MonstER Park - The Entity-Relationship-Diagram Learning Game

Johannes Schildgen

Technical University of Applied Sciences Regensburg, Germany
johannes.schildgen@oth-regensburg.de
https://www.monst-er.de, https://github.com/jschildgen/monster-park

Abstract. In computer science, game-based learning is an exciting and entertaining way to learn a programming language or coding fundamentals. MonstER Park is a game which applies this concept to entity-relationship models (ERM) an teach it in an easy, fun, and effective way. The plot of the game is about a theme park named MonstER Park that is opening soon, but it's not yet ready. The player of the game has to talk to little monsters and create an ER diagram step-by-step. The player gets instant feedback, and the game continues after correctly solving a task. On completion of a game, the player knows the following fundamentals of ER diagrams and can download a certificate: entity types, (recursive) relationships, (complex, multi-valued) attributes, (compound) primary keys, generalization. The game is free and available at https://www.monst-er.de without any registration.

Keywords: Game-based Learning · ER Diagrams · E-Learning

1 Introduction

Conceptual modeling is a core discipline in computer science. It's taught in database fundamental lectures together with the relational model, transactional concepts, and the query language SQL [2]. While students can try them in an interactive way, conceptual modeling is done either with pen and paper or with Computer-aided Software Engineering (CASE) tools like the SAP Power Designer or ERDplus. These tools are often complicated and have too many features, and they do not provide instant feedback to the student. Furthermore, these tools aren't the best way for learning the fundamentals of ER diagrams. These are good for applying the concepts learned through a lecture, a video, or a book. The motivation of creating the learning game MonstER park was to (1) teach the fundamentals of entity-relationship diagrams, (2) give instant feedback to students on their solutions, and (3) make the learning fun and entertaining through storytelling. The target group of this game is school and university students as well as IT trainees. As no prior knowledge in conceptual modeling or coding is required, anyone can play the game without any preparation.

The structure of this paper is as follows: Chapter 2 describes the game principle of MonstER Park, Chapter 3 focuses on implementation details, Chapter

4 shows the related work. Preliminary results are in Chapter 5, and Chapter 6 concludes the paper.

2 MonstER Park

2.1 Learning Games in Computer Science

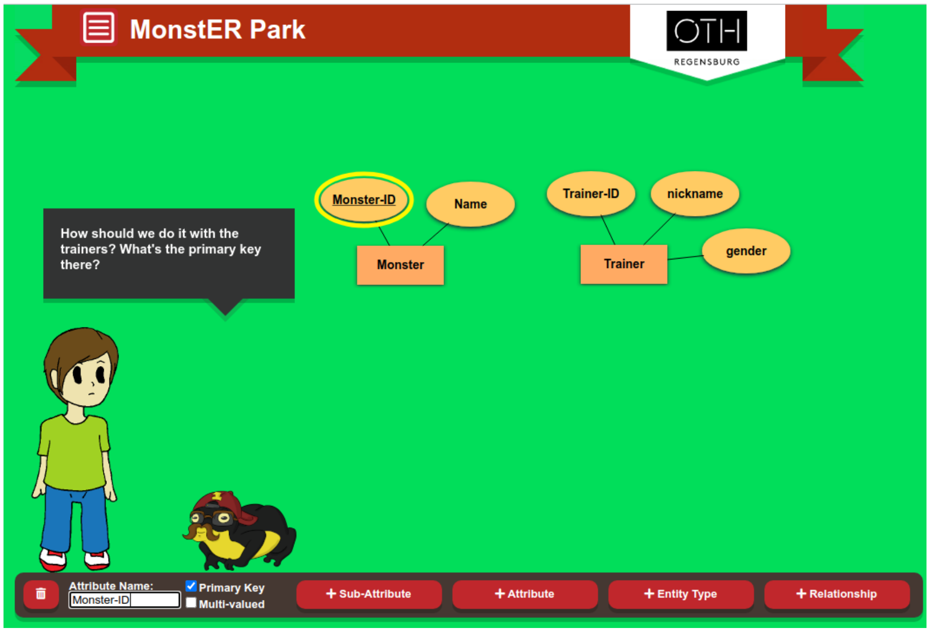
The word "game" in learning games means that the user who plays the game does not only do this to learn but also to have fun during the learning [9]. The key characteristics of games to achieve this are rules, goals, feedback, challenge, interaction, and a representation of a story [11]. The objective of MonstER park is to include all these elements.

Simple forms of learning games are quizzes or puzzles [1]. They are used for learning arbitrary knowledge topics ("what?") but not skills ("how?"). In computer science and other disciplines, practical exercises are very important. This way, the learners have to apply what they have learned on new and unknown problems. Typically, all learners make the same mistakes in the beginning and improve their skills gradually. In disciplines like learning a programming language, the learners get instant feedback when they code. They can immediately see whether their code is correct nor not. If the program shows the expected behavior and does not show an error, the learner was successful. In disciplines like conceptual modeling with ER diagrams, no instant feedback is given. When an ER diagram is drawn on a paper, only a reviewer (teacher, professor, or peer) can assert whether the diagram is syntactically and semantically correct or not. When the learner uses a CASE tool instead, these tools only check for syntactical correctness, but they cannot tell whether the ER diagram models a given scenario rightly. The game MonstER Park bridges this gap and instantly tells the player about the problem. The game continues only when the problem is solved by proper modeling.

2.2 Game Principle

Similar to any CASE tool, the layout of the game MonstER Park consists of a big paper area where the player draws an ER diagram and control elements to create, modify, and delete diagram elements like entities, relationships, or attributes (see Figure 1). Besides these modeling tools, the application shows pictures of game characters together with speech bubbles. One of the characters is always the player's avatar. Depending on the current story situation, this character has a happy, pensive, or surprised face. The second character is a monster or another human that tells the learner what to do. When one task is complete, the story continues and the next task is shown.

The story of the game is about a new theme park, similar to the movie Jurassic Park. But here, there are no dinosaurs, but little monsters. And similar to the story of the game Pokémon, the visitors of the park can also play with the monsters, collect them, train them, and participate in teams and contests.



How should we do it with the trainers? What's the primary key there?

Attribute Name: Monster-ID

☑ Primary Key
□ Multi-valued

Fig. 1. The game MonstER Park

The story begins before the park opens. The player has to model the elements of the theme park by creating an ER diagram step by step. For example, as the first task, the player must create an entity type "Monster", and afterward add the attributes of a monster and so on.

The game covers the following elements of ER diagrams: entity types, relationships, recursive relationships, attributes, relationship attributes, primary keys, compound primary keys, multi-valued attributes, sub-attributes, and sub-entity-types.

No prior knowledge in ER diagrams or databases is required to play the game. When a new concept appears in the game for the first time, it is explained to the learner. This way, MonstER park not only checks the knowledge in ER diagrams but also teaches them from scratch. The full game consists of 33 levels and 19 tasks. Each level of the game presents a story-line. For example, in one level, Fibi, a fire monster appears and mentions that his temperature has changed. The player has to create a new entity type "fire monster" which is a sub-entity-type of "monster" and has an attribute "temperature" to make progress in the game.

At the beginning of the game, the task descriptions are quite detailed, for example "Oh, so a monster belongs to a trainer. I'm going to create a relationship." At a later level, recurring modeling concepts are not explained again so that the player can directly apply the concepts they have learned to a new problem.

The control elements of the game are easy to use, self-explaining, and reduced to a minimum. The game displays buttons, text boxes, and checkboxes only when they make sense for the currently selected element. For example, when the learner clicks on an entity type, two buttons appear, one for deleting the entity type, and one for adding an attribute to it. Furthermore, a text box for the entity-type name and a drop-down list of an optional is-a relationship is shown. Different from other modeling tools, the users of MonstER Park can only create and modify cell elements, not the lines between them. When an attribute is added to an entity type or relationship, or when a relationship connects two entity types, the lines are automatically drawn in the diagram. This avoids incorrect usage of lines, e. g. by directly connecting two entity types without a relationship between them.

After completing the game, the learner can download a certificate. This certificate contains a unique certificate ID, a URL, the name of the learner, and an image of the created ER diagram. Learners can send their certificate ID to their professor or submit the ID in a submission form of an e-learning system so that the professor can check its validity. This way, students can, for example, get bonus points for an exam qualification, when they submit a unique and valid certificate ID after playing the game. An examiner can verify the certificate by either opening the certificate PDF via its URL or by using an API that only responds "valid" in case of a valid ID. The API has the following format: https://www.monst-er.de/cert.php?id=CERTIFICATE_ID&check

Limitations As a consequence of the limited number of control elements that make the editor easy to use, there are some limitations. The most important one is that it is not possible to create higher-degree relationships that connect more than two entity types. Recursive relationships are possible by selecting the same participating entity types. Cardinality selections allow for 1:1, 1:N, N:1, and N:M relationships, but 1:1 relationships are not covered within the game to set a focus on the most popular relationship types (1:N and N:M). Another limitation is the low number of supported properties of entity types and attributes. Weak entity types, extending primary-key attributes, or derived attributes are not supported. It is possible to extend the game by these concepts, but we decided to keep the game as simple as possible and focus on the main elements to avoid confusion. The same holds for special relationships like associations, aggregations, or compositions. These concepts can be explained after playing the game, for example in a lecture, to give an outlook of advanced concepts in ER modeling to the student.

3 Implementation Details

The game MonstER Park is playable in a web browser on any computer or tablet. However, it is not optimized for mobile phones due to their small screen size. The game runs on any operating system and does not require any installation or registration. This allows quick and easy setup in classrooms and at home. The game logic is completely client-based, except for the generation of the certificate. This way, the game can even be installed locally on machines without an internet connection by directly checking out the code from GitHub.

For the development of the game, we have used JavaScript game logic, jQuery and Foundation for the layout and controls, and the JavaScript library JointJS for the ER diagram visual elements.

Each user action, like adding a new entity type or changing its name not only results in performing a corresponding visual change within the editor, but also in maintaining an internal structure which represents the elements of the user's ER diagram. This internal structure is then used to check whether a level is solved correctly or not. It consists of two arrays: one array of entity types and one array of relationships. The objects in these elements store the name, the attributes, and other properties. An example diagram which consists of two attributes looks like this (simplified version):

```
_e = [
  { "name": "Monster",
    "_a": [ {"name": "Monster-ID", "options": ["primary"]},
            {"name": "Name"} ]
  }
]
```

The story of the game is also an array. One element corresponds to one level. A level is defined by a story text, the character images to display, and a solution. The solution looks similar to the array above but here, each element name is not a single string value but an array of all accepted solutions, for example: `["monsterid", "monsterno", "monsternumber"]`. All user input values for enitity-type, relationship, and attribute names are first normalized by removing all non-alphabetical characters and transforming it into lowercase format and afterward compared to the array of valid solutions. If for each element, its solution array contains the normalized user input, the level is marked as solved.

4 Related Work

As an earlier project, we created the learning game SQL Island [12]. Our statistics show that students from various universities and schools across the world play this game more than 6,000 times every month. MonstER Park uses the same game principle and follows similar goals: Both games can be played in a browser without any installation or registration, in both games, no prior knowledge is required, and both use an enjoyable story. Nevertheless, building a learning game for graphical modeling is quite different than for SQL queries. Checking the correctness of an SQL query is quite simple: The symmetric difference between the result sets of the user query and the sample solution must be empty. This is the case only if the two queries produce the same results. However, in ER diagrams, two different diagrams can be correct even if they differ in the naming of entity types, attributes, or relationships. Furthermore, there is no automatic way to check an ER diagram for semantic correctness or equivalence. Jajodia et al. [7] describe different forms of equivalence of two ER diagrams, for example, whether they allow to describe the same data, or whether they express the same constraints within the data. Within these definitions, two elements are treated as equal when their names are synonyms. Furthermore, Jajodia et al. focus on attribute domains. In MonstER Park, domains and data types are not specified, therefore, it is not possible to check for attribute-domain compatibility.

There are multiple interactive learning tools for ER diagrams, however, very few compared to programming languages or query languages like SQL. Furthermore, these tools can help in practicing ER-diagram drawing but not in learning the concepts. LabDER [8] is a virtual learning environment for automatically grading student responses in ERD exercises. This tool can be used by professors to define ER-based questions and automatically grade and provide feedback to student's answers. The feedback and grade are generated by a comparison of the student's ER diagram to a sample solution. Other two learning environments are KERMIT [14] (Knowledge-based Entity Relationship Modelling Intelligent Tutor) and EER-Tutor [4]. These applications present a task description to the user. After the user creates and submits an ER diagram using a built-in editor, the applications then check the solution and give feedback to the user. Thomas et al. [15] follow a similar approach. Their automatic grading tool checks for malformed ER diagrams, missing, and additional elements. In [13], a learner's ER diagram and the solution diagram are translated into an XML representation. The authors compare two approaches for computing the similarity between these XML documents, namely a tree-similarity and a machine-learning algorithm. ERM-VLE [5] uses a text-based interface to let the user add, drop, and modify elements in an ER diagram with input commands. When a command is wrong, an error is shown to the user. DATsys [6] is a tool for creating diagram-based assignment, not only for ER diagrams but also for flow charts, class diagrams, and more. This tool does not automatically check for correctness, nor it gives automatic feedback or grades. Instead, it offers a marking interface for the teacher to manually check the student submissions. Murray and Guimaraes [10] present a web-based tool that teaches the concepts and modeling notations by showing animated diagrams. This tool does not provide an interactive editor. The users can only see finished diagrams or select one of multiple solutions for a given problem in a quiz. A simulation game presented in [3] provides visualizations of different learning topics. At any point in the game, the user can pause the visualization and dive deeper into these topics and get more details. For example, to see the steps on how an ER diagram is created.

5 Preliminary Results

When the first version of MonstER was published, more than 350 students from various universities within the first three weeks. 68 of those students were attending the database-fundamentals lecture at the same time. They provided feedback that was overall positive and had many suggestions for improvement. Most of these suggestions, for example, for new features, better usability, or bug fixes are implemented in the current version of the game.

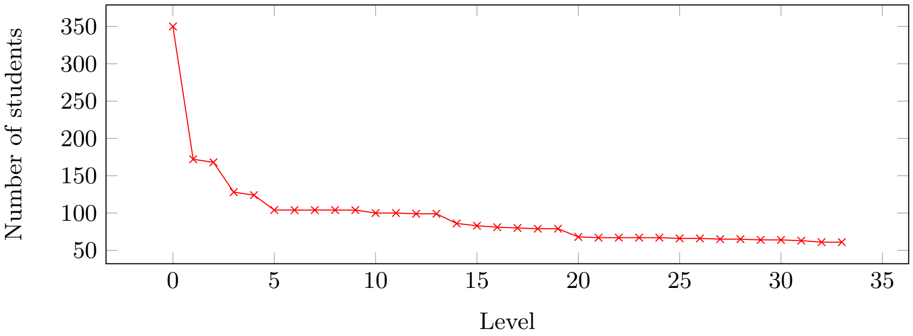


Fig. 2. Game Progress

As shown in Figure 2, out of 350 game sessions, 61 (12%) sessions lasted until level 33 (end of the game). 34 of those users downloaded their certificate. The figure also shows that most of the visitors who started the game just wanted to take a look at it and not to play it. One third of all visitors solved the first task (level 3). Another interesting result is that half of the users who solved the first task finished the game. Especially when level 20 is reached, they wanted to play until the end. This shows a low frustration rate during the gameplay.

Textual user feedback confirms this statement. The users wrote that they enjoyed playing the game, they had a lot of fun, and many of the players wished that the game had more levels. Although the game is too short, students liked the funny story of the game and the combination of learning and playing. They wrote that they feel good after solving level after level. They get a confirmation that they solved a task correctly. One student wrote that she used the game as a meaningful distraction while she was learning for an exam.

6 Conclusion

We presented the learning game MonstER Park that teaches the fundamentals of entity-relationship models in a game-based manner. Each level shows an exercise to the user as a part of bigger story. When the user completes an exercise successfully, the user gets feedback and the next exercise is shown. On completing the game successfully, the user gets a certificate.

Preliminary results show that students like the game and have no problems finishing it. However, these first results are based on students who already know the concepts of ER diagrams from a lecture. In future research, we will examine how well suited the game is for people without prior knowledge of ER diagrams.

MonstER Park uses the Chen notation. However, in many universities, the crow's foot notation is used primarily, which can be implemented in a future version of MonstER Park as an alternative in the diagram editor. Currently, the game is available in English and German. Supporting more languages is planned for future versions. As an extension of the game, a follow-up game is planned which uses the player's ER diagram from MonstER Park to step by step transform it into a relational schema and `CREATE TABLE` commands. And like MonstER Park and SQL Island, this game should also not require any prior knowledge of the relational model or SQL.

References

1. Al-Towirgi, R.S., Daghestani, L.F., Ibrahim, L.F.: Increasing students engagement in data structure course using gamification. International Journal of e-Education, e-Business, e-Management and e-Learning Increasing **8**(4), 193–211 (2018)
2. Joint Task Force on Computing Curricula, A.f.C.M.A., Society, I.C.: Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science. ACM (2013)
3. Connolly, T.M., Stansfield, M., McLellan, E.: Using an online games-based learning approach to teach database design concepts. Electronic Journal of e-learning **4**(1), 103–110 (2006)
4. Elmadani, M., Mitrovic, A., Weerasinghe, A., Neshatian, K.: Investigating student interactions with tutorial dialogues in eer-tutor. Research and Practice in Technology Enhanced Learning **10**(1), 16 (2015)
5. Hall, L., Gordon, A.: A virtual learning environment for entity relationship modelling. In: Proceedings of the twenty-ninth SIGCSE technical symposium on Computer science education. pp. 345–349 (1998)
6. Higgins, C., Symeonidis, P., Tsintsifas, A.: Diagram-based cba using datsys and coursemaster. In: International Conference on Computers in Education, 2002. Proceedings. pp. 167–172. IEEE (2002)
7. Jajodia, S., Ng, P.A., Springsteel, F.N.: The problem of equivalence for entity-relationship diagrams. IEEE Transactions on Software Engineering (1983)
8. Lino, A., Rocha, Á., Macedo, L., Sizo, A.: Labder-relational database virtual learning environment. In: ER Forum/Posters/Demos. pp. 42–55 (2019)
9. Malone, T.W., Lepper, M.: Making learning fun: A taxonomy of intrinsic motivations for learning. aptitude, learning and instruction iii: Cognitive and affective process analysis (1987)
10. Murray, M., Guimaraes, M.: Animated courseware support for teaching database design. Issues in Informing Science and Information Technology **6**, 201–211 (2009)
11. Prensky, M.: Digital game-based learning. Computers in Entertainment (CIE) **1**(1), 21–21 (2003)
12. Schildgen, J.: SQL Island: An adventure game to learn the database language SQL. In: Proceedings of the 8th European Conference on Games Based Learning (ECGBL 2014). pp. 137–138 (2014)
13. Simanjuntak, H.: Proposed framework for automatic grading system of er diagram. In: 2015 7th International Conference on Information Technology and Electrical Engineering (ICITEE). pp. 141–146. IEEE (2015)
14. Suraweera, P., Mitrovic, A.: Designing an intelligent tutoring system for database modelling. MJ Smith, G. Salvendy (eds) pp. 745–749 (2001)
15. Thomas, P., Waugh, K., Smith, N.: Experiments in the automatic marking of er-diagrams. In: Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education. pp. 158–162 (2005)