Lecture Notes in Networks and Systems 1027

Michael E. Auer
Reinhard Langmann
Dominik May
Kim Roos *Editors*

Smart Technologies for a Sustainable Future

Proceedings of the 21st International Conference on Smart Technologies & Education. Volume 1

Springer

Lecture Notes in Networks and Systems 1027

The series "Lecture Notes in Networks and Systems" publishes the latest developments in Networks and Systems—quickly, informally and with high quality. Original research reported in proceedings and post-proceedings represents the core of LNNS.

Volumes published in LNNS embrace all aspects and subfields of, as well as new challenges in, Networks and Systems.

The series contains proceedings and edited volumes in systems and networks, spanning the areas of Cyber-Physical Systems, Autonomous Systems, Sensor Networks, Control Systems, Energy Systems, Automotive Systems, Biological Systems, Vehicular Networking and Connected Vehicles, Aerospace Systems, Automation, Manufacturing, Smart Grids, Nonlinear Systems, Power Systems, Robotics, Social Systems, Economic Systems and other. Of particular value to both the contributors and the readership are the short publication timeframe and the worldwide distribution and exposure which enable both a wide and rapid dissemination of research output.

The series covers the theory, applications, and perspectives on the state of the art and future developments relevant to systems and networks, decision making, control, complex processes and related areas, as embedded in the fields of interdisciplinary and applied sciences, engineering, computer science, physics, economics, social, and life sciences, as well as the paradigms and methodologies behind them.

Indexed by SCOPUS, INSPEC, WTI Frankfurt eG, zbMATH, SCImago.

All books published in the series are submitted for consideration in Web of Science.

For proposals from Asia please contact Aninda Bose (aninda.bose@springer.com).

Michael E. Auer · Reinhard Langmann ·
Dominik May · Kim Roos
Editors

Smart Technologies for a Sustainable Future

Proceedings of the 21st International Conference on Smart Technologies & Education. Volume 1

Springer

Editors
Michael E. Auer
CTI Global
Frankfurt/Main, Germany

Reinhard Langmann
Edunet World Association e.V.
Blomberg, Germany

Dominik May
University of Wuppertal
Wuppertal, Germany

Kim Roos
Programme Director MScEng
Arcada University of Applied Sciences
Helsinki, Finland

ISSN 2367-3370 ISSN 2367-3389 (electronic)
Lecture Notes in Networks and Systems
ISBN 978-3-031-61890-1 ISBN 978-3-031-61891-8 (eBook)
https://doi.org/10.1007/978-3-031-61891-8

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Scratch-Based Exergame-Educational Approach in Teaching the Course "Physical Education" for IT-Specialties

Oleksandr Blazhko[1(✉)], Viktoriia Podhorna[2], Anastasiia Kokotieieva[2], and Nataliia Bashavets[3]

[1] Department of Information Systems, Odessa Polytechnic National University, Odesa, Ukraine
blazhko@op.edu.ua
[2] Department of Physical Education and Sports, Odessa Polytechnic National University, Odesa, Ukraine
{podgorna.v.v,kokotieieva.a.s}@op.edu.ua
[3] South Ukrainian National Pedagogical University Named After K. D. Ushynsky, Odesa, Ukraine
bashavetsnata@ukr.net

Abstract. I's proposed the approach of computer game control of human movements in sport computer games in order to increase the level of physical activity of stu-dents in online classes, which takes into account the initial knowledge of the spe-cialized disciplines of the curriculum. The approach is based on an open-access repository of computer programs created in the Scratch-based block language (47 games for 28 summer sports, 16 games for 10 winter sports), Scratch programs for human motion recognition based on the PoseNet neural network, which allow changing the principle of controlling game characters through the keyboard and mouse. The approach uses static gestures – unchanging position of the human body with fixed relationships of its various parts to each other, dynamic gesture – sequential transition in time between two static gestures and Scratch-block descriptions of recognition of gestures.

Keywords: ExerGame · motion recognition · Scratch

1 Introduction

Nowadays, there are means of interaction between a person and a computer through human movements' recognition, which allows controlling events in entertaining or educational computer games using ExerGame technologies ("exercise" + "game") [1]. Non-contact motion control sensors, such as the MS Xbox 360/One game console, which contains the MS Kinect sensor with an infrared camera [2] to support Kinect Sport 3D computer sports games have been used for many years to recognize human movements: Tennis, Table Tennis, Golf, Skiing, Baseball, Soccer, Basketball, Boxing, Track and Field (Sprint, Javelin, Discus, Long Jump and Hurdles). Due to COVID restrictions in 2020–2021 and military actions from 2022, students could not come to the sports

M. E. Auer et al. (Eds.): STE 2024, LNNS 1027, pp. 324–335, 2024.
https://doi.org/10.1007/978-3-031-61891-8_32

facilities of Odesa National Polytechnic University for "Physical Education" classes. Therefore, the "Physical Education and Sports" department together with the "Information Systems" department decided to adapt the discipline program through the ExerGame technologies' introduction. However, the use of these technologies in the first-term students' training program has limitations during online classes: (1) lack of special infrared cameras; (2) lack of access to commercial computer games; (3) lack of knowledge to independently create 3D computer games. The first limitation can be partially removed by using programs for processing human images from a Web-camera based on various machine learning models, for example, PoseNet [3]. The second limitation can be partially removed by using the Scratch programming environment [4] – a tool for imparting computational thinking skills from an early age in schools. Although Scratch allows the creation of computer programs with simple 2D graphics, which cannot be compared to commercial 3D games, Scratch provides access to working computer games created by pupils from around the world and hosted in open-file game repositories [5]. Such a repository allows choosing a game with an interesting scenario, exploring the open source code, and, if desired, making changes to it, and creating a new version of the game. Unfortunately, there are currently no examples of motion control sports games in the Scratch repository. However, the work [6] represents a special program for displaying human movement from the MS Kinect v.1 sensor in Scratch v.1.4 sensor blocks, which allowed the author to quickly create several games with different scenarios [7]: "Alien attack" controls body movements left/right and raising the arms above the head, "Hungry shark" controls the head up/down movements and joining the palms together, "Hungry ant" controls the right hand's movements. During the work [8], new games were created for the stroke patients' rehabilitation. These games can be easily adapted for the webcam instead of the MS Kinect infrared camera if you use the free neural network communication software libraries with the Scratch environment. The PoseBlocks web-program [9] was one of the first examples of free programs, which provided the Scratch environment with new sensory blocks based on the AImodel neural network for tracking body, and based on own images/poses/audio -models from the Google Teachable Machine (GTM) [10]. The PictoBlox [11] desktop program, also supports GTM using the Scratch language. The software library [12] allows representing the human body in the form of reference points based on the PoseNet neural network, as is done by the MS Kinect sensor, and transferring the coordinates of these points to the Scratch environment sensor blocks.

The analysis of works determined the need to modify the "Physical Education" curriculum in the first term for students studying IT specialties in the form of a Scratch-Based Exergame-Educational Approach (exergaming approach) using computer games based on the Olympic sports themes, which should take into account simple technical requirements for students during online classes and low requirements for knowledge of programming languages:

- the presence of a regular web-camera connected to the computer;
- the knowledge of the basics of programming in the Scratch environment;
- the access to the Scratch library for connection with the PoseNet neural network;
- the access to the open Scratch repository of games on the Olympic sports themes.

2 Stages of the Scratch-Based Exergame-Approach

The classes conduct by the ExerGame-approach, which includes three stages (see Fig. 1), were proposed.

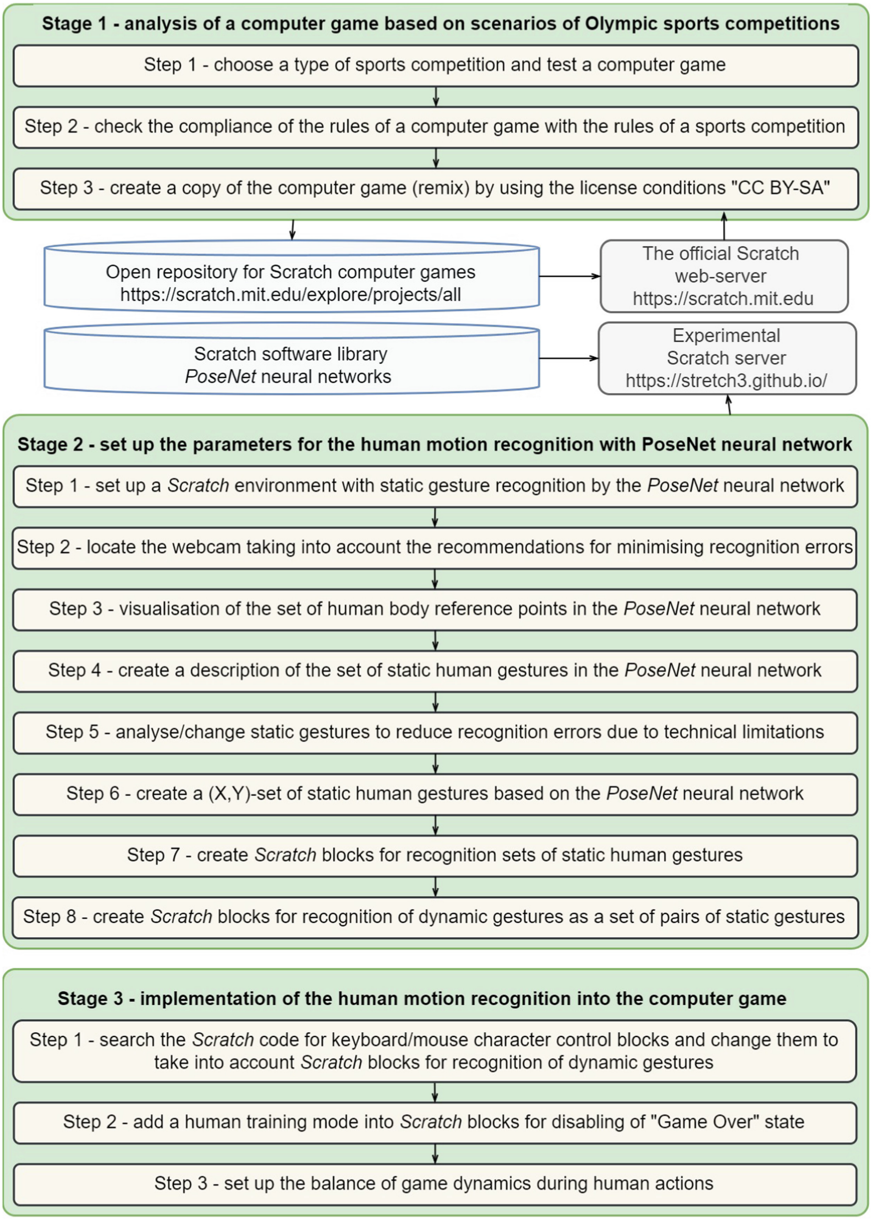


Fig. 1. The Exergaming approach of classes conduct based on 3 stages and a sequence of steps.

2.1 Stage 1 - Analysis of a Computer Game Based on Scenarios of Olympic Sports Competitions

In order to determine the list of computer games that students will explore and change during online classes, the authors of the ExerGame-approach in the Scratch-repository searched and selected computer games according to the following steps:

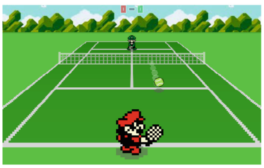
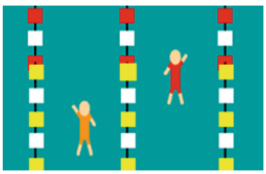
1) determination of the English-language names of official summer and winter sports;
2) search in the repository of the list of computer games by the English name of sports;
3) selection of found games by type of single player and by decreasing popularity of their use by the community;
4) selection of computer games by the kind of sports in different planes of the player's view projection: frontal (f-plane) – front/back view, profile (p-plane) – side view, horizontal (h-plane) – top view.

As a result of search the following was found:

- 47 computer games for 28 summer Olympic sports
- 16 computer games for 10 winter Olympic sports

For each game, the following was defined: the projection planes of the player's view, the project code in the Scratch repository, the computer keyboard/computer mouse keys controlling the movements of the computer athlete in the game, and the game screen example. Part of the description of the games found is presented in Table 1.

Table 1. Part of the description of the games found.

Type of sport, projection plane, project code	Computer keyboard/computer mouse control keys	Game screen example
Judo, 566395367, p-plane	A, D, W, S, W+Space, S+Space – move forward/backward, raise/lower defense, upper/lower attack	
Tennis, 520716879, f-plane	Left, right, forward, backward arrow keys – moving in directions, Space key – racket hits	
Swimming, 178358511, h-plane	Right and left arrow keys – movement with the right or left hands	

2.2 Stage 2 – Set up the Parameters for the Human Motion Recognition with PoseNet Neural Network

The Scratch environment can be hosted on any computer due to the open source code, and Scratch v.3 can be hosted on any web-server and new software libraries can be added. One of the web-servers hosts the PoseNet2Scratch software library for recognizing human movements based on the PoseNet [12] neural network.

When using the PoseNet2Scratch software library, students must consider the conditions for working with the webcam:

- the webcam is located at the level of human eyes, but the webcam can be placed lower;
- students stand in front of the even color wall, which contrasts with the person's clothes so that the PoseNet neural network can correctly distinguish the outline (silhouette) of the person's image against the wall image background;
- there are no extra objects in the area monitored by the webcam;
- the webcam is located at a distance sufficient to display the body parts of the students used in the sports' movements (2–5 m);
- the room is illuminated but without the light glare on the wall in front of which the students are standing.

The PoseNet neural network allows representing the human body in the form of 17 reference points (see Fig. 2).

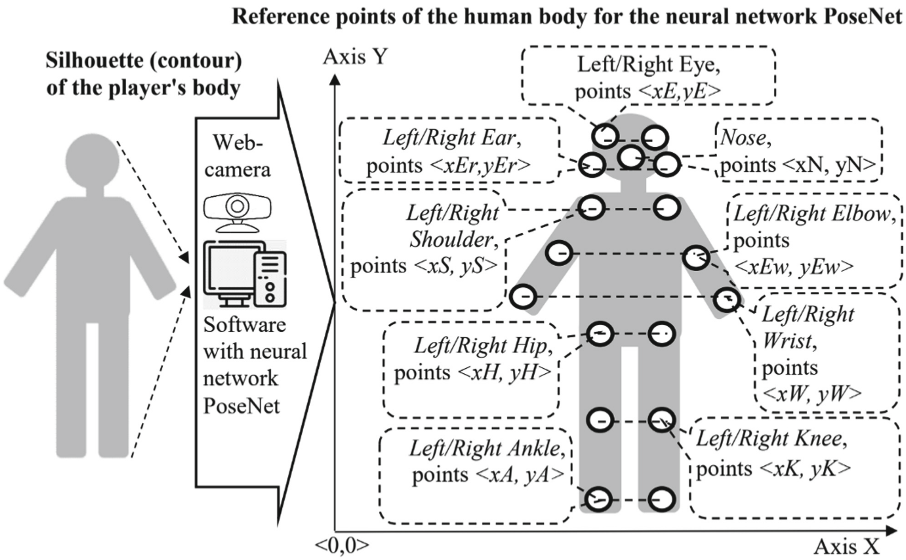
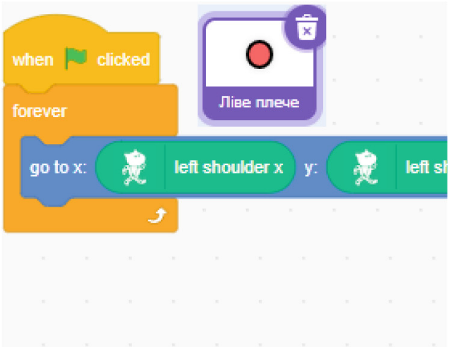


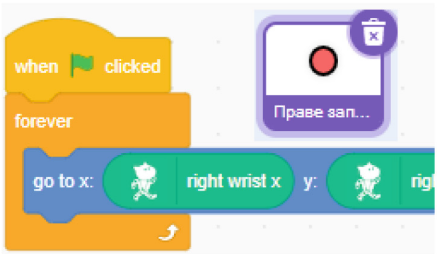
Fig. 2. The human body anchor points created in the PoseNet neural network

The PoseNet2Scratch software library adds a section of green blocks to Scratch to control anchor points. Figure 3 shows fragments of the sprite placement rendering software blocks associated with the anchor points of the nose, left shoulder, and right wrist.

(a) visualization of the location of the nose



(b) visualization of the location of the left shoulder



(c) visualization of the location of the right wrist

(d) examples of visualization of reference points

Fig. 3. Fragments of software Scratch-blocks for visualization of the location of sprites.

Each description of the athlete's movement can be considered as a sequence of poses—positions taken by the human body, head, and limbs in relation to each other.

If we talk about the process of human poses recognition by a computer, then a pose can be considered as a gesture – an action or movement of the human body or its part, which has a certain meaning or sense, that is, it is a sign or symbol.

The following terms will be used in the future:

- static gestures (sg) – unchanging position of the human body with fixed relation of its various parts to each other;
- dynamic gesture (dg) – a sequential transition in time between two static gestures.

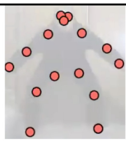
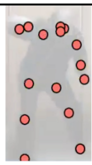
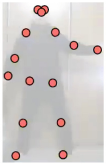
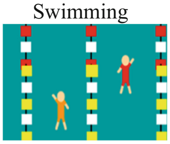
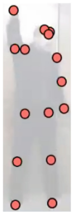
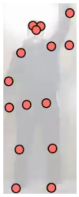
For static gestures, their coordinate description is proposed, which takes into account the following:

- for each pair of static gestures, anchor points that determine a significant difference between static gestures are searched, and a logical expression comparing the values of the coordinates of these anchor points is described;

- one should strive for a minimum number of comparisons between reference points, but guarantee a significant difference in their description between different static gestures.

Table 2 for three computer games presents examples of screenshots from the web-camera in the form of human body anchor points with static gestures in the Scratch program with a connected PoseNet neural network and a coordinate description of gestures.

Table 2. Table for three computer games presents examples of screenshots from the web-camera in the form of human body anchor points with static gestures in the Scratch program with a connected PoseNet neural network and a coordinate description of gestures.

Kind of sports	Screenshot of static gesture 1	Screenshot of static gesture 2	Static gesture coordinate description
Judo 	 *sg1* – defense	 *sg2* – attack	*sg1: (xK < xH) and (xW < xA) and ((ySL - yHL) = (xHR - xHL)) sg2: (yWL = yN) and (yWR < yEwR) and (xN = xH)*
Tennis 	 *sg1* – racket ready to hit	 *sg2* – hit with a racket	*sg1: (yWL > yEwR) and (xWL > xEwL) sg2: (yWL > yEwR) and (xEwL = xEwL)*
Swimming 	 *sg1* – right hand swing	 *sg2* – left hand swing	*sg1: (yWR > yN) and (yWL < yEwR) sg2: (yWL > yN) and (yWR < yEwL)*

To program the recognition conditions of the specified gestures, it is recommended to create variables, for example, *nose_y, left_wrist_y, right_wrist_y*, the values of which are set in the software modules for the anchor points visualization (see Fig. 4).

Figure 5 shows a fragment of the Scratch program where two static gestures "Both hands below nose" and "Both hands above nose" are first recognized using the following steps:

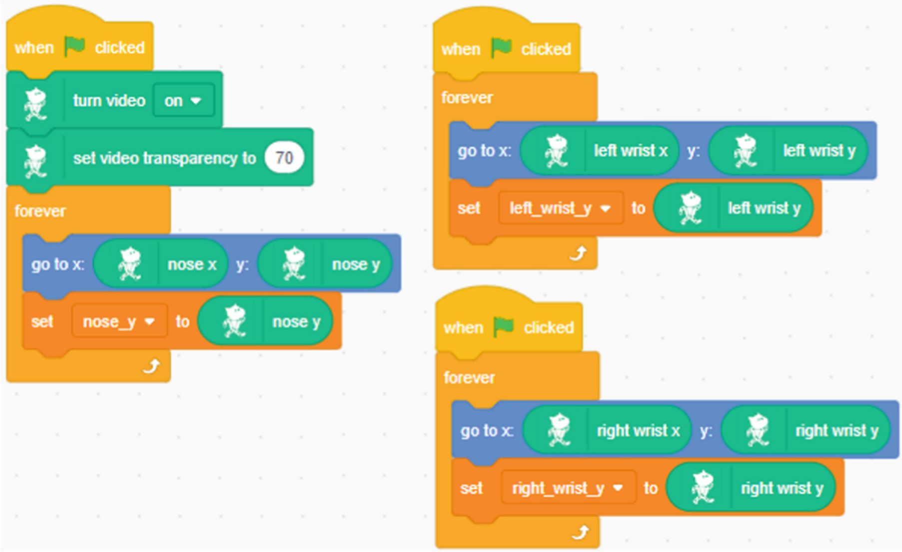


Fig. 4. Fragments of the Scratch program that set the values of the *nose_y, left_wrist_y, right_wrist_y* variables for programming static gesture recognition conditions.

1. checking the condition of the first static gesture recognition, which is defined by the logical expression "*sg1: (yN > yWL) and (yN > yWR)*", using the variables *nose_y, leftwrist_y, right_wrist_y*;
2. if the result of the logical expression check of static gesture returns true then set variable *new_static_gesture* = 1 as *sg1*;
3. checking the condition of the second static gesture recognition, which is defined by the logical expression "*sg2: (yWL > yN) and (yWR > yN)*", using Scratch blocks *leftwrist_y, right_wrist_y, nose_y*;
4. if the result of the logical expression check of static gesture returns true then set variable *new_static_gesture* = 2 as *sg2*;

Scratch blocks of dynamic gesture recognition take into account the transition between a previously recognized static gesture and a new recognized static gesture. Figure 5 also shows a fragment of the Scratch program, which recognizes two dynamic gestures: *dg1 – "hands raised above the nose"*, *dg2 – "hands lowered below the nose"*, through previously programmed recognition processes of two static gestures *sg1*, *sg2*.

The *dynamic_gesture* variable is used to establish the fact of dynamic gesture recognition, which can later be used in the program code of the game character controlling conditions analysis to replace the classic control version in the game through a computer keyboard or a computer mouse.

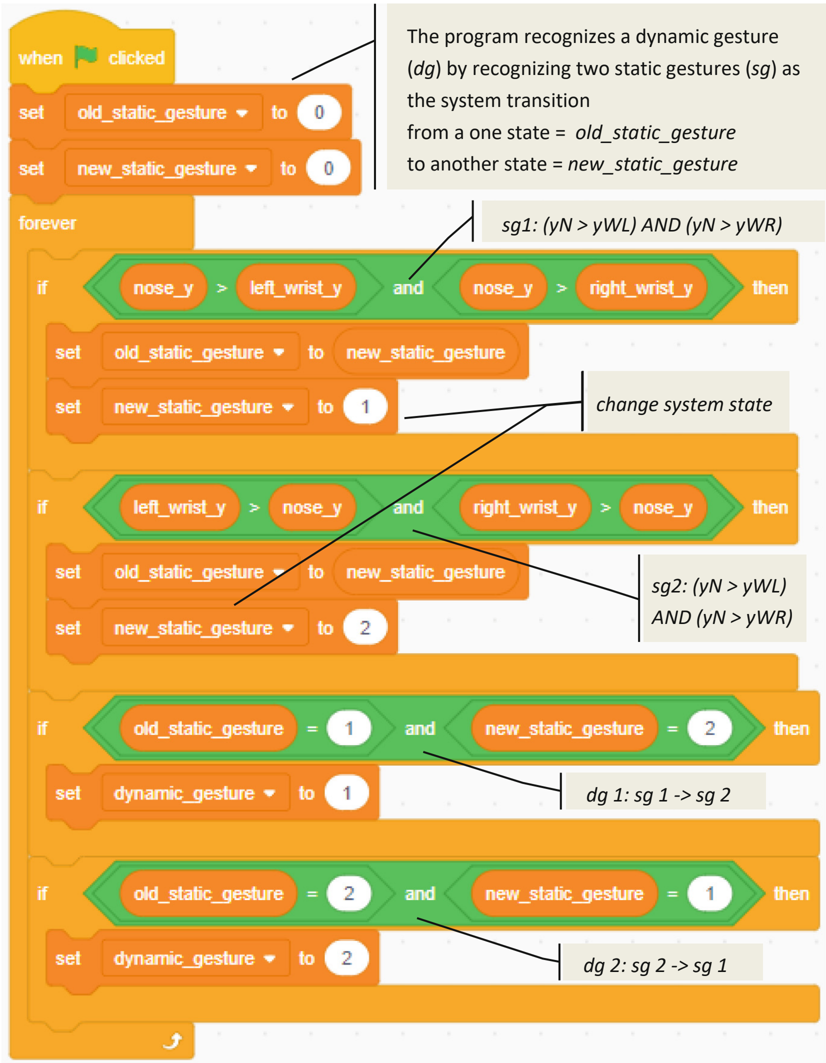


Fig. 5. A fragment of the Scratch-program that recognizes two static gestures and, based on them, subsequently recognizes two dynamic gestures.

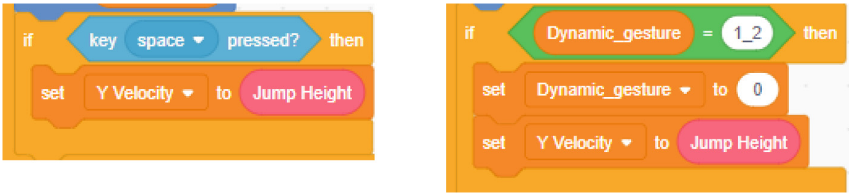
2.3 Stage 3 – Implementation of the Human Motion Recognition into the Computer Game

The existing Scratch program of a computer game uses two classic means to control the game character's movements:

- pressing the keyboard keys, which are programmed by the blue Scratch-block with the name "key pressed";
- pressing the keys of the "mouse" type manipulator, which are programmed by the blue Scratch block called "mouse down";

- horizontal movement of the “mouse” type manipulator, which is programmed by blue Scratch blocks with the names “mouse x”, and “mouse y”.

Therefore, at the beginning of the method’s third stage, it is necessary: (1) to find Scratch blocks with classic computer game controls in the computer game; (2) to replace classic blocks with blocks that take dynamic gesture into account. Figure 6 shows the example of fragments of a Scratch program with the means of a computer game control through the Space key of the keyboard, as well as its replacement with a variable that stores the number of a previously recognized dynamic gesture. After successfully checking that the dynamic gesture matches the specified number, the value of the variable must be cleared to start a new iteration of new dynamic gesture recognition.



(a) a computer game control means through the Space key of the keyboard

(b) replacing the control with a variable that stores the number of a previously recognized dynamic gesture

Fig. 6. Fragments of the Scratch program with the computer game control

One of the examples of the psychological and communication theory of games is the Mechanics-Dynamics-Aesthetics (MDA) structure as a tool for analyzing existing games and synthesizing new games based on three components [13]: Mechanics, Dynamics, and Aesthetics. “Mechanics” includes [14]: Game rules – the rules that determine the purpose of the game, for example, “create”, “destroy”, “avoid”, and “match”; Play rules – the rules of manipulation that determine the basic actions that the player can apply in the game, for example, “move”, “select”, “record”, “control”, “get lucky”, and “shoot”. “Dynamics” is a game result of using “Mechanics”, caused by:

- the players’ actions, which depend on their psychical and physiological abilities, for example, the brain’s reaction speed, the fingers’ motility speed when pressing the keys of the keyboard/mouse wheel;
- results of different mechanics’ interaction with each other.

After switching to game control from using dynamic gestures, the game may become unbalanced because the player does not have time to perform the appropriate movements for the dynamic gesture conditions to work. Therefore, students need to find Scratch blocks that affect the dynamics of the game, for example, determine the character’s movement speed, and the events’ change speed in the game.

3 Analysis of the Exergame-Approach Implementation

In October 2023, 92 first-year students majoring in Computer Science took part in the "Physical Education" online classes based on the Exergaming approach for five weeks, among them 68 students (74%) completed stages 1–3. In order to receive feedback, a questionnaire was conducted among students, the results of which revealed the following:

- 26% of students did not study programming languages at school;
- programming languages studied at school: Scratch – 50%, Python – 25%, Pascal – 15%, C/C++ – 15%, C# – 7%, Java – 6%, Basic – 4%.
- 30% of students did not experience difficulty with completing tasks;
- 20% of students indicated problems with the Internet speed;
- 17% of students had a "weak" computer;
- 15% of students did not have a webcam, but among them only a third could not complete the task, because others did it by temporarily renting webcams;
- 37% of students indicated that the Exergaming approach can be used as an aid for the motor activity increase;
- 26% of students believe that the Exergaming approach can be effective only in the online education period;
- 24% of students do not see the benefit in using the Exergaming approach;
- 57% of students felt motivated to continue practicing Exergaming approach and the desire to achieve a decent result;
- 28% of students felt oppressed while completing tasks, but this may be related to technical difficulties, which were noted in the corresponding item of the survey;

The questionnaire results also showed that the Exergaming approach developed the participants' Soft Skills as the students identified the following priorities for the Soft Skills types:

- 20% – adaptability – the ability to change ways of thinking according to the tasks and the conditions for solving them;
- 15% – creativity – the ability to create and find new original ideas, and solve tasks in a non-standard way;
- 13% – critical thinking – the ability to reason and effectively analyze information.

4 Conclusion

The analysis of the results of the first experimental implementation of exergaming approach showed that 1st-year students, starting to study "Physical Education" in the 1st term and not having sufficient skills in using professional programming languages, can already create computer games with 2D graphics where they will test their motor skills. Computer gamification of physical exercise support based on human movement control hardware and software can be introduced as an optional part of "Physical Education" with connections to other disciplines:

- the results of students' current studying in the disciplines they study in parallel, for example, in "Algorithmizing and Programming";

- the review of students' future learning outcomes in such disciplines as "Intelligent Data Analysis"/"Artificial Intelligence Methods and Systems", "Programming of IoT systems" and "Computer Game Systems Development".

For many years, the Scratch programming environment has been the basis for STEM in school education, therefore the proposed exergaming approach can be used in pupils' non-formal education.

References

1. Krause, J.M., Jenny, S.E.: Physical educators' exergaming integration experiences, attitudes, and self-efficacy beliefs. Phys. Educ. **80**, 91–107 (2023)
2. Kourakli, M., Altanis, I., Retalis, S., Boloudakis, M., Zbainos, D., Antonopoulou, K.: Towards the improvement of the cognitive, motoric and academic skills of students with special educational needs using Kinect learning games. Int. J. Child-Comput. Interact. **11**, 28–39 (2017)
3. Chung, J.-L., Ong, L.-Y., Leow, M.-C.: Comparative analysis of skeleton-based human pose estimation. Futur. Internet **14**, 380 (2022)
4. Grover, S.: Teaching and assessing for transfer from block-to-text programming in middle school computer science. In: Hohensee, C., Lobato, J. (eds.) Transfer of Learning. Research in Mathematics Education, pp. 251–276. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-65632-4_11
5. Scratch project Explore (2021). https://scratch.mit.edu/explore/projects/games/
6. Howell, S.: Install Scratch and Kinect2Scratch. In: St. Jean, J. (ed.), Kinect Hacks: Tips and Tools for Motion and Pattern Detection, pp. 184–206. O'Reilly, Sebastopol, CA (2012)
7. Howell, S.: Kinect2Scratch (Version 1.5) [Computer Software] (2012). https://stephenhowell.github.io/kinect2scratch/
8. Hung, J.-W., et al.: Comparison of kinect2scratch game-based training and therapist-based training for the improvement of upper extremity functions of patients with chronic stroke: a randomized controlled single-blinded trial. Eur. J. Phys. Rehabil. Med. (2019)
9. Jordan, B., Devasia, N., Hong, J., Williams, R., Breazeal, C.: PoseBlocks: a toolkit for creating (and Dancing) with AI. In: AAAI, vol. 35, no. 17, pp. 15551–15559 (2021)
10. Carney, M., et al.: Teachable machine: approachable web-based tool for exploring machine learning classification. In: Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems, CHI EA '20, pp. 1–8. Association for Computing Machinery, New York, NY, USA (2020)
11. Cruz, S., Bento, M., Lencastre, J.A.: Computational thinking training using PICTOBLOX. In: Kommers, P., et al. (ed.), Proceedings of International Conferences Internet Technologies & Society 2021, pp. 53–60 (2021)
12. Junya Ishihara: PoseNet2Scratch (2021). https://github.com/champierre/posenet2scratch
13. Duarte, L.C.S., Battaiola, A.L.: Distinctive features and game design. Entertain. Comput. **21**, 83–93 (2017)
14. Ben-Sadoun, G., Alvarez, J.: Gameplay bricks model, a theoretical framework to match game mechanics and cognitive functions. Games Cult. **18**(1), 79–101 (2023)

Author Index

M. E. Auer et al. (Eds.): STE 2024, LNNS 1027, pp. 357–358, 2024.
https://doi.org/10.1007/978-3-031-61891-8