



una

CS3216 Final Report

Group 3

Soon Chun Mun | Le Viet Tien | Soedarsono | Tay Yang Shun

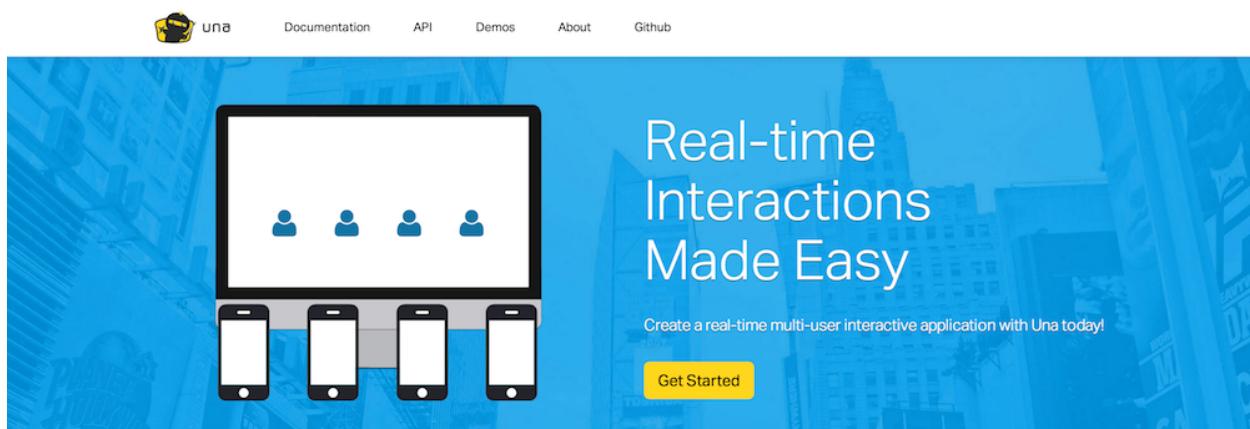
Description of the application you have developed.

Una

Product URL: <http://una-org.github.io>

Una is a free and open source JavaScript library that allows developers to create interactive applications where multiple users can interact with large central displays in real-time using mobile devices. We have developed a product site for interested developers to find out more about Una, try it out for themselves and potentially contribute to the development of **Una**!

Una Library Site Landing Page



What is Una?

Una is a **free and open source** Javascript library that allows developers to create interactive applications where multiple users can interact with large central displays in real-time using mobile devices.

Why use Una?



CROSS-DISPLAY INTERACTION

Una makes it easy for you to create interactive cross-display applications where users can interact with large displays using their mobile devices.

BUILT FOR THE WEB

Entirely web-based, no need to download any application from the app store; your web browser is the only app you need. No special hardware needed.

EASY INTEGRATION

Una handles the server-side for you, you only need to work on client-side code. This frees you up from working with low-level transport mechanism so you can focus on making your kick-ass applications.

What's Next?

Easy, [get started](#) with your new Una project, read up on the [API Reference](#) or support us on [github](#).

[Get Started](#)

© 2013 Una Project.

Una Library Demo Page



Una

Documentation

API

Demos

About

Github

Demos

The following apps are built on top of Una and demonstrates the ease of using Una.

Nutty Ninjas



[Nutty Ninjas](#) is a real-time shooter that brings the experience of social multiplayer gaming to a whole new level; it is a console-style game that can be played anywhere and anytime, simply with your computer and mobile devices. Multiple players can join a common gameplay screen just by using their mobile devices, and control their ninja character to unleash dangerous weapons at fellow players!

Shakeoff!



Support your favourite brand in this epic showdown of brands! Visit <http://shake.ltien.com> on your mobile device and choose your preferred brand to root for, then start shaking away!

© 2013 Una Project.

Using Una, we have developed 2 showcase applications - **Nutty Ninjas** and **Shakeoff**.

Nutty Ninjas

Product URL: <http://nuttyninjas.com>

Nutty Ninjas is a real-time shooter that brings the experience of social multiplayer gaming to a whole new level. **Nutty Ninjas** is a console-style game that can be played anywhere and anytime, simply with the user's computer and mobile devices. Multiple players can join a common gameplay screen just by using their mobile devices, and control their ninja character to unleash dangerous weapons at fellow players!

Game Concept

- Multiplayer top-down shooter
- Ninja themed
- Round-based deathmatch

Target Audiences

- Casual gamers, in a small social settings
- Group size of 4-6 players
- Age range of 18-24

Game Objectives

- Get as many kills as possible
- Destroy the boss in every stage

Game Flow

- User can input a room number on his mobile device's screen and join a game with that specified number.
- User will get to input his name, choose a ninja model, and start playing.
- Upon validation of his choices, the screen will display a controller and the user's controllable ninja will be added into the arena.

Game Landing Screen



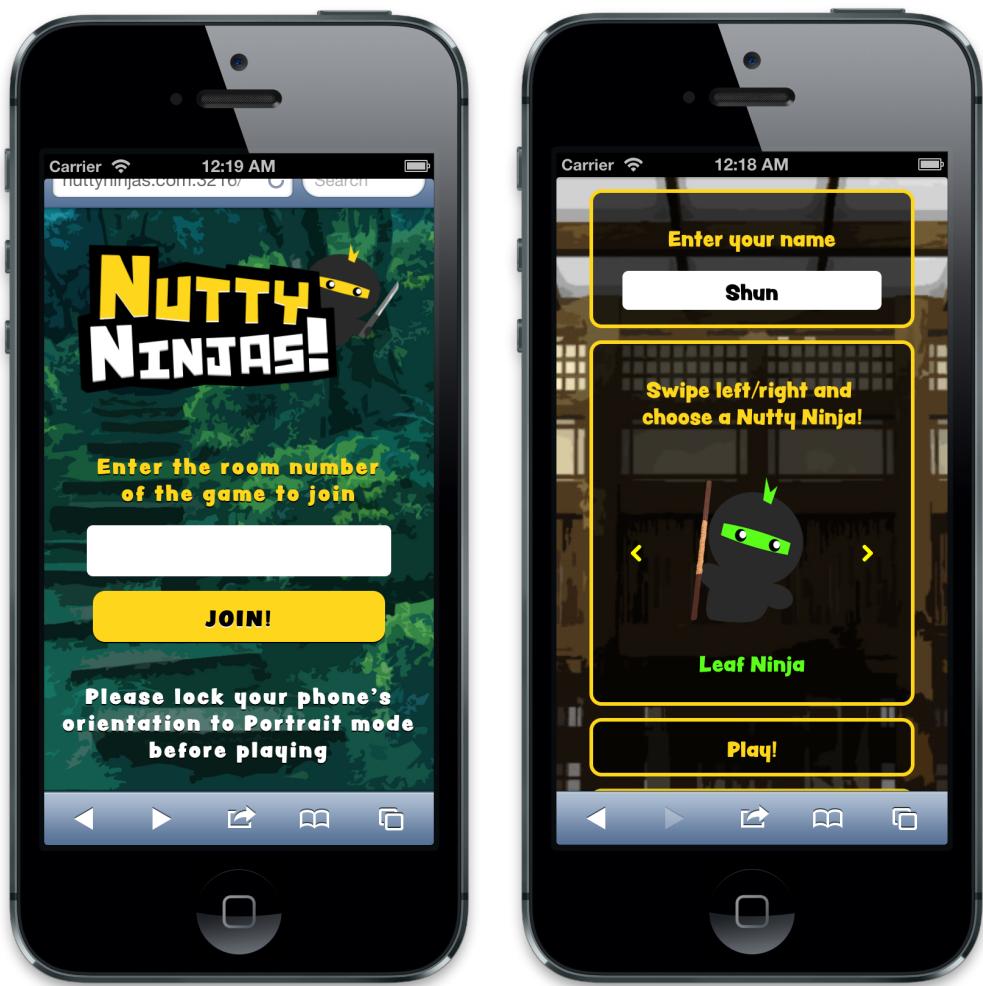
Gameplay Arena



Gameplay Arena 2



Controller



Shakeoff

Product URLs:

- Support Page(view on mobile): <http://shake.lvtien.com/>
- Results Page(view on desktop): <http://shake.lvtien.com/results>

Shakeoff demonstrates the advertising potential of Una; users can choose the brand that they like and start shaking to increase the popularity counter for that brand.

Results Page (Apple is winning, obviously)



Shaking Page (Let's support Apple)



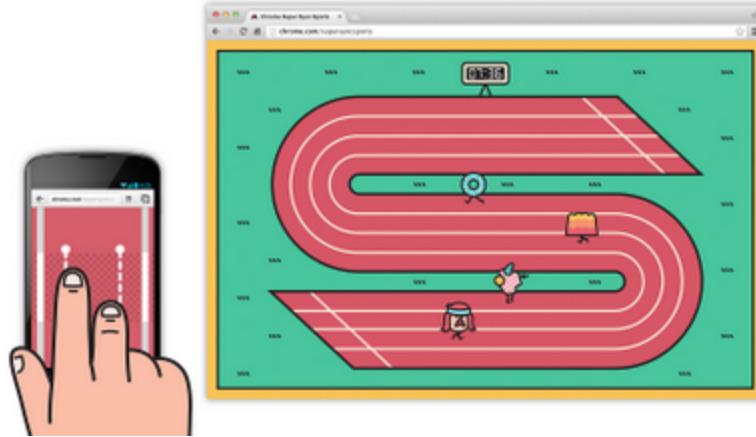
Are there any existing applications out there that are similar? What makes your application special?

There are in fact a few applications that adopt this style of gameplay. They are:

Chrome Super Sync Sports

Chrome Super Sync Sports is a web application that lets you sync your smartphone or tablet to your computer to play. It's easy, you just let your fingers do the running, cycling and swimming.

Pick a sport, connect your mobile device to your computer, choose from 50 athletes and if you're sitting with friends, invite up to three of them to join your race using their mobile device and your computer screen.



Chrome Maze

Turn your favourite site into a 3D maze, then find your way out using your mobile device as a controller and harnesses on its accelerometer. Tilt the device to move the ball.



Huntville

Huntville is a strategic slingshot game. Connect up to 4 iPhones and share in the fun! Multiplayer mode using your iPhone as a SlingShot!



However, these products are mostly experimental in nature, and are intended as technical demonstration of what is possible using Chrome web browsers. In a way, we wish to extend on these existing work, and we believe developers would be able to create products that leverages the similar technologies as these applications, but also provide a more engaging and meaningful gaming experience as compared to these demonstration applications.

Una is special because we place emphasis on player-to-player interaction between people while playing the game. Possible game systems include two-player cooperative games that requires live communication between players to meet the goals of the game. For example, in **Nutty Ninjas**, players can cooperate against the boss fire monster and kill it together for rewards. We try to bring out the fun and engaging elements of multiplayer games and enhance them to a whole new higher level.

We chose to make **Una** because we are all avid gamers ourselves and thoroughly enjoy the experience of console multiplayer games. However, it is tough for us to gather to play multiplayer console games nowadays because each individual's schedule is different and it is not feasible to bring a huge console

around with us to play. We want to bring the console gaming experience with us wherever we go. Una brings us (and other developers) one step closer to fulfilling this vision.

Review of milestone and timeline for project (which ones did you hit, what ones did you miss?).

Phase	Period	Details
1	1 week	Idea validation
	30/9 - 6/10	Research technology Prototyping
	Achievement	<i>Main technology stack finalized</i> <i>High level design completion</i>
2	2 weeks	“The basic game” - First iteration
	7/10 - 13/10	Network layer: <ul style="list-style-type: none"> • Setup server • Design APIs for resources: game, room, controller, screen • Implement controller, screen resources Game: <ul style="list-style-type: none"> • Concept + story • Basic graphical assets • Basic implementation
	14/10 - 20/10	Network layer: <ul style="list-style-type: none"> • Implement room resource • Implement user resource (join/create/search rooms) Game: <ul style="list-style-type: none"> • Combine with network layer • Display screen separated from controllers
	Achievement	<i>Network layer v0.1</i> <i>Game v0.1</i>
3	2 weeks	“Speed matters” - Second iteration
	21/10 - 27/10	Network layer: <ul style="list-style-type: none"> • Handle synchronization issue between client/server or between clients • General improvement Game: <ul style="list-style-type: none"> • Support multiplayers • General improvement

	28/10 - 3/11	<p>Network layer:</p> <ul style="list-style-type: none"> • Optimize for speed/real-time factor • General improvement <p>Game:</p> <ul style="list-style-type: none"> • Advanced graphical assets • Optimize for speed/real-time factor • General improvement
	<i>Achievement</i>	<i>Network layer v0.2</i> <i>Game v0.2</i>
4	1 week	“To the new world” - Third iteration
	4/11 - 10/11	<p>Network layer:</p> <ul style="list-style-type: none"> • Implement game resource • Abstract away common set of signals to an interface that other game libraries can follow <p>Game:</p> <ul style="list-style-type: none"> • Decouple from network layer • Modify to conform the new interface
	<i>Achievement</i>	<i>Network layer v0.3</i> <i>Game v0.3</i>
5	1 week	Finetuning
	11/11 - 17/11	Final user testing Bug fixing + code refactoring Documentation + final report Promotion + advertisement
	<i>Achievement</i>	<i>Ready for showcase!</i>

We hit every of our major milestones and even exceeded our own expectations by:

1. Abstracted out the **Una** library and open sourcing the code on Github:
<https://github.com/una-org/una>
2. Created a marketing page for **Una**: <http://una-org.github.io>
3. Built an additional non-gaming app, **Shakeoff** to demonstrate the advertising potential of Una.

We managed the project well, by sacrificing other modules and spending almost every weekend gathering in school together to increase the awesomeness of **Nutty Ninjas** and **Una**.

Individual contribution and roles. Acknowledgement of resources/help provided by external parties.

	Una	Nutty Ninjas
Soedarsono	<u>Overall Lead</u> <ul style="list-style-type: none"> • Implemented Una library • Library architecture design • Una marketing content • Liaising with external partners 	<u>Platform Ninja</u> <ul style="list-style-type: none"> • Integrated Nutty Ninjas game with Una • Network tester • Contingency I/C
Tay Yang Shun	<u>Product Lead</u> <ul style="list-style-type: none"> • Main product site design and implementation (HTML5/JS/CSS) • Una assets design • Una marketing content (posters, t-shirts) 	<u>UI/UX Ninja</u> <ul style="list-style-type: none"> • Gameplay design • Interface design and development • Controller implementation (HTML5/JS/CSS) • Game assets design • Marketing content (posters, design)
Viet Tien	<u>Server Lead</u> <ul style="list-style-type: none"> • Deployment to AWS instances • Server maintenance • Load tester • Backend support 	<u>Hacker Ninja</u> <ul style="list-style-type: none"> • Core game developer (JS) • Game architecture design and implementation • Character model design and implementation • Gameplay implementation • Quality assurance • Stress tester
Chun Mun	<u>Technology Lead</u> <ul style="list-style-type: none"> • Implemented prototypes to test the limits of HTML5 technology on mobile and socket.io • Library architecture design 	<u>All-star Ninja</u> <ul style="list-style-type: none"> • Core game developer (JS) • Game terrain/arena design and implementation • Gameplay implementation

Additionally, we would like to thank:

Ivan and **Abhishek**, our mentors from **LambdaMu Games** for their valuable insights and advice in deciding the product direction and suggestions on improvements for **Nutty Ninjas**.

Mr Tay Yang Heng for creating sprite assets (Ninjas, Projectiles, Cannons and Monsters) that were used in the game.

Application design (e.g. database schema, UML, etc, no codes please).

Una Architectures

Una provides developers with the following:

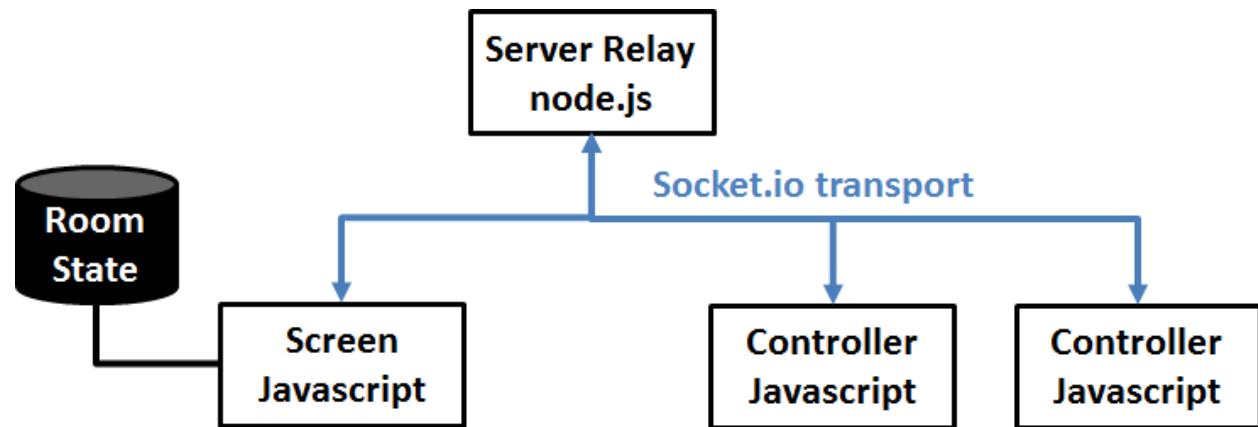
- Message passing server, written in node.js
- Client-side libraries, one for the screen and one for the controller, written in front-end Javascript.

Developers will designate Screen and Controller pages (or routes), using the screen and controller library respectively. The library allows the Screen to communicate with the Controller via the Server, and vice versa.

Una also supports the idea of a room, which is a collection of clients that can communicate with each other. Clients are only allowed to communicate with other clients in the same room, and this allows us to create independent instances of clients with its own states.

Una currently allows 2 different configuration, the default mode, and the server mode. The mode differs in where the state of the room is being maintained and modified.

Una Default Mode



In the **Una** default mode, the Server acts as a relay, transporting messages between various controllers and **one** screen. The Screen will keep track of the room state, and all interactions by any Controller will be relayed directly to the Screen. The Server does not maintain any state at all.

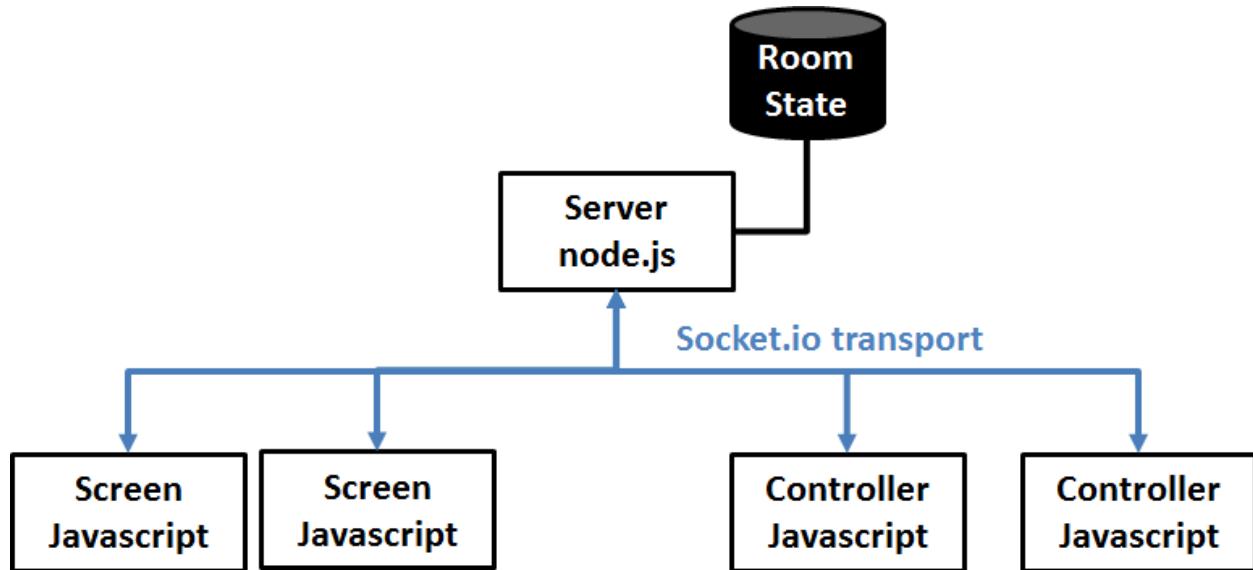
This mode is useful if there are a lot of processing needed to modify the room state. For a game, there could be many calculations needed in order to determine the next state of the game. By moving the responsibility of the state to the Screen on the client side, there will be minimal processing on the

server.

Unfortunately, this means that we are unable to support multiple concurrent screens, as they will not be able to be synchronized.

This is the mode used by **Nutty Ninjas**.

Una Server Mode

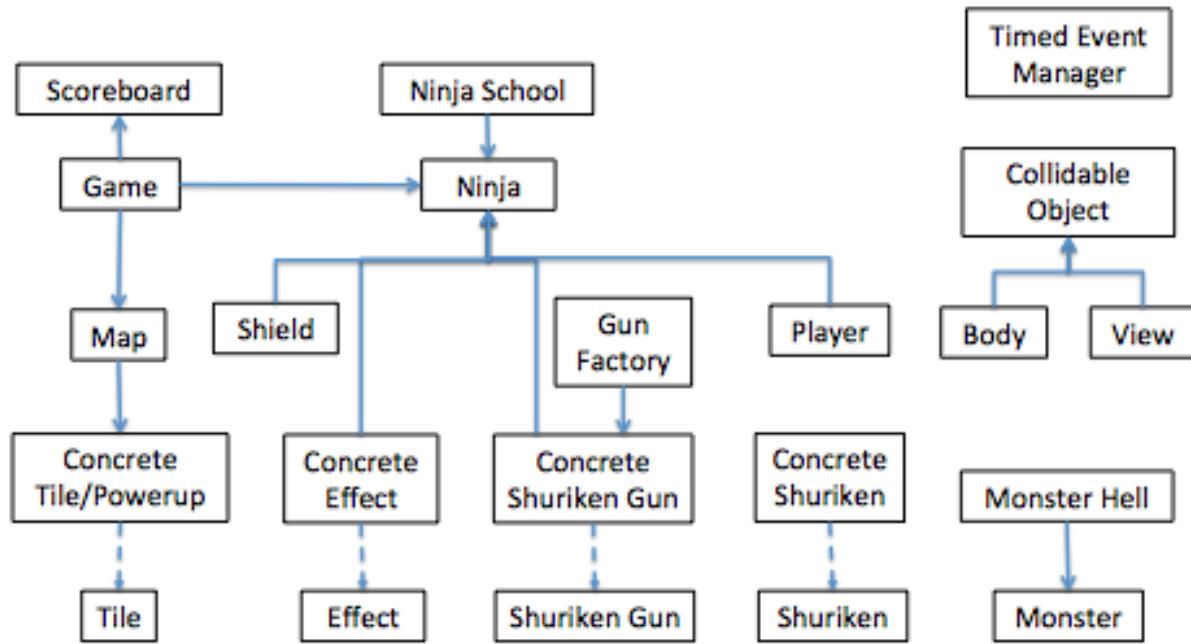


In the **Una** Server mode, the Server has a more traditional role, and it will maintain the room state for all the rooms. Controllers will be changing the state stored in the Server, and the Server will propagate the states that have changed to all the Screen.

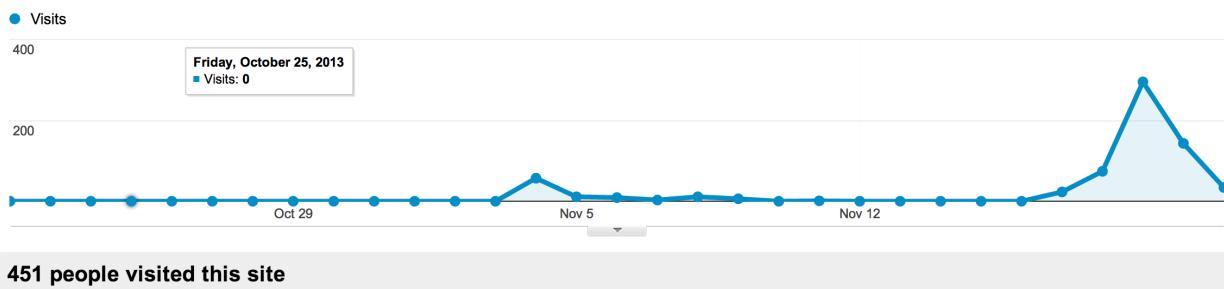
This mode should only be used if there is a trivial amount of logic involved in changing the each room state. The benefit of using such a mode is that since the state of the room is maintained on the Server, we can support multiple Screen instances. The Screen now becomes a viewer of the state stored in the server, and listens to changes of its own state by the Server.

This is the mode used by **Shakeoff**.

Nutty Ninjas UML Diagram



Report on the current number of users who have installed, active users, etc. Perhaps Google analytics data and screenshots (or similar analytics tools) to support your claims.



As **Una** is a development library, our target users are the developers. It is incredibly difficult to get developers to commit to use our library and to make a substantial product within the time frame of the final project, and as such we have decided to gauge the potential of our library by pitching it to existing developers, and gauging their response to their library.

We pitched our idea to our mentors at LambdaMu Games, a game studio that makes iOS games, and recorded down their feedback. The following is the summary of their feedback:

- Una should be a low-friction, low-cost, interactive platform.
- The greatest selling point of **Una** is that now everyone can play using their phone as a controller.
- The team at LambdaMu games have brainstormed multiple possible use cases for Una in just a short span of 15 minutes.
- The nature of the games that would be possible on the platform can largely be divided into 2 categories, those needing less than 10 people (e.g. **Nutty Ninjas**-styled), and those for more than 10 people (e.g. Bishi Bashi styled)
- Possible use cases beyond gaming are advertising and education.
 - Advertising: A novel method of getting people's attention. Should target advertising firm who wants to run a special/custom campaign.
 - Education: Possible a Q&A style application that could be used during lectures.

Future plans and strategies.

We have received strong interest from the crowd during STePS, and many were fascinated by the BYOD (Bring Your Own Device) gameplay concept. Many people got to try out **Nutty Ninjas** using their own mobile phones because there was low resistance to trying; there was zero hassle. No need to download or install anything and they could use their own mobile device straightaway. Judging from the feedback we received from the STePS event, we have 3 main target groups in mind:

Game Developers (Indie/Studio) - In the long term, we want to generalize our product so that game

studios can utilize our library in their game. We strive to make our library as easy to use as possible, such that game developers only need to plug their game engines into our framework and everything will fit in together seamlessly.

One possibility will be to include support multiple screen in **Una**'s Normal mode, so that games could be shown on more than one screen. If we can overcome the technical challenges needed to synchronise game state across screens, we believe the appeal that **Una** brings to the game developers will increase even more.

We will also have to work with game studios to determine the best genre of games that could be built using **Una**, and work towards reducing the effort needed to build these type of games.

Gamers - Our first priority will be to support multiplayer casual games that are similar to **Nutty Ninjas** - up to 4-6 players for casual gamers in a social settings. We will also strive for more diversity in the nature of our future games, and appeal to both male and female gamers.

In addition, we will seek to exploit more and differing capabilities of the mobile phone, such as the camera, microphone, and accelerometer. Our intention is create unique gaming experiences for games that is available right from their pocket.

Brands - There were some interest in utilizing the **Una** library to power interactive advertising for brands as seen in our **Shakeoff** demo. Low-barrier to entry means that everyone can join in the fun and generate consumer attention for the brand. Coupled with the multiplayer interaction, advertisements will leave a deeper impression on consumers as compared to static graphics. Best of all, the maintenance costs to keep the system up and running is really low.

Another possibility of utilizing **Shakeoff** as an advertising tool would be in bars during a major sporting event. Fans could be incited to show their support for their team by using **Shakeoff**, and this opens possibilities of incorporating brands on the main **Shakeoff** screen. An alternative would be to reward addition credits when fans purchase an item in the bar.

Insights gained from the project. What did you learn from doing the Final Project?

Team Synergy - We enjoy working with each other and do not mind meeting on weekends and spending time together to develop the product. There were not much division of roles and each of us could step in to help each other on their parts. Also, all of us enjoyed the process of working on the product and the game. We think that it is important for everyone to like what we are working and that intrinsic factors should be the main source of motivation for working on the project.

Agile (Anyhow Whack) Is Underrated - When we started brainstorming the game, we had some difficulties in "How to make the game fun and innovative", resulting in slow progress initially. However, during the development phase and after each iteration, new ideas seem to come naturally to us, to the point that we felt that it was a pity because we did not have enough time to implement all of them. It's impossible to have everything at the start and to know a product's full potential.

UX Is Absolutely Crucial - Through repeatedly getting our friends to play the game, we observed how they interacted with it. Initially, the UX of the product was pretty bad, and we ended up having to teach our friends how to use and play **Nutty Ninjas**. We learnt the importance of user testing and validation. Although the friends that we showed the product to found it difficult to use initially, once they were taught and after they started playing, we could see smiles on their faces while they were engrossed in the game.

With advice from Prof Colin, Su Yuen, Ivan, Abhishek and further user testing, we improved the UX with the following enhancements:

- (i) Removing 'Quickstart' from the main menu of the game
- (ii) Supported both orientations of gameplay for users who did not lock their device orientations
- (iii) Increased joystick hit area to left side of mobile device
- (iv) Reduced the amount of complicated textures in the terrain background
- (v) Gameplay messages to allow players and spectators to feel more involved in the game

As a result, on the day of STePS, we did not need to teach users how to play. When people saw existing players using their mobile devices to play with a huge screen, they intuitively pulled out their own devices and did the same. The whole process from joining to playing was a low-barrier, low-resistance one:

- (i) BYOD (Bring Your Own Device)-style. Their controller was in their pocket.
- (ii) The usage flow on the controller was linear. On each screen on the controller, there was either a next or a back button and input validation was in place to ensure that the user has performed the necessary steps before proceeding to the next screen.

From our Google Analytic data, we saw close to 300 unique users trying out **Nutty Ninjas** on the day of STePS. We believe a huge reason was due to the improvement in UX that we made after doing significant user testing and implementing their feedback.

Technology is Flexible - **Nutty Ninjas** was built using HTML5 canvas but the rest of the site was done using HTML5 DOM. Text and layout manipulation is a pain to do in canvas and DOM is a more suitable implementation for it. We could have done everything in canvas but we decided to use the DOM for the non-gameplay aspects of the game so that we could leverage on helpful front-end MVC toolsets like AngularJS to handle the auto-updating of the scoreboard and CSS3 transforms to do simple animations which are hardware-accelerated and crucial for smooth performance on mobile devices.

Execution over Ideas - The idea of having multiple users interacting with a large display is not an entirely new idea. While doing research on advertising use cases of our product, we found at least [10 over examples of big brands using interactive billboards for advertising](#). However, these brands probably spent a huge load on creating these advertising apps when they could have done so at a much lower cost.

We think **Una** and **Nutty Ninjas** were executed well. A month ago, we had nothing. We were busy with midterms for the other modules that we have neglected in the first half of the semester, so we started on the final project late. The iteration time between idea conceptualization, code and testing could last from a few minutes to a day. Thanks to the blazing speed at which we coded and designed, we had 2 solid products to showcase on STePS day. Certainly, good design and planning helped in the process.

Good Design (In code and graphics) - Good looks are absolutely critical to attracting the crowds we have seen in STePS. We have a member (Yang Shun) dedicated to the role in brushing up the look-and-feel of the product and that dedication have paid off in spades.

Ranging from the graphical assets, to the positions of the controller elements, great effort was made to ensure the whole game appealed to our target audience. 1-2 weeks before the actual product was launched on the STePS showcase, we managed to play-test the game with as many friends as possible to get their feedback and polish our design.

Even though the module focused on hacking a workable product together in a matter of weeks, good structuring of our code has helped enormously. It is unthinkable that so many features and libraries can be integrated into the Nutty Ninjas game without regression. Throughout the 6 weeks, the code base has been refactored several times to accommodate extensions.

With the final design of our **Una** library, we managed to create the **Shakeoff** demo within a few hours. this incredible speed can only be achieved since the library takes care of all networking routing for the developer.