Ecosystem Simulator

A Learning Game About Genetic Algorithms

Qianqi Huang, Heinrich Söbke(✉), and Tom Lahmer

Bauhaus-Universität Weimar, Marienstraße 15, 99423 Weimar, Germany
{heinrich.soebke,tom.lahmer}@uni-weimar.de

Abstract. Genetic algorithms (GA) are a subclass of machine learning methods that allow an automated determination of optimal points. Learning games seem to be very well suited for teaching GA principles, although learning games are still not state of the art. Accordingly, this article presents the development and evaluation of the learning game *Ecosystem Simulator* for teaching GA principles. The development cycle is described here, which includes the selection of a game theme, the identification of the learning content, the definition of the game mechanics to two subsequent iterations consisting each of development and evaluation. The evaluation of the second development iteration's prototype reveals attaining high scores both for learning motivation and intrinsic motivation along with a significant increase in knowledge. Thus, a learning game has been developed, which, in view of its rather young development timeline, seems to offer an appealing gaming experience combined with decent learning outcomes. All in all–as a motivating learning game–the *Ecosystem Simulator* enriches GA teaching.

Keywords: Serious game · Machine learning · Higher education · Game development · Development cycle

1 Introduction

Genetic Algorithms. Genetic Algorithms (GA)–first introduced by John Holland in the 1970s [1]–describe a search heuristic, which reflects the process of natural selection. GA are stochastic search algorithms, which are often used in machine learning applications for finding optimal solutions and which are widely applied in science, engineering, and business [2]. Katoch et al. (2021) [3] give an overview of GA and present the common types of GA operations *Selection*, *Crossover* and *Mutation* with their advantages and disadvantages. As an example of the diverse application areas of GA, Lee (2018) [4] considers the discipline of operations management and identifies the categories of process and product design, operations planning, and operations improvement. Furthermore, Drachal & Pawłowski (2021) [5] examine GA applications for predicting prices of commodities. Similarly, GA have been applied for a bankruptcy prediction modeling [6]. Moreover, GA have been successfully implemented in the optimization of wireless sensor networks (WSNs) regarding node placement, network coverage, clustering, data aggregation, and routing [7]. Further, GA have been applied in software testing [8, 9] for generating test data covering the most critical paths in the software and thus reducing

H. Söbke et al. (Eds.): JCSG 2022, LNCS 13476, pp. 182–197, 2022.
https://doi.org/10.1007/978-3-031-15325-9_14

the testing effort. Riechmann (2001) [10] investigates the connection of GA to evolutionary game theory by extracting rules using threshold values marking companies as being at risk. In conclusion, GAs have been widely applied in a variety of search and optimization problems due to their broad applicability, ease of use, and global perspectives. These affordances provide significant meaning to include GA in contemporary curricula, especially in engineering-oriented curricula.

Game-Based Learning. As media evolve, so do the requirements of learners for learning processes and the media used in those learning processes. The increasing popularity of digital games is also driving the adoption of digital games in learning processes [11]. When games are leveraged in learning processes, the learning processes are referred to as Game-based Learning (GBL) [12]; games themselves, which are applied for further purposes beyond entertainment, such as learning, are called Serious Games (SGs) [13]. SGs with the further purpose of learning, thus featuring GBL, are also called *learning games*. GBL approaches have been proved to be able to promote a student-centered environment, which provides more interesting, motivating, and engaging learning experiences [12, 14, 15]. Learning targets in GBL scenarios often focus on elementary knowledge, conceptual thinking, and social science [16–20]. Less often, computer science-related learning objectives, such as programming, algorithms and computational thinking skills, are featured in GBL scenarios [21–24]. Learning games with regard to GA are mentioned in literature rather sporadically [25].

Hence, the article describes the development and the evaluation of a learning game conveying knowledge about GAs. The remainder of this article is structured as follows: In the next section, the methodology is outlined. The section thereafter presents the results, which are divided into theme finding, first prototype evaluation and second prototype evaluation. Section 4 discusses the results and possible limitations. The final Sect. 5 contains the conclusions.

2 Methodology

The development of learning games requires the coordination of different disciplines. Harteveld et al. [26] suggest balancing pedagogy, game design, and reality into a functioning whole–a task requiring a structured approach. Accordingly, the work guided by Winn's [27] expanded Design, Play, Experience (DPE) framework, which differentiates in layers and aspects in matrix form. In particular, this article describes three specific phases of learning game development: The first phase is the conceptualization, in which especially theme selection (corresponding to the *Storytelling* layer) and target group selection (corresponding to the *Learning* layer) are described. The second phase is the development, which yields two prototypes and reflects the *Design* aspect. The evaluations of the two prototypes follow at the end of the *Experience* aspect, which were thirdly evaluated in both cases and thus form the basis for further development stages in the *Design* aspect.

2.1 Study Outline

This section outlines the methodologies applied in the three phases.

Phase 1: Theme Selection and Target Group Characterization. Two proposals were developed for the selection of a theme, which were offered for selection via an online questionnaire. A snowball sampling was used to acquire a total of 102 participants via social network profiles of the study lead.

Phase 2: Development of the Game Prototypes. The development of the game prototypes is generally an iterative process. Unity v2019.4.3fl [28], was used as the development environment. Most of the 3D models, particle effects, and sound effects are downloaded from Unity Asset Store [29] and Zapsplat [30]. Some of the artwork has been done from scratch using Adobe Photoshop [31]. Several iterations were performed, with the evaluation results being used in the development of the prototype for the next iteration. Methodologically, walkthroughs were conducted from iteration to iteration with experts in learning games.

Phase 3: Evaluation. Two prototypes were each subjected to a formal evaluation. The evaluation of the first prototype was conducted by a game test with 9 participants recruited from the pool of current and former research assistants of the institution independently of their possible participation in the questionnaire of phase 1. In the second evaluation, after an initial questionnaire including a pre-test, the 20 subjects were invited to play the prototype for a run of about 10 min and then to answer a final questionnaire including a post-test. Again, participants were recruited via social network profiles of the study lead announcing testing a serious game about GA. Also, no participants were specifically targeted or excluded regarding participation in the surveys of earlier phases.

2.2 Learning Outcomes

Essential for learning game development is the determination of intended learning outcomes. In detail, the players are expected to understand and be able to apply the following fundamental GA learning outcomes:

1) Chromosomes represent the **genes** of the individual, usually taking the form of bit strings and each locus in the chromosome has two possible alleles: 0 and 1. If the allele is 1, it dominates the other allele at the same locus and the gene feature presented by this locus is dominant, otherwise it is recessive.
2) The **fitness function** determines how fit (the ability of an individual to compete with other individuals) an individual is. It gives a fitness score to every individual. The probability that an individual will be selected for reproduction is merely based on its fitness score.
3) **Parent selection** ideally is to select the fittest individuals and let them pass their genes to the next generation. Normally, to avoid local maxima, two pairs of individuals (parents) are selected based on their fitness scores. Individuals with high fitness scores have more chances to be selected for reproduction.
4) **Crossover** is the most significant phase in a genetic algorithm. For each pair of parents to be mated, a crossover point is chosen at random from within the genes. Although other crossover strategies exist [3], the only strategy induced is “one-point crossover” at different positions of the gene selectable by the player.

5) **Mutation** is the final step of gene processing. After new offspring are formed, some of their genes may be subjected to a mutation with a low random probability. This implies that some of the bits in the bit string are flipped.

In general, there are four major steps of GA to be mirrored in the game, namely in this order **Parent Selection**, **Crossover**, **Mutation**, and **Breed**.

The second prototype is freely available for play [32]. The results of the three phases are described in the following.

3 Results

3.1 Theme

Two theme proposals were prepared, which were presented in the questionnaire and offered for selection. Both theme proposals were based on the principle of evolutionary development of parameter-controlled behavior of a game character group, which then had to measure its performance against an algorithm-controlled character group. One was a *Battlefield* theme using human-like characters. On the other, an *Ecosystem* theme was proposed using animal-like characters. 55% of the respondents chose the Ecosystem theme. The close decision might demonstrate the inhomogeneity of the target group regarding their gaming preferences.

For better characterization of the target group, the questionnaire included a few more questions. Accordingly, the participants were also asked whether they had already played digital games. 8% denied having played digital games. When asked about daily game consumption, 49% of respondents indicated that they used computer games for less than 1 h per day. 30% play 1 to 3 h per day, 12% up to 5 h. When requested about the game platforms predominantly used, 37% answered mobile games, 30% PC games, and 9% console games. 16% play on all platforms, 8% on none. Asked about attitudes towards whether games can support learning, 15% answered in the negative, 50% saw a requirement for appropriate game design, and 35% saw rather limited support in some games. Overall, the results show that the target group is reachable with digital games, but that the target group still has doubts about potential learning outcomes.

3.2 First Prototype

For the creation of the first prototype several specifications are essential, which are described in the following. In particular, the learning outcomes have to be defined, a semantic class model of the characters contained in the game has to be developed, and the game rules, which drive the game dynamics, have to be defined.

Semantic Class Model. Taking the learning objectives as a starting point, the characters of the game were defined by means of a semantic class model. Central characters for modeling a food chain for simulating an ecosystem are Wolf, Rabbit and Grass. In the following, the class Rabbit is introduced. The class Rabbit consists of four (class) components, each of which is expressed as a class (Fig. 1).

- **RabbitGene** contains gene information of rabbits, and the method to translate genes into specific features of the rabbit's behavior or the rabbit's attributes.
- **RabbitFOV** contains the attributes controlling the field of view of the rabbit, and the methods for searching closest foods, predators, and obstacles.
- **RabbitProperties** contains the attributes of the rabbit's properties controlling the status of rabbit, such as health point, alive or not, hunger point and speed. It also contains methods for updating the status of the rabbit.
- **RabbitMotion** controls the rabbit's behavior based on the attributes from RabbitFOV and the current status provided in the class RabbitProperties.

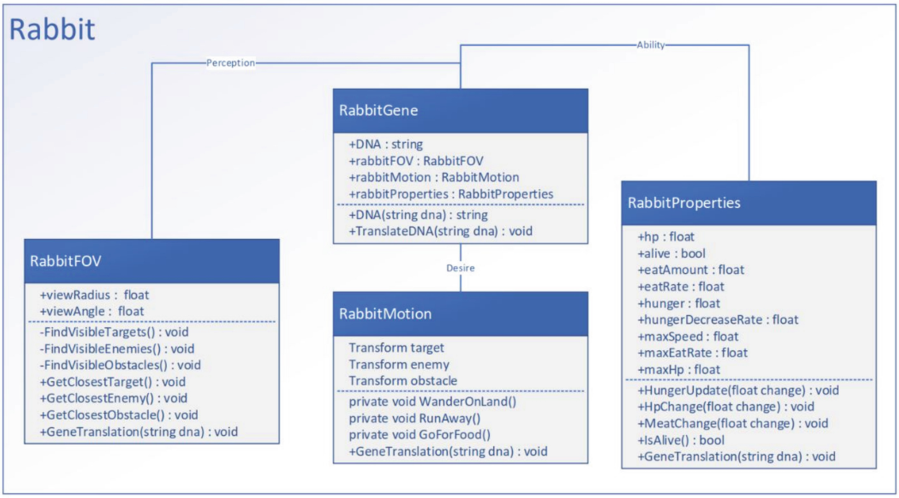

Fig. 1. Class diagram: rabbit

The genes essential for the game are mapped to a chromosome comprised of five genes. The five genes each influence the rabbit's behavior, appearance, and attributes. The value for each gene is either 0 or 1 with the respective meanings of no gene possession (recessive gene) and gene possession (dominant gene).

- **Vision**: this gene provides the ability of doubling the view angle and the view distance of the rabbit. Furthermore, the gene is adverse to the health of the rabbit: the maximum health points of the rabbit will be half of the health points of a normal rabbit.
- **Gut**: this gene prompts the rabbit to be more focused on food acquisition, which means the prior desire of the rabbit is switched from searching for the predator to searching for food.
- **Giant**: the rabbit with this gene grows into a giant individual having doubled in size, maximum hunger, and maximum health compared to the normal rabbit. However, in turn, the speed and eat speed are halved compared to the normal rabbit.
- **Agile**: the rabbit with this gene enters an excited state with four times move speed, eat speed, and hunger decrease compared to the normal rabbit.

- **Cure**: this gene provides self-cure ability by transferring hunger points to health points if the rabbit gets wounded and its current health points are lower than the maximum value.

The attribution of each gene is independent, which means that if a rabbit possesses all the five genes, it gains all the effects from these genes. By means of this principle, overlapping effects could occur, for example if the rabbit possesses the Giant and Agile gene, there will be a compensation for the rabbit's speed due to the Giant gene causing reduction of speed, while the Agile gene causes increase of speed. Thus, this simulation, mirrors natural selection and GAs without considering effects of dependent genes.

The Wolf is modeled as an NPC (Fig. 2), i.e. the player cannot influence controlling parameters, but the parameters are used statically to balance the game play during prototype development.

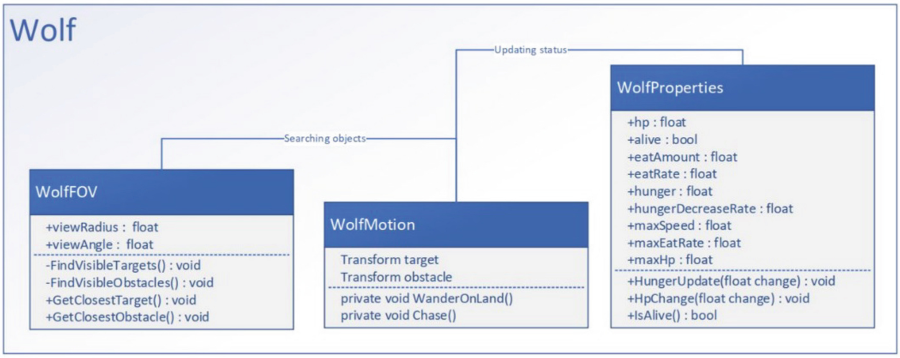

Fig. 2. Class diagram: wolf

Game Play. The game consists of multiple rounds of the interactive Parent Selection (Fig. 3), Crossover (Fig. 4), Mutation (Fig. 5), and Breed (Fig. 6) cycle ("mating season"), between each of which the food chain is simulated based on the actual genes (Fig. 8). In food chain simulation, rabbits and wolves are controlled solely by algorithms based on the actual set of genes (NPCs). The result of a food chain simulation is on the one hand whether the animals have survived, and if so, in which nutritional state they are. At the start of every mating season, there will be a maximum amount of four rabbits alive selected by the system for parent selection. For manipulating the gene combination of offspring, the players are required to select two rabbits among the chosen four as the parents for mating. The criteria for Parent Selection are based on the fitness value of each rabbit. The fitness value is calculated by a fitness function, which has been defined in a heuristic manner. To increase the suspense in the game, the environmental conditions are suddenly changed by randomly occurring scenarios. Besides the normal scenario that starts the game, there are the *Wolf Scenario* (for 40 to 60 s 2 or 3 wolves appear and hunt the rabbits), the *Thunder Scenario* (for 40 to 60 s random lightening are generated that damage all animals in the surrounding area), and the *Snowy Scenario* (besides the visualization of snow, for 40 to 60 s the number of grass tufts is reduced to 5, creating a food shortage for the rabbits). In each scenario, the fitness function is different from the others because a specific gene might deal a positive impact in one scenario but a negative one in another. In this way, the meaning of the fitness function might emerge to the players. Playing the role of prey in the food chain, there are three major behavior patterns designed for the Rabbit, which are implemented using the State Machine technique [33]. The algorithm's pseudocode is as follows:

```
Food found;
Predator found;
States {Wander, Escape, GoForFood}
state = Wander;
Switch (state) {
  case State.Wander:
    if there are no predators in sight do
      searching for food;
      if there are food in sight do
        Food = the closest food found;
        go for food;
      else
        escape from predators;
      break;
  case State.GoForFood:
    if there are no predators in sight do
      if Food still exist do
        go for Food;
        if distance to Food is inside the eat range do
          eat Food;
        else
          searching for food;
      else
        escape from predators;
      break;
  case State.Escape:


    if there are predators in sight do
      Predator = the closest predator found;
      if Predator is still insight do
        escape from Predator towards the opposite
       direction of the direction pointing to Predator;
      else
        searching for food;
      break;
end
```

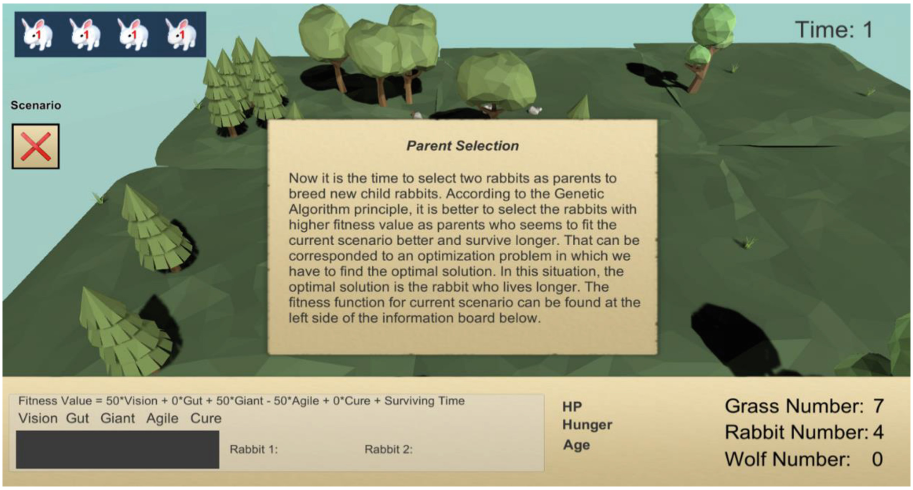

Fig. 3. Ecosystem simulator: explanation parent selection (Screenshot)

Playtest. The playtest for the first prototype involved a class of engineering students. Methodically, the playtest was accompanied in a one-on-one set-ting by the study lead, who also gave a short introduction, according to Fullerton (2008) [34]. Afterwards, participants were asked to play through a play cycle of approximately 10 min while vocalizing their thoughts (think-out-loud). The playtest was concluded with a semi-structured interview on the topics of game play, game mechanics, usability and learning. The observations during the playtest could be summarized as follows:

- The prototype is functional: Players are able to interact with the game unsupported by the study lead.
- The prototype is (partially) complete: Most players understand the winning condition of the game after introduction by the study lead. Only one player has trouble understanding the winning condition. There are no loopholes and dead ends in the game play.
- The prototype is not balanced: Most players are able to understand the characteristic of each gene and dynamically generate strategies to complete the game. However, there is a dominant species, the wolf, breaking the balance.

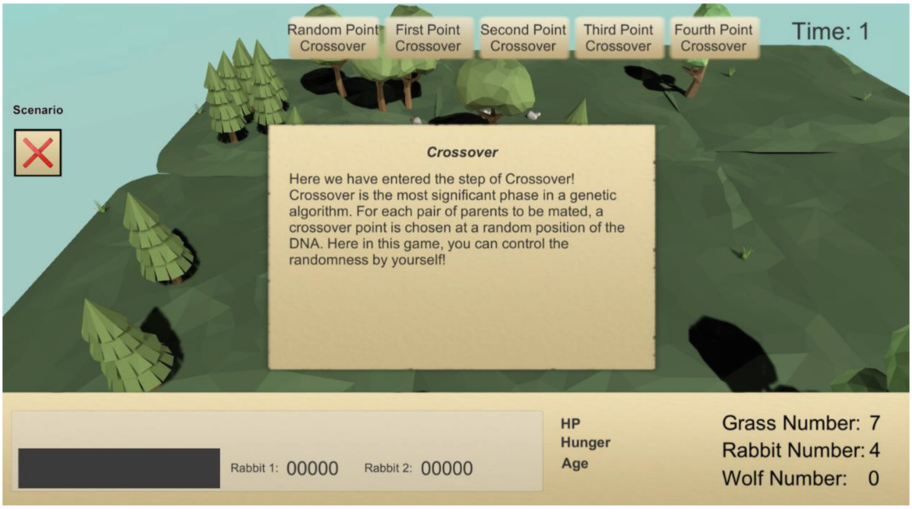

Fig. 4. Ecosystem simulator: explanation crossover (Screenshot)

Furthermore, the results of the semi-structured interviews were categorized (Table 1). Several software errors were indicated by all participants (#2, #3) and unclear visualizations were pointed out (#1, #4). Overall, the interviews confirmed the observations. In particular, a tutorial was missed, and the previous introduction was perceived as insufficient. Accordingly, the game's ability to convey knowledge still needed to be improved.

Table 1. Topics of interviews of first playtest (N = 9, ordered by Category and Frequency)

#	Category	Topic	Freq.
1	Gameplay	Display information of selectable rabbits unclear	9
2		Game end panel display bug	9
3		Mutation visualizer bug	9
4		Visualization of introduction on gene characteristics confusing	9
5		Motivation and interest decrease due to lack of rewards and player stats	4
6		Screen with technical terms unclear	3
7		Dominant object (wolf) causes imbalance	2
8	Learning	GA learning obstacles, lack of tutorial guide	7
9		Memorization of parent rabbits DNA impossible	5
10	Other	Adaption of different screen resolution	3
11		Desire better camera controlling	2
12		Better UI design required	2

3.3 Second Prototype

Based on the evaluation of the first prototype, a second prototype was developed, which included an extended player guidance through a tutorial as well as detailed explanations of the individual game steps. This second prototype was then subjected to further evaluation. Participants were recruited from GA-interested students in online forums for students at the institution. The evaluation of the prototype [32], took place online unattended using a pre-post-test design. In addition to pre- and post-tests, further standardized measurements were included in the two questionnaires. Due to the online unattended implementation of the study without any rewards for the participants, the number of participants slightly dropped from study step to study step. A total of 18 participants answered the pre-game questionnaire and 13 participants answered the post-game questionnaire in a valid manner, after for both questionnaires two answers had been excluded due to recognizable unsuitable answers (incorrect answer for a randomly inserted item "Please select '2' on this scale"). In the following, the results regarding the two standardized questionnaires used as well as the learning outcomes are summarized.

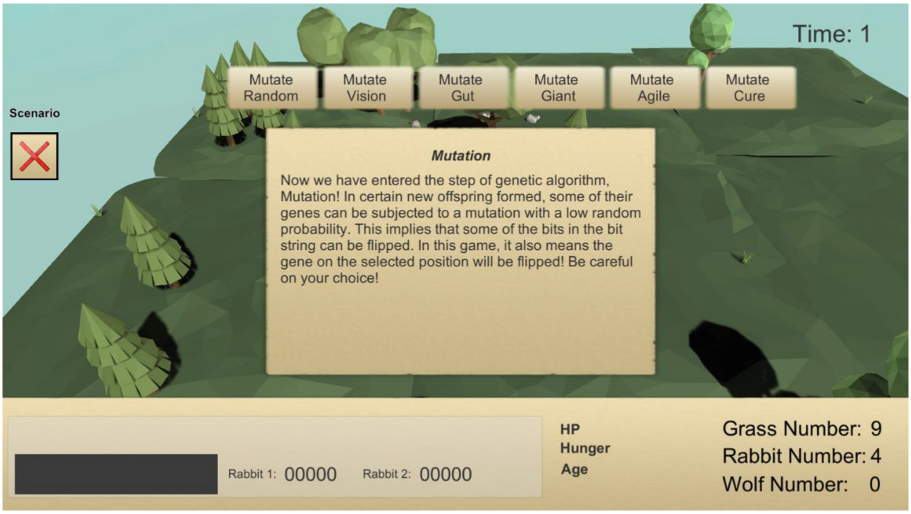

Fig. 5. Ecosystem simulator: explanation mutation (Screenshot)

Learning Motivation. Motivation in learning scenarios influences the success of learning significantly. The *Questionnaire of Current Motivation in Learning Situations (QCM)* [35] in its short form having 12 items [36] in four subscales was used to measure this motivation (Fig. 7). The scores show a good probability of success (4.5), interest in the game (4.4), and challenge (4.7) provided by the game. Anxiety in the lower half of the scale (3.4) is also important since higher values are seen as hampering learning. However, the scores achieved in unsupervised individual play are not at all competitive with those of a learning activity in groups [37] in which SimCity is facilitated–and which is consequently motivationally enhanced [38]. The comparatively strong drop of the subscale interest (4.4 vs. 5.3) is striking and could not have been predicted, since the participants were recruited on the basis of their interest in GA. Overall, however, the values of the QCM - also in view of the prototype stage - are to be considered positive and indicative of the game character.

Fig. 6. Ecosystem simulator: explanation breeding (Screenshot)

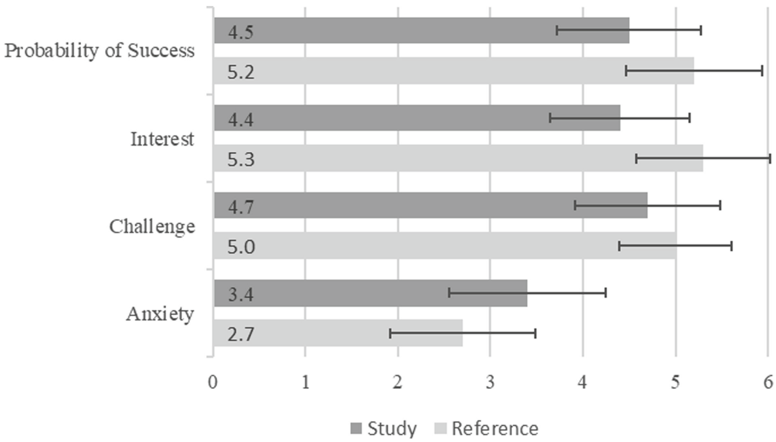

Fig. 7. QCM subscales: values of the study (N = 18, 7-point Likert scale), references adapted from a facilitated SimCity learning activity [37]

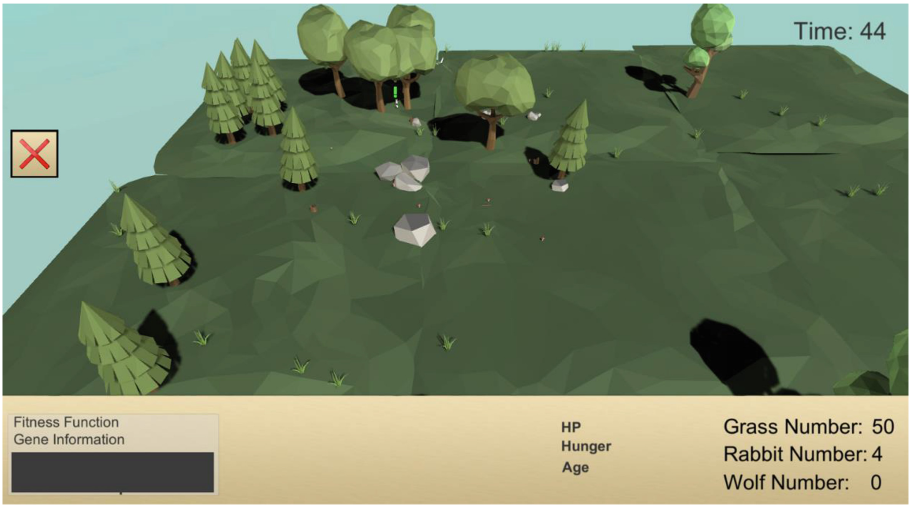

Fig. 8. Food chain simulation GA cycles (Screenshot)

Intrinsic Motivation. A game derives from triggering intrinsic motivation and being played for the sake of playing the game. Intrinsic motivation was captured during the post-game questionnaire using the *Measure of Intrinsic Motivation* (MIM) by Isen & Reeve [39], which consists of 8 items on a 7-point Likert scale. The MIM items were textually adapted to the game, i.e., the pronoun "it" was replaced by "the game." The score was 4.6 (N = 13), which is in the upper half of the scale. In comparison, for a likewise interactive app for rotating objects the scores 4.2 (static) and 4.7 (dynamic) were measured [40]. From this result, the conclusion is to be drawn that the Ecosystem Simulator is capable of arising intrinsic motivation.

Table 2. Pre- and post-test: questions (Correct answers are underlined)

Type	Question
Single-Choice	Genetic Algorithms are generally considered to solve what kind of problem? Evolutionary problem, Systematic problem, Numerical problem, Optimization problem
Multiple choice	Which practical problem can be solved by Genetic Algorithms? (Searching the shortest path between two locations, Ordering a series of numbers, Searching the maximum value of a function, Separating an area into average pieces)
Multiple choice	What are the main steps for Genetic Algorithms? (Parent Selection, Extension, Crossover, Mutation, Recursion, Breeding)
Open	If you know the steps of Genetic Algorithms, please write down the steps in the correct order

Pre- and Post-Test. The pre-test and post-test were conducted using four questions on factual knowledge regarding GA (Table 2). In the pre-test, the correct answers were not revealed, so no learning could occur from the pre-test on its own. Two questions were multiple choice questions, and one question each was a single-choice question and an open-text question. A total of 11 points could be scored, with one point de-ducted for each incorrect answer in the multiple choice questions. On average, partic-ipants improved 46% on the pre-test to 87% on the post-test (Table 3). Thus, a clear learning effect is observed overall.

Table 3. Results of the pre-test and the post-test

	N	Mean	SD	Correct
Pre	18	5.1	3.10	46%
Post	13	9.5	2.22	87%

4 Discussion

The development of the Ecosystem Simulator into a playable, online available learning game, which is able to impart knowledge, may be a solid base for teaching the basics of GAs. Nevertheless, there are limitations to be mentioned and open questions to be pointed out. For example, there may be a bias due to the attrition of participants in the post-game questionnaire, since presumably the most diligent and motivated participants dutifully participated in the second questionnaire.

In the initial questionnaire, there was a balanced choice between two theme candidates. To increase motivational effects of the game, it might be worth evaluating to what extent the both themes might be used at little cost as a kind of theme skin for the algorithms that have now been implemented. In this way, learners could select their preferred theme at the beginning of the learning activity and, on average, increased motivation might be achieved.

The fitness function used to evaluate the genome was first determined in a heuristic approach. At the same time, a natural fitness function results from the gameplay, i.e. depending on the genes, the rabbits have different survival times or different health states at the end of the game. Fitness functions reflecting the actual survival times of the rabbits were determined using a simulation analysis. The results of this simulation analysis still need to be incorporated into the fitness functions that are actually implemented.

5 Conclusions

This article outlines the development of the learning game *Ecosystem Simulator* for teaching the fundamentals of the machine learning sub-discipline Genetic Algorithms (GA). The complete development cycle was carried out, from the selection of a theme, the identification of the learning content, the definition of the game mechanics, to two development iterations with subsequent evaluations. The result is a learning game, which is available online and, which–in view of its comparatively short development history–already offers an encouraging gaming experience with good learning outcomes. Upcoming work includes the implementation of additional game mechanics, such as numerical player stats, to further increase the motivation generated by the game, as well as testing the game in real-world learning settings to approve appropriate didactic scenarios. Overall, the Ecosystem Simulator enriches GA teaching with a motivating learning game.

Acknowledgements. The authors gratefully acknowledge the financial support provided by the German Federal Ministry of Education and Research (BMBF) through grant FKZ 033W011B provided for the "TWIST++" project. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the institution mentioned above.

References

1. Holland, J.H.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. The MIT Press, Cambridge (1992)
2. Mitchell, M.: An Introduction to Genetic Algorithms. The MIT Press, Cambridge (1999)
3. Katoch, S., Chauhan, S.S., Kumar, V.: A review on genetic algorithm: past, present, and future. Multimedia Tools Appl. **80**(5), 8091–8126 (2020). https://doi.org/10.1007/s11042-020-10139-6
4. Lee, C.K.H.: A review of applications of genetic algorithms in operations management. Eng. Appl. Artif. Intell. **76**, 1–12 (2018)
5. Drachal, K., Pawłowski, M.: A review of the applications of genetic algorithms to forecasting prices of commodities. Economies **9**, 6 (2021)
6. Shin, K.S., Lee, Y.J.: A genetic algorithm application in bankruptcy prediction modeling. Expert Syst. Appl. **23**, 321–328 (2002)
7. Norouzi, A., Zaim, A.H.: Genetic algorithm application in optimization of wireless sensor networks. Sci. World J. **2014**, 286575 (2014)
8. Srivastava, P.R., Kim, T.: Application of genetic algorithm in software testing. Int. J. Softw. Eng. Appl. **3**, 87–96 (2009)
9. Xu, B., Xie, X., Shi, L., Nie, C.: Application of genetic algorithms in software testing. In: Advances in Machine Learning Applications in Software Engineering, pp. 287–317. IGI Global (2007)
10. Riechmann, T.: Genetic algorithm learning and evolutionary games. J. Econ. Dyn. Control. **25**, 1019–1037 (2001)
11. Squire, K.R.: Video Games and Learning: Teaching and Participatory Culture in the Digital Age. Teachers College Press, New York (2011)
12. Anastasiadis, T., Lampropoulos, G., Siakas, K.: Digital game-based learning and serious games in education. Int. J. Adv. Sci. Res. Eng. **4**, 139–144 (2018)
13. Michael, D.R., Chen, S.L.: Serious Games: Games That Educate, Train, and Inform. Course Technology, Mason (2005)
14. Yue, W.S., Zin, N.A.M., Jaafar, A.: Digital game-based learning (DGBL) model and development methodology for teaching history. WSEAS Trans. Comput. **8**, 322–333 (2009)

15. Coleman, T.E., Money, A.G.: Student-centred digital game–based learning: a conceptual framework and survey of the state of the art. High. Educ. **79**(3), 415–457 (2019). https://doi.org/10.1007/s10734-019-00417-0
16. Al-Azawi, R., Al-Faliti, F., Al-Blushi, M.: Educational gamification vs. game based learning: comparative study. Int. J. Innov. Manag. Technol. **7**, 131–136 (2016)
17. White, K., McCoy, L.P.: Effects of game-based learning on attitude and achievement in elementary mathematics. Netw. Online J. Teach. Res. **21**, 1–17 (2019)
18. Rose, S.P., Habgood, M.P.J., Jay, T.: Designing a programming game to improve children's procedural abstraction skills in scratch. J. Educ. Comput. Res. **58**, 1372–1411 (2020)
19. Hussein, M.H., Ow, S.H., Cheong, L.S., Thong, M.-K., Ebrahim, N.A.: Effects of digital game-based learning on elementary science learning: a systematic review. IEEE Access **7**, 62465–62478 (2019)
20. Cheng, C.H., Su, C.H.: A Game-based learning system for improving student's learning effectiveness in system analysis course. Procedia Soc. Behav. Sci. **31**, 669–675 (2012)
21. Kazimoglu, C., Kiernan, M., Bacon, L., Mackinnon, L.: A serious game for developing computational thinking and learning introductory computer programming. Procedia Soc. Behav. Sci. **47**, 1991–1999 (2012)
22. Gentile, M., et al.: The role of disposition to critical thinking in digital game-based learning. Int. J. Serious Games. **6**, 51–63 (2019)
23. Frankovic, I., Hoic-Bozic, N., Prskalo, L.N.: Serious games for learning programming concepts. In: Conference Proceedings. The Future of Education, p. 354. libreriauniversitaria. it Edizioni (2018)
24. Ouahbi, I., Kaddari, F., Darhmaoui, H., Elachqar, A., Lahmine, S.: Learning basic programming concepts by creating games with scratch programming environment. Procedia Soc. Behav. Sci. **191**, 1479–1482 (2015)
25. Cirkovic, I.: EvoEnvi: a collaborative serious game played on an interactive table for teaching evolution using genetic algorithms (2018). https://studenttheses.uu.nl/handle/20.500.12932/30527
26. Harteveld, C., Guimarães, R., Mayer, I., Bidarra, R.: Balancing pedagogy, game and reality components within a unique serious game for training levee inspection. In: Hui, K.-C., et al. (eds.) Edutainment 2007. LNCS, vol. 4469, pp. 128–139. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73011-8_15
27. Winn, B.M.: The design, play, and experience framework. In: Handbook of Research on Effective Electronic Gaming in Education, vol. 5497 (2011)
28. Unity Technologies: Unity - Game Engine. https://unity3d.com/
29. Unity Technologies: Unity Asset Store - Discover Top-Rated Game Assets. https://assetstore.unity.com
30. Zapsplat: Free Sound Effects. www.zapsplat.com
31. Adobe Inc.: Adobe Photoshop. https://www.adobe.com/de/photoshop.
32. Huang, Q.: Ecosystem Simulator. https://gamejamq.itch.io/ecosystem
33. Schneider, F.B.: Implementing fault-tolerant services using the state machine approach: a tutorial. ACM Comput. Surv. **22**, 299–319 (1990)
34. Fullerton, T.: Game Design Workshop: A Playcentric Approach to Creating Innovative Games. Morgan Kaufmann, Burlington (2008)
35. Rheinberg, F., Vollmeyer, R., Burns, B.D.: QCM: a questionnaire to assess current motivation in learning situations. Diagnostica **47**, 57–66 (2001)
36. Freund, P.A., Kuhn, J.T., Holling, H.: Measuring current achievement motivation with the QCM: short form development and investigation of measurement invariance. Pers. Individ. Dif. **51**, 629–634 (2011)

37. Söbke, H., Arnold, U., Montag, M.: Intrinsic motivation in serious gaming a case study. In: Marfisi-Schottman, I., Bellotti, F., Hamon, L., Klemke, R. (eds.) GALA 2020. LNCS, vol. 12517, pp. 362–371. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-63464-3_34
38. Baalsrud Hauge, J., et al.: Current competencies of game facilitators and their potential optimization in higher education: multimethod study. JMIR Serious Games **9**, e25481 (2021)
39. Isen, A.M., Reeve, J.: The influence of positive affect on intrinsic and extrinsic motivation: facilitating enjoyment of play, responsible work behavior, and self-control. Motiv. Emot. **29**, 297–325 (2005)
40. Zander, S., Montag, M., Wetzel, S., Bertel, S.: A gender issue? - How touch-based interactions with dynamic spatial objects support performance and motivation of secondary school students. Comput. Educ. **143**, 103677 (2020)