
Project Proposal

Terasology Launcher 4.0

Re-implementing of the Terasology launcher.

Azhar Ahamed

04-02-2019

Project Info

Project Title : Re-implementing of the Terasology launcher

Project short title: Terasology Launcher 4.0

URL of the project idea page:

<https://trello.com/c/D7NdkVUs/22-terasologylauncher-40>

Bio of Student

I am Azhar Ahamed a Computer Science undergraduate at Informatics Institute of Technology, Colombo, Sri Lanka. A creative front end developer with the mindset of a video game enthusiast.

My experience with front-end development begins with multiple online courses and many projects which were undertaken in order to contribute to the non-profit organizations, and to pass the undergraduate course modules with flying colors. I'm willing to learn and apply what I learnt. And most importantly help others with my knowledge. I'm currently leading a university group project which involves a hybrid system of convolutional neural network and natural language processing project following agile methodologies which is a web application solution for pre-owned product price evaluation. From these experiences I learned to build industry standard product feature by feature, testing each phase as it's built and to keep track of the progress by utilizing task management tools like Trello, GIT project, Microsoft Project. Which are I think utmost necessity to these kinds of task.

I'm really comfortable with web development languages like HTML5, CSS3, JavaScript, TypeScript, SQL, and PHP. Other than Front-end development my skills mainly lie in Java. Which had been my primary programming language for the past two years of my university life. I have also developed few full stack applications using java with frameworks such as Play, Spring as the backend and some front-end frameworks such as Angular, ReactJs. I'm also familiar with Python which is one of the robust languages. And also hardware technologies like Arduino and lot and also I'm an android programmer. All my projects can be found in [GitHub](#) repositories.

Outside programming, I am a graphic designer, video editor. I work on small projects whenever I get free time.

Contact Information

- Student Name – Azhar Ahamed
- Student Postal Address – 40/1A,
Chakindarama Road,
Ratmalana,
Western Province,
Sri Lanka
- Telephone – (+94)778804522
– (+94)112738586 (Home)
- Email – azhar.2017815@iit.ac.lk (Primary)
– zhranees@live.com (Secondary)

Student Affiliation

Institution: Informatics Institute of Technology
Program: Computer Science
Stage of completion: 2nd year of 4 years

Contact to verify:
Pradeepa Udakara
Registrar
Informatics Institute of Technology,
57, Ramakrishna Road, Colombo 06,
Sri Lanka
T: +94112360212

Schedule Conflicts

As my university offers a sandwich degree which means I have complete a 11 month compulsory internship starting from the first week of August. It will be a 40 hours/week internship. To compensate for that 3 weeks. I'm starting the coding three weeks earlier than the official coding time. It will not inter interfere the Community Bonding since usually apart from communicating with the mentors and organization, few other preparations are expected to be done during this time like setting up environments and familiarizing with code base. But for this task no such things are needed since the task involves creating a new launcher from scratch. And I have already familiarized with the current launcher. So as soon as Community Bonding starts I will be able to start the coding too.

Coding Plan and Methods

Introduction to Problem:

“The technology decision to use JavaFX for the launcher has brought up a couple of issues on the user site, partly relating to how JavaFX is shipped.”(Task Page)

Even though JavaFX is great to develop classic desktop applications and to develop business applications it's not so great when building fancy gaming client. JavaFX couldn't compete to the cutting edge design flexibility and animations of web frameworks such as Angular/React.

Apart from the Design, if we consider the deployment problems, problems related to differentiation in Java runtime environment. And should build a bundle for each and every target platform (Linux, Mac, Windows) also its heavy weight compared to web applications.

What we propose:

We propose to re-implement the launcher using angular 7 which is a great web application framework which uses Typescript, a superset of JavaScript. And which is maintained

by google and dozens of industrial apps which were built using it. (GitHub Community Forum, Delta, Even Google Summer of Code page).

And provide a desktop application using electron which is an open-source framework developed and maintained by GitHub which we will be using to build our desktop application.

Provide the essential features of a launcher with the existing features such as,

- Downloading the game
- Check for updates
- Check for the required files for the game to run
- To Launch the game
- Customize Launcher settings
- Customize Game settings
- Logging the changes
- Managing folders (Save games, Screenshots)

Features of the New Launcher

Features are tagged as,

1. **As is** – The features which are already implemented in the JavaFx launcher, features as it is or with minor changes
2. **Enhancement** – The features which are modified or enhanced.
3. **New** – These are the new features which I will be adding to the new launcher.

1. Downloading Game from the launcher.

Tag: **As is**

This feature will allow the launcher to download the game. User will be able to select the version he wishes to download, by default the launcher will download the latest version of the game prompting the user for a validation.

2. Checking for New Launcher update

Tag: **New**

This feature will check for new update every time the launcher starts. This will let the user be up to date with the game launcher version and up to date with the new features of the launcher.

3. Checking for New updates in the game

Tag: **As is**

This feature will check for game updates and notify the user on the current released version and the improvements on the latest version.

4. Checking for required files

Tag: **New**

This feature will check for the required file for the specific version of the game. It validates the local files with the selected version of the game which doesn't let the game to launch without all the required files.

5. Launching the game

Tag: **As is**

This feature will allow the user to start the game using the launcher which will only be available after the validation of files for that specific version.

6. Customizing the launcher settings

Tag: **Enhancement**

This feature will let user select from different set of settings for the launcher such as language selection, changing the download directory. Also give the user to customize the theme of the launcher from few of the predefined themes which will be included.

7. Customizing the Game settings

Tag: **Enhancement**

This feature will allow the user to change few of the game settings before even the game starts. Give a selection of the version the user wishes to play and re build features such as memory allocation, java arguments.

8. Logging the changes

Tag: **As is**

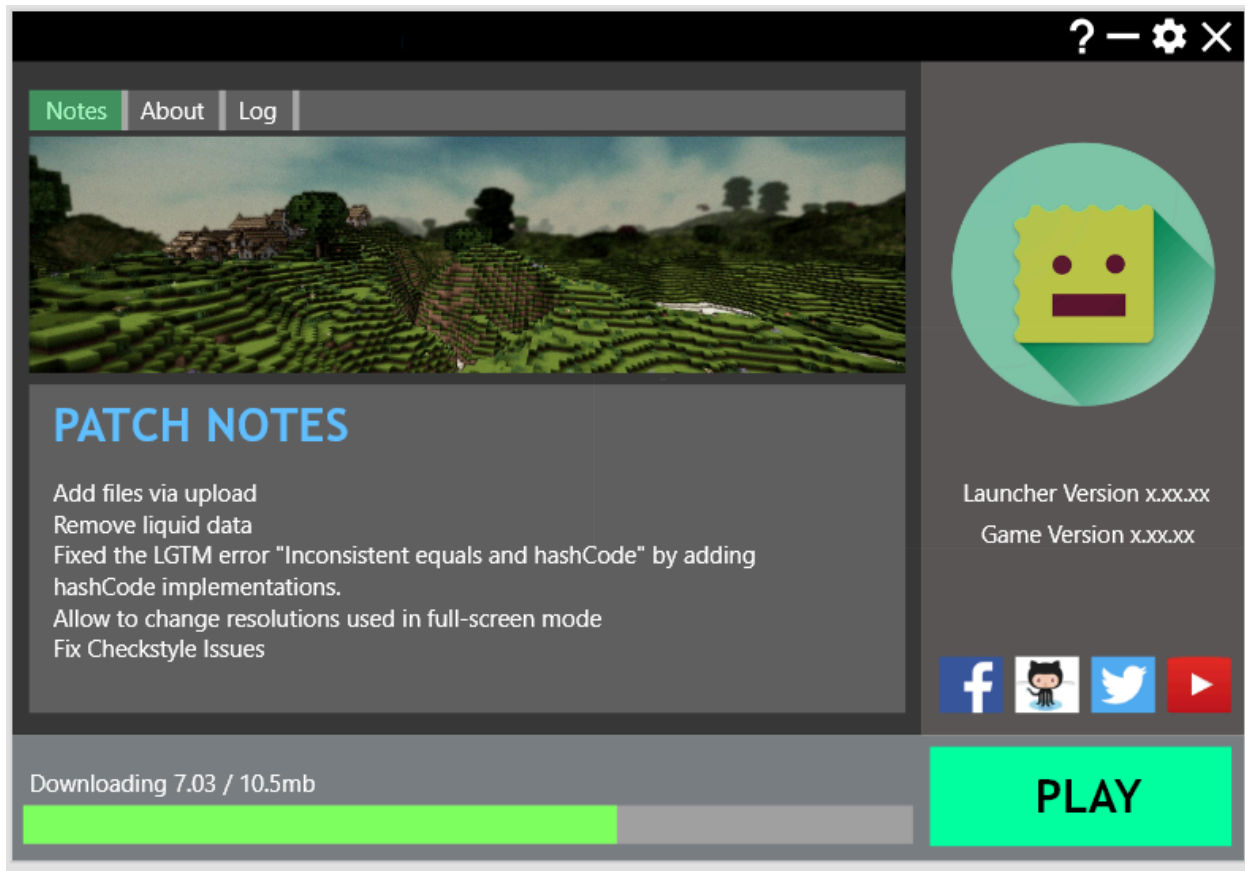
Provide user a predefined list to select from for the level of logging, log changes depending on the selected log level

9. Managing folders

Tag: **New**

Launcher will be managing all the folders where the saved games are, screenshots and it will be able to bind the version of the game with where it was installed giving the user more flexible options to install the game.

Wire Frame of the New Launcher



This is a basic wireframe of the launcher. Later we will look into more stylish and attractive design as we start developing.

Timeline

I have divided the timeline to smaller parts of weeks and each week has a specific objective. The week starts from Monday and ends on Sunday. Coding will be done in the weekday from Monday to Friday and testing, debugging and documentation of the component developed that week will be completed in the weekend. Each week this will be strictly followed to ensure the code meets the standards and documentations and vignettes are thoroughly documented. Apart from weekly objectives, each month will have a part of proposal as objective.

Month 01: Designing the Project and drawing the mockups.

Month 02: Implementing the design and the UI with components.

Month 03: Implementing the features.

Week 1, 2 [6 May - 15 May]

- Community Bonding, Getting to know about the community more.
- Designing the project, activity diagrams, state diagrams and wireframes.

Week 3, 4 [15 May - 31 May]

- Work on the Style Guide and get feedback from the community

-
- Prototyping the launcher views.
 - Defining the project structure
 - Finalizing and exporting the Prototype styles.

Week 5, 6 [1 June - 15 June]

- Identifying the reusable components and start building.
- Start implementing the static views.

Week 7, 8 [16 June – 30 June]

- Re-implementing the currently available features
- Plan the enhancements of the features which are currently available
- Enhance currently available features.

Week 9, 10 [1 July - 15 July]

- Strategize the new features which are to be added to the launcher.
- Implement the new features

Week 11, 12 [16 July - 31 July]

- Build the API for new features if possible
- Finalizing the features.

Week 13, 14 [1 August – 19 August]

- Re testing all the features and functions.
- Additional time allocation for unexpected delays and enhancements beyond proposal.

After the GSoC 209 coding period

- If I stuck somewhere in implementation, I will try to complete them after the GSoC period.
- I will continue my contribution and will be active in `Terasology` community. Also I will help new contributors.
- If possible, I want to contribute to other branches.