Hot Beats Setup Instructions:

1. Requires TH2 builds or later.
2. Set PC into developer mode
3. Install VS Code:
4. install node.js:
5. install git:
6. install “project my screen”
7. open powershell (or cmd prompt): npm install manifoldjs -g
8. create a new folder for generating app
9. in powershell, generate apps “manifoldjs http://btdj.azurewebsites.net “
10. type “cd HotBeats” then “Manifoldjs run windows10”. This should install and open the Windows10 app.
11. If you need to install the apps on the Windows Phone and Android phone, do this with the source packages in this folder. The APK can be downloaded on the Android phone from the GitHub repo: <https://github.com/nmetulev/buildtour2016/tree/HWA-keynote>
12. This next step is to make the install go super fast in the keynote
    1. Up platforms.json file in the npm folder: C:\Users\<**userAccount**>\AppData\Roaming\npm\node\_modules\manifoldjs
    2. Remove platforms for iOS, Android and Windows by deleting each line.
13. In the new “HotBeats” folder delete the folders for web, Chrome and FireFox. Rename windows 10 folder to “Windows”.
14. Got to GitHub and clone the source code from the project: http://github.com/boyofgreen/dj

Hot Beats Demo setup

1. Open HotBeats in the browser at <http://btdj.azurewebsites.net>
2. Open source code in visual studio code, open up manifest.json for illustration
3. Open powershell (or cmd prompt) and navigate to the folder which houses the “hotBeats” folder
4. Open Windows Phone and open “hot beats” app until it shows the sleigh Bells. Force Close App. This is important because the app will show the start screen every other time it’s open.

Hot Beats Reset for Demo

1. Find “HotBeats” folder where you generated the Manifoldjs app
2. Delete folders for “web”, “Windows10”, “FireFox” and “Chrome”

Hot Beats Demo Script

|  |  |  |
| --- | --- | --- |
| Step | Script | Screen Shot |
| 1. show in browser | > Many interesting Web Apps out there.> Hot Beats is a music performance app that lets you create tune old school style. - choose first four instruments then hit go.  - play each of the instruments to show different sounds |  |
| 2. generate apps | > lets move this into a store app so I can take advantage of some new APIs, and release to the store> I’m going to use command line tool ManifoldJS to generate these apps for each platform - in powershell, type “manifoldjs <http://btdj.azurewebsites.net> -l debug” > let the tool run - cd into Hot beats and either LS to show platforms or show in explorer |  |
| 3. show manifest | - in vs code, show manifest.json file> what’s happening is that Manfioldjs is going out to the site and downloading the W3C manifest, and from that it’s generating platform apps from it |  |
| 4. demo app | > now we have the app already installed here - open app, show smtc media controls with intigration |  |
| 5. demo rome | > now some of these instruments are better played on different devices. I want to play these jingle bells on my windows phone  - show project my screen for phone and open app  > you see here this is a UWP so it runs on all the platforms but it starts me over at the begging of the app  - force close app so it launches from closed  > now we have a new feature is Anniversary update that lets me transfer the state of this app to the one on other devices. It’s called project rome  - hit phone icon and then select windows phone  - go back to project my phone to show app open on windows phone with jingle bells  > Now you see we pick up right where we left off. And I can now use the motion sensors of the windows phone to play it right (shake phone)  -go back to phone and launch Rome icon again  > now lets go to the cow bell, which can get pretty rough. I don’t want to break my windows phone, so lets send it to the android phone  > project rome is cross platform as well, so we’ll use the android app we generated  - hit android phone, then open app on android phone  - pull out the mallet and play windows phone like cowbell. “it needs more cowbell!!’ (be careful, you can break screen) |  |