

GSoc'2022 Proposal

PyAr

PyZombis

About me

Some Details

Name: Shivam Shandilya

Email: shivam.stpaulsdarjeeling@gmail.com

Github: <https://github.com/shivamshan>

University: [Birla Institute of Technology, Mesra](#)

Address: Ranchi, India

Time Zone: IST (GMT +5:30)

Linkedin: <in/shivam-shandilya>

Background

I am a third year undergraduate pursuing B.Tech. in Electrical and Electronics Engineering at Birla Institute of Technology, Mesra. I can describe myself as hard working and passionate towards things that inspire me. My interests are in the fields of coding, physics and machine learning in general. I love devoting my time to problems that can really make a change. I also love to learn new skills and tools whenever I get time.

Past Experience

I have been following the PyZombis project for quite some time now and also actively contributing to it by fixing some bugs, working on previous pull requests and also adding proof-of-concept of some ideas and enhancements. I have also worked with the

maintainers to update the wiki section to help new contributors with the initial installation process.

I have also contributed to some other projects under PyAr such as PyAfipws and PyEmpaq. My contributions to these projects can be viewed [here](#) (PyAfipws) and [here](#) (PyEmpaq). Apart from these I have contributed to projects like [MSS](#).

Apart from open-source projects I have written some [blogs](#) mainly related to machine learning, participated in many hackathons and programs like [30 days of Google Cloud](#) and I also have a keen interest in [Quantum Computing](#).

I am proficient in git and github and can also adapt to new tools and softwares quite quickly.

Projects done

- **TextronAI:** (*Team-project*) A deep-learning based speech-to-text app that leverages state-of-the-art open-source engine [mozilla/DeepSpeech](#) to transcribe system audio to text-based output in real time. The idea was to integrate our app in meeting apps(e.g. Microsoft Teams) and provide features such as recording meetings offline, transcribing audio to text, highlighting important discussions/timelines/deadlines etc. This project is still under-development.

Project [link](#).

Tech-stack: Python, ElectronJS, Flask, DeepSpeech.

- **Facial Expression Detection:** This project makes use of Convolutional Neural Networks to detect 7 basic facial expressions.

Project [link](#).

Tech-stack: Python, Flask, Tensorflow

- **Vitarana Drone:**(*Team-project*) This project was made as part of e-Yantra/e-YRC - an annual international robotics competition hosted by IIT Bombay. We were assigned the theme of Vitarana Drone(Distribution Drone) and the idea was to simulate a quadcopter drone that would pick parcels from the source destination and drop them to the target destination. Our team was able to successfully implement the whole theme and was among the top few teams to reach the

semi-finals.Tech-stacks: Python,Gazebo Simulator, OpenCV, Path planning and Control systems.

Contributions to PyZombis

PR	Description	Status
#221	E2E tests for TWP33 to detect API issues	Open
#208	Fixed failing proxy in TWP45 lectures	Open
#206	Fixed failing proxy in TWP35 lectures	Open
#204	Fixed failing proxy in TWP33 lectures	Merged
#202	Update runestone install repo link	Merged
#201	Pygame POC	Open
#200	Refactor TWP42: Bases de Datos using SQL.js	Open
#199	added external api to replace failing api	Open

List of issues opened by me can be checked from [here](#).

Proposal

Project Abstract

[PyZombis](#) is a community course to teach Python to the spanish-speaking community. It is based on a Brazilian MOOC [Python para Zumbis](#). The idea is to make an improved version of the course in Spanish, with an interactive environment for the students, that allows them to visualize and try active code on the site (made possible thanks to the Runestone library). As per last GSoC, Brython interpreter support was added to active code, so that students could now write and execute Brython code. I plan to use this to add and improve lectures for the MOOC. Previously PyZombis was using [Skulpt](#) as its Python's online interpreter. Skulpt did not provide support for sqlite3 and PyGame, two python libraries that are used to teach SQL basics and pygame basics respectively in PyZombis lectures. Therefore, currently the lectures are not interactive. I also plan to help provide alternative solutions that will allow the lectures to be taught in an interactive way.

Project Description

These are some of the milestones that I plan to focus on:

Interactive SQL lectures

Currently the SQL lectures([TWP42: Bases de datos](#)) are not interactive, the reason being PyZombis' current web interpreter, [Skulpt](#), which has blocked the implementation of complex exercises like this one due to of its limitations with some libraries, specifically **sqlite3** in this case.

My proposal to fix this issue is based on [PR #184](#), which proposes to use [sql.js](#) for this purpose with the help of DOM features. I plan to design a wrapper for sqlite3 which will mimic its functions along with sql.js to handle the actual sql queries. An initial pull request which serves as a proof-of-concept for this idea can be found here, [PR #200](#). I plan to build upon this idea and make the lectures as interactive as possible. The

wrapper that I plan to design can also be expanded in the future if more functionalities of sqlite3 are required for teaching the lectures.

Interactive PyGame lectures

The PyGame lectures([TWP60:Pygame](#)), same as the SQL lectures are also not interactive due to the same reasons. Since brython is currently available as an interpreter in PyZombis, for this issue I tried using an old brython-pygame fork, which can be found [here](#). This fork is no longer maintained by the owners and hence many features are missing or do not work. After fixing some bugs I managed to get a working [proof of concept](#) for this idea that I further plan to explore.

I also plan to explore [pyjsdl](#) and [gamejs](#) as a javascript wrapper for this purpose. Pyjsdl library has a more complete implementation of pygame but it needs some modification to make it brython compatible for the canvas drawing methods to work. Gamejs on the other hand can be used as a wrapper to implement the necessary functions and features of pygame required to teach the lectures.

Web development lectures using web2py

If time permits I also plan to make the [Web development intro lectures](#) interactive using [web2py](#) educational framework. This would serve as a capstone project for peer- review final course assignment. This would include a full revision of the lesson and a Kubernetes deployment script to provide a full web2py multi-tenant server where the students can log-in and develop/execute their web exercises.

Timeline

Period	Work to be done
May 20 - June 12 (Community bonding)	<ul style="list-style-type: none"> Familiarize myself with the requirements of SQL and PyGame lectures. Discuss possible and available solutions with mentors Experiment and define the final solution for SQLite and PyGame Work on old and incomplete pull requests
June 13 - June 27 (Week 1 - Week 2)	<ul style="list-style-type: none"> Close previous PRs related to SQL lectures Implement the basic skeleton for the wrapper with sql.js and brython Start implementation of sqlite wrapper
June 28 - July 11 (Week 3 - Week 4)	<ul style="list-style-type: none"> Complete the sqlite wrapper Start implementation of the sql lectures in accordance with the original plan Complete the implementation of the lectures Write E2E tests with playwright
July 12 - July 25 (Week 5 - Week 6)	<ul style="list-style-type: none"> Decide upon the best option to implement the lectures(pyjsdl, games, old brython fork) Start implementation of the wrapper to build the lectures upon Complete the wrapper with all the features to teach the lectures
July 26 - August 8 (Week 7 - Week 8)	<ul style="list-style-type: none"> Start implementation of Pygame lectures using the wrapper Complete implementation of the lectures Write E2E tests with playwright
August 9 - August 29 (Week 9 - Week 10)**	<ul style="list-style-type: none"> If time permits, start implementation the web development introduction lectures using web2py Complete making the web development lectures interactive Write E2E tests with playwright
August 30 - September 5 (Week 11 - Week 12)**	<ul style="list-style-type: none"> If time permits, continue with web2py implementation Complete and wrap up any remaining work Document all the changes made Time slot in case of any delay



September 6 - September 12 (Final Week)	<ul style="list-style-type: none">• Wrap up any incomplete work• Time slot in case of any delay
--	--

(**) - Work done during this time is subject to change and can be adjusted as per requirement

Commitments

I don't have any prior commitments apart from GSoC for this summer. My semester exams are going to end by the first week of May and college will reopen in August, hence I will have enough time to contribute to my GSoC project. Even after my college opens I can still contribute my time as in the end year the workload from college is light. If accepted, I can consistently put in 40-45 hours a week throughout the summer.

Post GSoC

I believe PyZombis has a very noble cause of teaching Python to students for free. I am and will be more than willing to give back whatever I learn from my experience during GSoC. Post GSoC, I would love to stay back and help with the project further- make meaningful contributions, help new contributors get started with PyZombis, stay engaged with the community and most probably apply as a mentor for next year's GSoC.