GENERIC MODELLING USING UML EXTENSIONS FOR QUEENS CHALLENGE PUZZLE GAME FROM 1 TO 25 LEVELS SYSTEM

Hussain Mohammad Abu-Dalbouh, Ghadeer AlJibreen and Nehal AlDowighri

Qassim University, Computer Science Department, College of Sciences and Arts in Unaizah, Qassim, Kingdom of Saudi Arabia

ABSTRACT

The Unified Modeling Language (UML) is a language for the specification, visualization, and documentation of object-oriented software systems. Existing UML diagrams can be used to conveniently model behavior, but these diagrams can be hardly used to model games. However, UML cannot describe in an explicit manner the games requirements needed for modeling Queens Challenge Game. In this study, the modeling of queens challenge puzzle from 1 to 25 levels is discussed, the proposed extension to UML covering the use case diagram, sequence diagram, activity diagram and class diagram aspects of proposed at the various views and diagrams of the UML. The use of queens challenge game is illustrated using a queens challenge puzzle from 1 to 25 levels example. The purpose of this section is describing the extensions made in each of the UML diagrams, to allow the explicit representation of the proposed system. First, Context Diagram for proposed N-queen game system is introduced. Then, the proposed modifications to the UML are mentioned. Therefore, this study is aimed to showcase the game analysis, design of concept of game, a precise form of game-level operation specification, and an operation schema declaratively describes the effects of a game operation by using case model, actors, use case, relationships between the actors, the use case, interaction between the prototype and its user, sequence diagram, activity diagram and class diagram of queens challenge puzzle from 1 to 25 levels as defined by the Unified Modeling Language (UML).

KEYWORDS

Diagrams, Game, Queen, Solution, Unified modeling language, System, Analysis

1. INTRODUCTION

The UML prescribes a standard set of diagrams and a notation for modeling object-oriented systems and describes the underlying semantics of what these diagrams and symbols mean [1, 2]. Whereas there has been to this point many notations and methods used for object-oriented design, now there is a single notation for modelers to learn. The UML can be used to model different kinds of systems: Software systems, hardware systems, and real-world organizations. The invasion of software-intensive systems into nearly every domain of our life has seen the practice of software development stretched to combat the ever-increasing complexity of such systems, and to meet the increased demand. In such a development environment, the transformation from a concept for running implementation needs to rapidly meet the market demand, but at the same time the software should exhibit the necessary qualities of robustness, maintainability, and meet other requirements, such as usability and performance demands.

The Queens Challenge puzzle belongs to a series of famous puzzles. It is the problem of placing n chess queens on an n×n chessboard, so that no two queens threaten each other. A solution requires that no two queens share the same row, column or diagonal. Every column, row, and diagonal contains exactly one queen. For every level, there are many solution ways. For example, the 8×8 problem has 92 solutions. The game has many levels from 1×1 until 25×25, but every level is found in a separate game. This generates many difficulties for the players, because each level should be downloaded to complete the game. This leads to spend more time and effort by the players, which increases costs. This, in turn, decreases the number of players who want to play this game.

Therefore, having meaningful and standardized behavioral specifications of Queens Challenge Puzzle from 1 to 25 Levels would make it feasible to determine the properties, and enable more thorough and less costly. Unfortunately, such specifications are rarely used. Even less frequently is there a correspondence between a specification and the software itself. Currently, in the industry, much of what would be loosely classified as system specification is performed with use cases. The use cases are an excellent tool for capturing behavioral requirements of software systems. They are informal descriptions, almost always written in natural language, and consequently they lack rigor and a basis to reason about system properties. In this study, the benefits of behavioral specification techniques to queens challenge puzzle from 1 to 25 Levels system are shown by showcase the system analysis and design as defined by the Unified Modeling Language [3]. The diagrams that comprise UML and offer a Use-Case-driven approach on how these diagrams are used to queens challenge puzzle from 1 to 25 Levels system are reviewed. Moreover, the queens challenge puzzle from 1 to 25 Levels is presented. The proposed extension to the UML is covering aspects of the various views and diagrams of UML. The use of queens challenge puzzle from 1 to 25 levels is illustrated using the system.

2. LITERATURE REVIEW

According to [4] discussed that examining mind action amid commitment with various PC diversion sorts to comprehend standards of conduct. They examined the technique, investigation and results got from handling recorded mind movement information from various diverse clients, assembled amid playing time are introduced.

Artificial Intelligence (AI) scientists have for a considerable length of time chipped away at building diversion playing frameworks equipped for coordinating minds with the most grounded people on the planet. The accomplishment of such frameworks has to a great extent been a direct result of enhanced pursuit calculations, and long periods of steady learning building exertion in the interest of the program engineers, physically adding amusement particular information to their projects.The long-term aim of general game playing is taking that approach to the next level with intelligent agents that can automatically learn to play skilfully without such human intervention [5]. The general game playing competition [6] was founded as an initiative to facilitate further research into this area. Game playing agents connect to a game server that conducts the matches. Each match uses two separate time controls: a start-clock and a play-clock. The former is the time the agent gets to analyze the game description until play starts, and the latter is the time the agent has for deliberating over each move decision. The server oversees play, relays moves, keeps time, and scores the game outcome [7]. Game descriptions are specified in a Game Description Language (GDL) [8] a specialization of Knowledge Interchange Format (KIF) [9].A first-order logic based language for describing and communicating knowledge. The expressiveness of the GDL allows a large range of deterministic, perfect information, simultaneous move games to be described, with any number of adversary or cooperating players. Turn-based games are modelled

by having the players who do not have a turn return a no operation move [10]. A GDL game description specifies the initial game state, as well as rules for detecting and scoring terminal states and for generating and playing legal moves. A game state is defined by the set of propositions that are true in that state [7]. There are many computer games types, such as Strategy Games, Real-Time Strategy, Turn-Based Strategy, Helicopter view, Role-Playing Games, Single-Player, Multi-Player, Action Games, First-Person Shooters, First-Person Sneakers, Sports Games, Simulations, Adventure Games, and Puzzle Games.

Puzzles and logic games bring lot of smart entertainment which makes them my favourites. Puzzles could be played for hours, without getting bored, also a lot of logic games has random level generator (like Bubble Breakers) or are somehow randomized, so you do not play same levels again and again. They sharpen your mind and train patience, which is useful for everybody. There are some puzzle games such as: Sudoku, Gem Puzzle, Word Search, and Crossword.
The Queens Challenge Puzzle belongs to a series of famous puzzles. It is the problem of placing n chess queens on n×n chessboard, so that no two queens threaten each other. A solution requires that no two queens share the same row, column or diagonal. Every column, row, and diagonal contains exactly one queen. For every level, there are many solution ways. For example, the 8×8 problem has 92 solutions. The game has many levels from 1×1 until 25×25, but every level is found in separate game. This makes many difficulties for the players, and each level should be downloaded to complete the game. This leads to spend more time and effort by the players which increases the cost. This, in turn, decreases the number of players who want to play this game. Therefore, this study is aimed to incorporate several levels of the puzzle game in order to save time, effort and money from downloading each level separately. Then, develop a proposed puzzle game prototype and to display all solutions while playing the puzzle game at any level.

The following subsections presentsome background information on the Unified Modeling Language (UML) standardized by the Object Management Group (OMG) is provided, and Queens Challenge Puzzle game from 1 to 25 levels system as an example is discussed.

2.1. UML in brief

The UML is an accepted programming industry standard demonstrating dialect for imagining, determining, developing and reporting the components of frameworks when all is said in done and delicate product frameworks specifically [2]. The UML has a very much characterized language structure and semantics. It gives a rich arrangement of graphical antiquities to help in the elicitation and best down refinement of protest situated programming frameworks from prerequisites catch to the sending of programming parts [11].

The Unified Modeling Language (UML) has turned into the accepted standard for building Object-Oriented software. The UML expands on the as of now exceedingly effective standard, which has turned into an industry standard for demonstrating, outline and development of delicate product frameworks, and additionally more summed up business and logical procedures. The Object Management Group (OMG) determination expresses "The Unified Modeling Language (UML) is a graphical dialect for imagining, indicating, developing and reporting the relics of a product serious framework [11].

The UML offers a standard way to write a system’s blueprints, including conceptual things, such as business processes and system functions, as well as concrete things, such as programming language statements, database schemas and reusable software components. The UML defines thirteen basic diagram types, divided into two general sets: Structural modeling diagrams and

Behavioural modeling diagrams. This study deals with Use case diagram, Sequence diagram, Activity diagram, and Class diagram for Queens Challenge Game.

2.2. UML diagrams

A diagram contains model elements such as classes, objects, nodes, components and relationships, described by graphical symbols. Moreover, a diagram can be used to describe certain system aspects at different levels of abstraction. For example, a use case diagram in Queens Challenge Game specifies the functionality of the services provided by the game to its players. The player will have an initiative interface for communicating with the product. He will be able to generate puzzles of his choice of level. Additional interface controls will allow the player to see the solved puzzle. The UML diagrams used in this study are briefly described below:

2.2.1. Use case diagram

The use case diagram defines the functionality inside the system and determines the functions of a system and its users. It shows the external users/players and their connection to the functions/use cases that provided by the system. It also, a specific way of using the system by performing some part of the functionality, and to capture behavioral requirements of software systems. This diagram is used to model the static behavioral aspects of the use case view of the system to model.

2.2.2. Sequence diagram

This diagram describes how the blocks communicate, and how each use case is offered by communicating objects. It shows how the participation objects realize the use case through their interaction. The interaction takes place as blocks send stimuli between each other. The main purpose of the use case design is thus defining the protocols of the blocks. It also, describes a scenario involving various interacting objects.

2.2.3. Activity diagram

This diagram shows activities and actions to describe workflows. In the unified modeling language, an activity diagram represents the business and operational step-by-step workflows of components in a system. It also shows the overall flow of control.

2.2.4. Class diagram

This diagram shows the static structure of classes and their possible relationships (i.e., association, inheritance, aggregation and dependency) in a system. The class diagram shows the building blocks of any object-orientated system. It depicts a static view of the model, or a part of the model, describing what attributes and behavior it has rather than detailing the methods for achieving operations. Class diagrams are most useful in illustrating relationships between classes and interfaces.

3. QUEENS CHALLENGE GAME UML BY EXAMPLE

This section describes the extensions made in each of the UML diagrams to allow the explicit representation of the proposed system. First, Context Diagram for proposed N-queen Game system is introduced. Then, modifications to the UML are shown.

The context diagram is an overview of an organizational system that shows the system boundaries, external entities that interact with the system, and major information flows between the entities and the system.

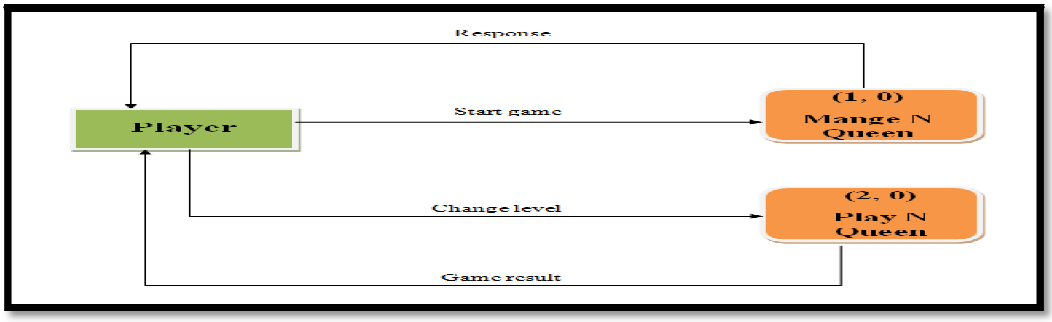

Figure 1.Context Diagram. (Figure 1 shows that when the player starts the N- Queen game, he/she gets the response, after that, he/she changes the game level, and then the game sends the result.)

The level-0 diagram represents a system major processes, data flows, and data stores at a high level of detail.

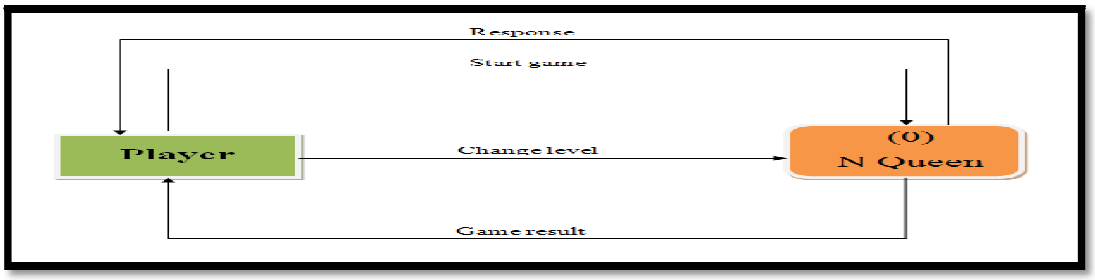

Figure 2. Level-0 Diagram. (Figure 2 shows that when the player starts the game, the game manages N-queen. When the player changes the level, he/she plays N-queen, after that he gets the result.)

The level-1 diagram shows the processes that comprise a single process on the level-0 diagram, and how information moves from and to each of these processes. Level-1 diagrams may not be needed for all level-0 processes[b][12].

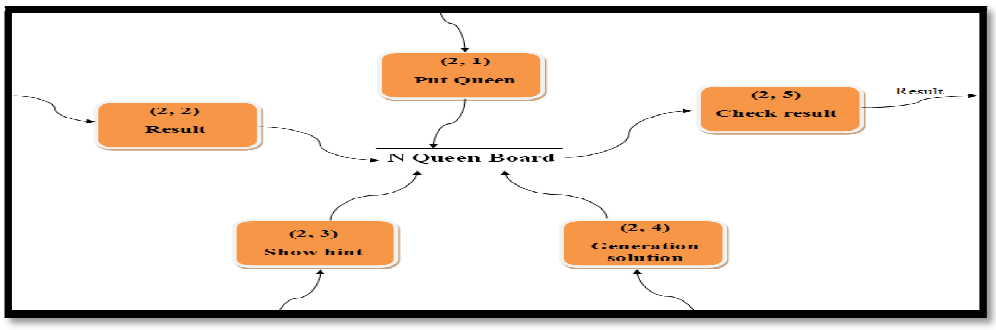

Figure 3.Level-1 Diagram. (Figure 3 shows that Level-1 diagram is more complex than Level-0 diagram.)

Use case Diagram: The primary actor is the general targeted audience. The primary user in the proposed system is the player. The precondition is the user has to start a new game. The main scenario is playing a new game, the game generator is called, and a new game based on the difficulty is generated.

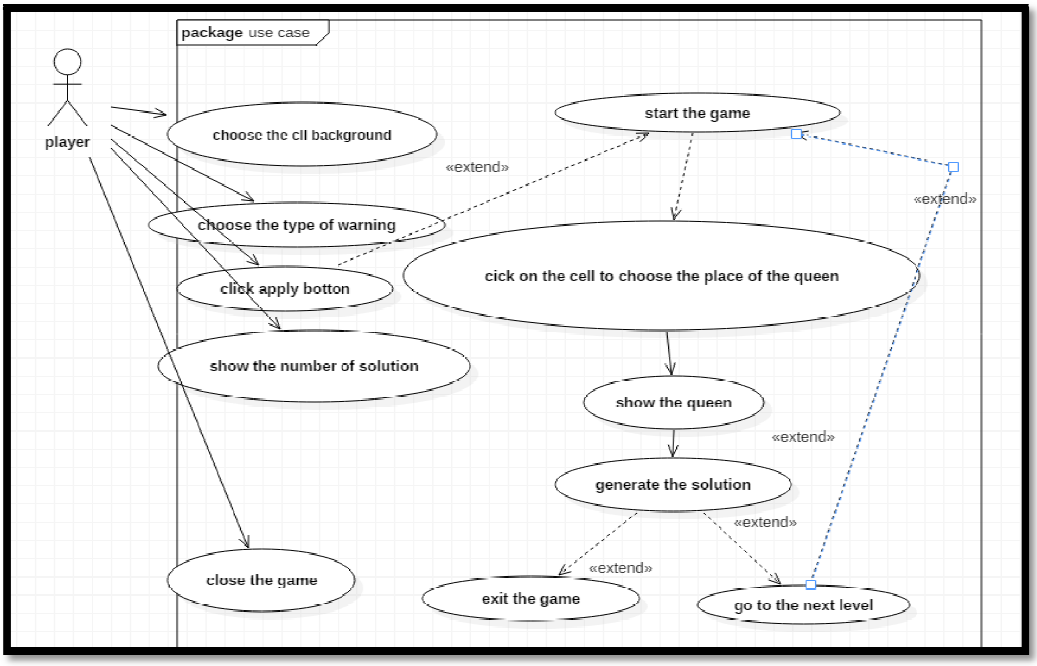

Figure 4.Use Case Diagram. (Figure 4 shows that Queens Challenge Game have many cases and some cases extended to other cases.)

Sequence Diagram: A sequence diagram is one kind of interaction diagrams that shows the interactions among systems while time progresses. Time is represented by a vertical timeline. How the blocks communicate after identifying the system architecture is described. This is done by designing Use case, which was described earlier. They define the external requirement on the blocks and they will be the basis for preparation of the manual. The interaction diagram describes how each use case is offered by communicating objects. The diagram shows how the participation objects realize the use case through their interaction. The interaction takes place as blocks send stimuli between each other. The main purpose of the use case design is thus to define the protocols of the blocks [2, 11-15]. Sequence diagram for Queens Challenge Game shown in Figure 5. The player first chooses the background of the game from the game interface. In addition, he selects type of warning which he wants to apply in his game. Then, clicks the apply button on the game interface. After that, the user can view the level of the game and choose the correct place to put the queen on it. Then, the queen is checked, if there is no two queens intersect with each other vertically, horizontally, or diagonally, that means the queen in the correct place, and the user should find others queen to go to the next level. Otherwise, the system exits the game.

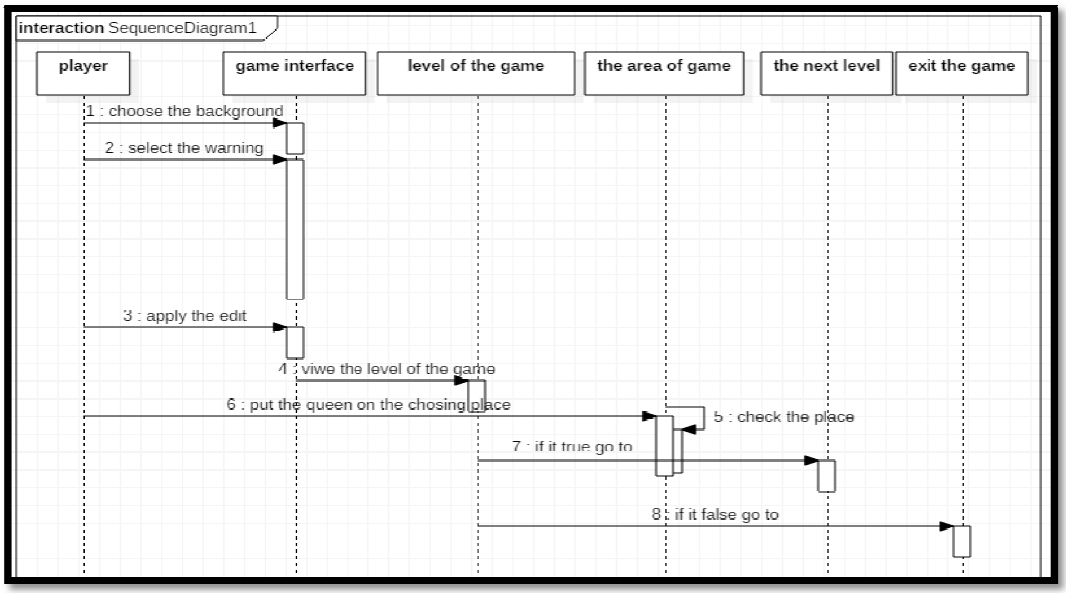

Figure 5. Sequence diagram for Queens Challenge Game. (Figure 5 shows that the sequence diagram for Queens Challenge Game depends on the sequence of many tasks.)

Activity Diagram: It shows the control flow between actions or complex actions called activities. The UML uses Activity Diagrams for describing the steps of processing in case of the use inside the system in a group of activities. It is a way of describing the logic of the process or workflow. It works in the same way the workflow do with the main difference between them, which is the activity diagram allows the operation and the parallel function. They are similar in the use of decision points and branches in describing the steps of the operation. This diagram is used a lot in the operation of analysis to describe the logic of process or the case of the use of the system as shown in Figure 6.

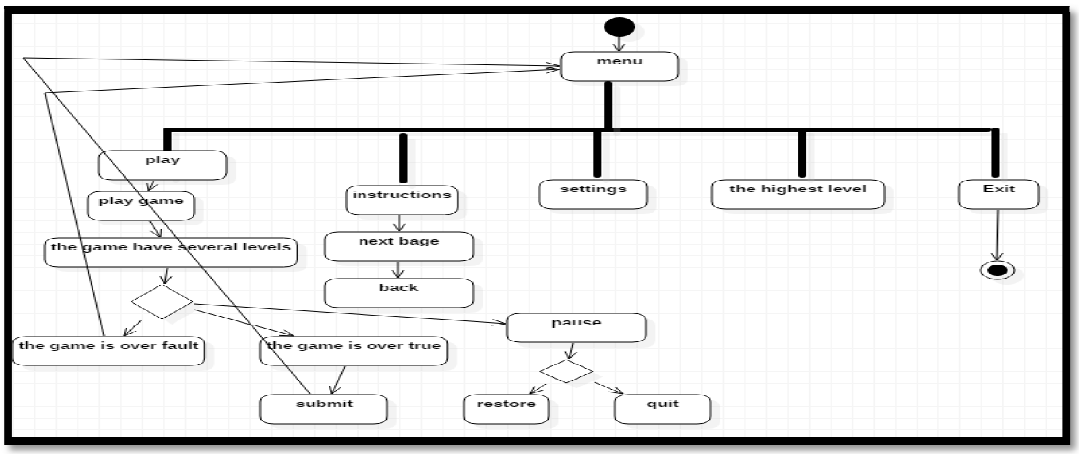

Figure 6.Activity diagram for Queens Challenge Game. (Figure 6(above) shows that Activity diagram for Queens Challenge Game starts with the menu and end with Exit. There are many activities among them.)

Class diagram: The class diagram shows the static structure of the system's software classes, and describes all relationships among those classes, including the association, aggregation and generalization relationships. Below is a high-level class diagram for Queens Challenge Game. This diagram depicts the relationships between different classes within the system, as well as the relationships between the player with the game and controller and proposed system interface. One can see the functionality each class has by looking at the associated functions. The multiplicity is also shown to help understand the system better. It is evident that Queens Challenge Game itself plays a central role in the overall diagram. Most of the functionality passes through it.

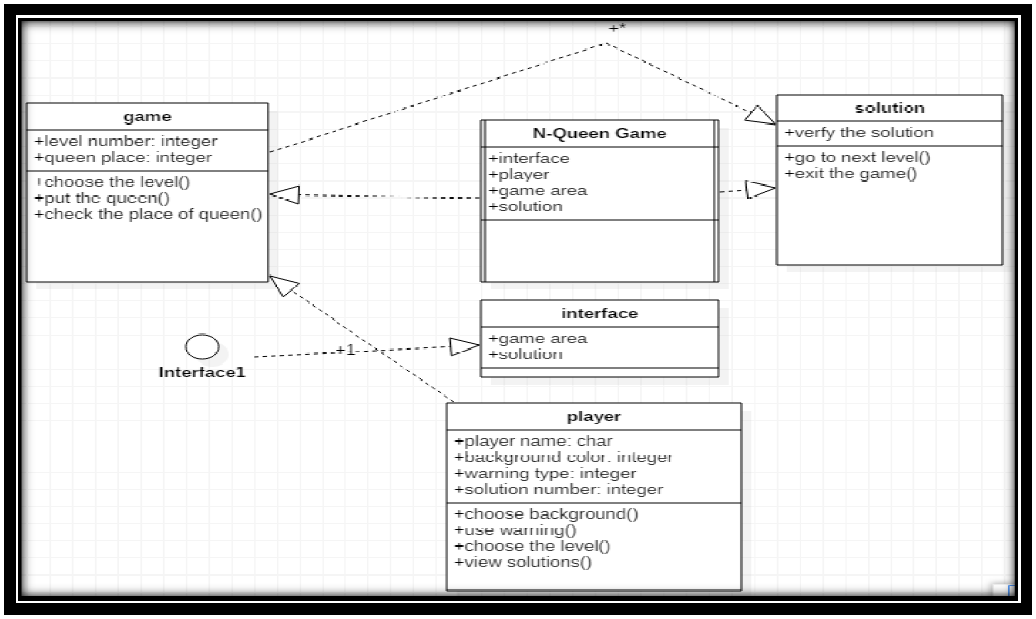

Figure 7: Class diagram for Queens Challenge Game. (Figure 7 (above) shows that Class diagram for Queens Challenge Game has many classes. The important class is interface, game, player, N-Queen game and solution.)

4. CONCLUSIONS

The unified modeling language is a powerful, object-oriented and visualized system analysis and modeling language. It uses a set of sophisticated modeling techniques and is widely applied to various areas. It allows specifying, building and documenting the devices of a system that involves a great quantity of software. It provides a standard form of writing the models of a system, covering so much the conceptual aspects, such as processes of the business and functions of the system, as the concretes, such as the classes written in a specific programming language, schemas of databases, and software components.

Currently, in the industry, much of what would be loosely classified as system specification is performed with the UML diagrams. The Unified Modeling Language (UML) prescribes a standard set of diagrams and a notation for modeling object-oriented systems and describes the underlying semantics of what these diagrams and symbols mean. In this study, Queens Challenge Puzzle from 1 to 25 Levels with the UML, an extension to the UML covering the main aspects of mobility at the various views and diagrams of UML are presented. This study illustrates the

Queens Challenge Game with the UML to have meaningful and standardized behavioral specifications of Queens Challenge Game would make it feasible to determine the properties and enable more thorough and less costly. Queens Challenge Game-UML helps to figure out what really needs for analyzing, designing and modeling mobility, and how to take something that is complicated, even messy and turn it into something precise enough that can do.

REFERENCES

[1] Wienberg, A., F. Matthes,& M. Boger, (1999)"Modeling dynamic software components in UML". Proceedings of the 2nd International Conference on the Unified Modeling Language: Beyond the Standard. Fort Collins, CO, USA, October 28-30.

[2] Booch, G., J. Rumbaugh, & I. Jacobson, (1998)"The Unified Modeling Language User Guide". 2nd Edn., Addison-Wesley, Reading, MA.

[3] OMG, (1999)"OMG Unified Modeling Language Specification", Version 1.3. Published by the OMG Uni-fiedModeling Language Revision Task Force on its WEB site: Retrieved from: http://uml. shl.com/artifacts.htm.

[4] Anastasios G. Bakaoukas, Florin Coada, FotisLiarokapis (2016)"Examining brain activity while playing computer games". J. Multimodal User Interfaces 10(1): 13-29 (2016).

[5] Bikramijt, B., Peter, S. (2007) "General game learning using knowledge transfer". In Proceedings of the International Joint Conference on Arti_cial Intelligence (IJCAI), p 672-677.

[6] Genesereth, M. R., Love, N., & Pell, B. (2005)"General Game Playing Overview of the AAAI competition". AI Magazine, 26(2), 62–72.

[7] Hilmar, F., Yngvi, B. (2010)"Learning simulation control in general game-playing agents". In Proceedings of the AAAI Conference on Arti_cial Intelligence, p 954-959.

[8] Love, N., Hinrichs, T., &Genesereth, M. (2006, April 4) "General Game Playing: Game description language specification".

[9] Genesereth, M. R., Fikes, R. E. (1992)"Knowledge interchange format, version 3.0 reference manual". Stanford: Technical Report Technical Report Logic 92-1.

[10] Stefan ,E., Peter, K. (2007). Symbolic exploration for general game playing in PDDL. In Workshop on Planning and Games at the International Conference on Automated Planning and Scheduling (ICAPS).

[11] Abu-Dalbouh, H. (2014) "m-TOPP-UML: An Extension to UML for the Modeling of Mobile Tracking on Patient Progress System". Research Journal of Applied Sciences, Engineering and Technology, Vol. 7, Issue 7, pp: 1202-1208. February 2014. Available: http://maxwellsci.com/print/rjaset/v7-1388-1394.pdf

[12] Jeffrey A., Hoffer, (2011) "Modern System Analysis and Design", sixth edition, Prentice Hall.

[13] Rumbaugh, J. and I. Jacobson, (1999)"The Unified Modeling Language Reference Manual". Addison.

[14] Oestereich, B., (2001)"Developing Software with UML: Object-oriented Analysis and Design in Practice". Person Education, London.

[15] Wesley, Boston. Scott, K., (2001)"UML Explained". Addison-Wesley, Boston, Massachusetts.