DESIGN OF A COMPETENCIES-BASED SOFTWARE TOOL FOR TRAINING IN THE RESEARCH OF BACTERIAL BEHAVIOR ON ROBOT SWARMS

Eyder A. Rodríguez C., Daniel M. Romero S. and Fredy H. Martínez S.
Facultad Tecnológica, Universidad Distrital Francisco José de Caldas, Bogotá D.C., Colombia
E-Mail: fhmartinezs@udistrital.edu.co

ABSTRACT

Higher education in Colombia since 2000 began to consolidate a training model oriented towards the competitive transformation of the student, in which processes are focused on problem-solving, which implies not only knowing the problem but also its conceptual and real approach. This trend has been structured over 20 years in what is known as competencies-based training and corresponds to the current state of higher education programs in the country. Although this approach has had a high degree of identification with the work environment, the truth is that the skills, knowledge, and abilities generated by the model are also applied to organizations and/or institutions. This article describes the design model of a specialized competencies-based software tool. The tool is part of the specialized training strategy of the research group and corresponds to the first of an integrated system of specialized research training. The design contemplates key elements at the pedagogical, didactic, and software engineering levels. The design architecture is made up of four stages which characterize each one of the tool's creations stages: educational design, computational design, production, and application. In the prototype, the problem of autonomous navigation of robots has been addressed, specifically, the implementation of a decentralized algorithm of the research group inspired by bacterial interaction. The preliminary evaluation of the tool has been advanced with undergraduate students linked to the research seedbed of the research group.

Keywords: bacterial behavior, competency, education, path planning, quorum sensing, research, robotics, skills, software, software engineering.

1. INTRODUCTION

Decrees 1860 of 1994 and 230 of 2002 of the Ministry of National Education (MEN) initiated the implementation of competencies in Colombia as a strategy for the improvement of the quality of education [1, 2, 3]. This training model began in basic and secondary education and was consolidated in vocational training. It sought to structure a strategy aimed at identifying minimum competencies to be promoted in young students, characterized by a direct relationship with work performance. It then led to the design of an integrated training process that involved the academic knowledge required throughout a set of training cycles. This led to the creation of an educational model called training by propaedeutic cycles [4, 5].

In principle, these state regulations are adopted over the years by public and private entities linked to the education sector [6, 7]. These adoptions imply adjustments at the level of curriculum, evaluation (of both students and teachers, academic programs and educational institutions), physical plant (classrooms, laboratories, equipment, etc.), but most importantly, at the level of integration of educational programs in coherence with the needs of the productive sector and the population [8, 9]. As for the programs, the old technical and technological training schemes were no longer static and terminal (only a level of training was achieved without the possibility of formal specialization), but became dynamic (constantly adjusted according to social needs) and continuous in the sense that they were integrated into the formal training model (continuity to professional and postgraduate programs was allowed) [10, 11]. As for the productive sector, in the new model, it is the industry and its needs for professional training that establish the competencies to be trained in young students [12]. And as for the population, the new model of training by propaedeutic cycles establishes a training by stages at the end of which a professional degree is granted, which allows a quick labor bonding as opposed to the traditional scheme of five years (a technician in two years or a technologist in three years can be labor bonded while continuing his professional training, which has a positive economic impact on his family nucleus) [13].

The model of training by cycles and competencies is characterized by a much more flexible structure that allows the student to be within the formal training scheme while beginning his work experience in the same field [14, 15]. But this is an advantage at the level of graduate training; the competency-based training model can also be used in other environments, such as at the level of companies, institutes, or groups, for example, allowing the advantages and flexibility of its structure to be transferred to these environments [16, 17, 18, 19]. In the framework of formal vocational training, the objective of competence-based design is to achieve a curriculum design that reflects a set of behaviors, knowledge, and skills of the productive environment. This curriculum is materialized in a set of training activities that are expected to positively impact both the new professional and the productive sector. This same design scheme can be combined, for example, with traditional software design strategies for the development of software tools used for specific training [20].

Although there are many approaches to address the concept of competency, they all agree that the central element is the student, who acquires specific knowledge through action [21, 22, 23]. In most definitions, this implies the use of specific resources related to the skills to be developed, or what some authors call *'action in context'* [24, 25]. These skills include attitudes, abilities, and knowledge, as well as their interrelationship, applied in specific situations and environments [26, 27]. This means that the definition of competencies for a certain activity must consider aspects such as relevance, level of meaning, the capacity to solve problems, and its orientation towards specific learning objects. In short, competency can be established as the way in which the student develops in specific situations to perform a task adequately [28, 29].

Software tools have become key elements of the educational process. They are tools capable of creating dynamic learning spaces with multiple contributions to both the student and the instructor [30]. Following the policies of training by competencies, it is possible to make an educational design of the tool that tends to a strong emphasis on the specific training of challenges identified within the work environment, or more generally, challenges inherent to the development of a professional profile within an organization or group. Educational software development models tend to be disconnected from the formal educational context, which does not allow them to be related to the competencies defined for a formal training process [31]. In general, the design tends to be more concerned with software engineering, which often limits its effective use [32].

This article presents the design of a software tool oriented to the specific training of undergraduate students within the research group [33, 34]. The design prioritizes the development of skills related to robotic navigation algorithms for a robot swarm through an algorithm proposed by the research group based on bacterial interaction [35]. The development of competencies is formulated together with the structural part of the tool, seeking to systematize the experience of researchers in working with this algorithm [36, 37].

2. MATERIALS AND METHODS

The ARMOS research group involves, semester by semester, undergraduate students interested in developing their final research work with the group. Among the fields of greatest interest is robotics, a research field in which the group has proposed solutions in terms of decentralized navigation topologies, assistance robot prototypes, and autonomous control schemes. Working with new students, the need for specialized training and updating related to the group's developments has been identified. In this sense, three topics have been identified that require special interest: the bacterial interaction algorithm proposed by the group as a decentralized navigation scheme for a robot swarm, the implementation of autonomous navigation schemes on the research group's robotic platform (ARMOS TurtleBot, Figure-1), and the implementation of estimation and control algorithms for Soft Bank Group's Nao robotic platform. We selected the first of these topics for the development of software training tools based on the competencies required to advance the research process.

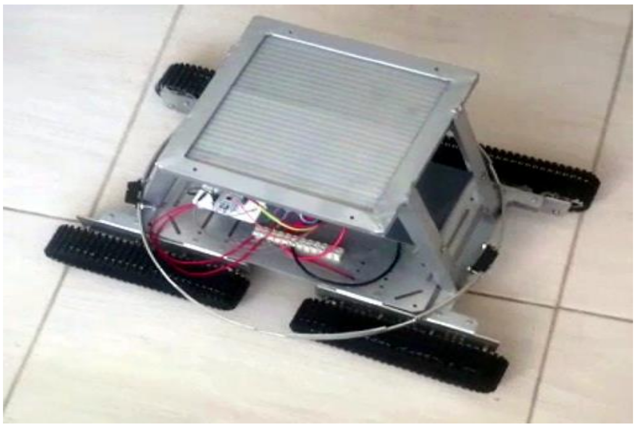

Figure-1. ARMOS TurtleBot mobile platform.

The selected software tool is a simulator that reflects the dynamics of the algorithm based on bacterial interaction, as well as the Quorum Sensing (QS) system proposed by the research group. As an initial condition, the design of the simulator takes into account the competencies required in line with the training needs detected by the research group. These competencies are the core of the design process; in this sense, each of the sections of the tool is structured progressively, in the same way that the competencies are linked to each other. Each section produces information that is used as input for other processes, linking the software architecture to the natural structure of the competencies. The design was structured along four stages defined according to the development stages of the tool: educational design, computational design, production, and application (Figure-2).

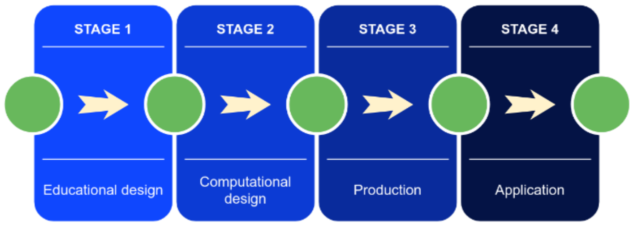

Figure-2. Competency-based model used in the development of the simulator.

Each stage of the design has a specific function. Stage 1 is focused on elements related to pedagogical structure and didactic scheme. This stage takes into consideration the student's educational background; the concepts shaped in his/her curriculum, and the student's expectations to design a family environment that uses previous concepts to introduce new competencies. Aesthetic and functional elements are also considered at this stage. Stage 2 develops the software engineering

VOL. 16, NO. 10, MAY 2021 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

fundamentals related to the development of a robust and user-friendly application. It is established as a base programming language to Python and Pygame modules due to the need for integration with tools of the research group, the ease of graphic implementation, open availability, robustness, and documentation. In phase 3 the different sections of the simulator are developed and integrated, and in phase 4 the tool is applied to the youngest members of the research group to determine its relevance and functionality.

For the educational design stage, we take as initial input the results of the training needs analysis carried out by the research group in the last year. Throughout the analysis, the research group detected specific deficiencies among their students in aspects such as:

- The real functioning of the mechanism of interaction between robots derived from the bacterial biological model, including QS activation.
- Programming of the control unit of the ARMOS TurtleBot robotic platform.
- Functional performance of the different geometric strategies of movement planning for indoor robots.
- Manipulation of schemes for emotion estimation implemented on the Nao robot.
- Manipulation of schemes of interaction man-machine implemented on the robot Nao.
- Manipulation of schemes of estimation of stereoscopic depth implemented on the robot Nao.

This first simulator focused on the QS-based bacterial interaction algorithm formulated by the research group. The implementation and manipulation of the algorithm on different platforms shaped the educational purposes of the tool. Consequently, the following specific competencies were designed:

- Ability to interact with the parameters of the algorithm to modify its performance in autonomous navigation tasks.
- Ability to design customized navigation environments for navigation tasks equivalent to those developed in the laboratory.
- Ability to analyze the performance of different conditions of the algorithm.
- Ability to document experiments related to the evaluation of the QS algorithm.
- Ability to evaluate the ARMOS TurtleBot model in different QS algorithm conditions.

Based on these initial competencies, the pedagogical design and the content design of the simulator were carried out. Among the basic characteristics of the tool was defined:

- The possibility of the software to work with customized environments.
- The tool's capacity to allow the selection of different robotic platforms, including the ARMOS TurtleBot.
- The possibility to configure all the parameters of the QS algorithm for the environment and robot selected.
- The possibility to observe the interaction of the robots in real-time.
- The ability to record the behavior of all parameters of the algorithm.

The consolidation of each one of these competencies is designed in a learning sequence oriented to the development of the young researchers' training process under the conditions defined in the pedagogical model, and in a sequential way. This sequence is verified by the final analysis of the strategy's performance.

In the computational design stage, two activities are carried out, firstly the definition and planning of the sections and modules to be developed, and which will make up the structure and interface of the software, and secondly the software engineering that relates the development of each of these sections and modules. As a first step, it is necessary to plan how the activities will be developed. This planning allows continuous control of the development process, linked to a schedule, and in coherence with the expected product. From this initial planning we proceed to the design of the learning environment, that is to say, the interfaces of the tool. For the software we define the following interfaces:

- Initial information window related to the purpose, objectives, and competencies.
- Window of the configuration of the simulation environment and the robots.
- interactive window of simulation and configuration of parameters of the algorithm.
- Status panel, with real-time information of the simulated process.

None of the interfaces should block the operation of the others. All the software is interactive and allows the continuous adjustment of all the parameters both before starting a simulation and during the simulations. The current status of the software must be evident to the user on the different interfaces. The specific architecture of each interface was defined by the research group through a technical script.

For software engineering, the development tools are considered according to the parameters defined in the educational design, in the planning of sections, and the functional requirements of the research group. For the functional requirements, use case diagrams were designed to specify the behavior of the system under specific operating conditions. A methodological process scheme was made with the functional description of the whole system, considering software tools already existing within the research group. This allowed selecting the most adequate development tools in coherence with the competencies and expectations of the group. The development was carried out in Python 3.9.0 and Pygame 2 with the idea of taking advantage of the language capacity and the graphic advantages of the library.

VOL. 16, NO. 10, MAY 2021 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

All the resources collected in the previous stages are used in the production stage for the development of the different sections of the software. In this stage, the implementation of the sections and their integration is carried out. Also, simultaneously, a user's manual is made that documents the handling of the simulator. In this stage the typical cycles of the development projects in engineering are presented, that is to say, a set of designs and tests that define the redesigns required for the adjustment of the software according to the design profile. The evaluations are carried out by the researchers of the group, who endorse the final performance of the tool. The general methodology used in this stage is shown in Figure-3. This methodology is structured in five phases defined according to the fulfillment of the competencies established in the educational design stage. Throughout these phases, a criterion is established for the evaluation of the competences, another criterion to verify the interrelationship of these competencies, the capacity of the software to stimulate the development of the competences is verified, redesigns to the structure of the software are proposed, and a final evaluation is set according to the resources available in the laboratory (robots and navigation environments).

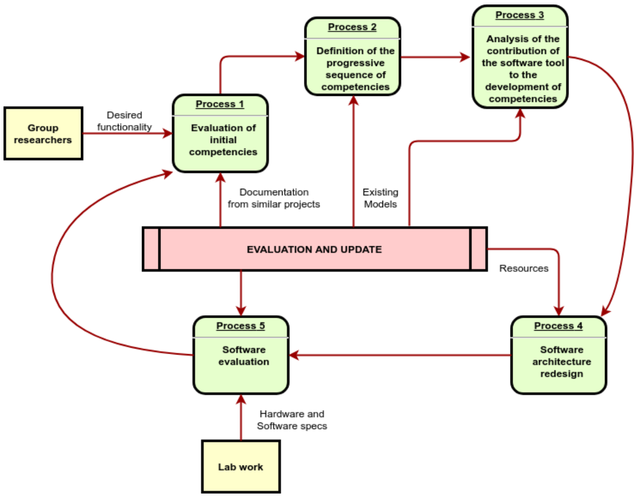

Figure-3. Strategy for evaluating and updating the simulator.

Finally, in the application stage, the simulator is provided to the students of the research group for their use and subsequent analysis of results related to specific training. This last part of the process is currently being evaluated by the research group. For its development, an initial profile of the students has been established, and a learning test to be applied after the use of the tool.

3. RESULTS AND DISCUSSIONS

Our software implements a robot swarm, which performs a wild movement in an environment selected by the user. According to the QS algorithm, the robots will have the quality to avoid collisions with different obstacles that affect their movements, since they are equipped with proximity sensors to fulfill this function. Thus, each time that the sensor informs the robot of the presence of an obstacle, it will randomly change the direction according to the wild movement of the automaton.

The user has the option to load an image with the environment in the desired design and adjust its dimensions. This environment will have a grid, which will have the function to define the regions where the automats will move. These regions will be loaded with weight values which define the place where the robots will converge to reach the quorum, with a threshold value also defined by the user. The quorum will be made by the robots by comparing the weight values with the number of robots in the region.

The simulator interface was designed in Qt Designer (Figure-4). This interface was designed with ease of user interaction in mind.

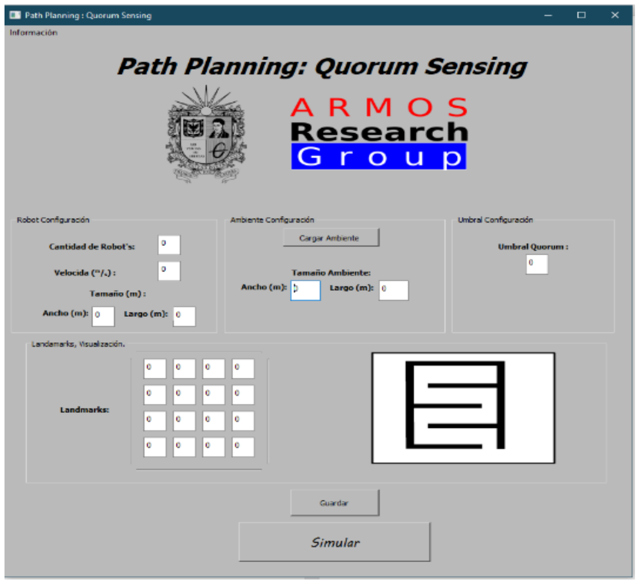

Figure-4. Software interface: Path planning simulator - quorum sensing.

To perform a simulation, the first step is to load the navigation environment in which the robots will interact (Figure-5). In this configuration, the simulator provides the option to load an environment as a graphic image, which will be available from a preloaded list of environments. This window also provides the ability to configure the actual size of the environment, which allows the possibility of using real environments scaled from the laboratory. The software already has some classic environments that can be used by the user (Figure-6). The loaded environments will be located in the *maps* subfolder.

ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

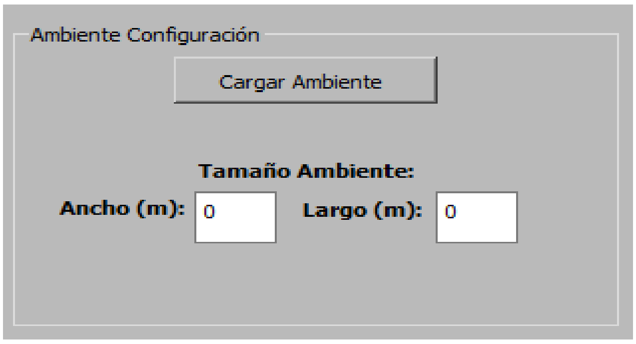

Figure-5. Navigation environment configuration.

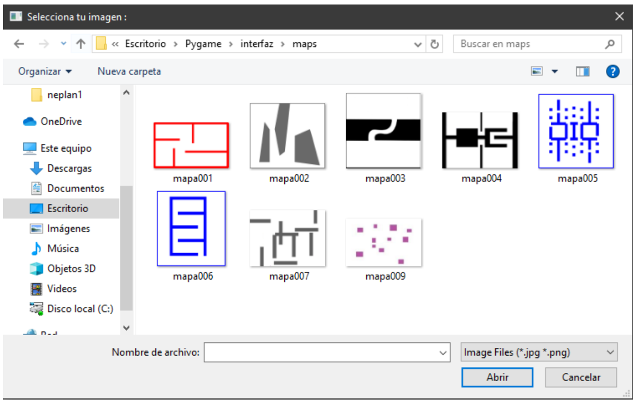

Figure-6. Environment loading, and preloaded environments.

By selecting the navigation environment, the software identifies obstacles and free space for navigation. Image processing is done with OpenCV. In general, dark regions are classified as obstacles and light ones as free space (according to the grayscale value). This creates an explorable environment for the robots, which is labeled as "1". The following is to define the parameters of the robot, this is, the number of robots, its speed, and size (Figure-7).

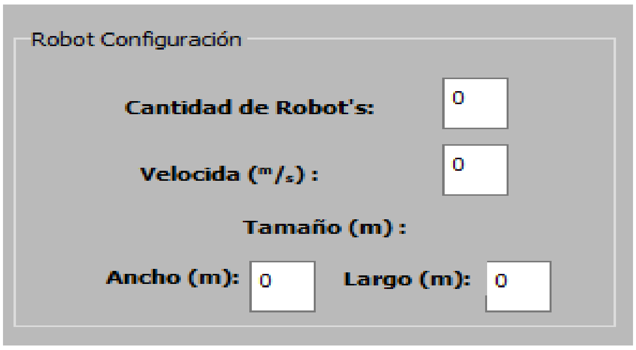

Figure-7. Robot parameter settings.

The user must define the desired QS threshold. However, this value is restricted to a maximum of 90% of the total number of robots, which guarantees the consistency of the algorithm. To define the place where the quorum will be established, we segment the environment into rectangular regions for which a weight value must be assigned, also configurable as landmarks, which will give local information to the robots about the performance of the region. The higher the value, the more attractive the region will be to the robots (Figure-8). Once the basic values for the simulation are configured, the user will click on save, to save these values in the base of the program, and then simulate to run the simulator.

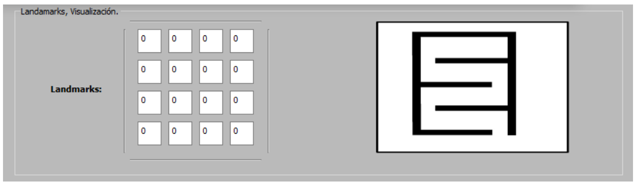

Figure-8. Assignment of landmarks with the preview of the loaded environment.

The example shown in Figura-9 corresponds to a navigation environment of 100 m x 50 m, and the one in Figure-10 to an environment of 75 m x 20 m. As it is observed, the size of the ARMOS TurtleBot robot is adjusted to the dimensions of the environment. The speed of the robot also adjusts to the conditions of the selected environment. In these two cases, we used the same design of obstacles and free space, but a different size was used for the environment. The robots changed when the size of the environment was modified, as having a larger area in the environment the robots look smaller in comparison, this means that the true size of the robots remains constant regardless of the size of the environment.

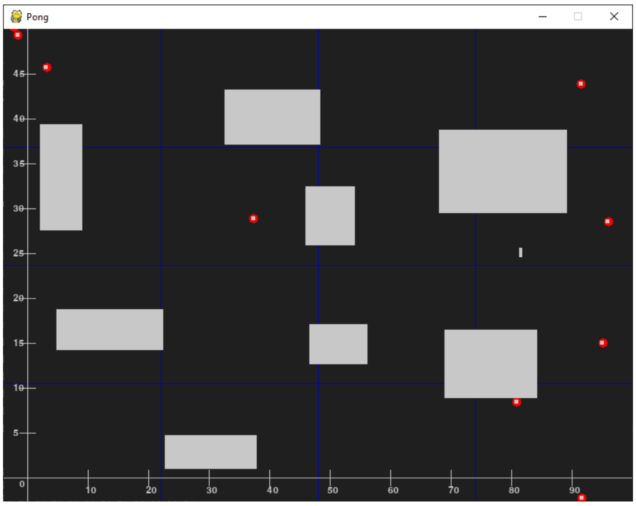

Figure-9. Navigation environment of 100 m x 50 m.

ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

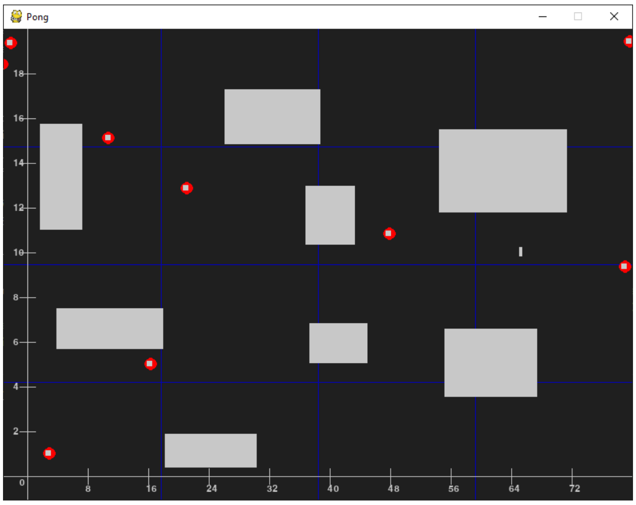

Figure-10. Navigation environment of 75 m x 20 m.

In the real world, robots have different sizes and different kinds of sensors. Our simulator has configured by default the parameters of the ARMOS TurtleBot, but it is possible to change the size of the robot and the range of the sensors. In Figure-10 the robots are 0.61 m x 0.61 m, with a sensor range of 0.8 meters. Figure-11 shows a new robot with a size of 1.2 m x 1.2 m with a sensor range of 1.5 m. The white square represents the robot, and the red circle represents the range of the distance sensors.

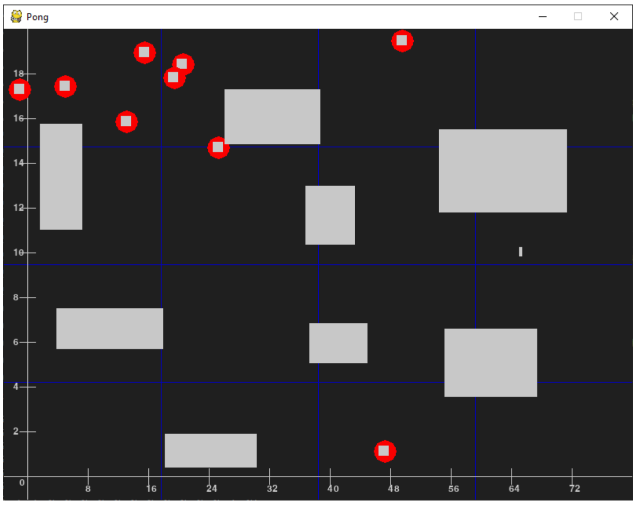

Figure-11. New robots with a size of 1.2 m x 1.2 m and sensor range of 1.5 m.

Another important parameter of the simulation algorithm is the size of the swarm, the number of robots that will perform the task. Our simulator allows us to implement the number of robots that the user wants for the algorithm as long as this does not exceed 70% of the area of the environment. For example, Figure-12 is implemented a simulation with 35 robots in an environment of 75 m x 20 m.

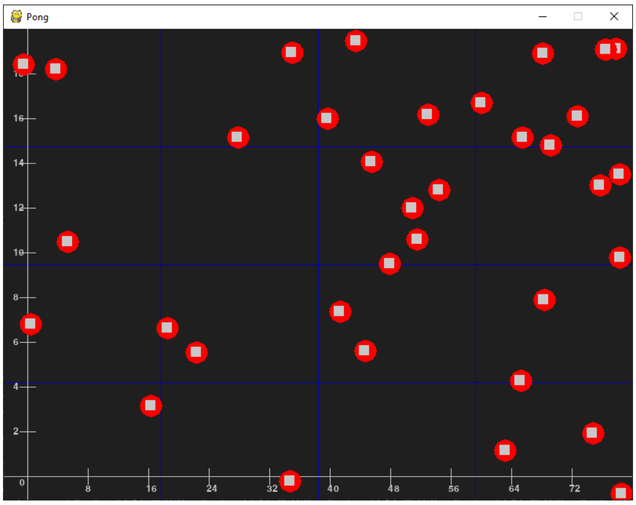

Figure-12. 35 robots defined in a simulation environment without obstacles.

To implement QS, the dynamics of the robots are governed by a wild movement. Through the initial interface, a certain quorum threshold is determined in a region chosen by the user, and a weight is given to this region. Depending on the weight specified by the user, robots that exceed the quorum threshold in a region will wait a time directly proportional to the weight of the region. Figure-13 shows how three robots in the upper-left grid reach the quorum threshold and begin waiting for more robots to arrive. This behavior is observed sometime later in Figure-14. Once all the robots are in the area of interest, the simulation will end showing the results which include the total simulation time and the time it took to exceed the quorum threshold. The simulator allows implementing the algorithm at a higher speed than real robots, which together with the scalability in the size of the swarm results in a powerful analysis tool.

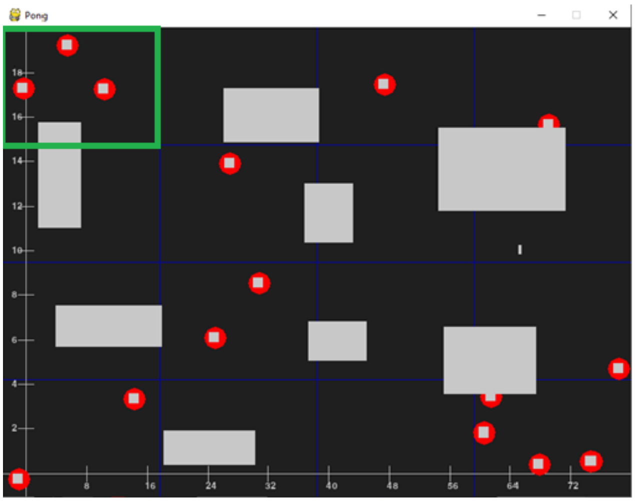

Figure-13. First robots reach quorum threshold.

ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

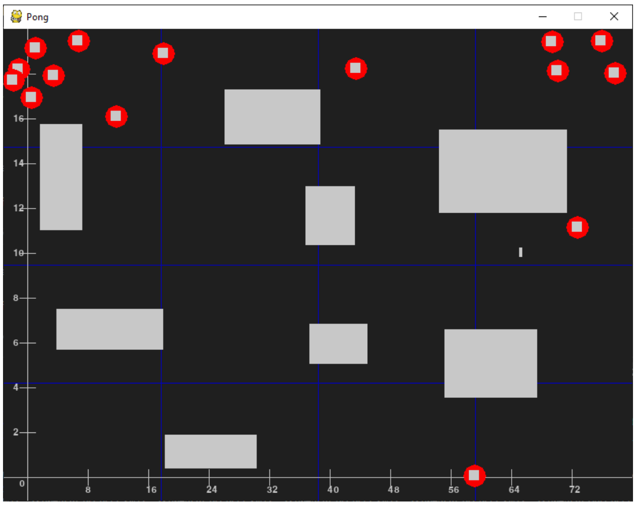

Figure-14. After some time, a large proportion of the robots are concentrated in the area of interest.

CONCLUSIONS

This article presents the design and development of a software tool designed for the specific training of the young researchers of the ARMOS research group. The design of the tool was based on the principles of competencies-based design in coherence with state policies of formal education. In that sense, an educational design was formulated that allowed to define the specific competencies to be developed, then a strategy of computational design was established that allowed to design the different sections of the software in a chained way, in coherence with the competencies to be formed, and according to the functional requirements established by the research group. This was implemented in a production phase that continuously verified the expected level of compliance of the tool throughout several cycles of redesign and testing. The developed tool corresponds to a navigation simulator for a group of autonomous robots through a movement strategy formulated and applied by the research group. This strategy derives its principle of operation of a simplified model of bacterial interaction; the tool software allows interacting with all the parameters of this algorithm, as well as of the robot and the environment. This tool is being applied to the students of the group, and its real impact is part of the research project that gives continuity to this project.

ACKNOWLEDGEMENT

This work was supported by the Universidad Distrital Francisco José de Caldas, in part through CIDC, and partly by the Facultad Tecnológica. The views expressed in this paper are not necessarily endorsed by Universidad Distrital. The authors thank the research group ARMOS for the evaluation carried out on prototypes of ideas and strategies.

REFERENCES

[1] G. Calvo. 1996. Los proyectos educativos institucionales y la formación de docentes. Revista Colombiana de Educación, (33). ISSN 2323-0134. doi:10.17227/01203916.5400.

[2] A. Montes and A. Gamboa. 2018. Miradas sobre la calidad de la educación básica en Iberoamérica: Visiones de España y Colombia. Revista Historia de la Educación Latinoamericana, 20(31). ISSN 2256-5248. doi:10.19053/01227238.8721.

[3] M. Bernal. 2017. Atreverse a evaluar más personal, más humano, más maestro. Revista Temas: Departamento de Humanidades Universidad Santo Tomás Bucaramanga, (11): 185-200. ISSN 1692-6226, 2422-4073.

[4] S. Tobón. 2007. El enfoque complejo de las competencias y el diseño curricular por ciclos propedéuticos. Acción Pedagógica, 16(1): 14-28. ISSN 1315-401X.

[5] C. Garzón. 2020. Educación inclusiva en las instituciones de educación superior que ofertan programas por ciclos propedéuticos en Bogotá 2010-2020: revisión de la literatura. Revista Boletín Redipe, 9(7): 71-90. ISSN 2256-1536. doi:10.36260/rbr.v9i7.1019.

[6] D. Forero, M. Majeed and P. Ruiz-Díaz. 2020. Current trends and future perspectives for medical education in Colombia. Medical Teacher, 42(1): 17-23. doi:10.1080/0142159X.2019.1659944.

[7] J. De Poorter and N. Aguilar-Forero. 2020. The emergence of global citizenship education in Colombia: Lessons learned from existing education policy. Compare, 50(6): 865-883. doi:10.1080/03057925.2019.1574558.

[8] P. Páez and C. Teelken. 2016. Perceived quality of private education and fears of stratification: Investigating the propositions of human capital theory by exploring the case of Colombia. Policy Futures in Education, 14(6): 597-616. doi: 10.1177/1478210316639416.

[9] E. Melo, J. Llopis, J. Gascó and R. González. 2020. Integration of ICT into the higher education process: The case of Colombia. Journalof Small Business Strategy. 30(1): 58-67.

[10] F. Diaz-Quijano, R. Martínez-Vega, A. Rodriguez-Morales, R. Rojas-Calero, M. Luna-González and R. Díaz-Quijano. 2018. Association between the level of education and knowledge, attitudes and practices

regarding dengue in the Caribbean region of Colombia. BMC Public Health, 18(1). doi: 10.1186/s12889-018-5055-z.

[11] J. Jaramillo, J. Pulido, J. Núñez, W. Bird and T. Komabayashi. 2010. Dental education in Colombia. Journal of oral science, 52(1): 137-143. doi:10.2334/josnusd.52.137.

[12] O. Guerra and J. Rojas. 2011. Knowledge identification, capture and dissemination: ECOPETROLS. A. real example of implementation. Proceedings of the European Conference on Knowledge Management, ECKM: 381-389.

[13] 2003. Tertiary education in Colombia: Paving the way for reform. World Bank Country Study, (APRIL): 1-205.

[14] T. Melguizo, F. Sanchez and T. Velasco. 2016. Credit for low-income students and access to and academic performance in higher education in colombia: A regression discontinuity approach. World Development, 80: 61-77. ISSN 0305-750X. doi: https://doi.org/10.1016/j.worlddev.2015.11.018.

[15] E. Hern Ã¡ ndez, J. Pimentel, D. Orozco and C. Carratalá. 2016. Inclusion of the equity focus and social determinants of health in health care education programmes in colombia: A qualitative approach. Family Practice, 33(3): 268-273. ISSN 0263-2136.

[16] P. Rowley and C. Walker. 2020. Adapt or perish: A new approach for industry needs-driven Master’s level low-carbon energy engineering education in the UK. Energies, 13(9). doi: 10.3390/en13092246.

[17] O. Prokopenko, I. Osadchenko, O. Braslavska, I. Malyshevska, M. Pichkur and V. Tyshchenko. 2020. Competence approach in future specialist skills development. International Journal of Management, 11(4): 645-656. doi:10.34218/IJM.11.4.2020.062.

[18] K. Chesnov, A. Chanchina, A. Fedorova, N. Zanozin, A. Makeeva, N. Bystrova and V. Trostin. 2019. Program and methodological support of the educational process in the magistracy. International Journal of Innovative Technology and Exploring Engineering, 9(1): 4691-4694. doi:10.35940/ijitee.A4728.119119.

[19] A. Bakanova, N.E. Letov, D. Kaibassova, K.S. Kuzmin, K.V. Loginov and A.N. Shikov. 2019. The use of ontologies in the development of a mobile e-learning application in the process of staff adaptation. International Journal of Recent Technology and Engineering, 8(2 Special Issue 10): 780-789. doi:10.35940/ijrte.B1144.0982S1019.

[20] C. Collazos and L. Merchan. 2015. Human-computer interaction in Colombia: Bridging the gap between education and industry. IT Professional, 17(1): 5-9. doi:10.1109/MITP.2015.8.

[21] A. Tahirsylaj and D. Sundberg. 2020. The unfinished business of defining competences for 21st century curricula-a systematic research review. Curriculum Perspectives, 40(2): 131-145. doi: 10.1007/s41297-020-00112-6.

[22] V. Molozhavenko, A. Abramovsky and A. Kovalev. 2020. Innovative technologies in higher education. Asia Life Sciences, Supp 22(2): 29-42.

[23] M. Barrett. 2020. The council of Europe’s reference framework of competences for democratic culture: Policy context, content and impact. London Review of Education, 18(1): 1-17. doi:10.18546/LRE.18.1.01.

[24] K. Gámez-Pérez, A. Sarmiento, H. Garcia-Reyes and J. Velázquez-Martínez. 2020. An international university-industry collaboration model to develop supply chain competences. Supply Chain Management, 25(4): 475-487. doi: 10.1108/SCM-08-2019-0317.

[25] N. El_haggar and F. Alshammary. 2020. Towards creativity and innovation by integrating employability competences into training programs for promoting graduates employability. International Journal on Emerging Technologies. 11(3): 873-880.

[26] E. Jacinto, F. Martínez and F. Martínez. 2016. Learning Strategies for Cryptography Using Embedded Systems. In: Smart Education and E-Learning 2016, edited by Vladimir L. Uskov, Robert J. Howlett and Lakhmi C. Jain. Cham: Springer International Publishing, Smart Innovation, Systems and Technologies. ISBN 978-3-319-39690-3, 495-505.

[27] F. Martínez, H. Montiel and E. Jacinto. 2016. Inductive Teaching and Problem-Based Learning as Significant Training Tools in Electrical Engineering| Springer Link.

[28] Y. Orekhova, L. Grebenkina, N. Suvorova and M. Stavruk. 2020. Pedagogical conditions of formation of

professional competence of students of a technical university. Periodico Tche Quimica. 17(34): 291-302.

[29] T. Shcherbakova, R. Malkarova, A. Sozaev, M. Anayev, R. Kulimova and S. Khakhetlova. 2020. Teacher as a subject of interaction: Communicative resources and risks. Journal of Advanced Research in Dynamical and Control Systems, 12(7 Special Issue): 163-175. doi:10.5373/JARDCS/V12SP7/20202095.

[30] C. Schmieder. 2020. Qualitative data analysis software as a tool for teaching analytic practice: Towards a theoretical framework for integrating QDAS into methods pedagogy. Qualitative Research, 20(5): 684-702. doi: 10.1177/1468794119891846.

[31] S. Demir and N. Manav Demir. 2020. Implementation of Activated Sludge Model No. 3 as an educational tool: bioXL3. Computer Applications in Engineering Education, 28(5): 1154-1173. doi:10.1002/cae.22292.

[32] S. Otarawanna, K. Ngiamsoongnirn, A. Malatip, P. Eiamaram, S. Phongthanapanich, E. Juntasaro, P. Kowitwarangkul, T. Intarakumthornchai, P. Boonmalert and C. Bhothikhun. 2020. An educational software suite for comprehensive learning of Computer-Aided Engineering. Computer Applications in Engineering Education, 28(5): 1083-1109. doi:10.1002/cae.22285.

[33] Annyi Saavedra, Esneider Vargas and Fernando Martínez. 2014. Diseño de un software para el aprendizaje de las medidas eléctricas en circuitos DC. Tekhnê, 11(2): 69-76. ISSN 1692-8407.

[34] D. Bello, L. García and F. Martínez. 2013. Planta robusta de levitación neumática para investigación y formación en control y visión artificial. Tekhnê, 10(2): 33-41. ISSN 1692-8407.

[35] A. Sánchez, and J. De Los Ríos R. 2015. Software didáctico para enseñanza de colores usado en niños menores de cinco años. Tekhnê, 12(2): 65-74. ISSN 1692-8407.

[36] P. Akiki, P. Akiki, A. Bandara and Y. Yu. 2020. EUD-MARS: End-user development of model-driven adaptive robotics software systems. Science of Computer Programming, 200. doi:10.1016/j.scico.2020.102534.

[37] N. Campos, M. Nogal, C. Caliz and A. Juan. 2020. Simulation-based education involving online and on-campus models in different European universities. International Journal of Educational Technology in Higher Education, 17(1). doi: 10.1186/s41239-020-0181-y.