L-Università ta' Malta

ARI2201

Individual Assigned Practical Task

Course Assignment 2022/2023

Play Games via Gestures

By Mark Dingli

mark.dingli.21@um.edu.mt
20703H

Table of contents

Plagiarism Declaration Form

Plagiarism is defined as *"the unacknowledged use, as one's own, of work of another person, whether or not such work has been published, and as may be further elaborated in Faculty or University guidelines"* (University Assessment Regulations, 2009, Regulation 39 (b)(i), University of Malta).

I, the undersigned, declare that the report submitted is my work, except where acknowledged and referenced. I understand that the penalties for committing a breach of the regulations include loss of marks; cancellation of examination results; enforced suspension of studies; or expulsion from the degree programme.

Work submitted without this signed declaration will not be corrected, and will be given zero marks.

Mark Dingli	ARI2201	12/06/2023
Student's full name	**Study-unit code**	**Date of submission**

Title of submitted work: Playing Games Via Gestures

Student's Signature

1. Introduction

Gaming is evolving at a rapid pace, changing them world of technology as we know it. As gaming continues to captivate a growing audience, the need for intuitive, accessible, and engaging games is greater than ever. One emerging frontier in the gaming industry is gesture-based control, which offers a novel way of interacting with technology through physical hand movements. Despite its potential, gesture-based control has been largely unexplored in domains such as gaming. Traditionally, gaming has been dominated by manual and button-based controls, often limiting the immersive experience and accessibility of games.

This method, while reliable, often fails to provide a truly immersive and engaging experience for the user. The reliance on such conventional control methods can limit accessibility, particularly for individuals who might find these controls physically challenging. In addition, the growing desire for more innovative techniques in gaming calls into question the sustainability of these traditional mechanisms.

One such game that has been a cornerstone in the gaming industry, yet still conforms to traditional control mechanisms is Pac-Man. With its simple gameplay and enjoyable mechanics, Pac-Man is a timeless game that offers an excellent canvas for exploring the potential of gesture-based control. However, integrating such a control system into an existing game poses a significant challenge. It requires not only an understanding of web development technologies but also the ability to leverage the power of machine learning in a creative and effective manner.

This project set out to tackle the challenges of implementing gesture-based controls into the Pac-Man game. It aimed to adapt the classic game of Pac-Man, integrating gesture-based controls in place of the traditional manual controls. The goal was to create a more immersive, interactive, and accessible gaming experience while exploring what is possible with current web development technologies and machine learning tools. By including more natural and intuitive control methods, it is possible to significantly enhance the user experience and open up gaming to a broader audience. This assignment thus acknowledges and addresses this gap, aiming to introduce better interaction through the integration of gesture-based controls.

The report that follows will detail the journey of this project, describing the initial stages of conceptualizing the idea, the selection and utilization of specific tools and technologies, and the eventual implementation and testing of the final product. The goal is that it will not only provide an insight into the process and challenges of such an undertaking, but also inspire further innovation in this exciting field.

2. Literature Review

Gesture-based control in gaming is a relatively new and unexplored area, with limited research available. However, there are several studies and publications that have provided insight into the challenges of integrating gesture-based controls into games, as well as the impact it can have on the user experience and the increase of accessibility.

One key area of research in gesture-based control is human-computer interaction (HCI). The HCI studies focus on understanding how users interact with technology and how to design interfaces that are intuitive, efficient, and enjoyable. Research in HCI has explored various input modalities, including gesture recognition, and has highlighted the advantages of natural and intuitive interaction techniques in improving user engagement and immersion in gaming experiences [1].

In the realm of gaming, there have been some notable examples of gesture-based control implementations. For instance, researchers have investigated the use of motion sensors, such as Microsoft Kinect, to enable gesture-based control in games. Studies have demonstrated the potential of using gesture recognition to enhance gameplay and create more immersive experiences [2].

Machine learning techniques play a significant role in gesture recognition and tracking. Deep learning algorithms, in particular, have shown promise in accurately recognizing and interpreting gestures from sensor data. These algorithms have been used in various domains, including sign language recognition and human pose estimation [3]. Applying machine learning to gesture-based control in gaming allows for the creation of personalized gesture classifiers that can adapt to individual users, enhancing the precision and reliability of the control system.

Accessibility is a critical consideration when exploring gesture-based control in gaming. Traditional control mechanisms often present challenges for individuals with disabilities or limited mobility, excluding them from fully engaging in gaming experiences. Gesture-based control has the potential to provide a more inclusive gaming environment by accommodating various physical abilities and preferences. Research in accessible gaming emphasizes the significance of considering diverse user needs and implementing alternative input methods, such as gesture recognition, to ensure equal accessibility [4].

In brief, the existing literature on gesture-based control in gaming demonstrates its potential to revolutionize the way users interact with games. The studies conducted in the field of human-computer interaction, gesture-based games, machine learning and accessible gaming provide valuable insights into the benefits and challenges associated with gesture-based control. This literature review highlights the need for further research and innovation in this area to refine the integration of gesture-based control into gaming experiences, ensuring enhanced user immersion, accessibility and engagement.

3. Design of the Solution

The design solution to incorporate gesture-based controls into the Pac-Man game is both systematic and user-oriented. The solution consists of several key components: the input module, gesture recognition module, game control module, and user interface module.

The input module serves as the interface between the user and the game, capturing the user's hand gestures via a webcam video feed. These gestures are processed in real-time, forming the foundation for the gesture-based control system.

The gesture recognition module utilizes a machine learning model, specifically a deep learning classifier, trained via Google's Teachable Machine interface [5]. The model has been trained on a dataset of hand gestures labelled as 'up', 'down', 'left', 'right' and 'nothing'. This module processes the video feed from the input module and interprets the user's hand gestures, mapping them to these specific commands.

Figure 1- Different hands gestures that are used as controls

The game control module integrates the recognized gestures into the mechanics of the Pac-Man game. Based on the detected gesture, this module alters the direction of Pac-Man in the game. For example, if a 'right' gesture is detected, the module will trigger Pac-Man to move right in the game. This ensures that the gestures from the users are accurately translated into movements within the game, enhancing interactivity.

The user interface module encompasses both the visual elements of the game and the rendering of the recognized hand gestures. The game elements include the Pac-Man character, ghosts, maze, and score. The user interface module ensures a visually appealing and intuitive interface that aligns with the gesture-based control system.

By integrating these components, the solution successfully incorporates gesture-based controls into the Pac-Man game. This approach to controlling the game creates an immersive and interactive gaming experience for the user, enhancing accessibility and engagement.

4. AI Techniques Used

The primary AI technique leveraged in this solution is machine learning, specifically employing Google's Teachable Machine interface. Teachable Machine is a web-based tool developed by Google that simplifies the process of creating, training, and deploying machine learning models. It abstracts away the complexities typically associated with these processes, making machine learning more accessible to non-experts and allowing for the rapid prototyping and deployment of models [5].

In the context of this project, Teachable Machine was used to create a model capable of recognizing and interpreting four distinct hand gestures: 'up', 'down', 'left', 'right'. Considering the constant motion of the Pac-Man character, a 'nothing' class was implemented. This allowed the Pac-Man to continue moving in the last given direction when no specific gesture was detected, supporting the fluidity and continuity of the game. The Teachable Machine was set to capture at a frame rate of 24 FPS, and the duration of input was set to 8 seconds for each class. This provided roughly 180 images per class for the model to learn from.

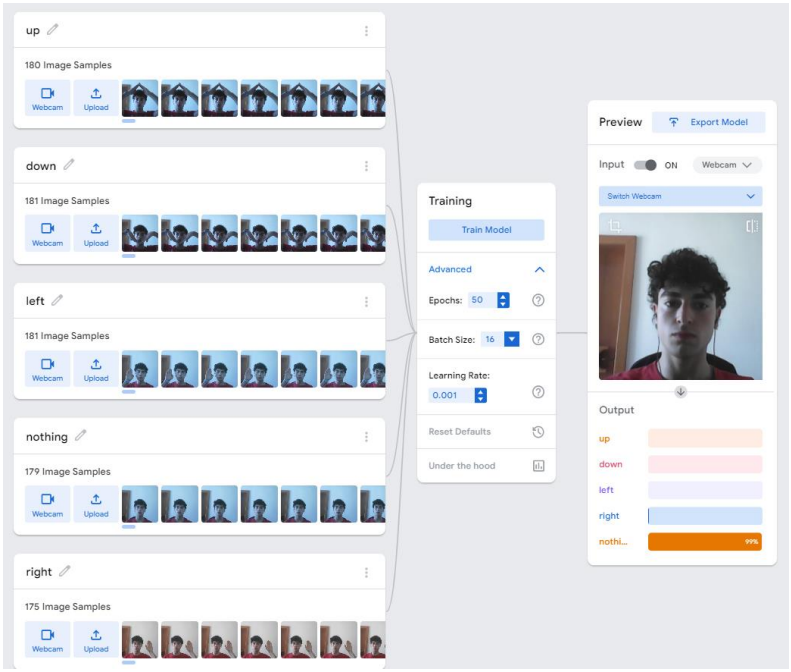

Figure 2- Screenshot of inputting and training the model

The Teachable Machine utilizes the power of TensorFlow.js, a JavaScript library for training and deploying machine learning models in the browser and on Node.js [6]. It uses techniques from deep learning, specifically convolutional neural networks (CNNs), to analyse and classify image data. CNNs are particularly adept at interpreting image data, making them ideal for a project focused on interpreting hand gestures.

The training parameters for the model were set with epochs at 50, batch size at 16, and a learning rate of 0.001. These parameters were chosen to balance the trade-off between training time and accuracy. The higher number of epochs ensures that the model has sufficient time to learn from the data, while the learning rate and batch size were set to ensure stable and efficient training.

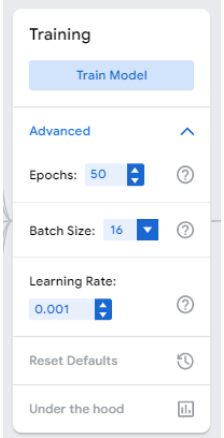

Figure 3 - Parameters used to train the model

Importantly, the interface also allows the trained model to be exported for use in other applications. This feature was essential for this project as it enabled the integration of the trained model into the game control module. The trained model could then interpret the webcam feed, recognize and classify the user's gestures, and send the appropriate commands to the game control module.

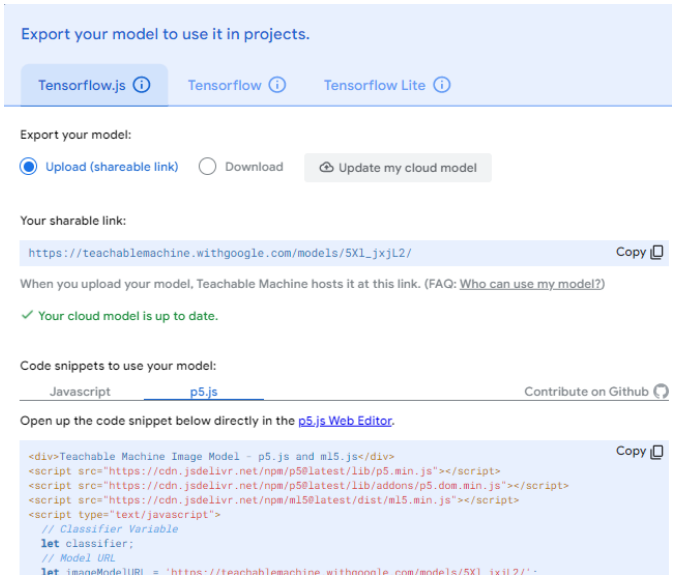

Figure 4 - Exporting the model

The use of the p5.js JavaScript library [7] was instrumental in capturing the webcam feed and interfacing with the model provided by Teachable Machine. The p5.js library provided the necessary functionality to create a canvas, draw the game elements, and capture the user's gestures through the webcam. The captured gestures were then fed into the model exported from Teachable Machine for real-time classification and interpretation.

The decision to utilize Teachable Machine was guided by several factors. First, the need for real-time interpretation of gestures necessitated a machine learning approach that could process and classify image data rapidly and accurately. Second, the desire to make the project accessible to other developers and non-experts encouraged the use of an easy-to-use, web-based interface. Finally, the simplicity of implementing and integrating Teachable Machine's exported models into the existing game infrastructure made it the ideal choice for this project.

5. Implementation

The transformation of the classic Pac-Man game into an AI-enhanced, gesture-controlled gaming experience unfolded through a series of carefully planned stages during the implementation phase of this project.

The process started with the generation of a machine learning model utilizing Google's Teachable Machine [5]. Various hand gestures representing directions (up, down, left, right, nothing) were captured in multiple positions and under different lighting conditions to train the model. Once trained, the model's performance was put to the test, testing its accuracy and real-time responsiveness to gestures.

Upon validating the model's effectiveness, it was primed for integration with the Pac-Man game. This step capitalized on the "shareable link" feature of Teachable Machine, a practical alternative to the conventional model download approach. This feature generates a unique URL for the trained model, enabling for easy integration into web-based applications.

The base for the Pac-Man game was adopted from an open-source project on the p5.js web editor [8]. This original code underwent a number of modifications to accommodate the machine learning model and the new gesture-based control system. A game control module was developed with the AI model. It processed the gesture data, translating it into game controls for the Pac-Man character.

Simultaneously, an input module was crafted using the p5.js JavaScript library [7]. This module captured a live video feed from the user's webcam, supplying real-time input to the Teachable Machine model for gesture classification.

The creation of a visually compelling and user-friendly interface was a top priority for this project. Featuring a retro arcade theme, the interface enhanced the immersion factor of the game. This theme extended to the game elements, achieved through the use of the p5.js library [7], as well as the menu and instruction pages created to guide users in operating the gesture controls.

Figure 5 - Menu Page

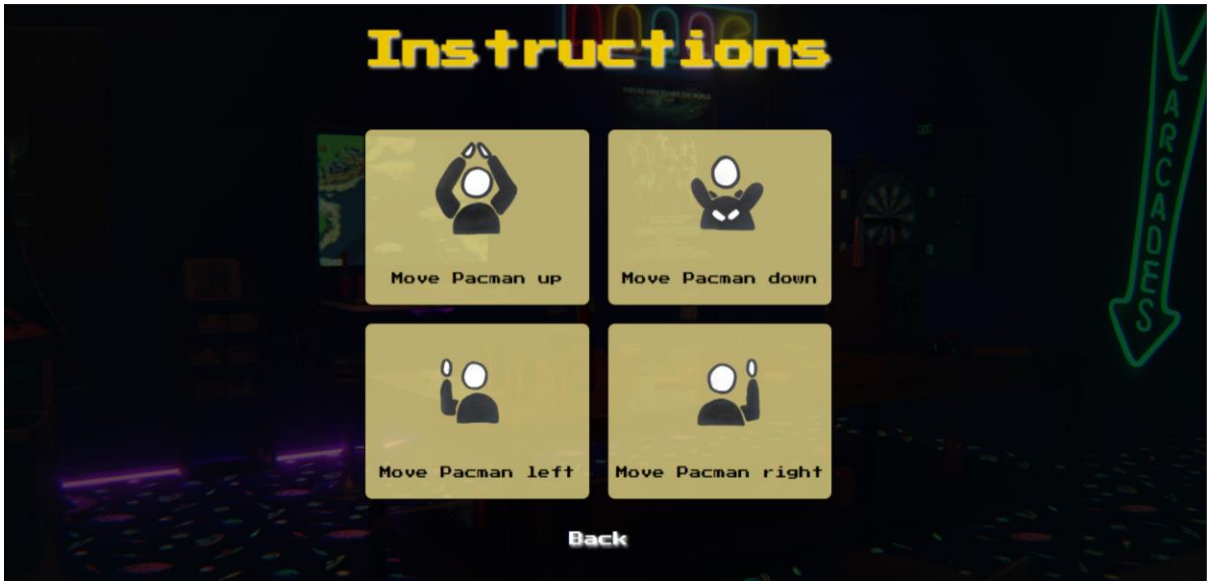

Figure 6 - Instructions Page

The menu page, acting as the entry point to the game, provided options to either start the game or navigate to the instructions page. The instructions page offers a visual tutorial using images which help users understand how to use hand gestures to control the game. Both of these pages were built using HTML and CSS to deliver an attractive interface.

On the game page, dynamic hand gesture images are displayed to the left, changing in sync with the player's movements. This visual feedback reinforced the connection between the player's hand gestures and the in-game controls. The game experience was improved with the addition of background music, which commenced with the start of the game, enhancing the arcade-like atmosphere. A restart button was also incorporated, enabling players to reset the game without having to reload the entire webpage. In-game scoring, dynamically updating to reflect player progress, added an element of challenge and competition. The implementation of victory and defeat screens added a polished feel, heightening the sense of achievement or challenge based on the game's outcome.

Figure 7 - Game Page

Continuous testing was carried out throughout the implementation process to ensure all components functioned cohesively. The end product is an AI-integrated, gesture-controlled Pac-Man game, taking the traditional gaming experience a step further by weaving together AI and gesture interactivity.

Figure 8 - Game Over Screen

Figure 9 - Dynamic hand gesture images

6. Evaluation & Results Obtained

The success of the implementation was evaluated on the basis of both the accuracy of the gesture recognition system and the overall gaming experience provided to the user.

After being trained and optimized, the Teachable Machine [5] model demonstrated a good level of accuracy in recognizing and interpreting the four distinct hand gestures: 'up', 'down', 'left', 'right' and ‘nothing’. The model was able to classify these gestures accurately in real-time, providing a smooth and helpful control experience for the user. Despite variations in lighting conditions and hand positions during the gameplay, the model remained relatively efficient in gesture recognition, validating the effectiveness of the training phase.

In terms of the gaming experience, the implementation of a menu screen, instructions page, and the incorporation of visual cues for gesture control significantly enhanced user engagement. The menu screen, serving as the starting point of the game, offered a user-friendly interface for beginning or exiting the game. The instruction page facilitated a smooth initiation for new players by providing a brief overview of the gesture controls. Furthermore, the dynamic display of hand gesture images on the game screen, altering in sync with the player's movements, added a level of immersion to the gameplay, enhancing the overall user experience.

The addition of a restart button allowed for uninterrupted gaming sessions by enabling users to quickly start a new game after a win or a defeat, thus adding to the game's replay value. The continuous updating of the score provided an exciting challenge for players, pushing them to improve their performance. The addition of a victory and a defeat screen gave a polished feel to the gaming experience, adding an extra layer of satisfaction, as per the game's outcome.

The project was organized in a structured folder system, making it easy to navigate and manage the different components. Testing was carried out using Visual Studio Code's built-in live server, which provided an effective and convenient way to preview the game and evaluate its functionality in a real-world environment.

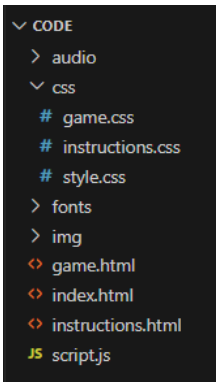

Figure 10 - Folder paths

Certainly, the evaluation of this project shows just how much AI can change the way we play games. By using the Teachable Machine model and creating an easy-to-use interface, the classic Pac-Man game became a more immersive and interactive experience. This project is a great example of how AI can be used creatively to take gaming to the next level.

7. Analysis of the results

The comprehensive analysis of the results acquired from the implementation of this project reveal crucial insights into its strengths, areas of weakness, and opportunities for improvement.

Among its primary strengths is the efficient use of Google's Teachable Machine [5]. The model exhibited impressive accuracy in real-time recognition and interpretation of the distinct hand gestures under various lighting conditions and hand positions. This not only validated the effectiveness of the training phase but also facilitated seamless game controls, significantly enhancing the gaming experience.

The user interface design also played a substantial role in the project's success. The implementation of a user-friendly menu screen, instructions page, and visual cues, coupled with the restart button and dynamic score updates, heightened the overall user engagement. The victory and defeat screens further added a polished feel to the gameplay, creating an immersive gaming environment.

Despite these strengths, it's important to acknowledge areas that could benefit from improvement. A key issue is the impact of individual variations in hand shapes, sizes, and movement speeds on gesture recognition. Through observation, it was found that the model works best with the individual who contributed to the training dataset, while other players experienced comparatively less success. This points to the need for a more diverse training dataset to better manage this variability, which could potentially enhance the model's overall accuracy.

Another influential factor observed was the environment in which the game was played. The model performed better in the environment where the training data was collected compared to different environments. This could be attributed to changes in background and lighting conditions which may affect the model's ability to accurately recognize gestures. This challenge also extends to the camera quality which can significantly impact the accuracy of gesture recognition.

Additionally, although the user interface is functional and intuitive, it does encounter a pause when the model is loading at the game's start. This could be improved with a loading indication or a "game starting" animation to enhance the user experience during this brief interlude.

The game's design could also benefit from some enhancements. The ghosts, for instance, could feature a better design and more sophisticated algorithms to trap and eliminate Pac-Man, thereby increasing the game's challenge and interest.

In summary, despite some areas requiring further refinement, this project stands as a successful example of the AI's transformative potential in gaming. It successfully combines the power of AI and user-centric design to create an engaging, interactive gaming experience, reshaping the traditional Pac-Man game.

8. Conclusion

To conclude, this project presents an exciting exploration of the integration of Artificial Intelligence and gaming, using the timeless Pac-Man game as a platform for implementing gesture controls. Through the use of Google's Teachable Machine and the p5.js library, an enhanced gaming experience was created, showcasing the immense potential of AI in redefining traditional gaming constructs.

This project provided a valuable learning experience on several fronts. Firstly, I gained insights into how to effectively incorporate AI into a game, navigating the complexities of real-time interaction and feedback.

Secondly, the task of designing a webpage to showcase the game honed my understanding of user interface design, web development and the significance of an eye-catching, intuitive user experience. The development of the menu page, instructions page, and the game page demonstrated how thoughtful design can enhance user engagement and enjoyment of the game.

Finally, the project helped me gain a deeper understanding of Google's Teachable Machine and its underlying principles. The process of capturing the hand gestures, training the model, and integrating it into the game control module provided a practical, hands-on experience with this powerful tool.

Ultimately, anyone intrigued by the intersection of AI and gaming should definitely explore this project. By visiting the website, they can engage with the reimagined Pac-Man game, featuring intuitive gesture controls powered by a machine learning model. This project showcases the transformative potential of AI in gaming, offering a glimpse into the future of interactive entertainment.

Video, Presentation & Deliverables Links

Important note:

When opening the folder to see the code and the **web application** make sure that you open the folder called *"code"*. The path should look like the following:

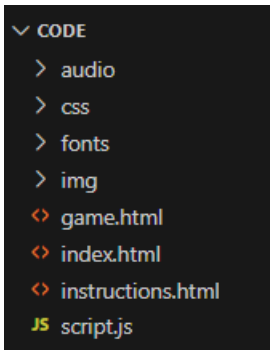

Figure 11 – Folder path for web application

Web Application Link:

https://drive.google.com/file/d/1Y5Vu_wY_8ivRQr21Cc1YqwAogkt2QghU/view?usp=sharing

Video Link:

https://drive.google.com/file/d/1e26WPxp03rxU_CKStsiDYMbAog7ebJWK/view?usp=sharing

Presentation Link:

https://docs.google.com/presentation/d/18zWYcY7pA01IYD7L-dBjzCKy0EEtgUs-/edit?usp=sharing&ouid=108751668891057001759&rtpof=true&sd=true

GitHub Repository Link:

https://github.com/markdingli18/IAPT

If you encounter any difficulties with any of the files do not hesitate to contact me:

mark.dingli.21@um.eud.mt

References

[1] K. W. Siovi, C. W. Kipruto, and A. Mindila, “Design thinking for gesture-based human computer interactions,” International Journal of Computer Sciences and Engineering, vol. 7, no. 3, pp. 919–925, 2019. doi:10.26438/ijcse/v7i3.919925

[2] D. M. Shafer, C. P. Carbonara, and L. Popova, “Spatial presence and perceived reality as predictors of motion-based video game enjoyment,” *Presence: Teleoperators and Virtual Environments*, vol. 20, no. 6, pp. 591–619, Dec. 2011. doi:10.1162/pres_a_00084

[3] B. Xie, B. Li, and A. Harland, “Movement and gesture recognition using Deep Learning and wearable-sensor technology,” Proceedings of the 2018 International Conference on Artificial Intelligence and Pattern Recognition, Aug. 2018. doi:10.1145/3268866.3268890

[4] S. Vickers, H. Istance, and M. J. Heron, “Accessible gaming for people with physical and cognitive disabilities,” CHI ’13 Extended Abstracts on Human Factors in Computing Systems, Apr. 2013. doi:10.1145/2468356.2468361

[5] “Teachable machine,” Google, https://teachablemachine.withgoogle.com/train/image (accessed May 18, 2023).

[6] “Tensorflow,” TensorFlow, https://www.tensorflow.org/ (accessed May 18, 2023).

[7] “p5.js,” get started | p5.js, https://p5js.org/get-started/ (accessed May 18, 2023).

[8] frutose, “Pacman game implementation,” P5.js web editor, https://editor.p5js.org/frutose/sketches/Vqru1IjAK (accessed May 18, 2023).