Project 2: Group 45

Names: Brian Huynh (bhhuynh), Shanmukh Chatadi (schatad), Manav Kamdar(mkamdar)

Repository Link: https://github.com/brianhhuynh38/slackpoint-v3

SP SlackPoint!

Turn tasks into triumphs... Make productivity fun!

Completed Features

SlackPoint is an innovative platform designed to boost user engagement and productivity by integrating gamified elements into daily tasks. Here's an overview of our recent developments:

- **Extensible Gamification Features**
  - We've enriched the platform with simple game mechanics to make task completion more engaging, also providing a simple, yet robust, framework for adding in more advanced game mechanics that could easily be incorporated into the system. Users now earn Task Points for each completed task.
- **Comprehensive Progression Systems**
  - Task Points contribute to character development within SlackPoint. Users can allocate these points to enhance their character's abilities, fostering a sense of growth and achievement.
- **Interactive Battle Systems**
  - We've laid the groundwork for text-based battles, where users can engage in friendly competitions. These battles utilize character stats, which are influenced by task completion and a structured growth system.
- **Task Tags and Filtering**
  - Task categorization is now available for better organization of large task lists. The groundwork is laid, with final checks pending.

Introduction

Lacking motivation? SlackPoint provides a fun and competitive way to ensure that you and your team get your work done while giving a sense of accomplishment and progression outside of your work:

- **Gamification of Productivity:** Incentivize your teammates with Task Points, earnable via task completion, and use those to get stronger.

- **Heavily-Customizable Character Progression:** Customize your character with any class and stat configuration you could want. Use Task Points to grow and develop your character!

- **Quick and Easy Task Planning:** Create tasks in no time at all using simple Slack commands!

- **Organize Tasks More Efficiently:** Tag your tasks so that you can better recognize and organize your tasks.
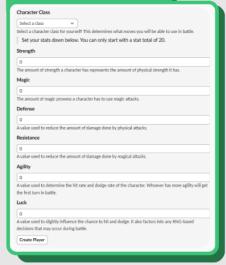
Potential Future Features

- **In-Depth Battle System:** Currently, there is only a foundation for the battle system ready. This is free to be changed as per the wishes of any developer due to the implementation's simplicity, in terms of game mechanics.
  - **Delay-Based:** Attacks use a delay-based battle system in which the type of attack determines how long it takes to take another action.
  - **Type-Weakness + Dynamic Turn-Based Systems:** Each player is able to take multiple turns and is able to expend turns to change their class. Turn count can be manipulated by exploiting weaknesses and resistances.
- **Task Delegation and Assignment:**
  - While the task-tracking system is useful, introducing the **ability to delegate them** to specific people or teams would be very useful for team leaders to designate and organize tasks.
  - **Introduce permissions** that only allow some people to have the power to resolve tasks for other people to distribute points, allowing for more order within the chat group.
- **Team Formations and Tasks:** The introduction of teams would be greatly beneficial as most software development operates in teams.
  - **Assign tasks to a group of people** and be able to distribute points across them all to make task delegation more accurate
  - **Teams can also be added to battle formations** within the game design aspect to introduce more advanced gameplay mechanics and features, increasing activity between teammates.

Character Creation

Character Class

Select a class ▼

Select a character class for yourself! This determines what moves you will be able to use in battle.
Set your stats down below. You can only start with a stat total of 20.

Strength

0

The amount of strength a character has represents the amount of physical strength it has.

Magic

0

The amount of magic prowess a character has to use magic attacks.

Defense

0

A value used to reduce the amount of damage done by physical attacks.

Resistance

0

A value used to reduce the amount of damage done by magical attacks.

Agility

0

A value used to determine the hit rate and dodge rate of the character. Whoever has more agility will get the first turn in battle.

Luck

0

A value used to slightly influence the chance to hit and dodge. It also factors into any RNG-based decisions that may occur during battle.

Create Player

Grow Stronger!

10:48 You have come to understand a new power...

Character Class: Fire Mage -> Swordmaster

STR: 0 -> 5 (5)
MAG: 20 -> 5 (-15)
DEF: 0 -> 5 (5)
RES: 0 -> 0 (0)
AGL: 0 -> 2 (2)
LUK: 0 -> 3 (3)

Test Cases

There are currently a total of 31 test cases (19 of which we added) in the current version.

These tests primarily ensure that the calculations for damage and hit rate are accurate to ensure a fair experience for all players.

Tech Stack

slack api

Flask   Python   PostGreSQL   ngrok

Check out the Repository here!

SCAN ME

GROUP 45: Brian Huynh, Manav Kamdar, Shanmukha Srinivas Sai Chatadi

NOTES	POINTS	EVIDENCE
Workload is spread over the whole team (one team member is often x times more productive than the others…)	Brian:3 Manav:3 Shanmukh:3	Everyone was assigned their own branches and completed their own work. https://github.com/brianhhuynh38/slackpoint-v3/compare/main...development
but nevertheless, here is a track record that everyone is contributing a lot)	Brian:3 Manav:3 Shanmukh:3	Everyone's commits can be seen via the commit log, as well as the amount of substance that each one contains https://github.com/brianhhuynh38/slackpoint-v3/compare/main...development
Number of commits	Brian:3 Manav:3 Shanmukh:3	The number of commits are 32. We each had our own style of committing: Brian made many smaller commits while Manav and Shanmukh did larger scale commits. https://github.com/brianhhuynh38/slackpoint-v3/compare/main...development
Number of commits: by different people	Brian:3 Manav:3 Shanmukh:3	Brian: 39 Manav: 7 Shanmukh: 4 https://github.com/brianhhuynh38/slackpoint-v3/compare/main...development
Issues reports: there are **many**	Brian:3 Manav:3 Shanmukh:3	https://github.com/brianhhuynh38/slackpoint-v3/issues

Issues are being closed	Brian:3 Manav:3 Shanmukh:3	https://github.com/brianhhuynh38/slackpoint-v3/issues
Docs: doco generated, format not ugly	Brian:3 Manav:3 Shanmukh:3	Github wiki and sphinx docs https://github.com/brianhhuynh38/slackpoint-v3/wiki
Docs: what: point descriptions of each class/function (in isolation)	Brian:3 Manav:3 Shanmukh:3	Sphinx docs
Docs: how: for common use cases X,Y,Z mini-tutorials showing worked examples on how to do X,Y,Z	Brian:3 Manav:3 Shanmukh:3	ReadMe.md, INSTALL.md, GitHub wiki https://github.com/brianhhuynh38/slackpoint-v3/wiki
Docs: why: docs tell a story, motivate the whole thing, deliver a punchline that makes you want to rush out and use the thing	Brian:3 Manav:3 Shanmukh:3	Mostly GitHub wiki which covers all the theory behind the Game Design portion. https://github.com/brianhhuynh38/slackpoint-v3/wiki
Docs: short video, animated, hosted on your repo. That convinces people why they want to work on your code.	Brian:2 Manav:2 Shanmukh:2	https://www.youtube.com/watch?v=ThgeL1FfqfM
Use of version control tools	Brian:3 Manav:3 Shanmukh:3	Yes, GitHub is our main version control

Test cases exist	Brian:3 Manav:3 Shanmukh:3	Yes in GH, mostly for BattleHelper.py (19 tests + some modified tests for help, viewCompleted, viewPending)
Test cases are routinely executed	Brian:3 Manav:3 Shanmukh:3	Yes, tests were run before committing major components
Issues are discussed before they are closed	Brian:3 Manav:3 Shanmukh:3	Somewhat, issues are discussed in some detail before being closed
Chat channel: exists	Brian:3 Manav:3 Shanmukh:3	Yes, using WhatsApp and Discussions and Issues
Test cases: a large proportion of the issues related to handling failing cases.	Brian:2 Manav:2 Shanmukh:2	Not really, most issues were used as a means of keeping track of what we needed to do. Though issues would still be
Evidence that the whole team is using the same tools: everyone can get to all tools and files	Brian:3 Manav:3 Shanmukh:3	Yes in GH (INSTALL.md)
Evidence that the whole team is using the same tools (e.g. config files in the repo, updated by lots of different people)	Brian:3 Manav:3 Shanmukh:3	Yes in GH (.venv)

Evidence that the whole team is using the same tools (e.g. tutor can ask anyone to share screen, they demonstrate the system running on their computer)	Brian:3 Manav:3 Shanmukh:3	Yes, but Mac has issues with certifications that we did not resolve
Evidence that the members of the team are working across multiple places in the code base	Brian:3 Manav:3 Shanmukh:3	Yes in GH
Short release cycles	Brian:3 Manav:3 Shanmukh:3	Not applicable for a small project
The file .gitignore lists what files should not be saved to the repo. See [examples]i(https://github.com/github/gitignore)	Brian:3 Manav:3 Shanmukh:3	Yes in GH
The file INSTALL.md lists how to install the code	Brian:3 Manav:3 Shanmukh:3	Yes in GH
The file LICENSE.md lists rules of usage for this repo	Brian:3 Manav:3 Shanmukh:3	Yes in GH
The file CODE-OF-CONDUCT.md	Brian:3 Manav:3 Shanmukh:3	Yes in GH

lists rules of behavior for this repo; e.g. see example		
The file CONTRIBUTING.md lists coding standards and lots of tips on how to extend the system without screwing things up; e.g. see example	Brian:3 Manav:3 Shanmukh:3	Yes in GH
The file README.md contains all the following		
Video	Brian:3 Manav:3 Shanmukh:3	https://youtu.be/Fu7Q7a5BYvw
DOI badge: exists. To get a Digitial Object Indentifier, regiser the project at Zenodo. DOI badges look like this:	Brian:3 Manav:3 Shanmukh:3	Yes in ReadMe
Badges showing your style checkers	Brian:3 Manav:3 Shanmukh:3	Yes in Readme, but is only run locally (flake)
Badges showing your code formatters.	Brian:3 Manav:3 Shanmukh:3	Yes in Readme
Badges showing your syntax checkers.	Brian:3 Manav:3 Shanmukh:3	Yes in Readme

Badges showing your code coverage tools	Brian:3 Manav:3 Shanmukh:3	Yes in Readme at the top
Badges showing any other Other automated analysis tools	Brian:3 Manav:3 Shanmukh:3	All badges at top

The following things are queries in the sustainability form, the ones above were in the rubric.

Does your website and documentation provide a clear, high-level overview of your software?	Brian:3 Manav:3 Shanmukh:3	Yes, database and game design theory
Does your website and documentation clearly describe the type of user who should use your software?	Brian:3 Manav:3 Shanmukh:3	Yes
Do you publish case studies to show how your software has been used by yourself and others?	Brian:3 Manav:3 Shanmukh:3	Gifs are shown in the ReadMe for functional
Is the name of your project/software unique?	Brian:3 Manav:3 Shanmukh:3	Yes
Is your project/software name free from trademark violations?	Brian:3 Manav:3 Shanmukh:3	Yes
Is your software available as a package that can be deployed without building	Brian:3 Manav:3	Yes

it?	Shanmukh:3	
Is your software available for free?	Brian:3 Manav:3 Shanmukh:3	Yes
Is your source code publicly available to download, either as a downloadable bundle or via access to a source code repository?	Brian:3 Manav:3 Shanmukh:3	Yes
Is your software hosted in an established, third-party repository likeGitHub (https://github.com), BitBucket (https://bitbucket.org),Laun chPad (https://launchpad.net) orSourceForge (https://sourceforge.net)?	Brian:3 Manav:3 Shanmukh:3	Yes , GitHub
Is your documentation clearly available on your website or within your software?	Brian:3 Manav:3 Shanmukh:3	Yes
Does your documentation include a "quick start" guide, that provides a short overview of how to use your software with some basic examples of use?	Brian:3 Manav:3 Shanmukh:3	Yes
If you provide more extensive documentation, does this provide clear, step-by-step instructions on how to deploy and use your software?	Brian:3 Manav:3 Shanmukh:3	Yes
Do you provide a	Brian:3	Yes

comprehensive guide to all your software's commands, functions and options?	Manav:3 Shanmukh:3	
Do you provide troubleshooting information that describes the symptoms and step-by-step solutions for problems and error messages?	Brian:3 Manav:3 Shanmukh:3	Yes, there are error messages outputted to the user when an invalid action is taken
If your software can be used as a library, package or service by other software, do you provide comprehensive API documentation?		N/A

Do you store your documentation under revision control with your source code?	Brian:3 Manav:3 Shanmukh:3	Yes
Do you publish your release history e.g. release data, version numbers, key features of each release etc. on your web site or in your documentation?	Brian:3 Manav:3 Shanmukh:3	Yes, releases are on github under releases, done from zenodo
Does your software describe how a user can get help with using your software?	Brian:3 Manav:3 Shanmukh:3	Yes
Does your website and documentation describe what support, if any, you provide to users and developers?	Brian:3 Manav:3 Shanmukh:3	Yes

Does your project have an e-mail address or forum that is solely for supporting users?	Brian:1 Manav:1 Shanmukh:1	No, but the discussions forum is there for developers
Are e-mails to your support e-mail address received by more than one person?	Brian:0 Manav:0 Shanmukh:0	No
Does your project have a ticketing system to manage bug reports and feature requests?	Brian:1 Manav:1 Shanmukh:1	Yes, technically, through GitHub Issues. No formal one though
Is your project's ticketing system publicly visible to your users, so they can view bug reports and feature requests?	Brian:3 Manav:3 Shanmukh:3	Yes, if GitHub Issues counts
Is your software's architecture and design modular?	Brian:3 Manav:3 Shanmukh:3	Yes
Does your software use an accepted coding standard or convention?	Brian:3 Manav:3 Shanmukh:3	Yes
Does your software allow data to be imported and exported using open data formats?	Brian:3 Manav:3 Shanmukh:3	Yes
Does your software allow communications using open communications protocols?	Brian:3 Manav:3 Shanmukh:3	Yes
Is your software cross-platform compatible?	Brian:3	Yes

	Manav:3 Shanmukh:3	
Does your software adhere to appropriate accessibility conventions or standards?	Brian:3 Manav:3 Shanmukh:3	Yes
Does your documentation adhere to appropriate accessibility conventions or standards?	Brian:3 Manav:3 Shanmukh:3	Yes
Is your source code stored in a repository under revision control?	Brian:3 Manav:3 Shanmukh:3	Yes
Is each source code release a snapshot of the repository?	Brian:3 Manav:3 Shanmukh:3	Yes
Are releases tagged in the repository?	Brian:3 Manav:3 Shanmukh:3	Yes (v.2.0, v.3.0, etc.)
Is there a branch of the repository that is always stable? (i.e. tests always pass, code always builds successfully)	Brian:3 Manav:3 Shanmukh:3	Yes, main
Do you back-up your repository?	Brian:3 Manav:3 Shanmukh:3	Yes

Do you provide publicly-available instructions for building	Brian:3 Manav:3	Yes

your software from the source code?	Shanmukh:3	
Can you build, or package, your software using an automated tool?	Brian:3 Manav:3 Shanmukh:3	Yes
Do you provide publicly-available instructions for deploying your software?	Brian:3 Manav:3 Shanmukh:3	Yes
Does your documentation list all third-party dependencies?	Brian:3 Manav:3 Shanmukh:3	Yes
Does your software list the web address, and licences for all third-party dependencies and say whether the dependencies are mandatory or optional?	Brian:3 Manav:3 Shanmukh:3	Yes
Can you download dependencies using a dependency management tool or package manager?	Brian:3 Manav:3 Shanmukh:3	Yes
Do you have tests that can be run after your software has been built or deployed to show whether the build or deployment has been successful?	Brian:3 Manav:3 Shanmukh:3	Yes
Do you have an automated test suite for your software?	Brian:3 Manav:3 Shanmukh:3	Yes
Do you have a framework to periodically (e.g. nightly)	Brian:0	No

run your tests on the latest version of the source code?	Manav:0 Shanmukh:0	
Do you use continuous integration, automatically running tests whenever changes are made to your source code?	Brian:3 Manav:3 Shanmukh:3	Yes
Are your test results publicly visible?	Brian:3 Manav:3 Shanmukh:3	Yes
Are all manually-run tests documented?	Brian:0 Manav:0 Shanmukh:0	N/A

Does your project have resources (e.g. blog, Twitter, RSS feed, Facebook page, wiki, mailing list) that are regularly updated with information about your software?	Brian:3 Manav:3 Shanmukh:3	Yes , GitHub issues and discussions and wiki
Does your website state how many projects and users are associated with your project?	Brian:3 Manav:3 Shanmukh:3	Yes
Do you provide success stories on your website?	Brian:0 Manav:0 Shanmukh:0	No
Do you list your important partners and collaborators on your website?	Brian:3 Manav:3 Shanmukh:3	Yes

Do you list your project's publications on your website or link to a resource where these are available?	Brian:3 Manav:3 Shanmukh:3	Yes
Do you list third-party publications that refer to your software on your website or link to a resource where these are available?	Brian:3 Manav:3 Shanmukh:3	Yes
Can users subscribe to notifications to changes to your source code repository?	Brian:3 Manav:3 Shanmukh:3	Yes
If your software is developed as an open source project (and, not just a project developing open source software), do you have a governance model?	Brian:0 Manav:0 Shanmukh:0	No
Do you accept contributions (e.g. bug fixes, enhancements, documentation updates, tutorials) from people who are not part of your project?	Brian:3 Manav:3 Shanmukh:3	Yes, with approval (if requested, though it hasn't happened)
Do you have a contributions policy?	Brian:3 Manav:3 Shanmukh:3	Yes
Is your contributions' policy publicly available?	Brian:3 Manav:3 Shanmukh:3	Yes
Do contributors keep the	Brian:3	Yes

copyright/IP of their contributions?	Manav:3 Shanmukh:3	
Does your website and documentation clearly state the copyright owners of your software and documentation?	Brian:3 Manav:3 Shanmukh:3	Yes
Does each of your source code files include a copyright statement?	Brian:3 Manav:3 Shanmukh:3	Yes
Does your website and documentation clearly state the licence of your software?	Brian:3 Manav:3 Shanmukh:3	Yes