**Progress Report**

**- Increment 1 -**

**Group #19**

# Team Members

Dylan McClure

* + **fsuid**: dmm18dk
  + **github**: dylnmc

Julian Sweatt

* + **fsuid**: js15f
  + **github**: juliansweatt

Caleb Smith

* + **fsuid**: cas16w
  + **github**: ninjanole5558

Lucas Zavalía

* + **fsuid**: lrz17
  + **github**: lzavalia

Michael Heron

* + **fsuid**: mph12e
  + **github**: mikeheroncode

1. **Project Title and Description**

**Plethora Py**:

Plethora Py resembles your typical arcade-style gaming suite. It is a collection of childhood games for the people who grew up playing board games. These digital versions allow a person to take their favorite games with them as long as they have a computer or a Raspberry Pi. Log in and try your hand at your nostalgic childhood games!

1. **Accomplishments and overall project status during this increment**

Each person developed one game. This allowed the developers to get their feet wet with pygame and become familiar with how to implement a game. Each game should be able to be played to completion. The UI and API have been developed and a test game has been created to demonstrate the interaction between a game and the ser interface. However, Each person’s game operates independently, and none of the games have been integrated into the API as of yet. Each game is run independently and handles all operations, some of which things will be handled by the UI and API in the future.

The games that were developed were chess, tic-tac-toe, connect4, multisnake, and tetris. Dylan used bitboards (which is the de-facto standard for representing chess games) to store the basic information for a game and intends to expand it to verify moves and create a chess engine. Lucas built tetris from scratch and used linear algebra to implement the movements for the pieces. Caleb used images to draw x’s and o’s to the screen. Julian implemented connect4 and handled win and draw conditions and indicated this as an overlay on the screen. Michael handled multiple users and implemented multi-person snake.

1. **Challenges, changes in the plan and scope of the project and things that went wrong during this increment**

* **packaging!** Python packaging is sometimes quite difficult. Since there is a central component of the game that needs to launch submodules (that are games) that can also import from parent directories, it was rather difficult to get everything set up properly. However, this was resolved by drawing from the wisdom of many other python developers who have encountered similar problems and using that wisdom to create a top-level package from which everything can import.
* **Learning Python**: Initial use of Python was difficult for some group members. Starting the code for a game was very slow until group member hit a learning curve. Help from more experienced group members was paramount.
* **Common Style Learning**: to develop with a common documentation style and code design pattern was a challenge we’ve had to overcome to ensure our individual games will work together on the Plethora Py game platform.
* **Game Design Fundamentals**: The are basic game design concepts that some of us were totally unfamiliar with. Things like event loops, blits, and handling basic physics were new concepts to some of us and needing those to implement even our most basic ideas was quite challenging.
* **Working in a Group**: Some members of the group had never worked on software in such a collaborative way so there was a bit of a learning curve involved in transitioning from writing code in solitary to in a group.

1. **Team Member Contribution for this increment**

* Dylan McClure:
  1. progress report
     + Team Member names - added fsuids and github usernames
     + Accomplishments and overall project status during this increment – helped put part of it together
     + Challenges, changes in the plan and scope of the project and things that went wrong during this increment - added “packaging!” section
  2. requirements
     + Improved Functional Requirements
     + Created Use Case Diagrams
  3. implementation and testing
     + Programming Languages
     + Execution-based Non-Functional Testing - second paragraph only
  4. source code
     + setup and packaging: setup.py and directory structure
     + API: src/arcade/plethoraAPI.py, src/arcade/games/test/\*
     + chess: src/arcade/games/
  5. video
     + chess screen recording
     + UI/API screen recording
* Caleb Smith:
  1. progress report
     + Team Members, Project Name/Description, Challenges, Plans
  2. requirements
     + Software Requirements and Design: Overview, Functional requirements, Non-Functional Requirements
  3. implementation and testing
     + Platforms, Functional Testing, Non-Fuctional Testing, Non-Execution-based Testing
  4. source code
     + tictactoe: src/arcade/games/tictactoe/\*
  5. video
     + tictactoe screen recording
     + general overview (intro)
* Julian Sweatt:
  1. progress report
     + Project Description
     + Challenges
  2. requirements
     + Operating environment
  3. implementation and testing
     + Execution-based Non-Functional Testing
     + Non-Execution-based Testing
  4. source code
     + connect4 : `src/arcade/gamesconnect4/\*``
  5. video
     + Recorded, edited, animated, and published the video
     + Created a demo of Connect 4 with voice over
* Lucas Zavalía:
  1. progress report
     + Challenges
     + Plans for the next increment
  2. requirements
     + Helped improve functional requirements
     + Added use a use case diagram
  3. source code
     + tetris: src/arcade/games/tetris.py
  4. video
     + tetris work through
     + change in scope
     + plans for next increment
* Michael Heron:
  1. progress report
     + Challenges
  2. requirements
     + Sequence Diagram
     + Operating Environment
     + Assumptions and Dependencies
  3. source code
     + MultiSnake Branch
  4. video
     + MultiSnake demo
     + Progress Report

1. **Plans for the next increment**
2. Integrate each game with plethora API
3. Develop 1-2 more games per person
   * This should be easier now that everyone is familiar with pygame
   * Maybe plethora API can help facilitate game development
4. Fix withstanding issues in the first game
   * Basic Functionality has been achieved but some features of each game could use another look in order to make the game flow better
5. **Link to video**

<https://youtu.be/N5Da6_acY4I>