**SCENARIO TITLE**

Smart Lighting for the Environment and Safety

**SUBJECT**Environmental protection

**1. SCENARIO IDENTIFICATION**

**1.1 Scenario Title**

*Smart Lighting for the Environment and Safety*

**1.2 Target Age Group**

Grades 7–9 (Junior High School)

**Curriculum**: Junior High School

**Thematic Area**: Computational systems, digital devices, networks

**Topics/Subtopics**: Automation and robotic systems – connecting computers to the physical world

**Expected Learning Outcomes:**

Students will be able to:

• Select and connect sensor-equipped devices and/or robotic systems to computers in order to control them or collect data.

• Program an application that controls a pre-assembled robotic or automation system using simple sensors and actuators, within the framework of a learning project.

• Design and program educational robotic and automation systems using physical computing to conduct experiments or creative learning projects involving design and construction.

• Explain how basic sensors work, attempt to construct them, calibrate them, and control them using code on a computer.

**1.3 Estimated Duration**

45 minutes

**1.4 Subject Areas Involved**

Computer Science, Technology

**1.5 Prerequisite Knowledge**

Students should be familiar with a visual programming environment. They should be able to create and debug programs. They must understand the concepts of loops and conditional statements and have applied them in a visual programming setting.

**1.6 Purpose of the Scenario**

By the end of the lesson, students should have:

1. become familiar with using Fossbot sensors
2. program sensors in combination with other Fossbot component
3. developed problem-solving skills, adaptability, and the ability to use new programming and graphical environments hrough teamwork, collaboration, and creativity

**1.6.1 Learning Objectives/Outcomes**

Students will be able to:

• Operate the FOSSBot interface environment (connection, project input, and editing)  
• Identify the basic functions and use of sensors (photoresistor and ultrasonic)  
• Program automation systems using logical structures (if, else, and)  
• Record and interpret sensor values in real environmental conditions  
• Create and improve a “smart lighting” system based on environmental data  
• Collaborate in groups, making decisions about robot function design and optimization  
• Recognize the role of technology in energy saving and safety enhancement

**1.6.2 Teaching Objectives**

Through teaching, the educator aims to:

• Develop problem-solving skills through programming activities with real-world applications  
• Link theory with practice by encouraging the exploration of sensor operation  
• Foster collaborative skills through group activities and role distribution  
• Promote creative and critical thinking, allowing space for experimentation and code modification  
• Strengthen environmental awareness by understanding technology’s role in energy saving  
• Apply a constructivist approach, where students build knowledge through experience, experimentation, and discovery

**2. LESSON PLAN DEVELOPMENT**

**2.1 Description of Teaching and Learning Activities**

This scenario employs experiential and inquiry-based learning using the FOSSBot and its visual programming environment.

The learning process focuses on exploration, encouraging students to experiment with commands and conditions by creating an interactive project. No ready-made solutions are provided, promoting knowledge discovery through observation and trial.

Collaborative learning is enhanced through teamwork, as students work in groups of 3–5 to develop their projects, strengthening social interaction and collective knowledge construction.

The scenario integrates gamification elements, adding motivation and engagement to the learning experience. Students create interactive projects using sensors, LEDs, and the computer screen, evoking gameplay experiences.

A worksheet is used for instruction, guiding students through five (5) consecutive activities:

**Activity 1:** Introduction to the FOSSBot interface and programming environment. Using an image with interactive hotspots, students identify the main functions of the environment.  
**Activity 2:** Understanding basic commands and timing parameters. Students examine the operation of a pre-built project ("black box"), explain the FOSSBot's behavior, and modify the code to change the LED activation/deactivation frequency.  
**Activity 3:** Connecting a sensor to an LED. Students collect data through the FOSSBot's photoresistor, evaluate it, and adjust the code to display messages on the screen and activate the LED.  
**Activity 4:** Adding a second sensor. Students combine photoresistor and ultrasonic sensor data, applying complex logic conditions to activate the LED—thus achieving the main scenario goal of creating a smart light that detects motion.  
**Activity 5:** Understanding check and enhancement of metacognitive awareness via a structured questionnaire (multiple choice and true/false questions) covering concepts from previous activities.

To implement the activities, the FOSSBot must be pre-synchronized with the local Wi-Fi network.

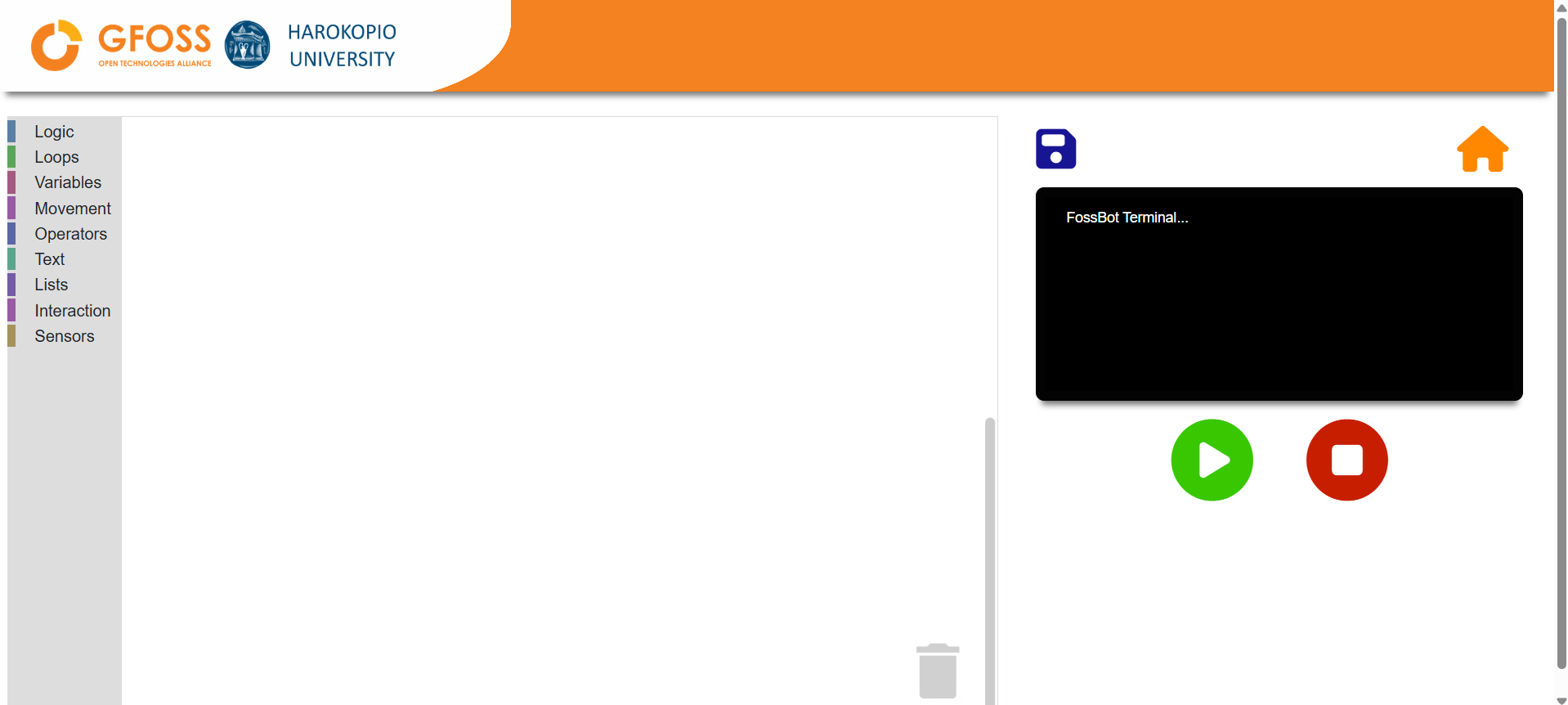
**2.2 Worksheets**

**Activity 1**

**Familiarization with the FOSSBot Programming Environment**

Using an Image Hotspots tool (such as[**https://h5p.org/image-hotspots**](https://h5p.org/image-hotspots)), interactive points will be created on the buttons of the image below to describe the function of each one:

1. Save Code
2. FOSSBot Interface Homepage
3. Run Program
4. Stop Program Execution
5. Trash Bin
6. Blocks Palette
7. Tile Drop Area

****

**Activity 2**

1. Open the FOSSBot. Wait a few seconds for it to connect to your local network.
2. On a computer or laptop connected to the same network as the robot, open Chrome or Firefox and go to the URL: [**http://fossbot-000.local:8081**](http://fossbot-000.local:8081) to access the FOSSBot interface.
3. Upload the project **“black box lights.xml”**.
4. Click **Edit**. 
5. Run the project.
6. What do you observe?

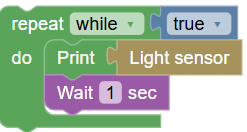
………………………………………………………………………………………………………………………..  
………………………………………………………………………………………………………………………..

1. Make the necessary changes so that the light toggles more quickly. What did you change?

……………………………………………………………………………………………………………………..

**Activity 3**

1. Go to the homepage of the FOSSBot interface.
2. Create a new project .
3. Name it **“Smart Lighting”**.
4. In the “short description for the new project,” you may write:  
   *“Creating Smart Lighting for the Environment and Safety.”*
5. First, we will need a sensor to detect light levels. This will be done using a **light sensor (photoresistor)**.
6. Find the range of values from your photoresistor that indicates darkness, using the blocks shown below and running the program:

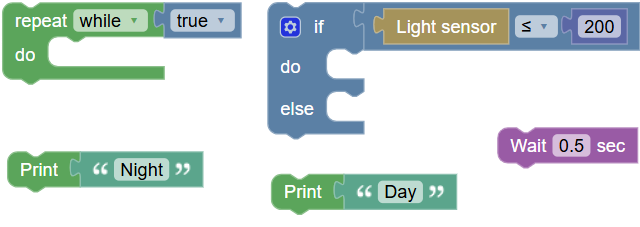


1. Record the value range:

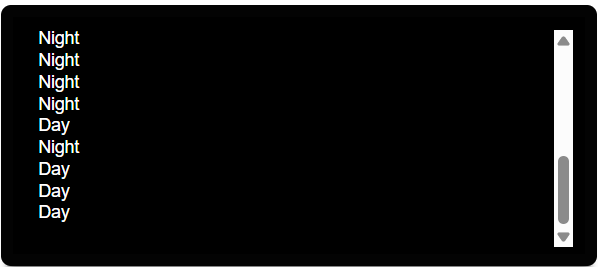
Daylight: …………………………………

Darkness: …………………………………

1. Stop the program execution .
2. Modify your program using the blocks below to display the messages **“Day”** and **“Night”** (the threshold values for the photoresistor can be based on your findings in step 7).



1. Run your program. The values will be displayed on your screen.



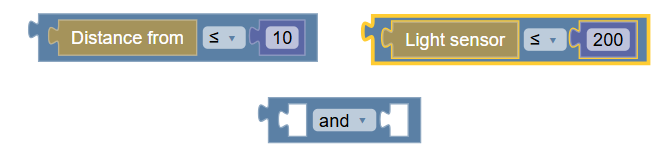
1. Modify your program so that the **FOSSBot's light turns on at night** and **turns off during the day**. You will also need the additional blocks shown below:



**Activity 4**

In this activity, we aim to turn on the lights in the dark only when an object (pedestrian or vehicle) passes by. To detect when an object passes, we will need to use an ultrasonic sensor.

1. We will enhance the code from Activity 3. To create a copy of it, go to the homepage  of the FOSSBot interface. Select **Export**  and then **Import** . Then, enter **edit mode** .
2. Modify the condition inside the **if structure** using the blocks below, in order to achieve greater energy efficiency (i.e., the lights should only turn on when there is an actual need).



1. To be more accurate, modify the night-time display text as follows:



1. Save your changes.

**Activity 5**

**Knowledge Assessment Activities**

1. Which of the following buttons creates a new project in the FOSSBot interface?  
   a)   
   b)   
   c)   
   d)
2. In which block palette can someone find the command “if… then … else”?  
   a) Logic

b) Loops

c) Interaction

d) Sensors

1. Which sensor detects light levels?  
   a) Ultrasonic Sensor

b) Light Sensor (Photoresistor)

c) Noise Sensor

d) Gas Sensor

1. Which sensor can detect the presence of an object?  
   a) Light Sensor (Photoresistor)

b) Noise Sensor

c) Gas Sensor

d) Ultrasonic Sensor

1. To evaluate two conditions at the same time, we use the logical operator...  
   a) or

b) and

c) not

1. The more light we have, the higher the value on the light sensor.  
   a) True

b) False

1. The command turns off the FOSSBot’s LED.  
   a) True

b) False

**Correct Answers:**

1.c, 2.a, 3.b, 4.d, 5.b, 6. False, 7. True