahadevan Subramaniam
University of Nebraska, Omaha
Omaha, USA
msubramaniam@unomaha.edu

***Abstract*—The design of a gamified instructional paradigm requires careful identification of concepts, concept dependencies, and concept flow in order to achieve maximum student proficiency, in a subject matter, while maintaining engagement. This is especially true for difficult and counter-intuitive fields such as quantum cryptography. In this paper, we present an abstraction of concepts that are needed to learn quantum key distribution in a gamified environment. This is coupled with a powerful adaptive navigation algorithm that guides students from one exercise to the next in the game such that maximum proficiency is achieved in various concepts associated with each exercise. The student traverses through different lessons in the game achieving the lesson outcomes in an efficient manner. This represents the first of its kind abstraction of quantum cryptography concepts and a navigation algorithm for a gamified paradigm.**

Keywords—serious games, engagement, exercise schedules, quantum cryptography

I. Introduction

In recent years gamification has become a trend covering a broad spectrum of multidisciplinary fields such as education, healthcare, defense, corporate training and advertising [1, 2]. Serious games have many definitions, the popular gist being games with a learning element along with an entertainment element [1, 2, 3]. [2] lists the advantages of serious games in various fields such as dance-pad for improving physical fitness through games, games for the purpose of rehabilitation, training games in corporate industry to minimize the teaching or the equipment costs and in education for improving logical thinking and so on. One of the challenging aspects of serious games is maintaining the player engagement, during the game, that might impact the learning outcomes [4][8-10, 21]. One of the factors that might affect the engagement is the cognitive load or application of knowledge. This is often related to design of exercises within the game and the distribution of relevant concepts, to be learnt, over these exercises [18, 19]. Failure to properly design the game may lead to lower engagement scores and lead to students spending long periods of time in navigational issues going from one exercise to the next. Navigational hints are a way of improving learning of player in the games also help in maintaining engagement [6] [7]. There are many ways to measure engagement of players in serious games depending on the game goals and various aspects inside the game [5].

Enhancing player learning abilities in serious games has become an area of research. In [11], authors try to enhance the learning ability of the players by implementing a pedagogical agent along with video tutorials in the game. These agents act as interactive assistants during the game play directing the player by providing necessary support through hints. [12] provides a design strategy for incorporating hints into the games where players leave the self-explanatory hints for the future players. [13] provides how serious games help in design and planning of a project to avoid accidents at workplace using a safedesign game.

An important feature of serious games is to adaptively improve the player performance [14]. The authors in [14] developed a virtual reality game to teach social engineering which assesses the player's performance by providing hints adaptively. Another way of adaptively improving the efficiency of the player is that prior game data is analyzed and hints are provided based on the current player performance [15] as well as update the assessments in the game dynamically. The authors used Bayesian network which is fed with the players game data, according to the data hints or feedback is provided in the help panel.

Quantum computing and cryptography is a growing field but remains inaccessible to a vast swath of student population because of the lack of courses at Universities and the lack of opportunities for hands-on training and experience. Given that this is a demanding field at the intersection of several disciplines such as computer science, Physics, mathematics and cybersecurity, it is difficult to maintain student interest, engagement and retention. One of the major challenges faced by educators, in this field, is the determination of appropriate breakdown of concepts and lessons and the flow between these concepts and lessons that must be followed for students, particularly, in a cybersecurity program. Multiple textbooks exist but most are written for students with strong Math, and Physics backgrounds. Furthermore, the flow that works in a textbook and traditional classroom setting does not necessarily translate to a gaming environment designed for teaching. This paper bridges this gap and presents a possible breakdown of concepts and lessons that may be used to develop a gaming environment for teaching quantum cryptography to cybersecurity students. Furthermore, we also present a navigation algorithm that can be used to direct a student between these concepts and lessons in order to gain proficiency in the subject. This flow was implemented and tested in an immersive gamified educational environment called QuaSim [11, 16, 17, 25, 26].

In a previous paper [7], the authors presented preliminary results on this concept flow graph. The previous results measured learning and engagement potential of the players in QuaSim using the knowledge concepts associated with the game and the player history. The paper attempted to improve the engagement and learning potential of the player by providing navigation hints thereby avoiding distractions to the player. [7] also introduced the concepts of hints in three different modes – manual, semi-automatic and automatic hints. In [7] QuaSim provided hints for the next exercise assuming the concepts that lead to learning goal in the game are independent of each other. In other words, we assumed that it is not required for the player to solve an exercise prior to solving another exercise. Each exercise has a value associated with it which determines the possibility of suggesting it as a next exercise to the player with reference to engagement potential. Exercises with new concepts (not yet encountered/solved by the player) in the current game session are given higher values and thus represent a higher engagement potential. Such exercises are suggested to the player iteratively. While, simultaneously learning potential is measured with respect to number of attempts the player takes before solving the exercise correctly. This paper, in part, presents further refinement of our previous results. In particular, we note that more often than not the concepts used and learned in one exercise depend on those in other exercises which may be deemed as pre-requisites by educators. In such cases, the next exercise maximizing the value metric must be chosen while taking these dependencies into account. This paper presents a novel algorithm that takes these dependencies into account while navigating through a landscape of concepts and lessons that are dependent on one another.

The paper structure is as follows: Section II discusses the guiding principles followed in identification of concepts and the abstract design of the game for quantum cryptography, Section III discusses our dynamic navigation procedure for traversing a concept graph for quantum cryptography, Section IV presents an overview of our game QuaSim and our updated navigation procedure with dependencies, Section V brings all the finer details together into a higher level view describing the various lesson dependencies in the game and Section VI concludes the paper.

II. A Systemic Approach to Teaching Quantum Cryptography

We distilled and identified the necessary concepts from quantum cryptography that a cybersecurity student needs to learn and internalize in order to understand the field. In doing so we employed a five-part model, called the Vowel Model, to develop the lesson plans [17]. This model helps to create learning efficiency and depth of content. The Vowel model consists of five instructional elements as described below:

A – Asking: In this first phase, the students can ask and get asked questions that are fundamental to completing a specific task, such as creating a quantum bit. This dialog phase allows for the creation of a testable hypotheses with a gaming environment for a cybersecurity student.

E – Exploring: The student is allowed to explore an immersive gaming environment and consult the oracle in the game to gain information about the challenge to be completed. This allows for the student to freely learn the content of a lesson before a formally structured instruction is delivered. This allows for the student to gain experience through trial and error and trying out different gaming elements.

I – Instruction: This allows for a formal instructional setting where the student in exposed to necessary concepts and ideas and completion of notational and symbolic exercises. This allows for the integration of important topics and demonstrate examples within the context of cybersecurity.

O – Organizing: This phase can be understood as a guided practice phase where the student actively engages with the game, completing gaming activities and exercises. The student often loops back to the Asking phase thereby addressing some of the early questions now based on formal instruction.

U – Understanding: This allows for the instructor to measure student progress in the game and proficiency in different concepts as the student attempts more and more exercises and proceeds through different lessons (levels) within the game.

A. Concepts, Exercises, and Schedules

Quantum cryptography involves an interplay of concepts from different inter-disciplinary domains including physics, mathematics, and computer security with subtle dependencies. A systematic approach including the organization of concepts, design of related exercises and a navigation approach enabling students to achieve proficiency in a stepwise fashion is crucial for effective instruction in this area. Such an approach can be adapted for several instruction modes including classroom lectures, educational games, and game-based teaching.

Informally we define a *quantum cryptography concept* (*qcc*) to be an indivisible unit of knowledge with a clearly stated learning objective(s) that are achievable by performing exercises and the learning progress (proficiency) can be assessed objectively. *A qcc* C is said to depend on *qcc* D if in order to achieve proficiency in C it is necessary to achieve proficiency in the concept D. A concept depends on a group of concepts if achieving proficiency in each member of the group is required to achieve proficiency on that concept. Quantum cryptography game unit consists of a set of *qccs*, $C = \{c_1, c_2, \ldots, c_k\}$ along with a set of exercises $E = \{e_1, e_2, \ldots, e_n\}$ that are designed to achieve proficiency over all the *qccs* in C. Achieving proficiency in all the *qccs* of C will result in meeting the learning objectives of the related unit. Each exercise in E is associated with one or more *qccs* from C. Relevancy of concepts to exercises is given by a tuple $R(e_i)$ which gives all the concepts that are hosted by an exercise e_i. Also, $E(c_j)$ gives the set of exercises which host a concept c_j. Note that each exercise can involve multiple concepts and a concept can be associated with multiple exercises. However, for any two concepts c_j and c_k and exercise e_i, if the two concepts belong to $R(e_i)$ then c_j and c_k must be independent of each other. Similarly, the sets of exercises hosting two dependent concepts must be disjoint. To measure the *concept proficiency*, each concept c_j is assigned a numeric value of 1 if the player successfully solves an exercise where the concept c_j resides else it is assigned with 0. The learning goal is measured as game unit *proficiency* $P(C)$ which is the Boolean tuple of concept proficiencies in all the concepts in C initially assigned with a value 0 for each concept. Hence, the player is said to achieve the learning goal if all the values in the tuple are 1.

The dependency among concepts introduces a dependency among the related exercises (see next subsection) and the order in which these exercises can be scheduled for a learner in a gaming session. For instance, if any *qcc* hosted by exercise e_j depends on that hosted by e_i then exercise e_i must be scheduled before exercise e_j. Several schedules of exercises are possible for a quantum cryptography game unit involving multiple concepts and multiple exercises. In order to achieve learning objectives in a robust manner devising schedules that can be completed with reasonable effort is crucial. Given potential variability in the learning styles and the uneven learning rates of learners, a navigation algorithm is described below to dynamically adapt schedules for individual learners.

III. Adaptive Dynamic Navigation Procedure

The navigation algorithm helps the player to navigate through the exercise space while maintaining engagement, and frustration is minimized. The next exercise is proposed to the learner based on a numerical value calculated for each exercise in the game unit. This is an iterative algorithm where the value of each exercise is calculated at every iteration. The value is calculated using,

$$v_i = \langle \frac{1}{|E(c_j)|}, c_j \in R(e_i) \rangle \qquad (1)$$

The value of each exercise is a tuple of values for each concept giving number of exercises the concept is relevant in. This value shows how essential is it to solve that exercise with respect to the concept and available exercises that cover the concept. The concept relevancy in all other exercises is updated with all the concepts that player has achieved proficiency in by setting the value to 0 as that concept is no more relevant to that exercise being achieved proficiency already. Also, the scenario proficiency is updated by assigning value 1 to the concepts where player achieved proficiency thereby tracking how far the player is from achieving the learning goal.

The above simple iterative procedure is sufficient to create customized dynamic schedule of exercises if there are no dependencies among the concepts in the schedule. When there exists dependency among the concepts, then an exercise involving a concept can be scheduled only after the learner has completed at least one exercise involving each of the concepts on which the original concept depends upon. A *concept graph* is a directed graph which represents relation between the concepts. There exists an edge from $c_1 -> c_2$ if concept c_2 is dependent on c_1. Using the concept graph and the set of tuples $R(e)$ relevant concepts for each exercise, an exercise graph is derived which is a directed graph. There exists an edge from $e_1 -> e_2$ if at least one concept in $R(e_2)$ depends on at least one concept in $R(e_1)$.

To select the next best exercise to the player taking into consideration the dependency among the concepts, edge weights are used. The value of each exercise (node value) is calculated using (1). The edge weight is calculated using the notion of *dependency overlap* and the *node value*. A *dependency overlap* from exercise e_i to e_j, $d(e_i, e_j)$ is

$$d(e_i, e_j) = N_{ji} - \delta * NR_{ji} \qquad (2)$$

N_{ji} = number of concepts in e_j depending on a concept of e_i

NR_{ji} = number of concepts common between e_j and e_i

δ = redundancy factor.

The dependency overlap $d(e_i, e_j)$ allows us to pick the next exercise node e_j such that maximum number of required dependencies among the concepts in e_j are met while minimizing the repetitive learning of concepts across the exercises e_i and e_j. For example, suppose $\mathrm{R}(e_i) = \{\mathrm{a, b, c, d}\}, \mathrm{R}(e_i) = \{\mathrm{e, f, c, d}\}$, $\mathrm{R}(e_k) = \{\mathrm{e, f, c, h}\}$, $\mathrm{R}(e_k) = \{\mathrm{e, f, c, h}\}$, then exercise e_k is preferred over e_i as the next exercise after e_i since it reduces the number of repetitive concepts (from 2 to 1) while meeting the same number of dependencies as e_i. To schedule exercises in increasing order of complexity while providing opportunities to learn new concepts in each exercise (freshness), we further refine edge weight to from node e_i to e_j as follows.

$$W_{e_i} -> W_{e_j} = \left(\frac{d(e_i, e_j)}{|R_{e_j}|} * v_{e_j} \right) \qquad (4)$$

Then, the next exercise is chosen based on maximal edge weight:

$$\text{Next exercise } (e_j) = \text{Max}\{W_{ei} -> W_{ej} \mid e_j\} \qquad (5)$$
$$\text{is adjacent to } e_i\}$$

Note that the parameter δ proportionately reduces the redundancy among non-dependent concepts. For example, if there are 200 concepts and 199 concepts are redundant between nodes e_i, e_j then the dependency overlap if δ is $1 - (0.01 * 199) = -0.99$, which reduces the node value excessively that the significance of the node carrying 1 dependent concept is lost. Hence, δ should be chosen in a way that preserves the significance of the node while reduces the priority compared to other nodes with fewer redundant nodes. Given N concepts, the parameter can be set as follows:

$$\delta = 10^{-\mathrm{ceil}(\log(N))} \tag{6}$$

The main steps of the dynamic schedule generation procedure is as follows. Initially, the source nodes (exercise with no dependent concepts) are queued. Once an exercise is successfully completed, edge weights of the nodes adjacent to the current node are calculated and updated into a weight matrix and the node with maximum weight is queued to the player as the next best exercise. The progress vector is updated, and the values of the exercises are recalculated using (1) and the process is repeated until all components of the progress vector are 1 or there are no exercises left that cover the remaining concepts. We use two queues named visited queue (V) that queues the exercises for the player, processed queue (P) which queues the nodes successfully solved by the player in order to explore further for the next exercises. Below, a node represents an exercise in the exercise graph.

1. Enqueue the base nodes into V. Base nodes are nodes that do not have any dependencies.
2. Dequeue each node, provide to the player for solving. and enqueue this node into P.
3. Repeat the process until V is empty.
4. After all the base nodes are processed, calculate all the *node weights* (1) using nhints (navigation hints algorithm)
5. Dequeue node from P say D_p and calculate *edge weights* (4) for all the nodes adjacent to D_p using *edge weight*.
6. Pick the child node with *maximum* edge weight and enqueue to V.
7. Dequeue the node D_v and provide it to player for solving.
8. *If* the player successfully solves and exercise (node) *then* enqueue the node D_v in to P
    a. *Else* re-calculate the node weights of all the nodes and the edge weights of D_p.
    b. Pick the node with maximum value and enqueue to V.
9. Remove node D_p from the exercise graph.
10. Calculate the node weights, if weight = 0, remove the node from the exercise graph.
11. Repeat from step 5.
12. If no child nodes exist for the current node D_p and all concepts are not learnt, go to step 1.
13. Repeat steps 1 - 10 until there are no exercises left or all the concepts are learnt.

IV. QUASIM

In this section we introduce the QuaSim serious game, its navigation hint system and the novel extended concept dependency handling mechanism. QuaSim is a virtual gamified education paradigm that teaches basic concepts of quantum computing and cryptography. The goals of QuaSim are three fold. First, it allows the internalization of counterintuitive quantum concepts that sit at the intersection of Physics, mathematics, computer science and cybersecurity [22, 23, 24]. Second, it provides a immersive environment for hands-on learning in the absence of expensive quantum equipment and field-training opportunities. And third, QuaSim enhances student learning and proficiency in relevant concepts while maintaining engagement through a gamified interface.

Figure 1 shows screenshots of the game. There are four major lessons in the game namely polarization, basis and measurement, quantum communication, and BB84 quantum key exchange. There are several sub-lessons such as matrix and Dirac notations, linear combination, quantum communication, channel noise detection, eve detection, etc.

Fig. 1. Screen shots from QuaSim game showing lessons 2 measurement (top) and 3 quantum communication (bottom)

QuaSim has been developed in Unreal Engine 4. The game includes several features to support student gameplay such as a narrator (including a mission statement), an oracle (when students want to seek help or the system detects a student needs help), several hint mechanisms, embedded videos and quizzes throughout the game, a web browser, calculator, various controls to fine-tune the environment and finally and most-importantly support for multiplayer scenarios for up to three players. All events and actions performed by the player are recorded in the game including the solutions of the player and their attempts to solve each exercise. This is used to analyze student performance and need for intervention (through the oracle) on the fly. The game is played in three versions categorized based on the hint mode namely manual, semi-automatic, and the automatic mode. Manual mode enables players to access hints manually, in semi-automatic mode hints are provided to the player with an option for the player to reject or accept the hint whereas in the automatic hint mode hints are displayed to the user without an option to reject. Hints of two different types – solution hints and navigation hints called *shints* and *nhints*, respectively.

A. A Navigation Example with Dependencies

Considering the lesson 1 from the game, the navigation algorithm is executed using the concept dependency graph shown in Figure 2. The concepts from lesson 1 being same angle qubit polarization (S), orthogonal qubit polarization (O), opposite quadrant qubit polarization (Q), vector notation (V), linear combination (LC), Ket notation (K), and the notion of a basis (B). The distribution of these concepts among 12 exercises of lesson 1 are denoted as P_1 – {S,V} ,P_2 – {O,V}, P_3 – {Q,V}, P_4 – {S,K}, P_5 – {O,K}, P_6 – {Q,K}, P_7 – {S,LC}, P_8 – {O,LC}, P_9 – {Q,LC}, P_{10} – {S,B}, P_{11} – {O,B}, P_{12} – {Q,B}. The dependency map for the exercises in the game is shown in Figure 3.

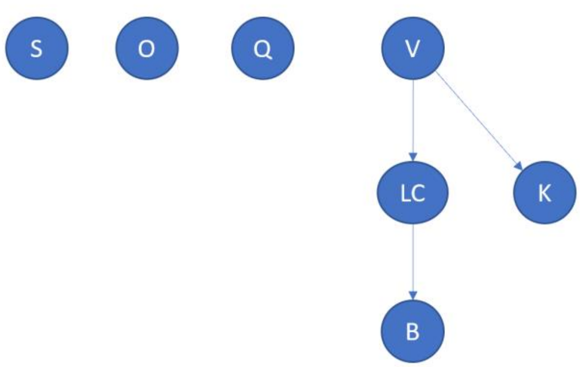


Fig. 2. Concept dependency graph of lesson 1.

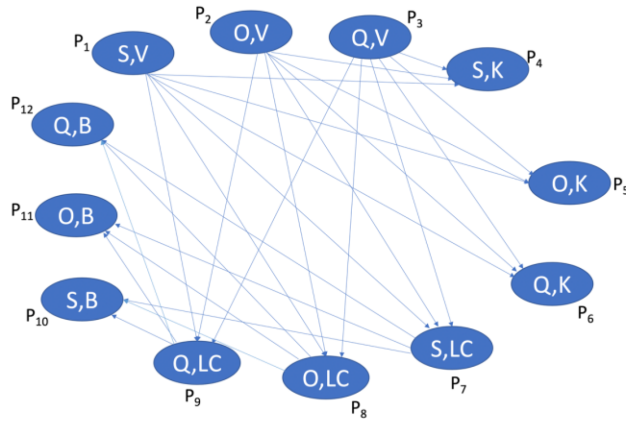
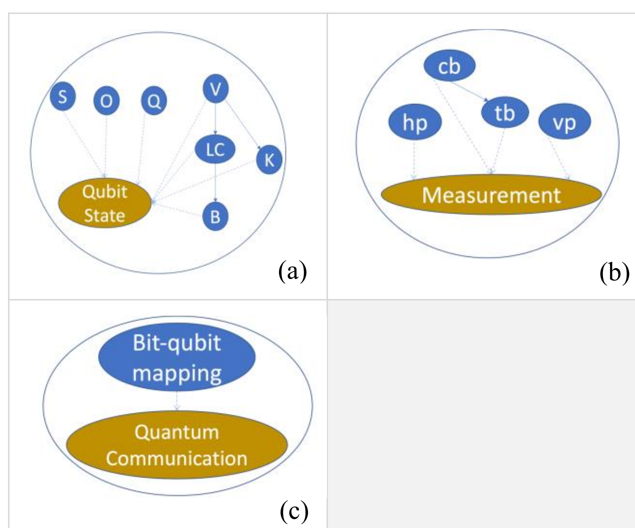


Fig. 3. Exercise dependency graph with 12 exercises of lesson 1.

Table I in Appendix shows a success scenario where the player successively solves the presented exercises in lesson 1. The edge weights are calculated using eq. 4 and next node is chosen using eq. 5. All ties are broken arbitrarily.

V. Super Concepts and Inter-lesson Dependencies

While the concept and exercise graphs provide a fine grained view of a game unit, we can abstract these dependencies to a higher (lesson) level in order to determine the inter-lesson dependencies. This abstraction is represented using the notion of *super-concepts*; each super-concept is an encapsulation of all the concepts and the lesson's end goal. Each lesson in a game unit may be associated with one or more super-concepts and the super-concepts in turn may depend on one or more concepts or other super-concepts. To achieve the lesson's goal, therefore, the player should achieve proficiency in all the associated concepts of that super-concept. The concept dependency graphs along with the corresponding super concepts are shown in Figure 4. The dotted arrows do not represent a physical dependency rather abstract association of every concept with the super-concept. Unlike in Figure 3, the super-concept does not correspond to a physical exercise inside the game in Figure 4.



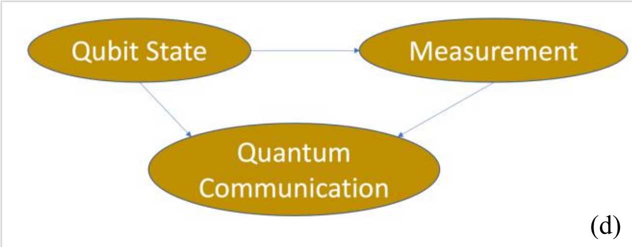


Fig. 4. Concept dependency graphs with *super-concepts* of lesson 1 (fig. a), lesson 2 (fig. b), lesson 3 (fig. c) and the lesson dependency graph (fig. d) of these *super-concepts*.

The concepts in lesson 2 Figure 4 (b) are computational basis (cb), basis transformation (bt), horizontal component (hc), and vertical component (vc) with the super-concept being measurement. The concept in lesson 3 shown in Figure 4 (c) include classical bits to quantum bit mapping with the super-concept of the lesson being quantum communication (qm). Similarly, Figure 4 (a) shows the super-concept for lesson 1 as qubit state (qs). As concepts dependency graphs are used to construct exercise dependency graphs, super concepts are used to construct lesson dependency graph shown in Figure 4 (d).

The lesson dependency graph provides the high-level overview of lesson navigation. In Figure 4, the measurement super-concept needs qubit state as a prerequisite. This dependency does not mean each and every concept of the super-concept are depending on each and every concept of the parent super-concept, rather the knowledge of the qubit state is necessary in-order to understand and solve measurement exercises on a higher level. For a game with multiple lessons, the lesson dependency graph is used to traverse through different levels in the game. Once a lesson node is being visited, the concept dependency graph of that super concept is used to form the exercise dependency graph to further traverse through that level in the game. The concepts in the graph could all be independent in which case navigation hints with independent concepts algorithm is executed instead.

Traversal through the lesson dependency graph can be done in two ways based on preserving engagement or freshness metric - a top-down approach guided by engagement or a bottom-up approach guided by modular concept structure. In top down approach, a schedule can be generated by moving to the next lesson in the graph even if the learning progress vector does not have a value 1 for all the concepts of the super-concept implying learning value of super-concept is not 1. This is to maintain the engagement by providing the player with new concepts for a change, and get back to the failed concepts later. Yet, the value of the current super-concept cannot be set to 1 even if all the concepts in the current concept are marked one, as the dependency is not yet satisfied. This will only be done once the player returns to the previous lesson node and successfully completed the remaining concepts. The second approach is a strict follow of the policy that parent node concept proficiency must be 1 before moving to the child node, else the player has to revert to the learning videos or tutorials to obtain the knowledge of the relevant concepts to continue further. There can also be a threshold proficiency of concepts, when met sets the super-concept proficiency as 1 to allow the player navigate through other lessons using the lesson dependency graph. These questions can be answered empirically by performing experiments, analyzing the engagement and learning potential of the players thereby reflecting on the efficiency of the approaches optimal way can be considered.

VI. Conclusions

The development of an educational gamified paradigm to teach counter-intuitive subjects such as quantum cryptography requires proper identification of concepts and concept dependencies. Equally important, in order to maintain student engagement, is an adaptive navigation mechanism to traverse the game such that the student achieves maximum proficiency in all the subject areas. This paper presented the first of its kind abstract model for designing a game for quantum cryptography. This model was implemented in our game QuaSim in order to dynamically schedule exercises in an adaptive and controlled manner. Future work will involve integrating the game in the classroom environment for easy adoption [20].

Acknowledgements

This project was partly supported by the National Science Foundation award #1623380.

References

[1] Djaouti D., Alvarez J., Jessel JP., Rampnoux O. (2011) Origins of Serious Games. In: Ma M., Oikonomou A., Jain L. (eds) Serious Games and Edutainment Applications. Springer, London.

[2] Susi, T., Johannesson, M., & Backlund, P. (2007). Serious Games : An Overview (IKI Technical Reports). Institutionen för kommunikation och information.

[3] Ritterfeld, U., Cody, M., & Vorderer, P. (Eds.). (2009). Serious games: Mechanisms and effects. Routledge.

[4] Petko, D., Schmid, R., & Cantieni, A. (2020). Pacing in Serious Games: Exploring the Effects of Presentation Speed on Cognitive Load, Engagement and Learning Gains. Simulation & Gaming, 51(2), 258–279.

[5] Hookham, G., & Nesbitt, K. (2019, January). A systematic review of the definition and measurement of engagement in serious games. In Proceedings of the Australasian Computer Science Week Multiconference (pp. 1-10).

[6] Veinott E.S., Whitaker E. (2019) Leaving Hints: Using Player In-Game Hints to Measure and Improve Learning. In: Stephanidis C., Antona M. (eds) HCI International 2019 – Late Breaking Posters. HCII 2019. Communications in Computer and Information Science, vol 1088. Springer, Cham.

[7] Bommanapally, V., Subramaniam, M., Chundi, P., & Parakh, A. (2018). Navigation hints in serious games. *iLRN 2018 Montana*, 115.

[8] Harp, S. F., & Mayer, R. E. (1998). How Seductive Details Do Their Damage: A Theory of Cognitive Interest in Science Learning. Journal of Educational Psychology, 90(3), 414–434.

[9] Mayer, R. E., & Johnson, C. I. (2010). Adding Instructional Features That Promote Learning in a Game-Like Environment. Journal of Educational Computing Research, 42(3), 241–265.

[10] Rowe, J. P., McQuiggan, S. W., Robison, J. L., & Lester, J. C. (2009). Off-task behavior in narrative- centered learning

environments. In Frontiers in Artificial Intelligence and Applications (Vol. 200, pp. 99–106).

[11] Abeyrathna, D., Vadla, S., Bommanapally, V., Subramaniam, M., Chundi, P., & Parakh, A. (2018, December). Analyzing and predicting player performance in a quantum cryptography serious game. In *International Conference on Games and Learning Alliance* (pp. 267-276). Springer, Cham.

[12] Atorf D., Kannegieser E., Roller W. (2019) Study on Enhancing Learnability of a Serious Game by Implementing a Pedagogical Agent. In: Liapis A., Yannakakis G., Gentile M., Ninaus M. (eds) Games.

[13] Din, Z. U., & Gibson, G. E. (2019). Serious games for learning prevention through design concepts: An experimental study. Safety Science, 115, 176–187. doi: 10.1016/j.ssci.2019.02.005

[14] Drey, T., Jansen, P., Fischbach, F., Frommel, J., & Rukzio, E. (2020, April). Towards Progress Assessment for Adaptive Hints in Educational Virtual Reality Games. In Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems Extended Abstracts (pp. 1-9).

[15] Shute V., Ke F., Wang L. (2017) Assessment and Adaptation in Games. In: Wouters P., van Oostendorp H. (eds) Instructional Techniques to Facilitate Learning and Motivation of Serious Games. Advances in Game-Based Learning. Springer, Cham.

[16] Ostler, E., Parakh, A. and Subramaniam, M. (2018). QuaSim: The development of a virtual simulator for teaching topics in quantum cryptography, Society for Information Technology & Teacher Education International Conference, March 2018, Washington, D.C.

[17] A. Parakh, P. Chundi and M. Subramaniam, An Approach Towards Designing Problem Networks in Serious Games, 2019 IEEE Conference on Games (CoG), London, United Kingdom, 2019, pp. 1-8.

[18] Annetta, L. A. (2010). The “I’s” Have It: A Framework for Serious Educational Game Design. Review of General Psychology, 14(2), 105–113. https://doi.org/10.1037/a0018985.

[19] Roungas, B. (2016). A Model-driven Framework for Educational Game Design. International Journal of Serious Games, 3(3). https://doi.org/10.17083/ijsg.v3i3.126.

[20] Bommanapally, A., Subramaniam, M., Parakh, A., Chundi, P. and Puppala, V. M. Learning Objects Based Adaptive Textbooks with Dynamic Traversal for Quantum Cryptography. Second Workshop on Intelligent Textbooks, In conjunction with 21st International Conference on Artificial Intelligence in Education, July 6th, 2020.

[21] Vadla, S., Parakh, A., Chundi, P. and Subramaniam, M. QUASIM: A Multi-dimensional Quantum Cryptography Game for Cyber Security, Journal of The Colloquium for Information Systems Security Education, Volume 6, Spring 2019.

[22] Parakh, A. and Subramaniam, M. Bootstrapped QKD: improving key rate and multi-photon resistance. SPIE Security + Defence, 10-13 September, Berlin, Germany, 2018.

[23] Parakh, A. Using fewer qubits to correct errors in the three-stage QKD protocol. SPIE Security + Defence, 10-13 September, Berlin, Germany, 2018.

[24] Parakh, A. Providing variable levels of security in quantum cryptography, SPIE Optical Engineering + Applications, 19 - 23 August 2018, San Diego, CA USA.

[25] McDermott, S., Vadla, S., Bommanapally, V., Parakh, A., Subramaniam, M. and Ostler, E. Teaching quantum cryptography using a virtual 3D educator: QuaSim, In Proceedings of 2017 National Cyber Summit (NCS'17), Huntsville, AL, June 6-8, 2017.

[26] A. Parakh, M. Subramaniam and E. Ostler, QuaSim: A virtual quantum cryptography educator, 2017 IEEE International Conference on Electro Information Technology (EIT), Lincoln, NE, 2017, pp. 600-605, doi: 10.1109/EIT.2017.8053434.

APPENDIX

TABLE I. EXECUTION OF DEPENDENCY ALGORITHM FOR LESSON 1.

Nodes in V	Nodes in P	Current node solved by the player	Node to explore children (E)	Edge Weights adjacent to node E	Node with Max value	Concepts learnt
P_1, P_2, P_3		P_1				$\langle S, V \rangle$
P_2, P_3	P_1	P_2				$\langle S, V, O \rangle$
P_3	P_1, P_2	P_3				$\langle S, V, O, Q \rangle$
	P_1, P_2, P_3		P_1	$P_1 \rightarrow (P_9, P_8, P_6, P_5) = \frac{1}{6}$, $P_1 \rightarrow (P_7, P_4) = \frac{0.99}{6}$	P_9	
P_9	P_2, P_3	P_9				$\langle S, V, O, Q, LC \rangle$
	P_2, P_3, P_9		P_2	$P_2 \rightarrow (P_4, P_6) = \frac{1}{6}$, $P_2 \rightarrow P_5 = \frac{0.99}{6}$	P_4	
P_4	P_3, P_9	P_4				$\langle S, V, O, Q, LC, K \rangle$
	P_3, P_9, P_4		P_3	$P_3 \rightarrow$ all nodes adjacent to P_3 have values 0		
	P_9, P_4		P_9	$P_9 \rightarrow (P_{10}, P_{12}) = \frac{1}{6}$, $P_9 \rightarrow P_{11} = \frac{0.99}{6}$	P_{10}	
P_{10}	P_4	P_{10}				$\langle S, V, O, Q, LC, K, B \rangle$
	P_4, P_{10}		P_4	<All concepts are learnt, end>		

Total of six exercises are solved in following sequence: P_1, P_2, P_3, P_9, P_4, P_{10}.