|  |
| --- |
| **TITLE: Project with voice-controlled object** |

|  |  |  |  |
| --- | --- | --- | --- |
| **LEARNING SCENARIO** | | | |
| ***School:*** | | ***Duration (minutes):*** | 90 |
| ***Teacher:*** |  | ***Students'***  ***age:*** | 13-14 |

|  |  |
| --- | --- |
| ***Essential Question:*** | How does voice-controlled object work? |

|  |
| --- |
| ***Topics:*** |
| * Interpreting outputs of voice-controlled object algorithm |
| ***Aims:*** |
| * Practical use of voice-controlled object |
| ***Outcomes:*** |
| * Exploring the possibilities of Speech to Text extension in Scratch |
| ***Work forms:***   * *work in pairs, group work*   ***Methods:*** |
| * *presentation, talk, discussion, interactive exercise* |

|  |
| --- |
| **ARTICULATION** |
| **Course of action (duration in minutes)** |
| **INTRODUCTION**  **Announcement of the goal of the lesson:**  Understanding voice-controlled object algorithm and its usage through a practical project. |
| *Ask your students if an object can be voice controlled.*  *Is it possible to benefit from voice-controlled objects in some way?* |
| **MAIN PART**  Through a project, the teacher instructs students on Scratch commands and skills as well as on how to train a model to convert speech to text.  Step 1: Open your Chrome web browser and go to: <https://machinelearningforkids.co.uk/scratch3/>  Step 2: Load extension Speech to text (STT - Google Chrome browsers only)    Step 3: Delete Cat sprite by clicking on sprite trashcan icon  Step 4: Download maze.png from: <https://drive.google.com/file/d/11YBBhQcIhVfHYMWeLkhgAYKSwfv33pT5/view?usp=sharing> and upload it to Scratch as custom sprite  Graphical user interface, application, Word  Description automatically generated  Step 5: From sprite gallery choose Retro Robot and use the second costume (Retro Robot b)  Graphical user interface, application, Word  Description automatically generated  Step 6: From sprite gallery choose Home Button  Graphical user interface, application  Description automatically generated  Step 7: Make 3 variables (for all sprites): - **gameover** (shows how the game ended) - **xm** (x position of robot) - **ym** (y position of robot)  Step 8: Make a list and change its name to:  **endtalk** (displays the game over message, depending on how the game ended)    Step 9: Switch to **maze sprite** and start coding. First few blocks are setting the position, size, and visibility of the maze. Forever loop that follows is a collision checker. In case of collision, the **gameover** variable is set to 1.  Graphical user interface, application, table, Word, timeline  Description automatically generated  Step 10: Switch to **Home Button** sprite. The code is almost the same except that in case of collision a **pop** sound is played and the **game over** variable is set to 2.  Graphical user interface, application, Word  Description automatically generated  Step 11: The main code is assigned to Retro robot code and there are 5 threads which start simultaneously.  11.1 Flag (start) code sets the variables, list entries (and deleting all previous entries), position and visibility. At the end there is a listening block (from STT extension) in the loop.  Graphical user interface, timeline  Description automatically generated with medium confidence  11.2 The next 4 threads are triggered by speech recognition and each recognized voice command is handled by a sequence of blocks.  If no collision is detected (gameover is 0) robot moves in specific direction (by changing xm or ym value and moving robot to calculated position).  If collision is detected (gameover is 1 or 2) game over message (from the list) will be displayed and the program will stop.    Step 12: Try to use your native language voice commands and see how it works.  Discuss how (in)correct spelling affects the speech recognition.  Turn on (check) speech recognition results.  A screenshot of a computer  Description automatically generated |
| Object tracking robots, if they can be controlled smartly through voice, can be of tremendous help for the physically handicapped people. A speech recognition system is used to recognize a set of predefined commands such as forward, backward, left, right and rotation at a particular turning angle. The robot navigates its way as per the voice-command signal, while also tracking the desired object. The voice-command signal processing is carried out in real-time, using an on-time cloud server that converts it to text form. The command signal text is then transferred to the robot via Bluetooth network to control its differential drive. The prototype smart robot consists of three sub-systems: speech recognition system, object tracking system and the differential-drive based movement's control system. The accuracy and efficiency of the speech recognition system is examined through a set of experiments. The effect of factors such as noise and distance etc. are examined, with encouraging results. The prototype robot can recognize the voice-commands within a Bluetooth range, i.e, 10 m. Possible extensions are also discussed that could lead to a wide range of further applications. |
| **CONCLUSION** |
| We can control objects in a computer program with our voice. |

|  |  |
| --- | --- |
| ***Methods*** | ***Work forms*** |
| ***presentation***  ***interactive exercise / simulation on the computer*** | ***work in pairs***  ***group work*** |

|  |
| --- |
| ***Material:*** |
| * <https://machinelearningforkids.co.uk/scratch3/> * <https://drive.google.com/file/d/11YBBhQcIhVfHYMWeLkhgAYKSwfv33pT5/view?usp=sharing> |

|  |
| --- |
| ***Literature*** |

|  |
| --- |
| **PERSONAL OBSERVATIONS, COMMENTS AND NOTES** |
|  |