REALISTIC BICYCLE SIMULATION

VELOVENTURE

FINAL DEMONSTRATION REPORT
SPRING 2024

GEBZE
TECHNICAL UNIVERSITY

CONTENTS

GEBZE
TECHNICAL UNIVERSITY

TEAM

Kubilay Yazman

Ömer Sarıçam

Reşit Aydın

Sare Nur Aydın

Buket Gençer

Ahmet H. Sevinç

Enes Patır

ABOUT

OUR STORY

We developed a sensor-equipped bicycle system for interactive gaming, seamlessly integrating hardware and software. Users enjoy immersive game modes and maps, enhancing their cycling skills. Our team focused on sensor integration and facial recognition for easy access, creating a dynamic and user-friendly experience. We proudly combine technology with physical activity for a unique cycling adventure.

OUR VISION

Our vision is to innovate and inspire through cutting-edge interactive cycling technologies. By pushing boundaries, we enhance user experiences and advance immersive systems. Committed to collaboration and continuous improvement, we bring creative ideas to life, making a positive impact in fitness, entertainment, and beyond.

TECHNOLOGY

Our project leverages the power of C++ and Unreal Engine to create an immersive cycling experience. Users engage with interactive maps and racing modes through precise sensor integration, enhancing realism and responsiveness. Seamlessly blending hardware and software, we elevate the user experience. Our team is dedicated to exploring technology's potential and delivering impactful innovations.

GEBZE
TECHNICAL UNIVERSITY

INTRODUCTION

The user sits on a bicycle system equipped with sensors and logs into the game. Registered users can log in by simply scanning their faces with the facial recognition system, without needing a password. For unregistered users, a facial scan is conducted, a photo is taken, and a username is obtained to provide access to the system.

In the game, the user encounters two different maps and two different game modes:

1. **Free Roaming Mode:** In this mode, the user can freely roam the selected map. This allows the user to develop their cycling skills and abilities. The first map offers an area where the user can easily navigate and freely experience the cycling journey.
2. **Race Mode:** In this mode, the user tries to complete the lap in the shortest time possible and competes with their own best time. The second map is designed to better convey the race atmosphere.

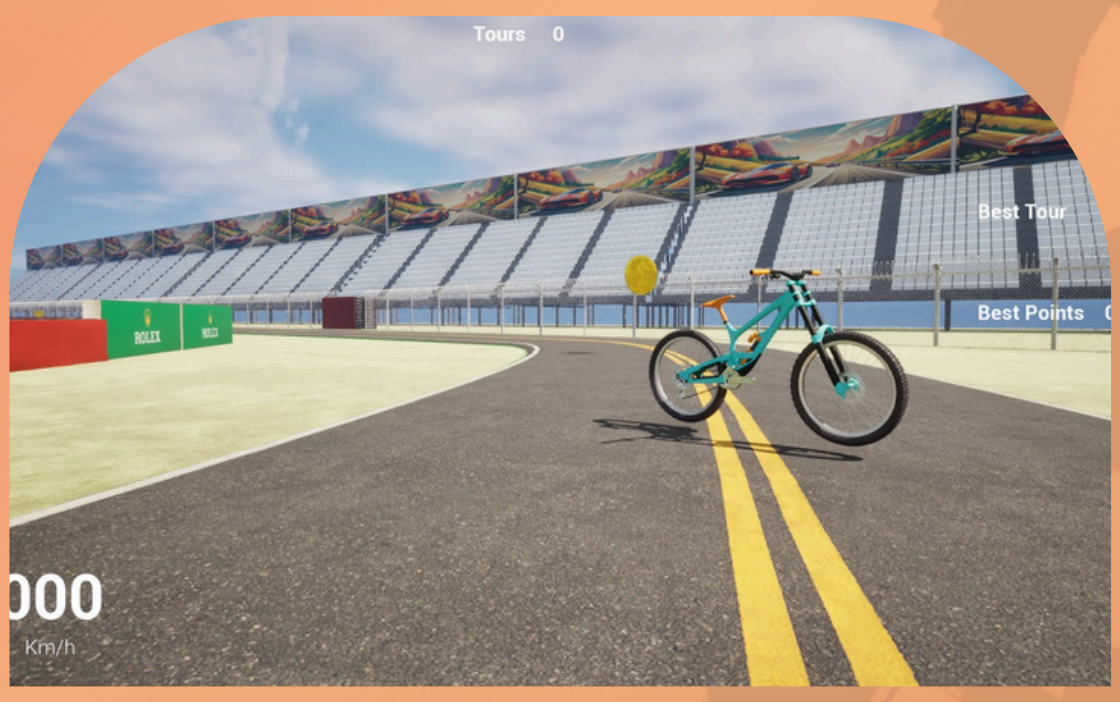
After selecting the game mode and map, as the user pedals on the bicycle system, the half effect magnetic sensor reads the data and moves the bicycle. The rotary encoder sensor placed on the handlebars allows the user to steer the bicycle.

For a more realistic experience, the pressure system placed at the back of the bicycle makes it harder to pedal on slopes and ramps in the game, providing a genuine riding experience.

GEBZE
TECHNICAL UNIVERSITY

OUR SIMULATION

FREE ROAMING MODE



RACE MODE

GEBZE TECHNICAL UNIVERSITY

MODULES

SENSOR PROCESSING MODULE

DESKTOP SIMULATION MODULE

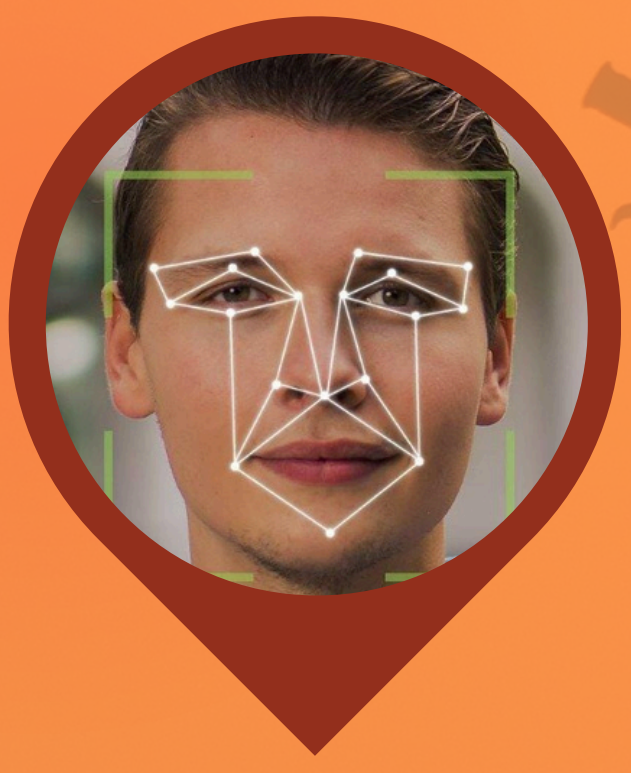
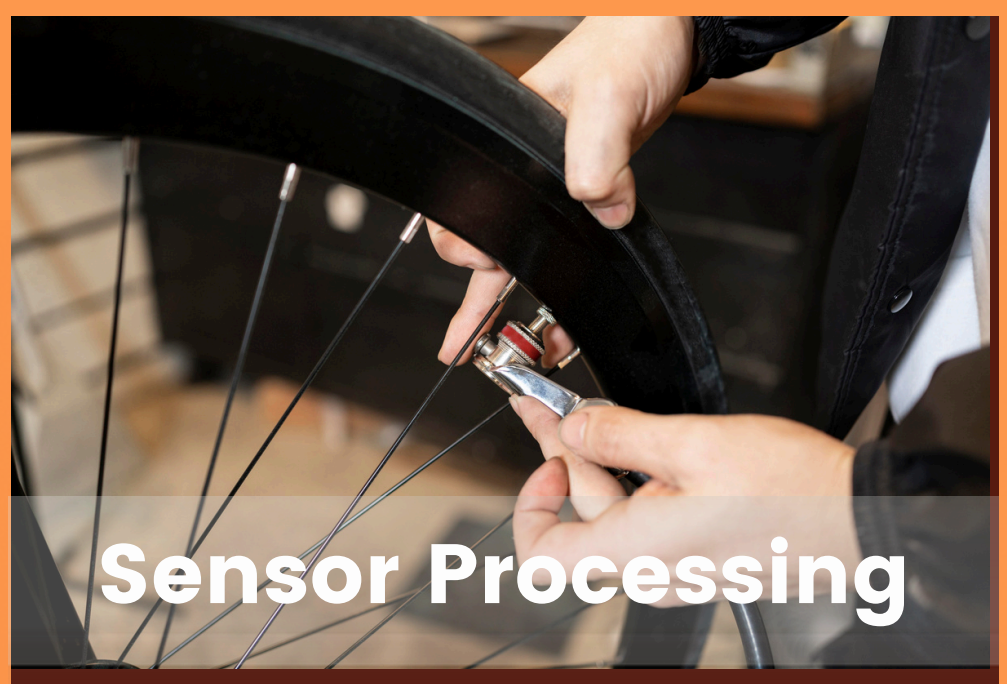


IMAGE PROCESSING MODULE

MOTION CONTROL MODULE

GEBZE TECHNICAL UNIVERSITY

MODULE'S TEAM



Sensor Processing

Kubilay Yazman
Ömer Sarıçam
Sare Nur Aydın
Buket Gençer
Reşit Aydın

Desktop Simulation

Kubilay Yazman
Enes Patır
Ahmet Hakan Sevinç
Sare Nur Aydın
Buket Gençer
Reşit Aydın

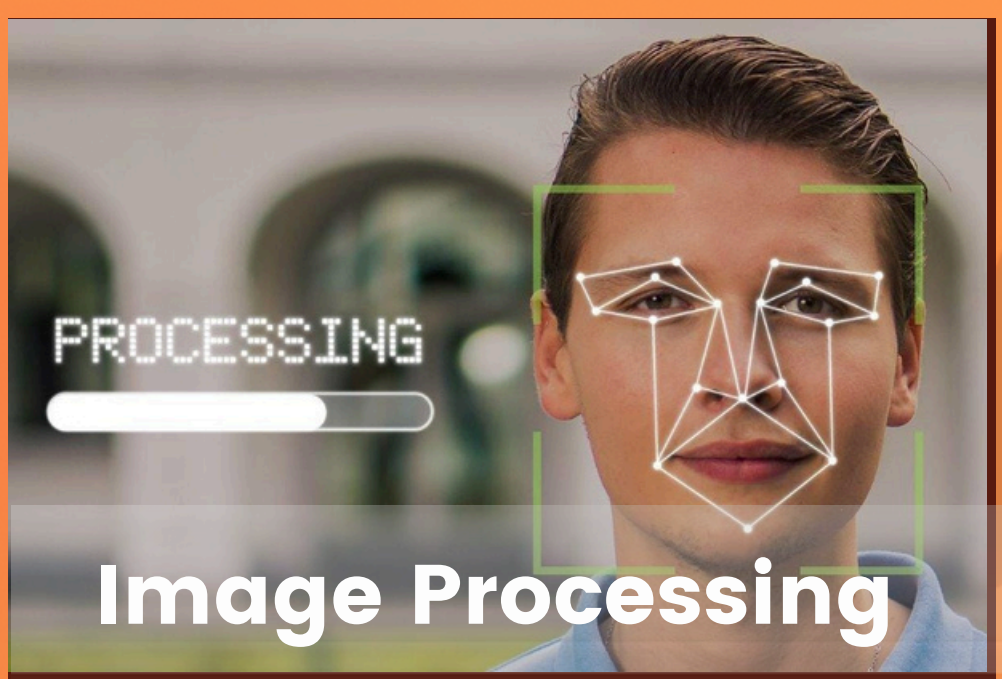


Image Processing

Ahmet Hakan Sevinç
Ömer Sarıçam
Sare Nur Aydın
Enes Patır
Reşit Aydın

Motion Control

Kubilay Yazman
Ömer Sarıçam
Ahmet Hakan Sevinç
Enes Patır
Buket Gençer
Reşit Aydın

GEBZE TECHNICAL UNIVERSITY

DESKTOP SIMULATION

MODULE OVERVIEW

The Bike Simulator module is responsible for providing a realistic biking experience. within the larger project. This module encompasses bike animation and physics, customization options, map creation, and processing incoming data to ensure smooth gameplay.

MODULE OBJECTIVES

- Implement realistic bike animation and physics for a lifelike biking experience.
- Develop tools for creating diverse and challenging maps.
- Process incoming data to handle player inputs, environmental changes, and game events effectively.

REQUIREMENTS

Bike Animation and Physics

- Develop realistic animation for bike movement, including pedalling, turning, and leaning.
- Implement physics simulation for bike dynamics, considering factors such as friction and inertia.
- Ensure smooth transitions between different biking actions to enhance realism.

GEBZE
TECHNICAL UNIVERSITY

REQUIREMENTS

Map Creation

- Develop tools for creating diverse terrains and environments
- Allow designers to place obstacles, ramps, and other interactive elements within the map.
- Provide options for adjusting terrain features and lighting settings.

Processing Incoming Data

- Handle player inputs, including steering, acceleration, braking, and interaction with the environment.
- Process environmental changes such as terrain variations, and obstacle collisions.
- Manage game events such as race objectives, time limits, and player progress tracking.

ACCOMPLISHMENTS

It is selected the most suitable 3D model for the bicycle simulation. Some modifications were made to the downloaded model to make it compatible with Unreal Engine 5. Blender application was used to make these modifications.

Under this module, our team members were divided into two main groups responsible for visual design and bicycle physics. Those in the visual design group were responsible for the game map and UI design.

The other group worked on the animation and physics of the bicycle, adjusting these features to provide a realistic and enjoyable experience.

GEBZE
TECHNICAL UNIVERSITY

HEADER FILES

Input:

- **Steering angle:** The angle at which the player is steering the bike.
- **Acceleration:** Information regarding the acceleration inputs from the player.

Output:

- **Terrain feedback:** Feedback data related to the terrain conditions, such as the presence of ramps and the inclination indicating whether the player is going uphill or downhill.

SIMULATION

GEBZE
TECHNICAL UNIVERSITY

IMAGE PROCESSING

MODULE OVERVIEW

The Image Processing module is designed to recognize images using the Dlib and OpenCV libraries. Its primary function is to analyze input images and return the corresponding names of the recognized objects within those images. This module is an essential component of our broader project, facilitating face recognition to serve personalized experience.

REQUIREMENTS

Data Input

Raw Image Data: Input images in various formats.

Pre-trained Models: Models for object detection and classification.

Data Output

Recognized Object Labels: Labels corresponding to objects detected within input images.

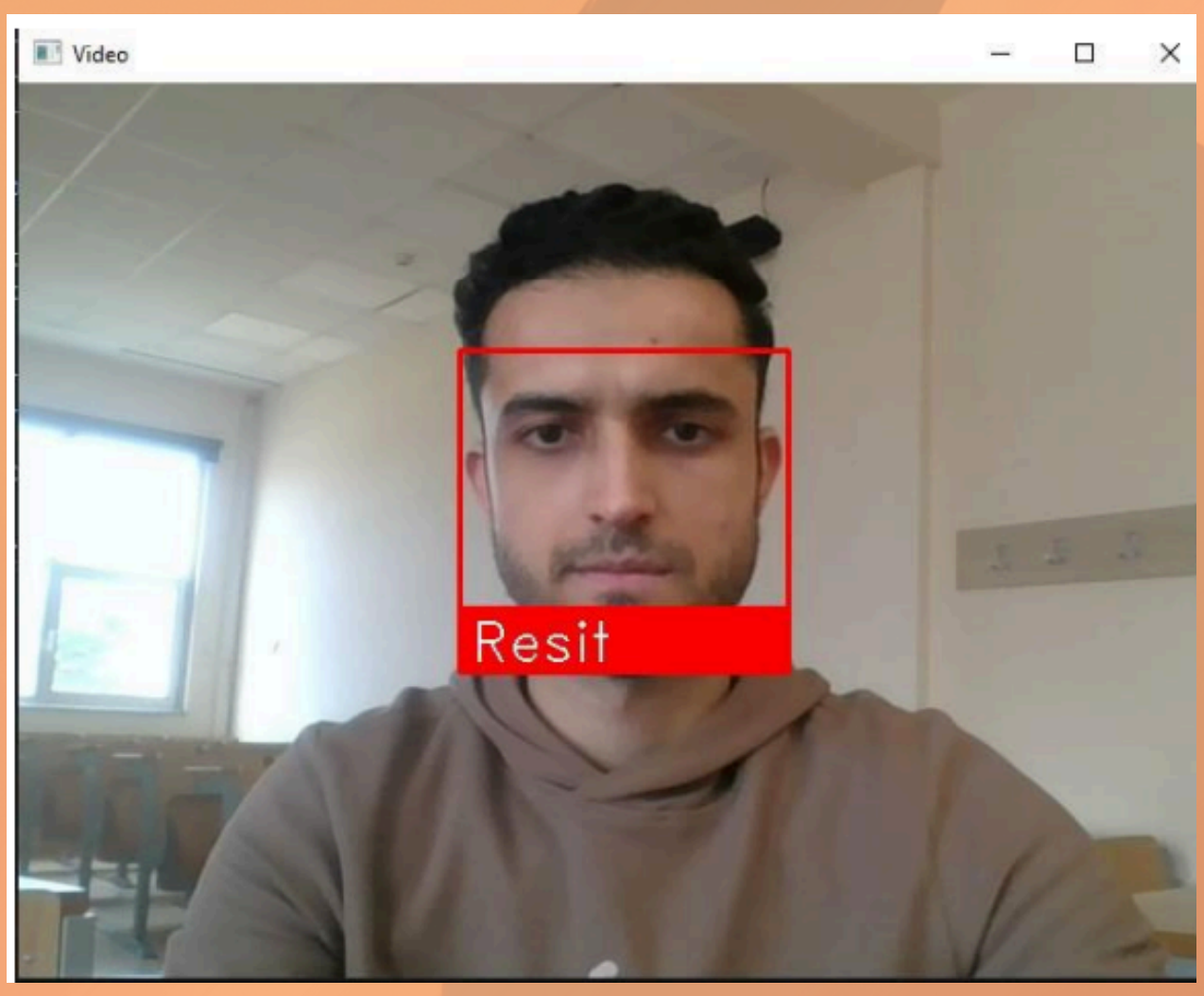
Confidence Scores: Confidence scores indicating the certainty of object recognition.

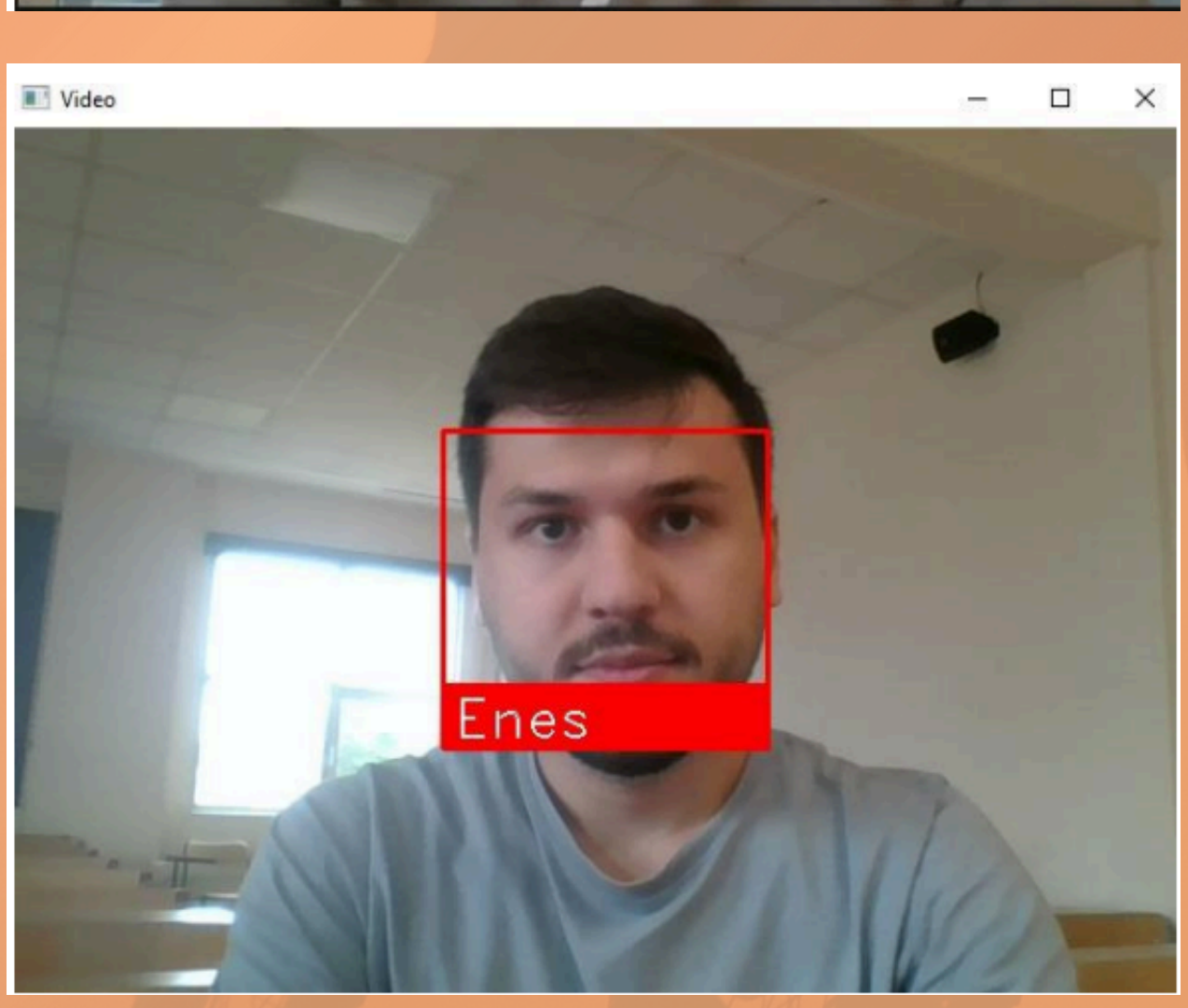
GEBZE TECHNICAL UNIVERSITY

ACCOMPLISHMENTS

The core functionality of the Image Processing module revolves around image recognition. Using the Dlib and OpenCV libraries, we implemented algorithms capable of identifying objects within images with a high degree of accuracy. These algorithms leverage deep learning techniques and pre-trained models to detect objects across various categories.

IMAGE PROCESSING





MOTION CONTROL

MODULE OVERVIEW

The main goal of our bicycle simulation project's Motion Control Module is to provide an immersive experience that accurately mirrors real-world cycling conditions within an Unreal Engine game environment. This is accomplished by using servo motors and signal readers incorporated into the bicycle setup.

The servo motor adjusts the resistance felt by the user, simulating scenarios like uphill climbs. Signal readers interpret game data, like terrain steepness, to adapt the resistance accordingly.

MODULE OBJECTIVES

Simulate physical resistance encountered during outdoor cycling, such as inclines or tough terrain, through the Resistance for Back Wheel component.

Accurately track and interpret the cyclist's directional steering and pedaling speed with the Direction and Speed Sensing component.

GEBZE
TECHNICAL UNIVERSITY

REQUIREMENTS & ACCOMPLISHMENTS

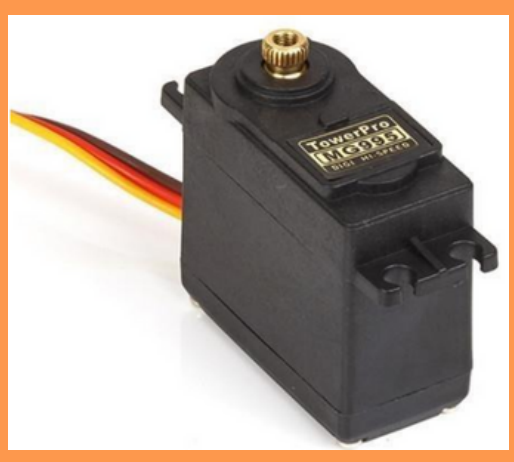
Resistance for Back Wheel

In this section, we detail the implementation of a simulation enhancement aimed at replicating the real-world difficulty experienced when pedaling uphill. By applying this enhancement, we strive for a more realistic simulation environment.

To achieve a more realistic simulation of uphill pedaling resistance, we devised a method involving the use of a servo motor connected to the bicycle's rear wheel brake cable. This servo motor, controlled by an ESP32 microcontroller, applies tension to the brake cable, thereby increasing the difficulty of rotating the rear wheel.

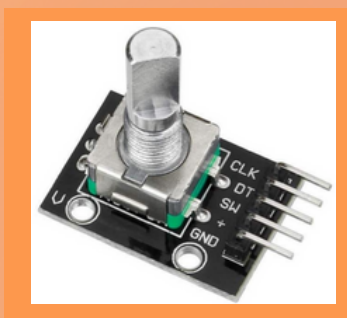
Initially, we experimented with manually pulling the rear wheel brake cable to assess its feasibility. However, we found that using both brake pads made this operation challenging and tended to halt the bicycle rather than providing resistance. To address this, we removed one brake pad, allowing us to exert force on the rear wheel without excessively impeding its rotation. Subsequently, we tested servo motors of varying torque ratings and found that a servo with a 25kg torque capacity provided adequate tension.

After selecting the appropriate servo motor, we identified an optimal location on the bicycle frame for mounting it, considering both available space and the force it would apply. Following mounting, we conducted tests to determine the servo's ability to generate resistance corresponding to different inclines of the simulated hill.



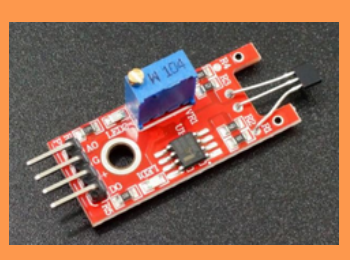
REQUIREMENTS & ACCOMPLISHMENTS

Our experiments demonstrated that by controlling the servo motor at specific angles, we could effectively simulate varying levels of resistance corresponding to the gradient of the virtual hill. This method successfully replicated the sensation of pedaling uphill in the simulation environment. In conclusion, the integration of servo-controlled resistance for the back wheel represents a significant enhancement to our simulation platform. This innovation contributes to a more realistic portrayal of cycling conditions, enabling users to train and analyze performance more effectively.



Direction Sensing

An attempt was made to use an IMU sensor initially to measure the bicycle's direction. However, due to the inability to obtain clean and stable data, the IMU was abandoned. A rotary encoder was then used to measure the bicycle's direction. The rotary encoder was placed at the head of the bicycle's steering tube. A counter was kept. When the bicycle turns right, the counter value increases; when it turns left, the counter value decreases. When the bicycle's handlebar is straight, the counter becomes 0. This way, directional data is obtained.



Speed Sensing

A linear Hall effect sensor was used to measure the bicycle's speed. Three magnets were placed on the bicycle's wheel at equal intervals. The circumference of the bicycle wheel was calculated to be 175 cm. Therefore, it is understood that the bicycle advances 58.3 cm with each magnet detection. The speed is calculated based on how many magnets are detected per second.

Header Files

Input Data: Sensor data capturing the cyclist’s speed and steering direction.

Output Data: Signals to the gaming software representing the real-world cycling dynamics, including resistance adjustments and environmental interactions.

SENSOR PROCESSING

MODULE OVERVIEW

The Sensor Processing module serves as a critical bridge between the physical bicycle sensors (speed, direction) and the Unreal Engine-based game environment. This module's primary function is to translate realworld sensor data into meaningful inputs for game mechanics, such as bike speed, direction, and resistance level in the game environment.

REQUIREMENTS & ACCOMPLISHMENTS

Data acquisition and Data Processing

The data obtained from the linear Hall effect and rotary encoder sensors on the bicycle were sent to the computer via serial communication over COM ports using ESP32. These data were parsed to obtain meaningful speed and direction information. Powershell scripts were written to display the data.

Through the script, degree data was sent to the ESP32 via serial communication to ensure the motor moves at the desired angle.

GEBZE TECHNICAL UNIVERSITY

REQUIREMENTS & ACCOMPLISHMENTS

Data Integration with Unreal Engine

The connection between the sensor data and Unreal Engine has not been established yet. However, a study was conducted where data from the IMU sensor was printed in the Unreal environment to learn how the connection could be established

Header Files

Input Data

Raw Sensor Data: Includes velocity (speed) and angular orientation (direction) data captured from the bicycle's sensors.

Game Environment Signals: Instructions received from the Unreal Engine game that dictate the resistance level to be simulated by the hardware, corresponding to in-game activities such as uphill or downhill riding.

Output Data

Processed Speed and Direction Data: Calibrated and synchronized data ready for use in game mechanics, ensuring that the player's physical actions are accurately reflected in the game.

Resistance Adjustment Commands: Signals sent to the bicycle's resistance mechanism to alter the physical effort required to pedal, based on virtual terrain and environmental conditions within the game.

GEBZE
TECHNICAL UNIVERSITY

OUR GALERY

GEBZE
TECHNICAL UNIVERSITY

OUR GALERY

https://veloventure1.wixsite.com/home

https://www.youtube.com/watch?v=K-N3bh40SfM&t=6s

https://www.youtube.com/watch?v=egQUI4KQWJs&t=2s