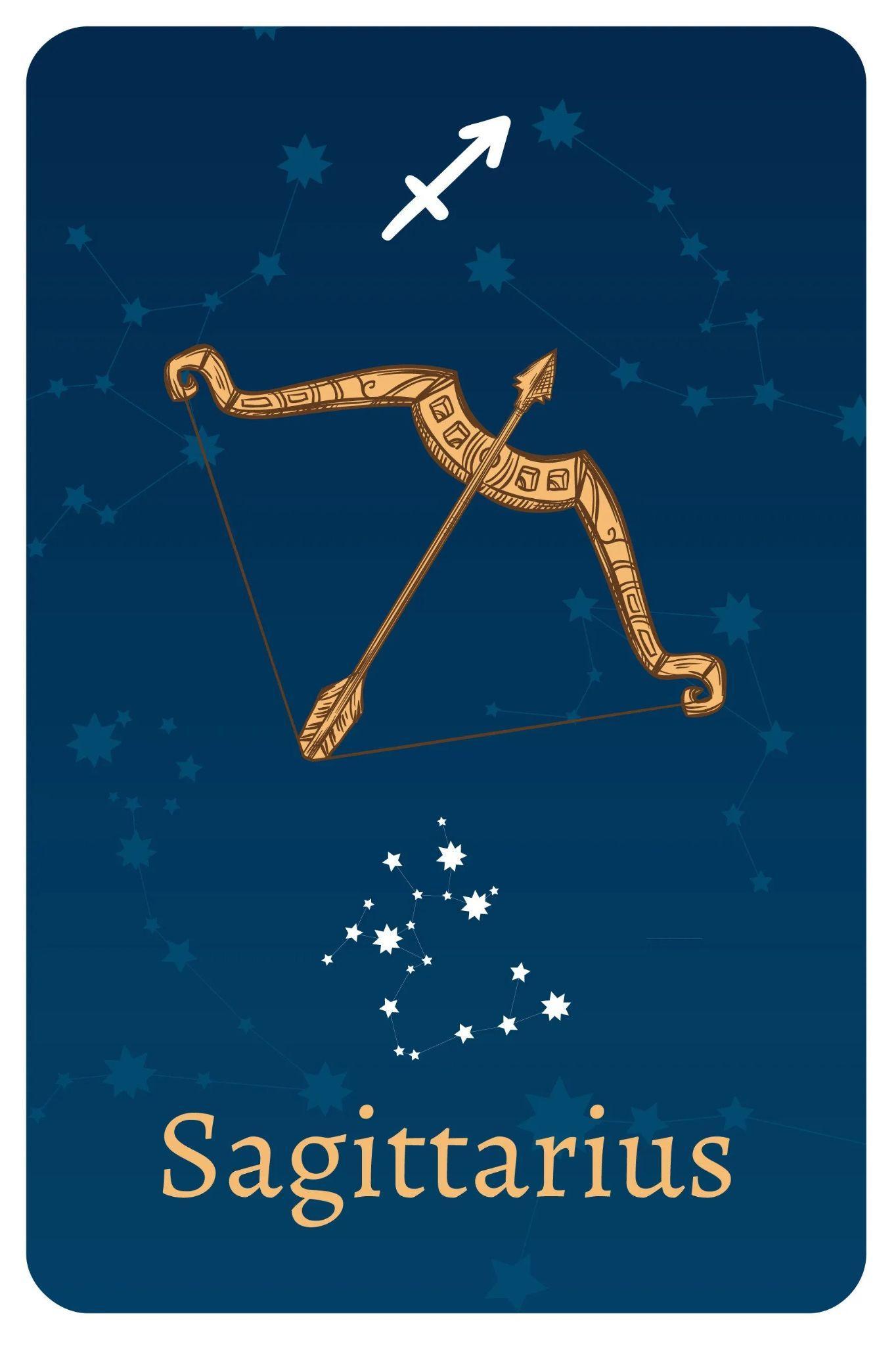
****

TEAM MEMBERS

Shivansh Shukla

Nathan Lee

Edwin Kulakkattolikel

Contents

[PURPOSES AND GOALS 3](#_heading=h.gjdgxs)

[HOW TO INSTALL THE PROGRAMS](#_heading=h.30j0zll) 3

[DEPENDENCIES](#_heading=h.1fob9te) 3

[HOW TO RUN THE PROGRAMS](#_heading=h.3znysh7) 3

[DESIGN ARCHITECTURE](#_heading=h.2et92p0) 4

[PROCESS & WORKFLOW](#_heading=h.tyjcwt) 4

[TEST DATA 5](#_heading=h.3dy6vkm)

[VIDEO RECORDINGS](#_heading=h.1t3h5sf) 9

[CONCLUSIONS](#_heading=h.4d34og8) 9

[REFERENCES](#_heading=h.2s8eyo1) 9

# PURPOSES AND GOALS

This lab aims to provide students with practical experience and knowledge in Python programming across various real-world domains.

The goals include:

1. **Heart Rate Analysis:** Analyze heart rate data from a WAV file converted to CSV file, calculate time-domain measurements, and visualize the results.
2. **COVID-19 Modeling:** Implement the SIDARTHE mathematical model to simulate COVID-19 spread and study different states and dynamics.
3. **EMS**: Design an EMS vehicle lighting control system, manage LED patterns based on inputs, and experiment with customization.
4. **Red Alert:** Enhance skills in game development by adding features to "Red Alerts" and contributing to the gaming industry.

# 

# HOW TO INSTALL THE PROGRAMS

## DEPENDENCIES

**Heart Rate Analysis:** pip install matplotlib, heartpy

**SIDARTHE**: pip install numpy, matplotlib, scipy

**EMS:** KRIA KV260 Board, Jupyter Notebook, pynq library

**RED ALERTS:** pip install pgzero, pgzrun, and random library

# HOW TO RUN THE PROGRAMS

**Heart Rate Analysis:** Make sure .csv spreadsheets are in the same folder as python code, and have pip installed with powershell

**SIDARTHE**: Adjust model parameters and run code.

**EMS:** Connect LEDs, buzzer, and slider to PMOD of KRIA board. Power up KRIA board. Connect KRIA board to a laptop. Load Jupyter Notebook from Laptop. Run code.

**RED ALERTS:** To play the game, ensure you have Python, Pygame, and pgzero installed. Run the script, press 'Space' to start, and click on stars. Red stars advance you to the next level. Winning the final level leads to victory. Restart with 'Space' if you lose. The game becomes tougher with each level.

# DESIGN ARCHITECTURE

**Heart Rate Analysis:**

Hardware: Computer system, peripherals (mouse, keyboard), display

Software: Python, matplotlib, heartpy

**SIDARTHE**:

Hardware: Computer System, Display, and Input Devices.

Software: Python, numpy, matplotlib.pyplot, and scipy.integrate

These software components collectively execute the code, model the system, and generate visual plots.

**EMS:**

Hardware: KRIA Board, Laptop, LEDs, Buzzer, and Slider

Software: Jupyter Notebook, pynq, and ps\_gpio\_kv260.hwh

These software components collectively create a bitstream for the KRIA board to send signals to activate the LEDs and take the Slider signal to activate the buzzer

**RED ALERTS:**

The game's architecture combines user hardware, including the computer and input devices, with key software components. This includes the Python interpreter, Pygame, and game-specific code for logic and rendering. User input triggers actions, Pygame manages graphics, and the game logic determines game state and animation. In cloud scenarios, additional components handle multiplayer and data storage. This architecture orchestrates a smooth gaming experience.

# PROCESS & WORKFLOW

**Heart Rate Analysis:** The heart rate analysis code does an analysis of a .csv file, with all the measurements from an audio file of the heart beat. A .wav to .csv conversion is needed, and can be done using the provided python script. It first graphs the original heart rate from the data values of the .csv file. Then, the library “heartpy” is a tool that analyzes the heart rate by providing the BPM as well as the rejected peaks in the graphs. It also provides additional metric measurements in print.

**SIDARTHE**: This code simulates a disease outbreak using the SIDA model. It sets parameters, defines the model, integrates differential equations, extracts results, and plots the dynamics of susceptible, infected, deceased, asymptomatic, and recovered individuals over 365 days. The code provides insights into the impact of parameter changes on disease progression.

**EMS:** The EMS Vehicle has a left and right turn signal that can be modeled as either input 0(00) or 1(01). The Turn signal LEDs either light up in a sequence from left to right or right to left based on these inputs. The input 2 (10) will result in the breaking output, which turns on all three turn signal LEDs. The input 3 (11) will result in the Emergency output, which triggers the emergency LEDs to alternate flickering. The Buzzer is also enabled by the Emergency input and the slider, which will ring an alarm when both inputs are active.

**RED ALERTS:** The process of playing the game begins with launching the game, followed by an introductory screen, and the user initiates gameplay by pressing 'Space.' During gameplay, stars appear, and the player interacts by clicking on them, aiming for red stars to advance levels. The game becomes increasingly challenging as levels progress. It concludes with either a victory upon completing the final level or a potential game over. After either outcome, players can choose to restart by pressing 'Space.' This workflow involves initiating the game, interacting with it, and responding to game results, all while allowing for continuous gameplay.

# 

# TEST DATA

| Normal Heart Rate Graph |  |
| --- | --- |
| Heart Rate Analysis Graph |  |
| Sidarthe Modeling Plot |  |
| Sidarthe Modeling Plot |  |
| Sidarthe Modeling Plot |  |
| Sidarthe Modeling Plot |  |
| RED ALERTS: Intro |  |
| RED ALERTS: Game win |  |
| RED ALERTS: Game Over |  |
| RED ALERTS: In game |  |

# VIDEO RECORDINGS

| **Recording Title** | **URL** | **Notes** |
| --- | --- | --- |
| Heart Rate Analysis | <https://youtu.be/klNC2v7V7zc> | Video explaining heartrate analysis |
| Sidarthe Model | <https://youtu.be/iohBIN8kjK4> | Video Showing Sidarthe Modeling Code and Plots |
| EMS Hardware | <https://youtube.com/shorts/UEPLUot_Bz8> | Video Demoing EMS Hardware Functionality |
| RED ALERTS | <https://youtu.be/j9n9eDR7SFA> | Video demo for game red alerts |
| EMS Software | <https://www.youtube.com/watch?v=u7eff03fpy4> | Video demo of EMS hardware with JupyterLab |

# CONCLUSIONS

The lab's main goal was to help students learn practical Python programming in real-world scenarios. It included four different tasks. In the Heart Rate Analysis task, we successfully analyzed heart rate data and used a helpful library called 'heartpy' to get important metrics. In the COVID-19 Modeling task, students simulated disease outbreaks to see how different factors affect the spread of diseases. In the EMS (Emergency Lighting Control System) task, students designed a system that controls the lights on an emergency vehicle. And in the Red Alerts (Game Development) task, students improved their game development skills by creating a challenging game.

Throughout the lab, valuable lessons were learned by tackling the specific challenges of each task. Clear and well-documented code proved essential for successful disease modeling, while organized resources and concise instructions were critical for the EMS system design and game development. The lab's key takeaways underscore the importance of understanding how different factors impact a system's performance. To improve future iterations, the team will focus on enhancing coordination and resource organization. Providing accessible and up-to-date materials is vital, and staying attuned to industry trends and technological advancements will ensure the lab remains relevant. Task-specific nuances, such as data processing in Heart Rate Analysis and the intricacies of game mechanics in game development, will guide efforts to make the lab even more effective. In conclusion, this lab has successfully imparted practical Python programming skills, and the knowledge gained will pave the way for a more enriching educational experience in the future.

# REFERENCES

* What Is Agile Reporting? (Definition and How To Complete) <https://www.indeed.com/career-advice/career-development/agile-reporting>
* Test Report <https://strongqa.com/qa-portal/testing-docs-templates/test-report>