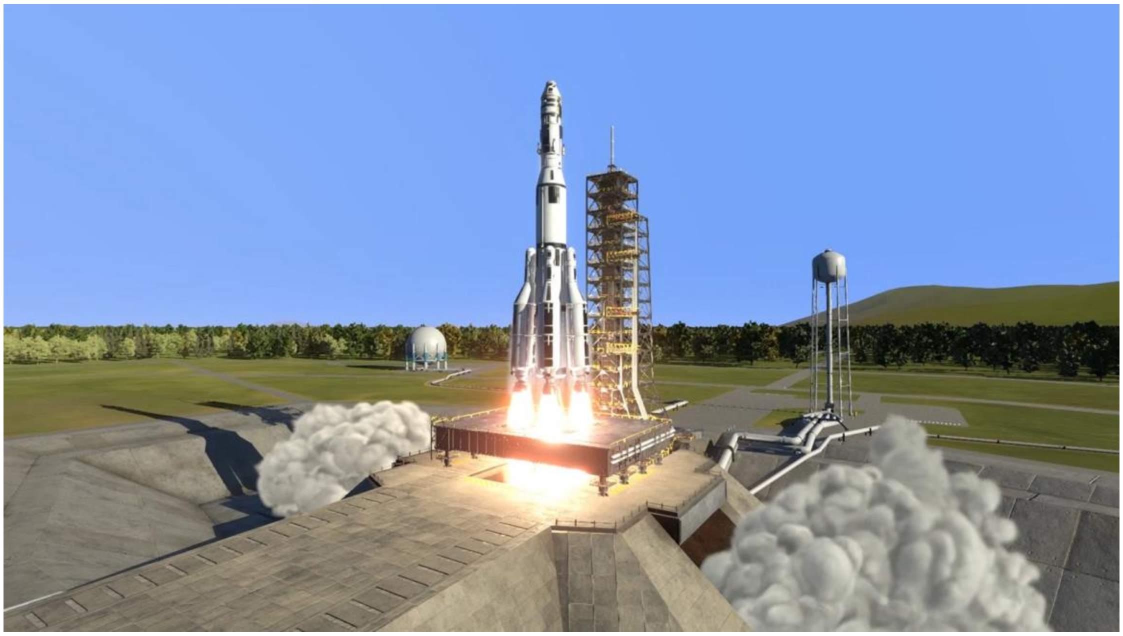
PROJECT III – CTUSR SPACE PROGRAM

INTRODUCTION

This project is primarily intended to assess your abilities, and for most of you, it will be a formality. This project is a bit different from other tasks for newcomers. Unlike your colleagues, your project is not focused on microcontrollers. Instead, you will process data taken from the first project's team using Python. However, this does not make the project any less interesting—in fact, it might be quite the opposite.

This document describes the project task (what you are supposed to do) and how to execute it (how you should do it). The execution section will contain numerous links to tutorials, so even if you're unfamiliar with something, don't worry—you'll learn it all. If you get stuck, don't hesitate to contact me (via Teams or email), and I'll be happy to help. Also, try to help each other as much as possible within your team—we need as many members as we can get. Plan weekly meetings to organize task completion and seek help from senior team members when necessary.



REQUIREMENTS

Your task is to write a Python program that will receive data from the microcontroller of the first team and send a rocket into space. Unfortunately, we don't have the budget for a real rocket capable of reaching such heights yet, so you'll have to settle for launching a virtual rocket in a relatively sophisticated simulation program, Kerbal Space Program (KSP). You'll receive data via Bluetooth and communicate with KSP using a specially modded API.

EXECUTION

1. Your first task will be a Hello World in Python. A formality for most of you, just to verify that you have Python installed on your PC.
2. The second task is to set up a Git account and create a repository for your project. There will be a seminar on this.
3. Download KSP onto your computer; we recommend the Steam version. We assume you already own it. If not, one or two copies should suffice for the group. If you don't have a copy of the game yourself, consider purchasing one. Make sure that the device where you will be testing and demonstrating your program has no mods installed.
4. As a group (especially if not all of you are familiar with KSP), or individually, spend a few tens of minutes in the game environment. Build a few rockets and fly them. Try to get into orbit around Kerbin (the home planet). The end product of this step should be a simple single- or multi-stage rocket capable of flying at least a few kilometers high, which you will later use to learn the basics of control through Python.
5. Download the KRPC mod for KSP from the link in reference [1]. This mod allows you to communicate between the game environment (simulation) and Python (control environment/flight computer). Extract the downloaded .zip file into the game's GameData folder.
6. Open your favorite programming environment (we strongly recommend VS Code) and create a new program. Before starting, you'll need to install the KRPC library (instructions in documentation [2.]). Follow the documentation to connect KSP with your Python program.
7. Test the connection between KSP and your program by launching your test rocket in KSP from Python after running the control program (or pressing a button).

8. Gradually familiarize yourself with the functions that the KRPC library supports and try them out.
9. The previous points were just preparation. Your task now is to use the experience gained to write a control program that automatically launches a rocket into space upon detecting a start instruction, preferably into a stable orbit around Kerbin. How you approach this task is entirely up to you, but note that the test rocket you've been using may not be ideal for your program. You must carefully think about how you plan to achieve the goal and design both the rocket in KSP and the simulation program accordingly.
10. The tenth task involves connectivity with other projects. You must be able to receive data in the format that your colleagues from Project I will provide. You will process the data locally, and based on that, the rocket launch will be initiated. You'll receive the data via Bluetooth.
11. The eleventh task is to create documentation for the entire project. Your colleagues are limited to a maximum of 3 A4 pages, but we'd like you to also explain the principles on which you based your control program and possibly describe some basic physical laws associated with it, even if that means exceeding the 3-page limit by a page or two. The documentation will be in English. I recommend creating the documentation gradually during your meetings.

REFERENCES

https://spacedock.info/mod/69/kRPC

https://krpc.github.io/krpc/