

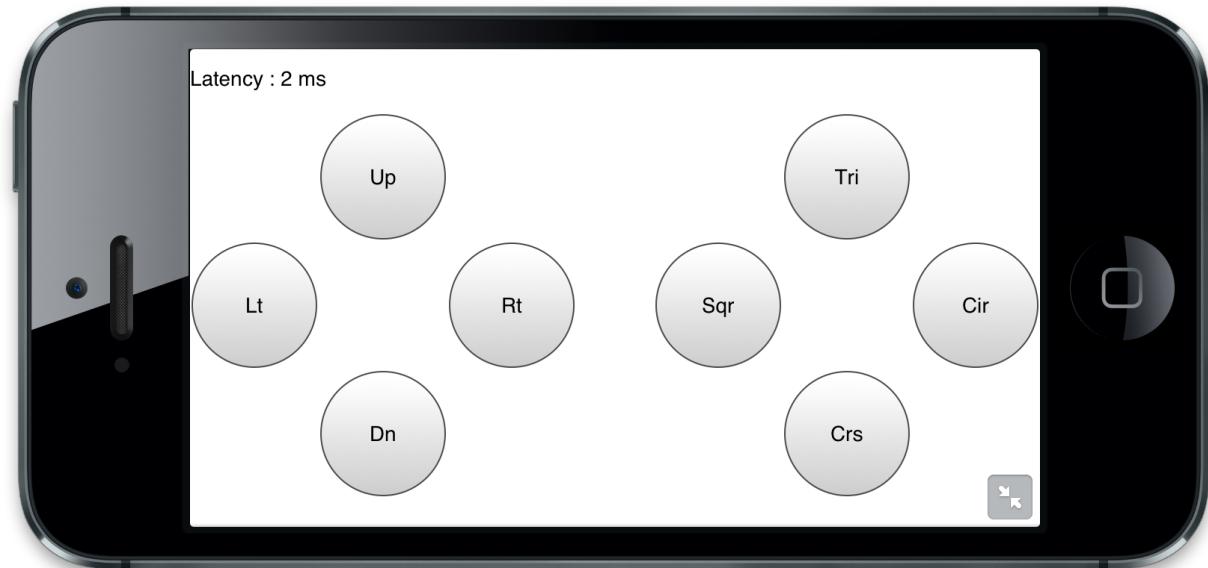
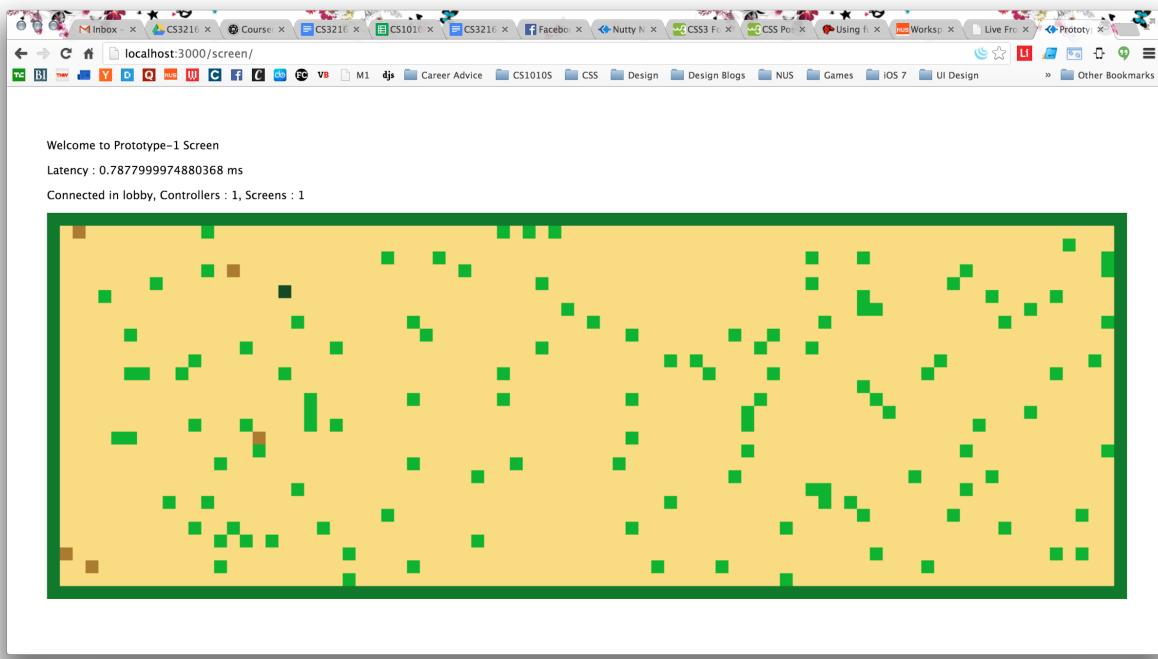
CS3216 Progress Report 1

Team 3 - Soon Chun Mun, Le Viet Tien, Soedarsono, Tay Yang Shun

1. Application Prototype

Prototype 1: Chun Mun's Initial Prototype (Server architecture, and latency)

User connects to the screen using a mobile device and controls the movement of the black pixel.



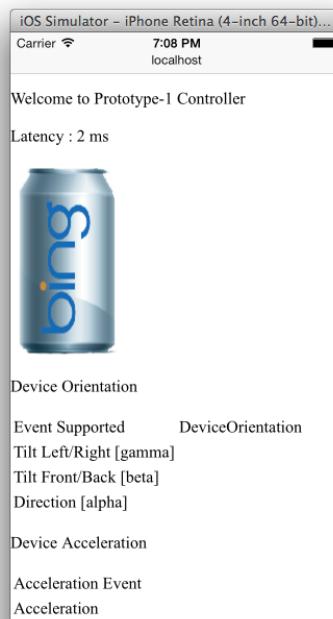
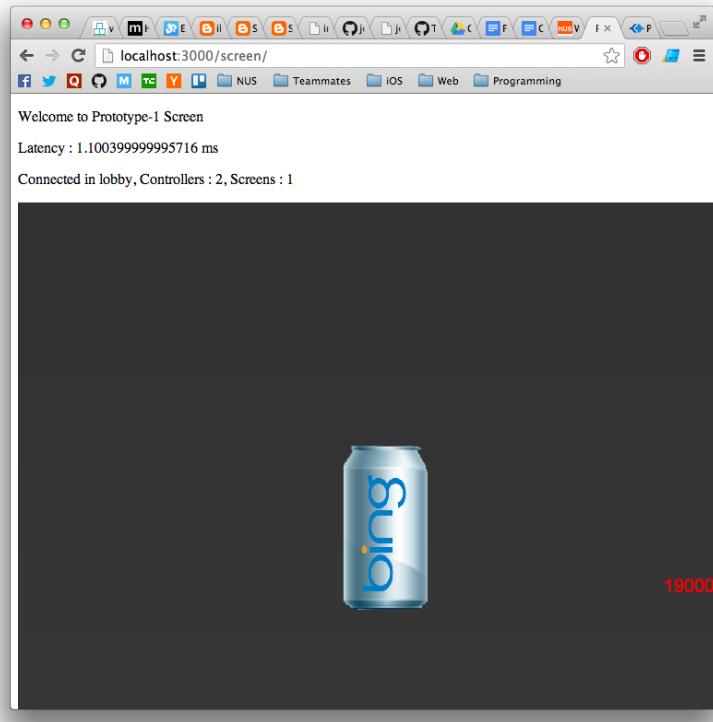
Prototype 2: Chrome Rotation (Touchy.js, Gestures latency)

Sliding one's finger in the orange section will cause the chrome logo to spin.



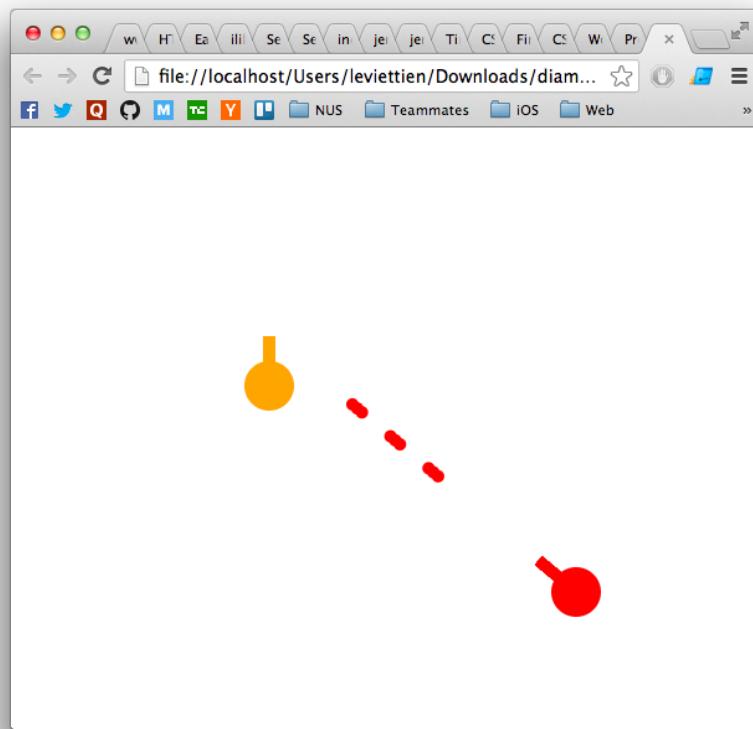
Prototype 3: Coke Shaker (Accelerometer and latency)

Shaking the phone will cause the coke can to vibrate and eventually jump into the air.



Prototype 4: Tank Game (whether the game would be fun or not)

Game whereby user moves a tank and shoots projectiles.





Entry screen of our game on a 50-inch display

2. Is your project on schedule according to the milestones and timelines submitted in the initial project proposal? If you are on schedule, great! If not, why not? What is your team doing about the slip in your project schedule?

We are 1 week behind our proposed schedule. There are a couple reasons for this delay. First, we tested more prototypes than we initially conceived during planning. The prototypes were instrumental in figuring out what constitutes a good game to play in this particular fashion. Secondly, the workload following the midterms started increasing as deadlines for other modules are approaching. This cost us many days of catching up in other modules as opposed to working on the final project

3. What were the problems/difficulties your team has encountered? How have you overcome them, or what plans do you have to overcome them?

a) Scheduling of build time

It is difficult to predict the amount of free time that we have to code together. With various deadlines across our classes and also almost everyone in the team is involved in undergraduate tutoring commitments, it has become obvious that it is impossible to schedule a large amount of time for all team members to meet up.

The splitting down of our main tasks has become more important so that we can get work done

without too much communication overhead.

b) Technical Difficulty - Synchronization of Game States between multiple Screens/Games

The architecture that we have chosen requires the “Screen” to host the actual game state within the browser. “Controller” browsers will connect to the game in the “Screen”. The server that simply keeps track of rooms where the screens and controllers belong to, and relay the messages. This helps to keep the load on the backend light.

Once the game has begun in one screen, the game state in the screen basically progresses as it receives input from the various controllers. It is difficult for another game screen that has just joined to synchronize their game state from the running game and queue up the controller commands that are coming in at the same time.

There are 2 suggestions on how to handle the problem - Sidestep it, or a barrier in time. We can sidestep the problem by only allowing one “Screen” to host any game in the room. Any other screen that tries to enter the room while the game has started will not be able to join.

Alternatively, we can stop all the games across every “Screens” when a new screen joins. Then all will wait for all game state information to synchronize to the new one.

4. Any changes to the application since the initial project proposal?

Initially we decided to build a casual game like [Gyro](#). However after some trials with our friends with our prototypes, it became apparent that the game needs to be changed to improve the user experience of playing on the phone.

We settled on building a ninja-themed multiplayer shooting game similar to the classic retro game - BattleCity. Hopefully, it will be well-received when we do our alpha-testing.

5. Assuming that your team has already deployed a prototype of your application, how has the response been? Any other new insights, plans or strategies your team has come out with?

So far, we have been concerned about verifying the technical feasibility of the project. As a result, we have built several technical prototypes to verify our proposed architecture, and verify the performance.

We were primarily concerned with the latency of our architecture. As a command originating from the user’s mobile phone controller would have to make 2 network hops before reaching the screen (one from the mobile phone to the server; one from the server to the screen), we were worried that the latency would be too high to make any of our proposed game unplayable.

We conducted our testing with up to 4 mobile phone controllers with different makes and with

Prototype 1 and Prototype 2. All the mobile phone controllers and screen were connected to the NUS Network during the test, and the server was hosted on an AWS instance located in Singapore. We have discovered that with a fairly good internet connection, the latency was often insignificant to cause any disruption to the gameplay. However, there were instances of lag spikes that occurred sporadically. In addition, we discovered that the make of the mobile phone and mobile browsers actually had an impact on performance.

We would be looking into the possibility of creating a native wrapper application that would allow players to deploy a game locally on their local network. With a more reliable local network, we expect that there would be lesser instances of lag spikes in this configuration. We would also suggest that players use a better performing browser when playing the game.

6. Updated project schedule

Phase	Period	Details
3	2 weeks	Nutty Ninja v0.1 - second iteration
	21/10 - 27/10	<p>Network layer:</p> <ul style="list-style-type: none"> • Join and leave the lobby • Message passing between controllers, server and screens <p>Game:</p> <ul style="list-style-type: none"> • Lobby system • A copy version of Battle City • Basic integration with Box2D • Splash screen • Ninja themed assets
	28/10 - 3/11	<p>Server:</p> <ul style="list-style-type: none"> • Manage all running rooms <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 features (traps, power up, etc) • Advanced graphical assets • Optimize for speed/real-time factor • General improvement
	<i>Achievement</i>	<i>Network layer v0.1</i> <i>Game v0.1</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

		<ul style="list-style-type: none"> • 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 newly interface
	<i>Achievement</i>	<i>Network layer v0.2</i> <i>Game v0.2</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>