

POCKETPAD

CONTROLLER EMULATION PROJECT

PROJECT TEAM 8

- Benjamin Dravis – Computer Science major at Purdue University
- Jack Fang – Computer Science major at Purdue University
- Krish Shah – Computer Science major at Purdue University
- James Burrows – Computer Science major at Purdue University
- Bautista Tedin – Computer Science major at Purdue University
- Joshua Irizarry – Computer Science major at Purdue University

PROBLEM STATEMENT

Our project will develop an iOS application that will allow users to emulate different types of gaming controllers with the portability and accessibility of their phone/iPads. This application will allow users to connect an emulated controller to their computers over network or Bluetooth, which will allow them to play games through services such as Dolphin. A user of this application would no longer need to rely on expensive emulation services or physically have different types of controllers to play certain games, but rather, they could use our application to emulate several different types of controllers with a singular iOS application to play a wide range of games. While similar applications may exist, our mobile application will be provided for free and support far more controller types.

BACKGROUND INFORMATION

AUDIENCE OF THE PRODUCT:

Controller emulation is a process by which different types of controllers can be used for games that they initially were not intended for. This allows people within the gaming community to be able to play some outdated or even newer titles in the style in which they want to and prefer to play them. Gamers have been doing this using any number of formats for years now, which has allowed them to continue to play the games that they love the way that they love. As such, a controller emulation project such as this will be targeted and intended for the gaming community allowing them to do this easier and cheaper than they have ever been able to do in the past.

HISTORY OF SIMILAR TECHNOLOGY:

In the past, controller emulation has primarily been done by connecting different controllers to different devices, such connecting an Xbox 360 controller to a play station. Recently a lot of controllers have the capability to connect to their intended device via a wired or wireless connection. PlayStation for example allows users to connect their controllers through a wired connection to the PlayStation using a USB cord plugging into the controller and into the PlayStation, but they also allow for the connection over Bluetooth to the PlayStation.

As such, this connection can be extended to other devices other than their original intention because it is just a simple wired or Bluetooth connection which most modern devices are well equipped for. One can plug in the USB connector to their intended device to allow the user to connect their controllers to it, or they can use Bluetooth connections. Some platforms and games allow for this to be done seamlessly with easy connection of different controllers for playing games, which will allow for users to use the connection as is with the game that they are intending to play.

However, some devices and games do not allow for that connection to be done so seamlessly because some devices or controllers, especially older ones, don't have the ability to connect wirelessly through Bluetooth or the type of physical connection that most modern technology requires, so they are reliant upon outside software to work as intended. Emulation services such as Dolphin allow users to reconfigure their controller's inputs to match the inputs or types of inputs necessary for the user's chosen game. Software such as this has provided users with a wider range of games that they could play and emulate due to this feature. However, depending on the type or age of the controller or connection device, users may not even have the capabilities to connect their emulated controllers to play their games because of the lack of a Bluetooth connection or reliant upon a wired connection. Controllers would not only need to be purchased for connection, but also adapters would become necessary to simply connect the controller to a given device.

LIMITATIONS OF OTHER TECHNOLOGIES AND OUR SOLUTIONS:

Despite past technologies or the different implementations, our controller emulator will provide a novel way for users to play their games using emulated controllers:

- Controller emulators in the past were reliant upon physically having a given type of controller to play games with it. However, that can be expensive depending on the type of controller that is gotten as well as the number of controllers that a person gets. For instance, the current cost of a GameCube controller is sitting around \$20. The current cost of a PS5 controller is around \$80, and the current cost of an Xbox 360 remote is around \$30-\$40. Given this, it can become very expensive to play with a user's preferred controller, especially if you still need to purchase an adapter for the already expensive controller, but our product will make this a lot more affordable and easily accessible to users. The developed iOS application will allow users to emulate those different types of controllers without the price tag of the \$20-\$80 that the user would have to spend otherwise to play their chosen games whilst having access to a library of types of controllers that they can emulate from the comfort of their iOS device.
- Controller emulators have been reliant on outside technology and software such as adapters or services such as Dolphin to meet their intended purpose. They have been used to connect the controller to the intended device or needed to remap the controller's inputs so that it works, but our product will not be as reliant upon that because of our application's ability to connect via network or Bluetooth regardless of the user's chosen type of controller in the application. Our application, at first, will still be reliant upon free services such as Dolphin to fully emulate the controller, but there is a lot of room for growth and development within the functionality and capabilities with our application that can be made in the future.

Our application will make controller emulation more affordable and easier for users through the aforementioned functionalities we intend to include, which will allow for a better solution to something the gaming community has been doing for a while. Our application will improve upon the functionality as well as

include a lot of quality-of-life features, such as controller customization, that other current implementations and technologies for controller emulation don't allow for or have.

FUNCTIONAL REQUIREMENTS

- As a user of the application, I would like to have a tutorial for the application so that I can have a guide for general understanding of how to use the application and its numerous different functionalities properly.
- As a user of the application, I would like to be able to have an emulated a movement joystick like that of the left stick of common Xbox and PlayStation controllers so that I can control the movement of my characters in the different games that I am playing.
- As a user of the application, I would like to be able to have an emulated perspective joystick like that of the right stick of common Xbox and PlayStation controllers so that I can control the perspective of my characters in the different games that I am playing.
- As a user of the application, I would like to be able to have an emulated directional pad so that I can use the directional movement functionality of the D-Pad needed for the menus in some newer games and game functionality in most older games.
- As a user of the application, I would like to be able to have emulated bumpers so that I can use the functionality that games I like to play have tied to the left and right bumpers on most common controllers such as blocking or displaying emoticons.
- As a user of the application, I would like to be able to emulate the diamond of buttons so that I can interact with the different menus and activities such as going through dialog, attacks, option selection, and more that are commonly bound to those buttons and are necessary to play the games.
- As a user of the application, I would like to be able to have emulated triggers so that I can use the functionality that games such as shooter or adventure games commonly have bound to those buttons for interaction with certain game mechanics necessary to play certain games.
- As a user of the application, I would like to be able to emulate a settings button on controllers so that I can access the settings on certain games that I am playing in order customize my in-game settings whilst playing games.
- As a user of the application, I would like my emulated controller to include an option for motion controller so that I can run games such as Mario Kart which can be played with or require using motion control features rather than buttons or joysticks. (i.e. rotating the phone to steer).
- As a user of the application, I would like my emulated controller to have vibrational haptics so that I can access and options for the immersive tools that exist in the games that I am playing as well as play certain games or minigames such as those in Mario Party that rely on vibrational haptics to function properly.
- As a user of the application, I would like my emulated controller to have sound haptics so that I can access and options for the immersive tools that exist for setting the environment or style of the game that I am playing as well as play certain games or minigames that make use of the sound haptics to function properly.
- As a user of the application, I would like to be able to construct custom controllers using a custom and modular layout of buttons, bumpers, joysticks and more so that I can create and experiment with my own unique style of controllers when playing games.

- As a user of the application, I would like to have an accessible controller file within the application to emulate an Xbox controller so that I don't have to spend the time creating my own custom version of an already pre-existing controller for easy access to it.
- As a user of the application, I would like to have an accessible controller file within the application to emulate a PlayStation controller so that I don't have to spend the time creating my own custom version of an already pre-existing controller for easy access to it.
- As a user of the application, I would like to have an accessible controller file within the application to emulate a GameCube controller so that I don't have to spend the time creating my own custom version of an already pre-existing controller for easy access to it.
- As a user of the application, I would like to have an accessible controller file within the application to emulate a Nintendo switch controller so that I don't have to spend the time creating my own custom version of an already pre-existing controller for easy access to it.
- As a user of the application, I would like to be able to access a library of the commonly used controller files that are on the market as well as the custom controllers that I make myself so that I don't have to manually remake them each time I want to switch between the types of controllers I want to use and simply be able to save and load them as I want.
- As a user of the application, I would like to be able to make some modification to the types of controls so that I can use my preferred style of controller such as split versus conjoined D-Pad with my chosen controller.
- As a user of the application, I would like to be able to customize the color layout of any given so that I can apply my own style and color preferences to at a given time to the controller that I am using.
- As a user of the application, I would like to be able to customize the name of my controller or device when I connect so that I can clearly identify which controller is connected to me and is accepting my inputs.
- As a user of the application, I would like to have a settings menu in the application so that the different options for customization and functionality are easily accessible at any given time as well as not taking up space to ruin the layout and functionality of the emulated controller.
- As a user of the application, I would like a selection option to be able to connect to my chosen device via network so that I can play my chosen games with more people than I would be able to with connections such as Bluetooth.
- As a user of the application, I would like a selection option to be able to connect to my chosen device via Bluetooth so that I can play my chosen games with higher connection speed, allowing the controller inputs to run faster than it would with other connections such as network connections.
- As a user of the application, I would like to be able to see the eligible devices to connect to in the area in a nice menu so that I can select which specific device that I would like to connect to.
- As a user of the application, I would like to be able to disconnect from what would be then the connected device so that I can stop gaming at any time or potentially switch between different devices if there is another game that I would like to play somewhere else.
- As a user of the application, I would like to be able to pull up a GUI on the connected device so that I know that I can see the processes and what is happening on the server side of the connection as well.
- As a user of the application, I would like there to be a portion of the GUI that acts as a connection list so that I can see which devices, if any, are connected to the given server at any moment.

- As a user of the application, I would like to be able to see a mock-up of my controller on the GUI so that I can see that it is properly recognizing my controller and the customizations that I potentially put into it beyond the act of simply connecting my phone.
- As a user of the application, I would like the mock-up of the controller to highlight the button, joystick, feature, or whatever input that the user is putting into their emulated controller displayed on the GUI so that I can see that my controller is properly sending inputs to the device that it is connected to before playing a game.
- As a user of the application, I would like a clear way to differentiate between different controllers and inputs on the GUI so that if I am using the application to play with a group of my friends, we won't get confused by which person's controller corresponds to which when playing our games.
- As a user of the application, I would like to be able to customize the GUI's color scheme so that I can make the application personal and to my preferences.
- As a user of the application, I would like to be able to access a settings menu in the computer GUI so that I can toggle wanted or unwanted features from the GUI display to meet my wants/needs at a given time provided by the application regardless of the type of phone/device that I have.
- As a user of the application, I would like to be able to import and export controller configurations so that I can share my custom layouts with friends or use their presets.
- As a user of the application, I would like to adjust the sensitivity of my emulated analog inputs so that I can fine-tune the speed and responsiveness of movement based on those preferences.
- As a user of the application, I would like a way to fine-tune dead zones on my emulated analog inputs so that I can eliminate unintended movement or input drift when I am trying to use the controller.
- As a user of the application, I would like the application to ask me to automatically switch between different layouts I manually set depending on what game I am playing so that I won't have to go through the process of switching it each time I am wanting to play a different game especially if I intend on playing multiple games in a single session.
- As a user of the application, I would like the ability to emulate turbo button functionality so that I can automate repeated button presses in certain games.
- As a user of the application, I would like to enable macros for specific button combinations so that I can execute complex in-game actions with a single press.
- As a user of the application, I would like a visual indicator for connection strength displayed alongside the connection list in the QT GUI when using Bluetooth or network mode so that I can troubleshoot input lag or disconnections more easily.
- As a user of the application, I would like a help menu that will provides quick access to more in-depth troubleshooting guides and FAQs so that I can quickly resolve any issues or questions I may have when using the application.
- As a user of the application, I would like an option to use my phone as a remote mouse/keyboard hybrid in addition to a controller so that I can navigate game menus or settings that require text input.
- (If time allows) As a user of the application, I would like to have the option to see an overlay over my games so that I can see my inputs on the screen while playing.
- (If time allows) As a user of the application, I would like an option to set up multi-touch gestures to trigger specific in-game actions, allowing for more advanced inputs beyond standard buttons.
- (If time allows) As a user of the application, I would like to be able to sync my controller settings across multiple devices using a cloud backup so that I can access my custom layouts anywhere.

- (If time allows) As a user of the application, I would like the application to be android compatible as well as iOS compatible so that I or users like me would be able to use the services and features.

NON-FUNCTIONAL REQUIREMENTS

ARCHITECTURE AND PERFORMANCE

We plan to develop the application with a distinct separation between the mobile client, handled within our iOS application, and the computer server, which users will host locally on their devices. This structure will allow us to effectively divide our work while minimizing compatibility issues between the client and server. Communication between the two will be facilitated using sockets, ensuring seamless data transfer even when there are multiple clients connected. We will implement a TCP-based socket connection to enable fast, real-time communication between the client and server, while securing the connection with TLS to protect user data.

Our iOS mobile application and its client will be built using SwiftUI utilizing its features for developing both a user-friendly application and client to connect with the server. For Bluetooth connectivity, we will use CoreBluetooth, Apple's built-in framework for managing Bluetooth connections. Network communication will be handled with the Network library. Using Swift will also allow us to easier integrate the networking features that our application requires as well as allowing us to use some built in iOS features to make the best product possible in terms of service and function.

The server will be written in Python with the socket library built in. It both reads and transmits data sent from our swift client to allow the user to interact with the game that they are trying to play as well as interact with the computer GUI. Like the server, our GUI will also be written in Python with the QT framework, which will allow us to build a robust interface with all the functionality that is needed to ensure good functionality and quality-of-life. Building it using this framework will allow us to implement the robust functionality mentioned in our functional requirements.

SCALABILITY

This application will be very scalable in nature depending on the resources of the people using it. Our final product will include the availability of both network and Bluetooth connections. This diverse functionality would be able to handle different numbers of people at any given time. Connection via Bluetooth would have a maximum number of people set to 7 people joining due to preset limitations with Bluetooth connections. Network connections on the other hand would, in theory, be able to handle thousands of people at any given time securely transmitting and receiving data, which is a whole lot more than would ever reasonably be able to connect because this project will be run and hosted locally on a person's personal computer. We plan on ensuring good and secure functionality with the connection allowing for developed scalability within our product.

Going into the future, there is a whole lot of room for growth and development within the project. Initially our intention will be to use our application in tandem with other software such as Dolphin to fully emulate the controller. So, like the current technology implementation it would still be reliant upon outside technology. However, this necessity could be removed in the future given the right investment in time and energy to the project. This would make the application be able to be self-sufficient for its intended function to be used in tandem with games being able to regulate the emulation for different games beyond using the server to send

and receive responses, and with this functionality, it could be further developed to be used in tandem with software such as Steam allowing for an even wider variety of games accessible to the user through the use of our application. Depending on the time and effort devoted to this, it could grow in any way the developers see fit or could be useful to the gaming community.

USABILITY

Both the iOS application on the iOS device and the Qt PySide6 interface on the computer should be very user friendly and easy to navigate. With all the different features of our final product, it can become unnavigable if we are not careful about what we are doing, so a lot of what we are doing during its development is to ensure the usability of the program. To ensure that we retain good usability, we are going to be careful with how we structure the different features of our applications, which is a huge part of the reason that we are using SwiftUI for this. SwiftUI has a lot of built in functionality allowing us to ensure that our program will work across different sized devices as well as more built-in user-friendly features as well as allowing us to update our computer interface every 10ms for minimal input delay from the iOS application to the QT interface. We also plan on making our application using clean and concise menus as the way navigate to and use different functionalities, and the applications will also have the ability to toggle on and off certain features from those same menus which will help us to ensure that usability of user-friendly nature our team seeks to achieve in our product. Also, we are going to be implementing an in-app tutorial to help to explain a lot of the different features and functionalities of the project as well as general knowledge about how the application can be used. We believe that all these different functionalities will help to ensure our programs' usability.

HOSTING/DEPLOYMENT

The application will be hosted in two different ways with the Swift application and pyQT interface being hosted separately. The entire codebase will initially be stored entirely within its own GitHub repository that users can access due to our desire for this to be a free and accessible tool of the gaming community. Using our codebase, the users can then go use our pyQT interface and server locally on their own device so that they or their friends can easily connect using the iOS application that would run and exist on their chosen iOS device. It would allow the users to use their own personal devices/technology in order to run our free application at their convenience.