QUASIM: A Multi-dimensional Quantum Cryptography Game for Cyber Security

Srikanth Vadla, Abhishek Parakh, Parvathi Chundi,
Mahadevan Surbamaniam

Abstract - Discovering and predicting a gamer's behavior and adapting the game environment to improve the learning is a challenging task in any game-based learning environment. QuaSim is a gamified intelligent tutoring system (ITS) developed to teach quantum cryptography. In QuaSim, students solve problems related to quantum cryptography through different lessons/game plans. In this paper, we provide an overview of QuaSim, and our approach to analyzing students' performance and gameplay behavior based on activity sequence modelling and clustering. We present the results of our analysis and identify different student groups having distinct gaming patterns and problem-solving behaviors. Finally, we discuss the pre- and post-game survey results.

Keywords

Quantum Cryptography, Serious Games, Unsupervised learning, Clustering, Intelligent Tutoring Systems

1 INTRODUCTION

Serious games have been extensively used in academia, government, and industry to provide experiential learning in several aspects of cybersecurity education [16-21]. In this paper, we present an adaptive 3D game, QuaSim, that enables the users to learn quantum cryptography and its applications in designing various cybersecurity protocols. QuaSim is developed in Unreal Engine platform integrated with SQLite database. QuaSim provides a learning and gaming environment with tutorials and quizzes that are followed by the gaming scenarios where students solve different problems. QuaSim monitors the student activities in the game session and records it in the database, which is used to analyze and identify players behavioral patterns and areas of improvement for the intelligent tutoring environment.

Cybersecurity has seen several game based intelligent tutoring systems (ITS) [16-21]. Benjamin et al [20] present a study with multiplayer quantum games where, entanglement shared among multiple players enable different kinds of cooperative behavior. Situ et al [21] present a quantum approach to play asymmetric coordination games and show that quantum entanglement can help players to coordinate their strategies. Cone et al [17,18] present a highly interactive video game, CyberCIEGE, which is a security awareness tool and enables the users of an organization to achieve security training objectives. Boopathi et al [19] introduce a gaming approach to test students' knowledge in various cybersecurity concepts with an overall goal of providing computer security training. Labuschagne et al [16] show an effective interactive web-based game that informs and tests users about security threats and vulnerabilities, with the goal of creating cyber security awareness.

An extensive review of video games, video game studies, important developments, influential perspectives and the relations between them can be found in [1, 11]. Arnab et al [2] propose a model that supports analysis and design of the serious games and enables the replication of various educational and game elements in a serious game. Slimani et al [15] compared various game design methodologies based on different classifications of game elements and discussed the differences in the use of serious game design methodologies following the comparative study.

Gaydos et al [10] conclude that most of the ITS studies focus on the game developments aimed at learning only the subject rather than how such games work or how they can be consistently developed. They discuss the importance of design and serious considerations in defining and sharing a design and its implications. Dicheva et al [3] discuss gamification design principles, game mechanics, context of applying gamification based on educational level, academic subject and type of application, implementation and finally the evaluation of the gamified application. Aslan et al [5] discuss the digital educational game life-cycle and propose a methodology called GAMED which provides a modular approach for overcoming the complexity in the development and guides the developers throughout the lifecycle.

Hamari et al [4] define flow, engagement and immersion in game-based learning and investigate their impact on learning in the game-based learning environments. Gauthier et al [14] compare between voluntary use of a game and a non-game study aid by medical students and found that studying with the game led to better predictability of learning results. Smith et al [13] discuss the challenges in designing and developing STEM games for higher education and showed how the development of their game for STEM education overcome the challenges by utilizing the popular game mechanics to increase player engagement. Divergently, Long et al [12], based on their analysis with the experiments using commercial equation solving game and a research based ITS, showed that the perceptions about what works educationally can be wrong and they believe that there is no replacement for rigorous experiential evaluation of educational technologies.

Klingler et al [8] propose an evolutionary clustering pipeline which may be applicable to any sequential learning data and targeted to improve the cluster stability by multiple training sessions of the students. Kock et al [6] explain an approach to model learner's problem-solving activity sequences and used those models for the automated clustering and found new information about learners and their behavior. The methodologies presented [6, 8] are applicable for any kind of sequential data analysis.

The rest of this paper is organized as follows. Section 2 describes QuaSim architecture, the codification of quantum computing basics, and the cryptographic protocols as single and multi-player game scenarios. Section 3 describes user studies where users from the government (STRATCOM), undergraduate, and graduate students played QuaSim. This section also describes the methods used to collect and analyze player data in these experiments. The results of data analyses are described in section 4. Finally, section 5 concludes the paper and discusses future work.

2 QUASIM: A QUANTUM CRYPTOGRAPHY VIRTUAL EDUCATOR

QuaSim is a multi-dimensional interactive game that is built using the Unreal Engine 3D platform that has been extended to embed predefined instructional components including videos, audio dialogues, auto-graded quizzes, and tests. Additionally, QuaSim supports dynamic import of media-rich instructional components from certified open source and social platforms to further peer-peer learning. QuaSim game scenarios are played in the game at different locales over a city and the locales include different levels of high-rise building as well as open spaces in the city.

QuaSim game is divided into two single player lessons targeting quantum cryptography basics, one single player lessons targeting quantum communication and two multi-player lessons involving the quantum key exchange protocols. Each lesson consists of several exercises that introduce new concepts and mathematical notations. In turn, each exercise consists of multiple problems that are variations on the exercise. The first of these lessons introduces the concept of mapping qubits onto polarized photons. In lesson1 (fig. 1(a)), several qubit receptors located at different levels of a building must be activated by programming qubits with proper orientation. The receptors accept qubits programmed at an angle that is either the same, orthogonal to or in the opposite quadrant (but equivalent qubit) as the angle specified at the receptor. These different types of programming of the qubits make up three problem types: Same Angle, Orthogonal and Opposite Quadrant respectively, in different exercises of the lesson. The programming of the qubits must be performed using the notations that are commonly used to program qubits in quantum computing/cryptography. These notations can be either the matrix, ket or linear combination of vector representations of the photons, (see the player workspace on the left in Fig. 1) each type correspond to an exercise. The other single player exercises (Fig. 1(b)) introduce the notions of quantum superposition and the use of various orthonormal bases. Here instead of the target being a qubit detector it is a qubit "decomposer" that decomposes the incoming photons into orthogonal components. These components are then sent to a detector to check their correctness.

Figure 1 (a): Lesson 1 screenshot shows a player firing a photon (qubit) at 255 degrees to activate the detector. (b): Lesson 2 screenshot shows player firing a qubit such that the vertical component is greater than or equal to 0.9.

As can be seen in figure 1, the player workspace also consists of a calculator and a mini-map to the left of the calculator. The mini-map is used to locate the various detectors/targets in different game levels. In addition, on the top right of the screen are health bar and score. Score increments when students get an answer correct and health decreases upon an incorrect answer.

Figure 2(a): Lesson 3-part 1 screenshot shows a door that can be activated by sending a code 0010 encoded using qubits. (b): Lesson 3-part 2 screenshot shows two bases being used for the same purpose.

"Qubits left" denotes the number of qubits (photons) that a student can fire. Initial number is set to 100 and can be changed by an instructor to adjust the difficulty level of the game. Health can be replenished by watching and re-watching tutorial videos and taking embedded quizzes.

The top left of the screen provides various tools including game settings, media player, web browser and a navigator used to jump to different levels within the game.

Once the students understand mapping qubits onto polarized photons with respect to a basis, they are led into a portal where successive doors need to be opened by entering a code into them (fig. 2(a)). To enter the code, students map the code onto a series of qubits (photons) in appropriate basis (information mapped onto quantum objects) and transmit the photon towards the door that is equipped with a photon detector (this simulates quantum communication between a player and a computer). Finally, the scenario gets more complex with the use of two bases (see Fig. 2(b)). Most quantum key distribution protocols make use of at least two bases.

The final two lessons are multi-player scenarios, where players play the roles of Alice and Bob alternately. The only way they can advance through the game is if they can collaboratively solve a given problem. These lessons feature a chat window (fig. 3) that simulates a classical public channel vulnerable to passive eavesdropping that players use to discuss basis information and other protocol parameters. Actual code word is only transmitted using qubits.

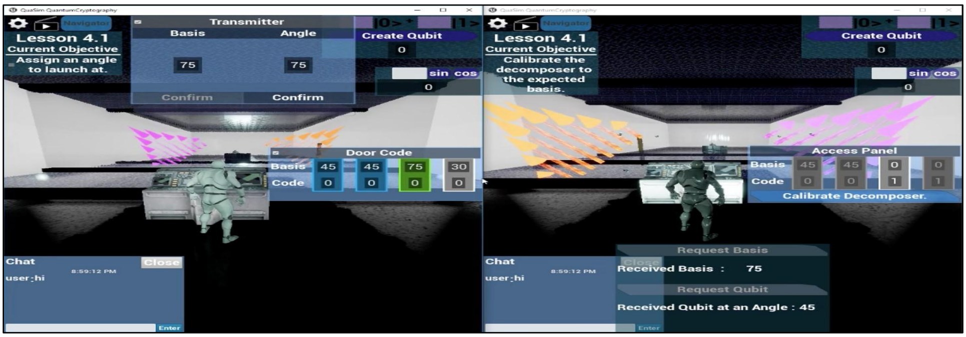


Figure 3: The screenshot shows a multiplayer game scenario used for quantum communication and secure transmission.

While the first of the two multi-player scenarios assume absence of Eve, in the second lesson Eve (a third player) actively eavesdrops on the communication channels. In either case, Alice and Bob do not know about Eve's presence and must take all precautions to minimize leakage of information. Fig. 3 shows a screenshot of the multiplayer scenario of the game. Alice (left) is given a code word and corresponding bases for each bit in the code word. She first sets the basis on the Transmitter (top center of the screen) and maps the bit value onto a photon polarization angle corresponding to that basis. The receiver similarly configures his qubit decomposer (detector) to the same basis and receives the photon, once transmitted. Based on the polarization detected, Bob decodes it into a bit value and enters the value into the access panel. The chat window can be used to reconcile the bases used in transmission and reception and correcting any errors. Eve is not shown in the screenshot but has an interface like Alice and Bob's.

3 EXPERIMENTS AND RESULTS

3.1 Experimental Setup

QuaSim game was played by 60 players (14 females and 46 males, 57 graduate students, 2 seniors and 1 freshman) in a media lab. Each computer had an Intel i7 @ 3.40 GHz processor, 16GB RAM running Windows 10 Enterprise 2016 and NVIDIA GeForce GT 730 graphic card with 2 GB of graphics memory. Each student was given a medium-quality noise cancelling headphones to listen to game-music, lectures and dialogues in the game and to avoid distractions. To play the game, players used an optical mouse and a keyboard. Each player took approx. 50-60 minutes (excluding the introduction and lesson videos) to complete the game. Data was collected from the database connected to the local QuaSim server on each machine.

All the players signed an IRB consent form and were administered a pre-game test testing their knowledge of quantum cryptography. The players registered in QuaSim using a given random anonymous id and email address, completed a background demographics survey, and watched a short mandatory video introducing the game features. The players subsequently logged in and entered the city with the receptors and began playing the game. QuaSim assigns each player a unique player id and logs every action of the players in the QuaSim database (SQLite3). The data has been collected from the database connected to the local QuaSim server on each machine. Students were also given a post-game test along with qualitative surveys.

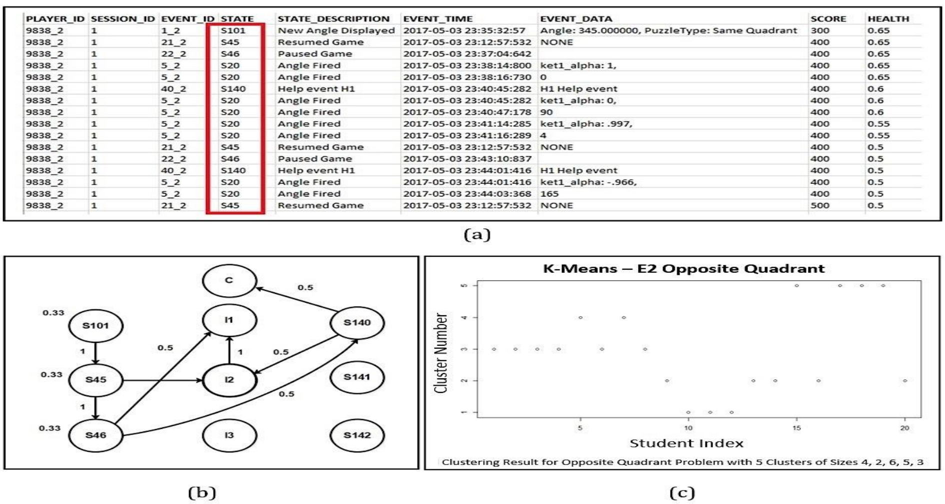
QuaSim logs every interaction of a player during a game session as *timestamped events* and stores them in the database (see Fig. 4(a) for an example of logged sequence of events). In QuaSim, Session id is a unique identifier which identifies entire instance from a login to logout action of a player. Each player can have multiple game sessions where each session has multiple events. An event is triggered whenever an action is performed by the player. Player can attempt a problem multiple times and each attempt could contain multiple event interactions. Event data in each row holds all the information related to a specific event. For example, if Event id is 1 which display the problem and problem type information then the corresponding Event data will be 'Angle:90 Problem Type: Orthogonal'. Event Time has the time at which an event is triggered and it is stored in format yyyy-mm-dd hh:mm:ss:SSS]. QuaSim supports over 100 distinct events and we have observed close to 65 events of the total events in the logs produced in our experiments/gaming sessions. These events can be mainly classified into – actions manipulating gaming elements and controls for navigation, actions pertaining to search for a problem solution attempt, and actions to formulate and submit a problem solution attempt. The logged events for each player in each session are analyzed to classify player game behaviors. The results of such analyses can then be used to identify hints in QuaSim to enhance the player experience and learning.

4 METHODOLOGY: DATA ANALYSIS

Our data analyses methodology produces labeled clusters of student behaviors by analyzing the logged sequential activity data in a push-button manner starting with the logged timestamped events. The main steps are explained in the following subsections.

4.1 Generated Timed Activity Sequence

The events logged in a player game session (raw data, fig. 4(a)) are translated into a timed activity sequences. Each element (row in fig. 4(a)), consists of – player id, session and event ids, game state, event name, time of occurrence, event data, followed by player score, and health at the time of the event. The game states in the timed activity sequence are automatically generated by the translation process by either mapping them directly to the events or computed as outputs of function applied to the events. For example, the activity sequence from Fig. 4(a) depicts two failed attempts followed by a third correct attempt by a player to program a qubit at an angle that is in the opposite quadrant of the receptor angle. The attempts can be succinctly modeled in terms of three game state transition sequences: *S101 -> S45 -> S46 -> I1; S140 -> I2 -> I1; S45 -> S46 -> S140 -> C*, where states *S101*, *S45*, *S46*, and *S140* (highlighted in red color bordered rectangle in Fig. 4(a)) correspond to problem display, resume, pause, and help events. The game states I1, I2, and I3 and C are generated using functions on events. The states I1 (gross error) I2 (medium error), and I3 (almost correct) denote failure levels whereas C denotes a correct solution.



PLAYER_ID	SESSION_ID	EVENT_ID	STATE	STATE_DESCRIPTION	EVENT_TIME	EVENT_DATA	SCORE	HEALTH
9838_2	1	1_2	S101	New Angle Displayed	2017-05-03 23:35:32:57	Angle: 345.000000, PuzzleType: Same Quadrant	300	0.65
9838_2	1	21_2	S45	Resumed Game	2017-05-03 23:12:57:532	NONE	400	0.65
9838_2	1	22_2	S46	Paused Game	2017-05-03 23:37:04:642		400	0.65
9838_2	1	5_2	S20	Angle Fired	2017-05-03 23:38:14:800	ket1_alpha: 1,	400	0.65
9838_2	1	5_2	S20	Angle Fired	2017-05-03 23:38:16:730	0	400	0.65
9838_2	1	40_2	S140	Help event H1	2017-05-03 23:40:45:282	H1 Help event	400	0.6
9838_2	1	5_2	S20	Angle Fired	2017-05-03 23:40:45:282	ket1_alpha: 0,	400	0.6
9838_2	1	5_2	S20	Angle Fired	2017-05-03 23:40:47:178	90	400	0.6
9838_2	1	5_2	S20	Angle Fired	2017-05-03 23:41:14:285	ket1_alpha: .997,	400	0.55
9838_2	1	5_2	S20	Angle Fired	2017-05-03 23:41:16:289	4	400	0.55
9838_2	1	21_2	S45	Resumed Game	2017-05-03 23:12:57:532	NONE	400	0.5
9838_2	1	22_2	S46	Paused Game	2017-05-03 23:43:10:837		400	0.5
9838_2	1	40_2	S140	Help event H1	2017-05-03 23:44:01:416	H1 Help event	400	0.5
9838_2	1	5_2	S20	Angle Fired	2017-05-03 23:44:01:416	ket1_alpha: -.966,	400	0.5
9838_2	1	5_2	S20	Angle Fired	2017-05-03 23:44:03:368	165	400	0.5
9838_2	1	21_2	S45	Resumed Game	2017-05-03 23:12:57:532	NONE	500	0.5

Figure 4: (a) Raw Data from Play Session (b) Discrete Markov Model (DMM), and (c) Clustering

4.2 Map Timed Activity Sequences into DMN

The timed activity sequences are mapped to a Discrete Markov model (DMM, see fig. 4(b)) over the game states [6]. A DMM describes how likely a player is to visit and move from a state to another and allows us to succinctly capture patterns of player behavior while attempting a problem. The prior probability of each state in a DMM is obtained based on the count of first actions of the timed activity sequence of an attempt to solve the problem. Transition probability in a state is calculated by counting the number of transitions from that state to other states. From the diagram (see fig. 4(b)), it can be observed that the prior probabilities of the first action of each attempt (Prior_S101 = 0.33 etc.) are mentioned beside each node (state) and the transition probabilities (Trans_I2_I1 = 1 etc.) are mentioned along the directional edges (transitions) between two nodes [6].

4.3 Generate Labeled Player Clusters

Finally, generate the matrix of prior and transition probabilities for each problem and given it as an input to the K-Means clustering algorithm [7] which groups the students into different clusters. K-Means clustering algorithm [7] is a type of unsupervised learning which takes numeric matrix of data and the number of centers (k clusters) as input and cluster the data into k groups based on the feature similarity. We have given the matrix of probabilities and number of clusters (5) as the input to the K-Means algorithm in R Studio and it generated the clustering results which are plotted as shown in the fig. 4(c), where X-axis represents the students index and Y-axis represents the cluster number. Each cluster consists of students who has similar activity sequences to solve the problem. Finally, we label the clusters based on the activity sequence patterns of the students in a cluster for each problem. Finally, to label the clusters in a clustering, the timed activity sequences of each player in each cluster in a clustering were processed to identify all the common activity subsequences with support threshold (Σ) and length threshold (Λ) were identified and the subsequence with maximum value of the product of Σ and Λ was assigned as the label of that cluster.

4.4 Results

In Lesson1, students goal is to complete four exercises, each having three different problems (Same Angle, Orthogonal and Opposite Quadrant) (see Table 1) with different qubit notations as mentioned in section 2. Analysis of data produced 12 clustering's (for 12 problems) of the 60 players. Each of these clusterings was generated with the optimal number of clusters set to the value 5 for the K-Means algorithm. The number of clusters was determined manually by running the K-Means several times for different values for number of clusters. The least skew in the cluster sizes in the clusterings was observed when the value was set to 5 and hence this was chosen as the number of clusters for all the clusterings. For each cluster in each clustering, the activity sequences of the players belonging to that clusters were analyzed to maximize the value $\sigma \quad \lambda$ described in Section 3 to label the clusters.

The labels assigned to the clusters across the 12 clusterings can be mainly classified as – *Correct with Minimal Effort*, *Repeated Incorrect*, *Helpseeking*, *Hesitant*, and *Unsolved*. Players in the *Correct with Minimal Effort* group obtained solution in the very first attempt or after very few incorrect attempts. Those in *Repeated Incorrect* group had a string of consecutive failed attempts whereas those in the *Helpseeking* group produced failed attempts but after looking for help a significant number of times. Finally, the *Hesitant* group just repeatedly played with game elements involving pause, resume and other non-problem solving actions and the *Unsolved* group simply skipped the problem. The clusterings for the Exercise1-Orthogonal problem and Exercise1-Opposite Quadrant problem depicting these groups are given in fig. 5(a) and fig. 5(b) respectively. In Fig. 5(a), 17 players are in the group *Correct with Minimal Effort* (highlighted in the red color bordered ellipse in Fig. 5(a)) and in Fig. 5(b), 23 players are in group *Helpseeking* (highlighted in the red color bordered ellipse in Fig. 5(b)). The overall distribution of the players into these groups across all the problems in all the exercises is given in Table 1.

As can be seen the two significant groups in the table are *Correct with Minimal Effort* and *HelpSeeking* groups. The performance and the engagement of these players can be improved if the game can provide appropriate aids in a timely fashion. The student's problem-solving patterns and clustering results helped us to understand the need for the different types of hints in the game using an effective hint design approach [9] to improve the player's engagement and continual learning of subject.

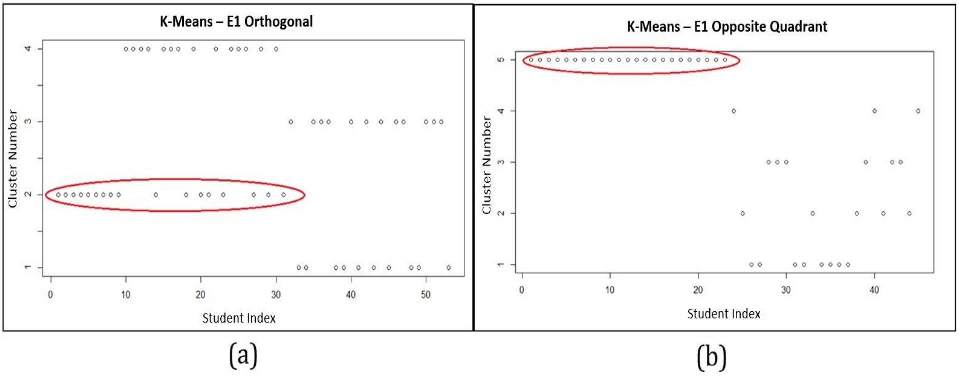


Figure 5: (a) K-Means clustering result for Exercise1 Orthogonal Problem. (b) K-Means clustering result for Exercise1 Opposite Quadrant Problem

Table 1:
Distribution of Students across various clusters

Problems	Correct with Minimal Effort	Repeated InCorrect	HelpSeeking	Hesitant	UnSolved
E1-SameAngle	10	10	19	15	6
E1-Orthogonal	17	9	27	0	7
E1-Opposite Quadrant	16	5	23	7	15
E2-SameAngle	17	2	20	5	16
E2-Orthogonal	25	2	21	0	12

Table 1:
Distribution of Students across various clusters

Problems	Correct with Minimal Effort	Repeated InCorrect	HelpSeeking	Hesitant	UnSolved
E2-Opposite Quadrant	21	7	11	4	17
E3-SameAngle	22	3	5	9	21
E3-Orthogonal	19	3	17	4	17
E3-Opposite Quadrant	21	0	10	3	26
E4-SameAngle	14	0	14	3	29
E4-Orthogonal	22	2	6	1	29
E4-Opposite Quadrant	22	0	9	0	29

4.5 Pre And Post Game Survey Analysis

Table 2:
Pre-Game and Post-Game test results

Scores	Pre-Game (% of students)	Post-Game (% of students)
High (85-100)	49.09	72.73
Medium (70-85)	36.06	18.18
Low (<70)	14.55	9.09

The pre and post-game tests consisted of multiple-choice questions on knowledge of quantum basics and quantum secure communication. Each answer was graded on a decreasing five step-scale where the correct answer received maximum points and other answers received reduced points based on their "distance" from the correct answer.

The distance was an expert input. Table 2 indicates the percentage of students scoring high, medium or low scores in the quizzes. As seen, students showed significant improvement in scores after playing the game.

Qualitative surveys: As a part of game registration process, each student filled out a survey that indicates their self-assessed knowledge levels, on a scale of 0-100 (fig. 6). It is to be noted that, from the figure 6 and table 2, although large number of students reported low knowledge in classical and quantum cryptography they could significantly improve their test scores after playing QuaSim.

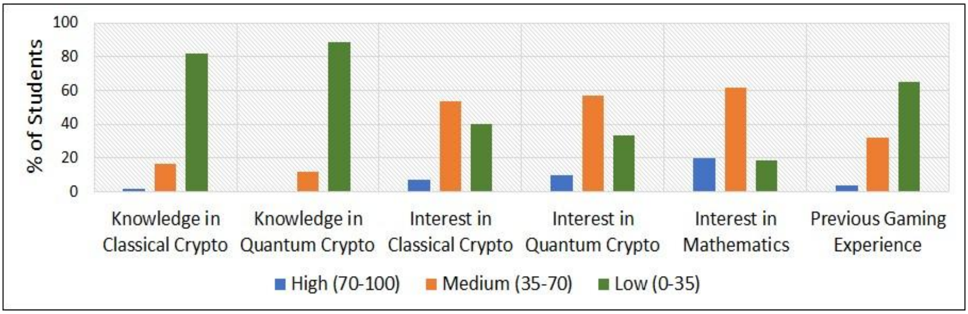


Figure 6: Demographics Information reported by Students

Post-game qualitative survey included 10 questions designed by an education/curriculum design expert on our team. The questions measured (self-reported) player's engagement, frustration and interest levels, usability of the game etc. on a scale of 1-10. These are summarized in fig 7.

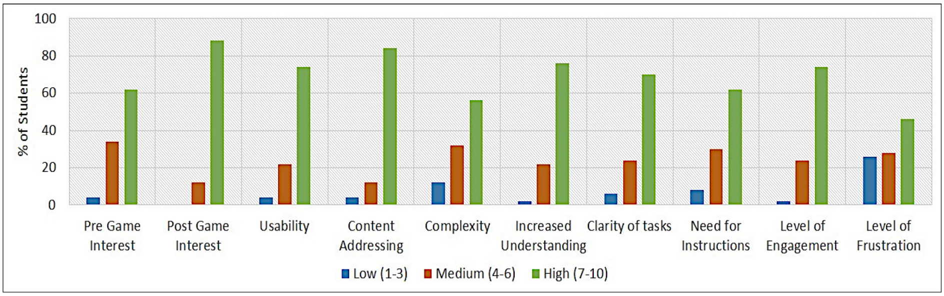


Figure 7: Post-game qualitative feedback from students

We observe that students reported a significant increase in their interests in quantum concepts after the game. Remaining feedback indicates overall positive educational gaming experience for the students.

Figure 8 summarizes the qualitive feedback on gaming elements of QuaSim. As seen in the graph, majority of the students provided high ratings with educational content receiving the highest rating. Four elements were rated somewhat lower –

graphics/animation, directions for the puzzle, gaming directions quality and pace of the game. The graphics had suffered because of the low-end graphics card available in the lab where the game was played. The lower rating for directions for the puzzle is correlated with the somewhat lower rating in "need for instructions" in fig. 7. We plan to address the areas of deficiency in our future versions of the game.

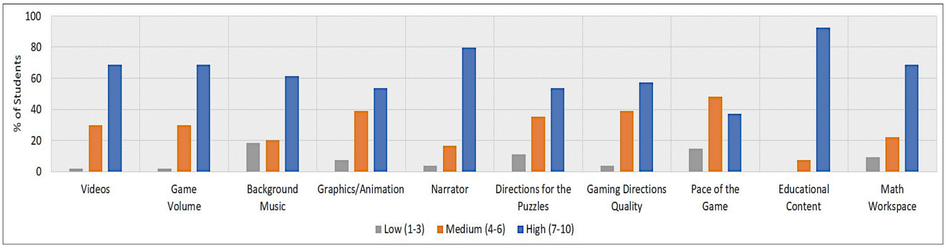


Figure 8: Feedback on Quality of Gaming Elements

5 CONCLUSION AND FUTURE WORK

We have studied the behavior of students and the way they are learning the basic concepts of Quantum Cryptography through the gamified environment QuaSim. We observed various group of students having unique game playing and learning behavior. On the other hand, from the results and feedback, we understood the need for more instructional components in the game to make the QuaSim more effective for the students continual learning.

For this we are in the process of developing three types of hints: manual, semi-automatic and automatic and will be the subject of future research and publications.

REFERENCES

[1] Egenfeldt-Nielsen, Simon, Jonas Heide Smith, and Susana Pajares Tosca. Understanding video games: The essential introduction. Routledge, 2015.

[2] Arnab, Sylvester, et al. "Mapping learning and game mechanics for serious games analysis." British Journal of Educational Technology 46.2 (2015): 391-411.

[3] Dicheva, Darina, et al. "Gamification in education: a systematic mapping study." Journal of Educational Technology & Society 18.3 (2015): 75.

[4] Hamari, Juho, et al. "Challenging games help students learn: An empirical study on engagement, flow and immersion in game-based learning." Computers in Human Behavior 54 (2016): 170-179.

[5] Aslan, Serdar, and Osman Balci. "GAMED: digital educational game development methodology." Simulation 91.4 (2015): 307-319.

[6] Köck, Mirjam, and Alexandros Paramythis. "Activity sequence modelling and dynamic clustering for personalized e-learning." User Modeling and User-Adapted Interaction 21.1-2 (2011): 51-97.

[7] Hartigan, John A., and Manchek A. Wong. "Algorithm AS 136: A k-means clustering algorithm." Journal of the Royal Statistical Society. Series C (Applied Statistics) 28.1 (1979): 100-108.

[8] Klingler, Severin, et al. "Temporally Coherent Clustering of Student Data." EDM. 2016.

[9] Wauck, Helen, and Wai-Tat Fu. "A Data-Driven, Multidimensional Approach to Hint Design in Video Games." (2017).

[10] Gaydos, Matthew. "Seriously considering design in educational games." Educational Researcher 44.9 (2015): 478-483.

[11] Adams, Ernest. Fundamentals of game design. Pearson Education, 2014.

[12] Long, Yanjin, and Vincent Aleven. "Educational Game and Intelligent Tutoring System: A Classroom Study and Comparative Design Analysis." ACM Transactions on Computer-Human Interaction (TOCHI) 24.3 (2017): 20.

[13] Smith, Katherine, et al. "Overcoming challenges in educational stem game design and development." Simulation Conference (WSC), 2017 Winter. IEEE, 2017.

[14] Gauthier, Andrea, Michael Corrin, and Jodie Jenkinson. "Exploring the influence of game design on learning and voluntary use in an online vascular anatomy study aid." Computers & Education 87 (2015): 24-34.

[15] Slimani, Abdelali, et al. "Improving Serious Game Design Through a Descriptive Classification: a Comparation of Methodologies." Journal of Theoretical and Applied Information Technology, 2016, vol. 92, núm. 1, 130-143(2016).

[16] Labuschagne, W. A., et al. "Design of cyber security awareness game utilizing a social media framework." Information Security South Africa (ISSA), 2011. IEEE, 2011.

[17] Cone, Benjamin D., et al. "Cyber security training and awareness through game play." IFIP International Information Security Conference. Springer, Boston, MA, 2006.

[18] Cone, Benjamin D., et al. "A video game for cyber security training and awareness." computers & security 26.1 (2007): 63-72.

[19] Boopathi, K., S. Sreejith, and A. Bithin. "Learning cyber security through gamification." Indian Journal of Science and Technology 8.7 (2015): 642-649.

[20] Benjamin, Simon C., and Patrick M. Hayden. "Multiplayer quantum games." Physical Review A 64.3 (2001): 030301.

[21] Situ, Haozhen. "A quantum approach to play asymmetric coordination games." Quantum information processing 13.3 (2014): 591-599.