Overlord

Vision
Create a Discord bot that helps with tasks an RPI student would likely need (such as degree/course planner), with the ability to scale out into features across a multitude of categories, such as web crawling, mini games, and server moderation.

The bot library is designed to allow many people to work on separate components at the same time and enable students of any experience level to participate. Beginners can work on easier functions, while more experienced students can work on complex functions.

Stack
Python is the only language required. However, subprojects are allowed to use different languages as they see fit.

Goals

1. Learn to use GitHub with proper conventions in an environment with a big team.
2. Proficient in discord.py library and capable of running an independent discord bot.
3. Degree planner generates an outline of the courses they want to take through an interactive graphical interface.
4. Web crawler capable of retrieving and storing course information from RPI's catalog.
5. Chatbot to converse with users, format messages, manage threads and permissions and provide an API for other features to use to display graphical content.

Milestones

- Project will be run in two week sprints.

Sprint 1 (9/20 - 10/4)

- Form groups of 2-4 students with each group working on a specific sub-feature of the bot. Each group should be able to perform basic interactions through the Discord API.
- Every member should be capable of independently running the bot from their own computer and making commits to the common github project.
- Proposals for additional projects submitted by members.
- Finalize project scope for all subteams before sprint 2

Sprint 2 (10/4 - 10/18)

- Complete architecture design for all subprojects
- Prototype of basic functionalities

Sprint 3 (10/18 - 11/1)

- Revision of functionalities, meeting with each group to discuss any changes to original goals
- Switch members working on short term sub-projects (i.e. Calendar) to their secondary projects. Switch will only take place if said sub-project is completed, if not, its timeline will be extended.

Sprint 4 (11/1 - 11/15)

- Full functionality implemented based on architectures designed in Sprint 2
- Testing and bug fixes
- Features complete (all planned functions are working, but maybe a few bugs remaining)

Sprint 5 (11/15 - 12/9) - Final Sprint (Longer sprint due to Thanksgiving break)

- Cross team assembly, if any subprojects can combine functions, merge them together
- Code freeze (no critical bugs, ready for presenting)
- Look into hosting options and decide whether the bot is worth hosting and if this project should continue into the next semester.

Milestones for sub-projects

- Each subproject will have at least one member dedicated to it. If there are 3+ people working on said subproject, the dedicated team member(s) will take on the responsibility of leading the rest on feature development. Project leads will monitor overall progress and development in all subprojects.
- unitalicized names represent primary projects for those members
- *italicized names represent secondary projects for those members*

1) **Degree Planner**
    a) **Members:** Alan, Dell, Peter, Zhehua, *Stan*
    b) **Description:** Generates a catalog of all RPI classes and attributes to allow the user to generate a plan that both fulfills all degree requirements and maximizes the desired electives they wish to take. Will take input from the web crawler.
    c) **Milestones:**
        i) Sprint 1: Create an outline of the data structures needed to represent and calculate the scheduling.
        ii) Sprint 3: Prototype capable of generating outlines of test data that fulfills all requirements.
        iii) Sprint 4: Develop methods for displaying graphical interfaces to use on Discord to demonstrate the various possibilities of displaying information, such as using SVGs, buttons and select menus.
        iv) Sprint 5: Capable of fetching data automatically from RPI catalog and fully accounts for all degree requirements of single and dual majors offered by RPI.
    d) **Possible Additional Stack:** SQL
2) **Web Crawling**
    a) **Members:** Jack, Timothy, *Peter, Sam*
    b) **Description:** Works with the degree planner team to web crawl the data from RPI's course catalog. Also pulls Finance data and Wiki data from Yahoo Finance and Wikipedia's apis.
    c) **Milestones:**
        i) Sprint 1: Familiarize with using web interaction APIs such as selenium and beautifulsoup to parse HTML/XML documents.
        ii) Sprint 3: Capable of fetching data from RPI catalog through web crawling.
        iii) Sprint 5: Display data fetched from websites such as yahoo finance on Discord when requested by a user, integrate with other modules in our project.
    d) **Possible Additional Stack:** SQL

3) **Minigames**
    a) **Members:** Suhan, Sam, Christian
    b) **Description:** game like Go or Chess with a board made of emojis
    c) **Milestones:**
        i) Sprint 2: Display board states and take input from Discord chat.
        ii) Sprint 3: Functional gameplay using existing game engines for Go and Chess.
        iii) Sprint 5: Custom gamemodes.
4) **Chat/Moderation**
    a) **Members:** Xihong, Bijun, *Sam, Christian, Stan, Arslan*
    b) **Description:**
    c) **Milestones:**
        i) Sprint 1: Use existing open source chatbots to converse.
        ii) Sprint 2: Moderation features planned
        iii) Sprint 3: Integration with other bot features, such as developed as an information display API and permission control
        iv) Sprint 5: Fine tuning
5) **Polling**
    a) **Members:** Tyler, Viano
    b) **Description:** Creates polls
    c) **Milestones:**
        i) Sprint 2: Configure polls through both text-based and graphical interface with bot and display on selected channel of server.
        ii) Sprint 4: Integrate with external input such as Google forms.
6) **Math Module**
    a) **Members:** Arslan
    b) **Description:** Parse math expressions and evaluate using various algorithms such as the Shunting-Yard Algorithm or reverse polish notation.
    c) **Milestones:**
        i) Sprint 2: Parser implemented using the Shunting-yard algorithm.
        ii) Sprint 3: Evaluate any algebraic expressions.
        iii) Sprint 4: Evaluate simple calculus problems.
7) **Calendar**
    a) **Members:** Stan
    b) **Description:** Private/shared calendar on Discord
    c) **Milestones:**
        i) Sprint 1: Creating commands and familiarizing with using the Discord API.
        ii) Sprint 3: Working calendar that can be used in both private and server chats to record events and notes. This subproject should be completed before the end of sprint 3. Member(s) will switch over to their secondary projects.

Blockers

- It is possible that certain sub-projects are abandoned or heavily modified from their original description throughout the course of the semester. To ensure we stay on track, we expect at least 75% of the functionalities we described to be implemented regardless of which sub-project they are listed under.

  For example, the web crawler may merge with the degree planner without affecting overall functionality, and similarly the chatbot and moderation can split into two different projects and still preserve all original goals.

  If a sub-project is removed entirely due to lack of interest, members originally assigned to them would still be expected to perform tasks of equivalent workload and intensity in another sub-project.

- **Dependency Blockers:** It is possible that certain API or tools do not function correctly, especially smaller open source projects that may not have received through testing. For each piece of external software we use, there will be several alternatives listed to ensure the project will not halt due to a single faulty external program.

- **Time Blockers:** The timelines of all sub-projects are adjustable depending on the speed at which a milestone was finished. If a milestone was not completed on time, its timeline can be pushed back. If pushing back will mean the sub-project won't be completed on time before the end of the semester, then the goals of said project will be rewritten to focus on what *can* be completed within this semester.

- **People Blockers:**
    - **Guidelines for switching sub-projects:**
        - Some projects, such as the calendar or poller, are not designed to take all semester. The people working on them have expressed secondary choices for their subproject selection and they will be automatically moved over to those projects when their primary project ends.

        - If someone wants to switch projects after sprint 1, a meeting will be required with a project lead to discuss any issues they've encountered and a suitable switch will be determined. Each sub-project may receive a maximum of one new member after the end of sprint 1.

    - **Members leaving:**
        - All no-credit and external members were not heavily considered when planning the milestones since they may switch or leave the project at any time. As such, their leaving the project will have little impact.

        - For-credit members leaving may result in a time blocker for the project. However, we have left enough buffer time to allow 2 or 3 members to drop out of the project without affecting the overall timeline. If they were part of a single person sub-project then that sub-project would be removed (but may be brought back a different semester).

        - A mid-session discussion will be held after sprint 3 to discuss the progress in detail with all members. If the pace is moving significantly slower than expected to the point where it is no longer feasible to complete the original goals, we will either adjust the timeline, move members around or remove sub-projects entirely.

          Note that the 75% goal completion rate would still be maintained, so the removal of sub-projects will be kept to a minimum to ensure no one feels burned out or unmotivated from their current project trajectory.

Team Members (Name, RCSID)

1. Jack Wang (wangj49)
2. Alan Zhang (zhanga10)
3. Tyler Chan (chant7)
4. Viano Arcery (arcerv)
5. Bijun Wu (wub8)
6. Xihong Xie (xiex11)
7. Andy Chen (chena19)
8. Samael (Sam) Collins (collis10)
9. Stanley (Stan) Guo (guos4)
10. William (Arslan) Lau (lauw)
11. Dell Zheng (zhengd)
12. Suhan Gui (guis2)
13. Christian Stec (stecc)
14. Zhehua Zhang (zhangz45)

1. Peter Capriotti (caprip2) *
2. Timothy Liakh (liakht) *
3. Tom Wang (UC Irvine) **

*Taking RCOS for 0 credits
**External