TAHA AKHLAQ

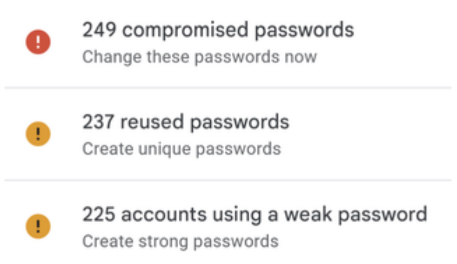
BROOKLYN, NY

takhlaq04@gmail.com
linkedin.com/in/taha-akhlaq
github.com/TahaAkhlaq
(347) 556-9525

PASSWORD GENERATOR - JAVA

What?

- Programmed a reliable password generator using Java that provides a user with a strong and unique password that is not vulnerable to cybercrime such as brute force or dictionary attacks.
- Protects users' personal information from unauthorized individuals to avoid identity fraud, fraudulent transactions, and other illicit activities.



249 compromised passwords
Change these passwords now

237 reused passwords
Create unique passwords

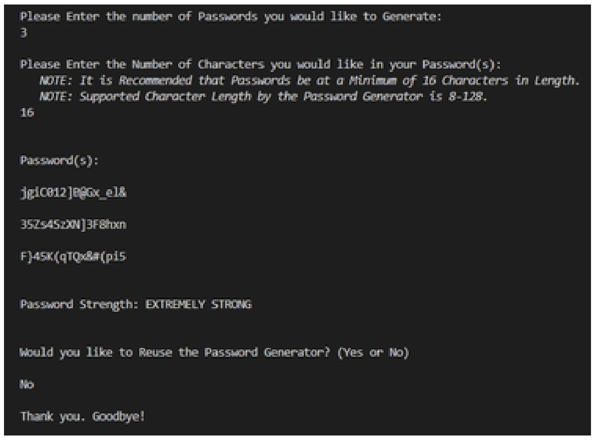
225 accounts using a weak password
Create strong passwords

Potential Additions:

- A password manager that works with the password generator to accessibly store all of a user's passwords.
- A frontend interface coded in HTML, CSS, and JavaScript in addition to the Java backend to improve the user interface.
- Introduce cryptography for additional security.

How?

- A user inputs the number of passwords they would like to generate for their account(s). The user then specifies the number of characters for their password(s). There is a minimum of 8 characters and a maximum of 128 characters as supported by most websites.
- The password generator recommends a minimum length of 16 characters as it is an appropriate character length, but the user still has the ability to choose.
- The password generator also conveniently comes with a password strength indicator based on the length of the password.
- The user then has the option to reuse the password generator if they wish to.



```
Please Enter the number of Passwords you would like to Generate:
3

Please Enter the Number of Characters you would like in your Password(s):
   NOTE: It is Recommended that Passwords be at a Minimum of 16 Characters in Length.
   NOTE: Supported Character Length by the Password Generator is 8-128.
16


Password(s):

jgiC012]B@Gx_el&

35Zs4SzXN]3F8hxn

F}45K(qTQx&#(pi5


Password Strength: EXTREMELY STRONG


Would you like to Reuse the Password Generator? (Yes or No)

No

Thank you. Goodbye!
```

Pros:

- Quickly supplies a password to a user upon request.
- Provides a strong password, ensuring the user's safety.

Cons:

- Tedious to enter in the password when needing to log into a particular account.
- A password manager is needed to be used alongside the password generator to facilitate saving all of a user's passwords in one place.

Learned:

- Extensive Control Flow
- To keep user experience in mind to visually improve the program by incorporating features such as newline characters and code reuse
- Leveraged Pseudorandomness

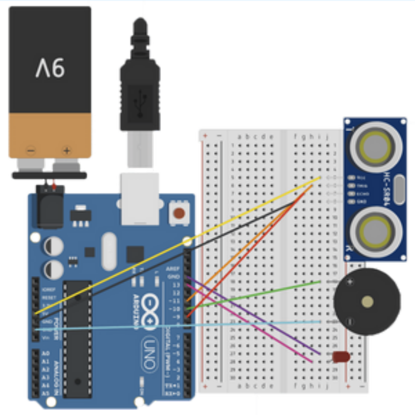
SMART BLIND STICK - ARDUINO

What?

- Programmed and designed a sophisticated smart blind stick in C/C++ using Arduino.
- Enables a visually impaired user to safely and independently navigate to his or her destination.
- Conveniently attaches to an existing cane.

How?

- An ultrasonic sensor detects obstacles in the user's path.
- If an obstacle is detected, the active buzzer activates to audibly alert the user, and an LED glows to visually alert others nearby so that they may assist if needed.
- The circuit, including the Arduino and a battery, sits on a breadboard.



Code Is Available On My GitHub

TAHA AKHLAQ

BROOKLYN, NY

takhlaq04@gmail.com
linkedin.com/in/taha-akhlaq
github.com/TahaAkhlaq
(347) 556-9525

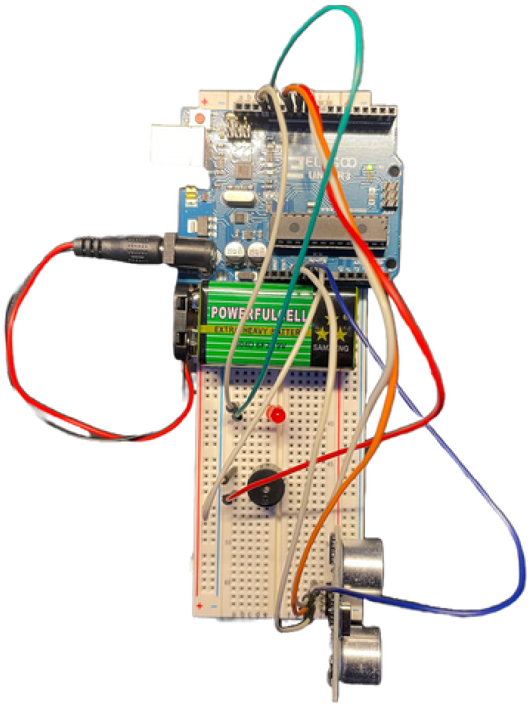
SMART BLIND STICK - ARDUINO

Potential Additions:

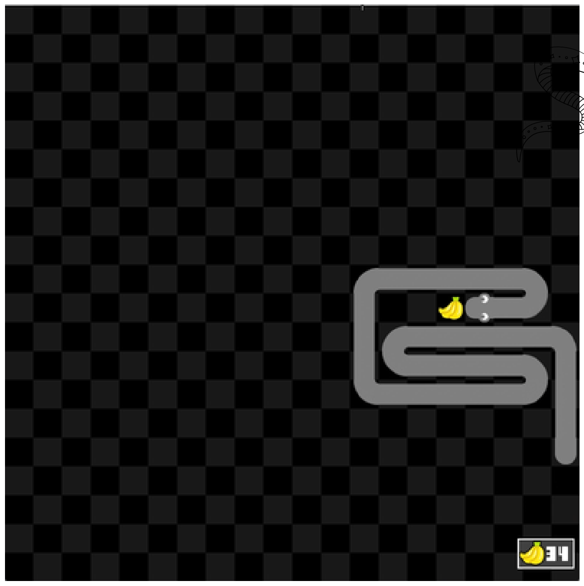
- Additional buzzers to increase the volume. Also, a vibrator to grant the user numerous forms of feedback.
- More ultrasonic sensors pointed in different directions to provide the user with a better sense of his or her surroundings. Also, the addition of a water sensor to detect the presence of water in the user's path.
- An RF remote that triggers the smart stick to beep to allow the user to relocate it with ease.
- A greater power supply to effectively power the innovative device.
- 3D printed compartment composed of PLA to house the device and securely attach it to the cane as well as not expose the device to the elements.

Learned:

- Incorporate an ultrasonic sensor to create a functioning circuit on a breadboard.
- Power the Arduino using an alternative power source as opposed to a USB cable.



SNAKE GAME - PYTHON



How?

- Notoriously, the objective of the snake game is to navigate the head of the snake to the fruit (a banana in this case) and to avoid hitting the snake's body as well as the walls.
- Pygame is a versatile, portable open-source Python library primarily designed to create 2D video games as well as applications. It allows programmers to efficiently implement multimedia and a graphical user interface (GUI).
- The snake game utilizes multi-threaded CPU rendering to ensure a smooth gaming experience.

What?

- Created my own iteration of one of the most popular arcade games, Snake, in Python using Pygame.
- Incorporated advanced game logic, mechanics, graphics, and sound to optimize player experience.

Code Is Available On My GitHub

TAHA AKHLAQ

BROOKLYN, NY

takhlaq04@gmail.com
linkedin.com/in/taha-akhlaq
github.com/TahaAkhlaq
(347) 556-9525

SNAKE GAME - PYTHON

Learned:

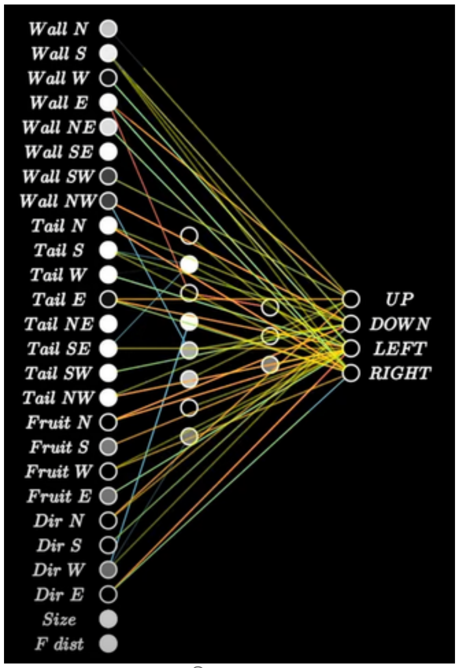
- Set a maximum frame rate to ensure the game runs optimally across various systems.
- Construct the game's layout effectively by utilizing the display surface and other surfaces.
- Colorize a surface using an RGB tuple.
- Use rectangles to accurately position and center surfaces on the display surface.
- Change the x and y coordinates effectively by storing 2D data as vectors.
- Organize the game by utilizing methods and combining classes into a single main class.
- Eliminate repetition in code by creating variables that behave as constants.
- Use lists and elements to simulate the movement of the snake.
- Implement player input, sound, and reset games automatically.
- Improved debugging in game development.

Potential Additions:

- Main menu to control in-game settings such as the music's volume and the snake's speed.
- Generate powerups that spawn randomly to grant the snake certain abilities, such as the ability to travel faster or immunity from going over itself for a short duration of time.
- Display and save a high score so that the player can attempt to achieve a better score each time they play to introduce competitiveness.
- Allow the player to alter the aesthetics of the game, including the fruit and the color of the snake, as well as the background.
- Offer a variety of map sizes that the player can select to play from.
- In-game rewards upon reaching a certain score, such as badges and titles, to keep the player engaged.

NEAT:

- A **neural network** is composed of layers that make up its topology. The input layer, which is the first layer, receives information through neurons. This information consists of data such as the distance between the snake's head and its closest tail segment, the direction of the fruit relative to the snake's head, and the overall size of the snake. It feeds the data to neurons in the subsequent layer (the output layer) through connections with weights and biases. The weights and biases allow the artificial intelligence to identify and assess the strength of each particular connection. Changing these values dictates the performance of the artificial intelligence and ultimately determines which of the four directions is the most optimal for the snake to move in.
- **NEAT** (NeuroEvolution of Augmenting Topologies) is a complex genetic algorithm that enables a neural network to artificially evolve and improve in a particular game by replicating the process of natural selection. It controls the weights and biases of the neural network by setting parameters that favor the snake's survival, thus enhancing the artificial intelligence.
- **Initialization:** Implementing the NEAT algorithm into my snake game generates a population of snakes, each with its own neural network with random weights and biases, known as genomes.
- **Evaluation:** The NEAT algorithm evaluates and tests each genome based on the score they achieve, which determines each snake's fitness. The algorithm uses fitness as a constructive system to select the best-performing snakes.
- **Crossover and Mutation:** The best-performing genomes combine to create a new, improved population consisting of a generation of their offspring, with some having mutations. Depending on NEAT's parameters, these may create hidden layers between the input and output layers to improve the functionality of the neural network.
- The algorithm repeats this testing process of evaluation, crossover, and mutation to find the best-performing genome by simulating genetic evolution (neuroevolution).



Source

Code Is Available On My GitHub