xrdesktop GSoC Proposal

- April 10th, 2021

Dev Pant

Github : Sigsev-Dev

Email : devpant.btech.cs19@iiitranchi.ac.in

Discord nickname : Dev_p #3543

Phone number : +91 6397118669

Time Zone : UTC + 05:30

Town, Country: Dehradun, India

Meeting with mentors

- Before 30th April : Reachable anytime between 5:30 am to 5:30 pm (UTC) [11:00 am to 11:00 pm IST] through Email/ Discord.

- After 30th April : Reachable anytime between 2:30 am to 8:30 pm (UTC) [8:00 am to 2:00 pm IST] through Email/ Discord.

Can join any pre-planned video session if required.

About Me

I am a Sophomore at I**ndian Institute of Information Technology, Ranchi. I am majoring in Computer Science Engineering.** My areas of interests include **Operating Systems**, **Creative Programming**, **Virtual Reality** and **Machine Learning**. I developed a passion for programming in Grade 10 (High school) when I started with a book called ‘Let us C’ which was my first experience with a coding language continued with the ‘Introduction to Algorithms’ and later developing interests in Competitive programming which gave me more proficiency with how to understand-code-debug.

I’ve been programming ever since, but became aware of open-source a few months back, with not a vast experience in version control systems but seeking as many opportunities as i can to explore my way through. I have contributed to houseofgeeks in the past which helped me through the basics about git and how to contribute and understand projects and databases.

Coding Skills :

- Programming Languages

    -Fluent in C, C++

    -Sound knowledge of OOP

-Development Environment

    -Vim Text Editor

    -Gnome

    -OS : Ubuntu 20.04 LTS , Windows 10

- Version Control

-Git

The Project

Implement a Virtual VR Keyboard

Current Status-

The OpenVR API that is supported by SteamVR is currently being used by xrdesktop's Client Class for text entry with no functionality of a direct equivalent keyboard concept built in OpenXR. So using OpenXR currently will mean giving up the keyboard for text entry. Also staying in OpenVR runtime, to support text entry, limits any further UI/UX improvement to be done by xrd regarding the virtual keyboard as it is majorly controlled by SteamVR and their patch updates.

Primary Goal-

The main goal will be to implement xrd's native api for virtual keyboard that will be an overlay window with a rectangular frame consisting of many trigger buttons (just like implementing a quit button but here it will pass on characters to the event poll which can be further processed and displayed in the output box) which can be selected by the user through a simple mouse/controller click over the particular character showed in the frame.

The key points here will be:

- Implementing all character buttons
  (available in a standard Qwerty keyboard)
- Integrating window manager for the frame
- A show overlay function
  (to be called by the user)
- An event poll
  (to process the input character)
- Input trigger

  (ext. hardware input)
- Replacing current api calls in steam vr

  +

  Integrating function calls for xrd's internal keyboard in runtime

Implementation Details-

The implementation will include three main layer :

- Creating an API for I/O hardware validation and accepting user entries

- Processing characters in event queues
- Re-raising the character to the display box and work on callback

A full combination of backend and frontend work would be to start from the user side through the runtime to our framework and back to display.(I'll create a user-centric application with a complete pipeline from UI/UX design to the corresponding backend frameworks. This is more clearly explained in the flowchart below.)

Process-

User calls a function *xrd.show_overlay*

The window manager provides a virtual 3D vr keyboard

User clicks on a character button

The output keypress goes into event queue with an event_poll function being called

The input character will be passed on to the window manager

using kwin

It will be linked to the output comment box in our keyboard frame

Implementation Strategy-

The main goals for the project are:

- ➢ To implement a complete working in-VR Keyboard
- ➢ Replace all the existing API calls for external libraries for successful in-VR keyboard integration
- ➢ Connect our own input with the libraries to window output.

Since most of the other work is already done in xrd's framework, mostly what we will be needing there, are going to be improvements only.

My project workflow would be-

- ➢ I'll start by writing codes for the all character buttons, integers and special inputs like reverse key,enter key,space bar e.t.c and adding .png files to support their images, integrate all *keyboard_press_event* signals with all the character buttons. I'll then add UI for sounds and sprites. Functions like changing key color on press.
- ➢ After I'm done with all the trigger buttons. I'll start implementing the UI Overlay for the keyboard frame that is with constraint positions and placements for all the keys, calling 3D *scene.objects* in *Gulkan* along with functions like *_kwinwindow_to_xrdwindow* and also integrating the *quit_window* functions. By this time I'll be done with a virtual keyboard frame ready to integrate.
- ➢ I'll then start by writing user end functions like *_create , _destroy, _show, _hide, _show_keyboard, _set_position e.t.c* and passing the

character pressed to the already existing pipelines where this keypress will produce a signal back to gxr containing that character. I'll integrate these calls with the event poll functions and then finally connect these event polls back to *keyboard press callback* functions in the window manager for the display.

- ➢ In the end when all this is done I'll start by adding improvements to the already existing libraries (wherever needed) and finishing up with a code cleanup before shifting on to other secondary goals I've planned for this project.

Secondary Goals-

In case the project idea is a full finish on an early note, and there is still time left to contribute further improvements to the API, I would like to work on following ideas :

- ❖ Improving UI/UX for the Virtual Keyboard
  - ➢ Integrating "auto-correct/ auto-suggest word" algorithms to improve typing speed.
  - ➢ The Levenshtein distance based auto correction helps the users to correct spelling mistakes, while the N-Gram model offers word predictions..
- ❖ Implementing a slide-over keyboard to suggest words (though this will be a tough call and might take some post Gsoc time too for implementing). This will enhance the typing speed for users to a very great extent. This feature is already being used in smartphone's touch typing.
- ❖ I have also noticed in a presentation given by Lubosz Sarnecki at FOSDEM 2020 that we at xrdesktop already have techniques like hand tracking and this will only add up a lot of project opportunities i can work on in future on the ways of user input after i build the base of a virtual keyboard in our framework with standard click inputs.

Any further eye-catching subject and need I learn whilst working on the project idea and which i might seem to be able to contribute to.

Proposed Deliverables-

(during GSoC)

- A full functioning in-vr keyboard
- Integrated ui/ux improvements for window and I/O installation.
- Detailed documentation; for users as well as developers
- Fortnightly blogs on development advances, milestones and problems incurred

Milestones-

There will be four major milestones.

(Although not making too rigid targets or timelines, which are most likely to be altered upon discussions with the mentors during the community bonding period and understanding the difficulties better and the changes that might be suggested later.)

Milestone 1-

(Deliverable before coding period begins)

I'll start by familiarizing myself with core xrd's functionality. Having an overall idea of the codebase will be necessary when I go ahead. Though I've already started using xrd and studying the codebase for how different things are implemented in it and trying to implement small issues like creating a *'quit button'* just to get a hand on using the database to improve the api.

Milestone 2-

(Deliverable before Phase 1 Evaluation)

I'll start by writing the basic functions in the .c .h files to create an input frame with all working keys input and *event queues,*to access the input and integrate it through the pipelines back to *libinputsynth (*or any better alternative). I have already started writing the draft functions for a basic trigger button. I'll also start writing/improving the UX for dev/uinput to access VR controllers.

Milestone 3-

(Deliverable before final Evaluation)

I'll have the input triggers using external hardware clicks , finalised and integrated function calls with xrd's framework that are to be used in runtime. I'll be leaving bare minimum work for this time period so as to adjust any delay or errors(improvement) if needed.

Milestone 4-

(Wishlist, if time permits/ Post Gsoc work)

In case I'm able to complete the first three milestones within time I'll move on to secondary goals, stated in the implementation details.

Detailed Timeline-

Present – April 20 (Homework period)	Get more familiar with xrd and learn other techs, that will help/will be needed, like Glib and Vulkan.

April 20 - April 30 **(Hiatus)**	End semester examinations will be near so I'll be inactive, but will be available for communication through Email/Discord
April 30 – May 17	Continue the homework period
May 17 – June 7	Community Bonding Period: Discuss with mentors any important points missed and plan the work ahead
Coding Period Starts:	**Milestone 1 reached**
June 7 – June 17	Write backends for all char. buttons
June 18 – June 27	Build the keyboard frame with ui
June 28 – July 8	Integrate all function calls with our framework and complete the headers.
July 9 – July 12	Do a general code cleanup done till now and start integrating functions for external input.
1st evaluation	**Milestone 2 reached**
July 13 – July 23	Complete integrating all functions and linking external uinput to event queues.

July 24 – August 5	Work on integrating the event queues with callbacks to kwin, writing function calls on the window manager.
	Milestone 3 reached
August 6 – August 16	Do a general code cleanup. Make sure there is nothing left undone and everything is tidy. Prepare for Final Evaluation.
Final Evaluation	
After August 16	Keep contributing by completing leftover secondary goals, and extending multiple ui/ux backends as necessary.

Why me for the project?

I've been studying about window management for a while and have developed a great interest in Virtual Reality recently. This project seems to be a great start for all the things I desire to explore. I have been programming in C/C++ for almost 3-4 years by now and I'm fairly acquainted with the specifics of OOP and Software Development Techniques. I've been a Linux user for almost 2 years now and have struggled with the lack of some basic functionalities which were common in other OS.

This project caught my eye at once around mid march when I finally decided this has to be the one I'll be contributing to. I am fairly new at many techs

that I will be needing but I have planned to learn all of them before the coding period starts. I'm a quick learner and have learnt quite a bit while exploring/researching about the project along with building & understanding xrd environment on my device. I'm confident I'll be able to polish my skills, and will be able to contribute even more further ahead.

I can easily devote around 50 hours per week during my timeline. I believe that the allotted work per week is completely doable for me and is neither overloaded nor slacked. This makes me eligible to apply for the Virtual Keyboard Project. I'll also try to maximise the contributions I can give to xrd even after I complete my base GSoC project and be an asset to the organisation.

Other Commitments

I have my end term examinations between 20 April - 30 April. So I'll be inactive for that time period which is way before the official coding period. My vacations are tentative to start in the first week of May and end on August start, and the official GSoC period is from 17 May to 23 August . I can easily devote around 56-63 hours a week until my college reopens and 35-42 hours per week after that. I plan to complete most of the work before my college reopens. During the vacations I have no other commitments and shall keep my status posted to all the community members.

References-

- https://gitlab.freedesktop.org/xrdesktop/xrdesktop
- https://gitlab.freedesktop.org/xrdesktop/gxr/-/tree/78828bf3dfcecef9aceb349e0919a08958b35421/src/openvr

- https://github.com/ValveSoftware/openvr
- https://github.com/rjth/Punchkeyboard
- https://github.com/galaktor/gostwriter