Setting up
Cordova Build
for Construct games
with CLI (command line interface)

Version 1.1

Author: saiyadjin

Contents

0. Changes 2
1. Intro 3
2. Prerequisites 4
3. Installing Visual studio and Android SDKs 5
4. Installing Java SDK (JDK) 7
5. Installing NodeJS 10
6. Installing Gradle and Git 11
7. Building and testing your game with Cordova 12
   7.1. Setting up Cordova 12
   7.2. Construct 2/3 games 12
   7.3. Non Construct 2/3 games (or apps) 14
8. Advanced techniques and help 15
   8.1. IPhone X fix 15

0. Changes

Version 1.0:

- Initial version

Version 1.1:

- Improved intro
- Made text more readable
- Changes to Java from 10 to 1.8 since Cordova still uses 1.8 version
- Updated installer part for Visual studio
- Updated installer part for Android SDKs
- Updated Java installer parts, since Version 1.8 HAS to be used
- Updated links to download Java
- Updated chapter 7 – for non C2/3 games/apps and for C2/3 games/apps
    - Also split the thing in 2 parts, since basically building is always the same
- Added chapter 8 – fixes and advanced stuff
- Fixed wording on some sentences, generally easier to read

1. Intro

So you guys want to make Cordova games for your mobile phones but you have no clue what and where to do it?

You want full control, but you don't know where to start from? Want to be able to edit and tweak your configuration files, build the game for different architectures and devices?
You are using some of the HTML game making tools like Construct 2/3, Game Maker or other that give you just projects that you don't know how to compile and use?

Good, then this is the guide for you. I will show you how to build the game for Android and IOS using default options. For some *advanced techniques* you will have to ask me on Discord or Google them yourself. Some of those are described in later chapters, but there's much more.

So let's keep this simple and fast. This guide is primarily intended for Windows users, since I don't use Linux or anything else by that matter, but it can be used as a handy guide and some customizations for Linux and MacOs users since this is all JS based (games).

2. Prerequisites

What, you thought you won't need anything? ☺ Anyway let's go (**install nothing for now, just download**):

A) NodeJS:
   - download latest LTS version (LTS is 8.11.3 at the moment of writing)
   - Make sure it's x64 Windows installer (.msi)

B) Java SDK V1.8:
   - Accept the agreement and pick the Windows x64 version file.
   - Do not download Demos and samples

C) Android studio or Visual studio:
   - **This guide will focus on Visual studio!**
   - Click free download button on the Community version

D) Android SDK:
   - Installable through Visual studio (or Android studio)

E) Gradle:
   - our build tool

F) Git:
   - Tool for downloading projects and binaries and stuff from sources like GitHub.
   - Click the button on the right side to download

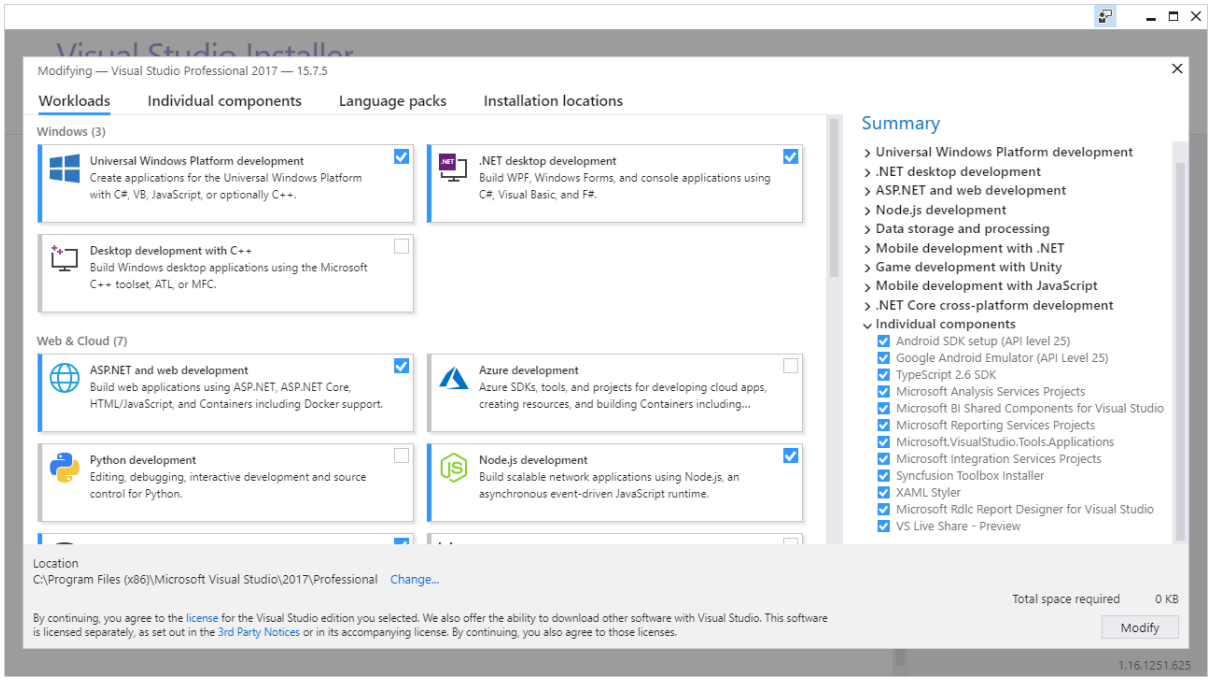
G) Notepad++ :
   - Download x64 version
   - Pick installer version

Make sure you download everything in x64 variant, I doubt anyone has x86 Windows.
Ok, once you've downloaded it all, let's get started. We'll go through installation first.

For those on Linux or MacOS – make sure you pick installers for your platforms.

3. Installing Visual studio and Android SDKs

Run the Visual studio installer. Once you've pressed install, this menu appears:



You will have to choose the next stuff to install:

A) *Node.JS* development (in case you get interested in NodeJS)

B) Select *Mobile development with .NET* – make sure you select in the right panel *Google android Emulator (API 27).* Also untick *Java SDK*

C) Finish installation

After installation run Visual Studio. Follow these steps:

a) Click the toolbar *Tools*

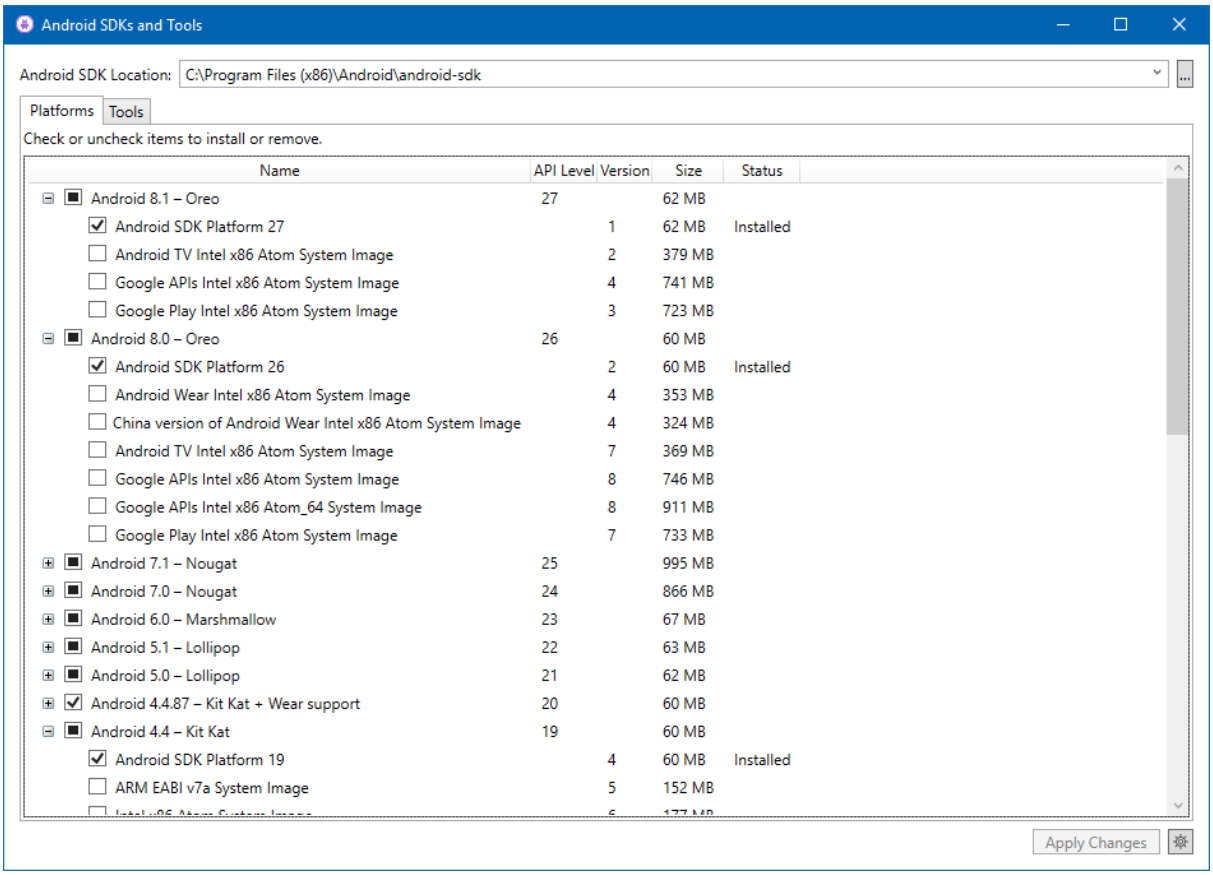
b) Move mouse to Android

c) Click the *Android SDK Manager* option

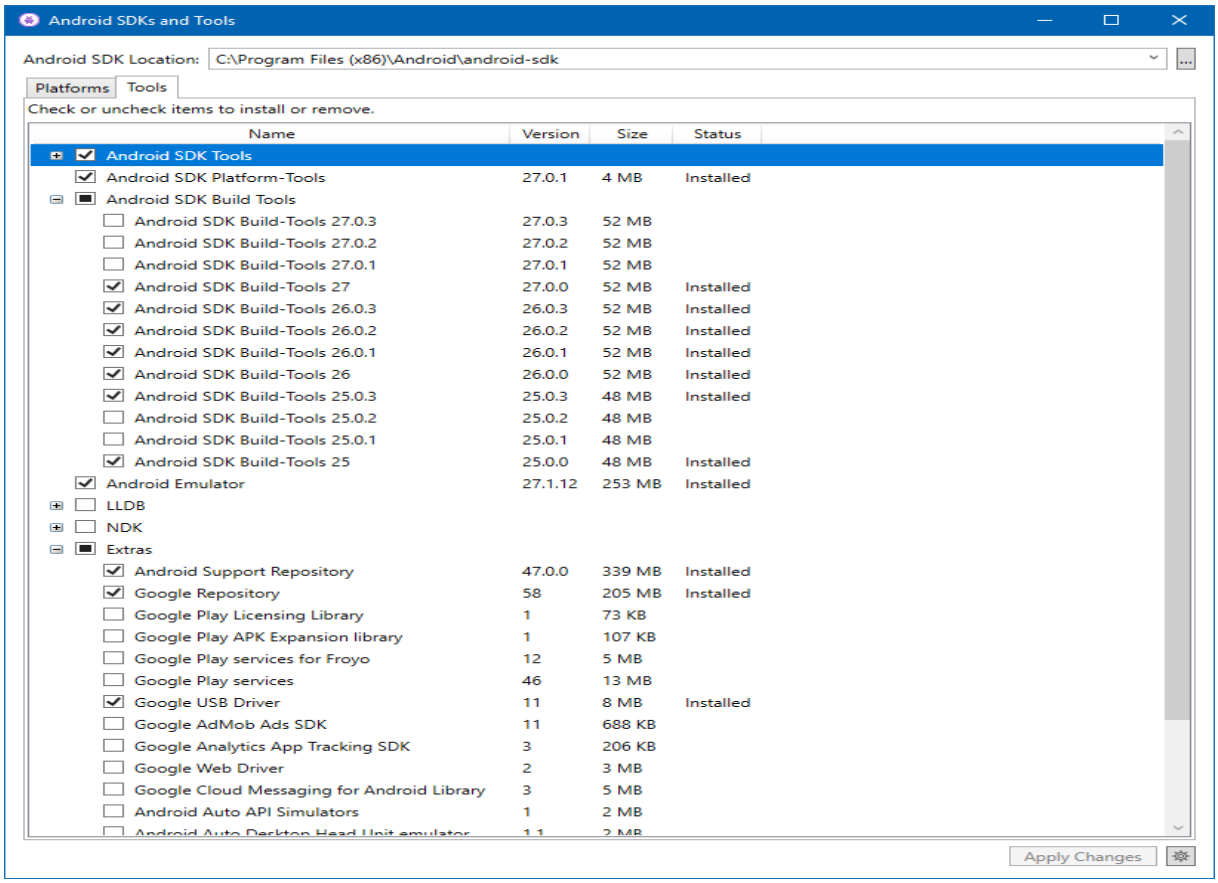
d) Select the Android SDKs to install

For point d) check the image on the next page. You should select all the SDKs that you want to support. I recommend from 19 (KitKat) upwards.

You can also install additional tools as noticed on the second image on the next page. Google USB driver, build tools and repositories are useful so do install them.



Click on apply changes to install all those SDKs. You can also install these tools:

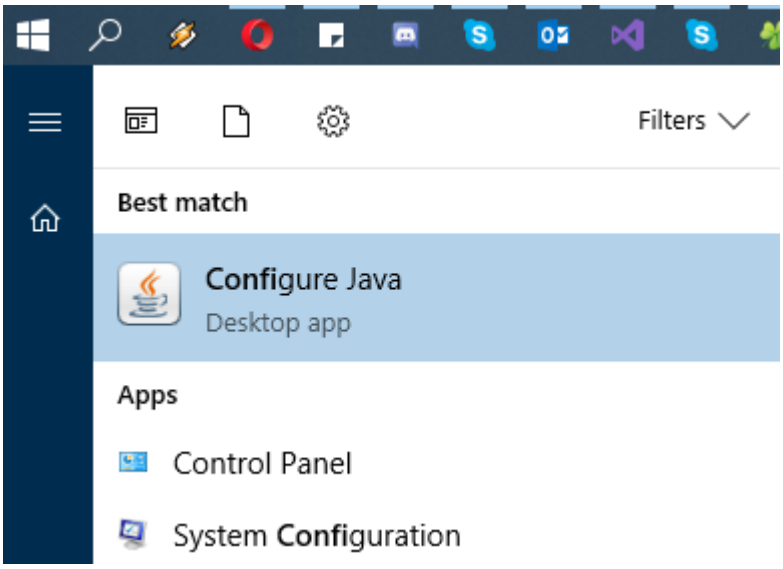


Once that is done, we can go to the next step. You've installed Android SDKs.

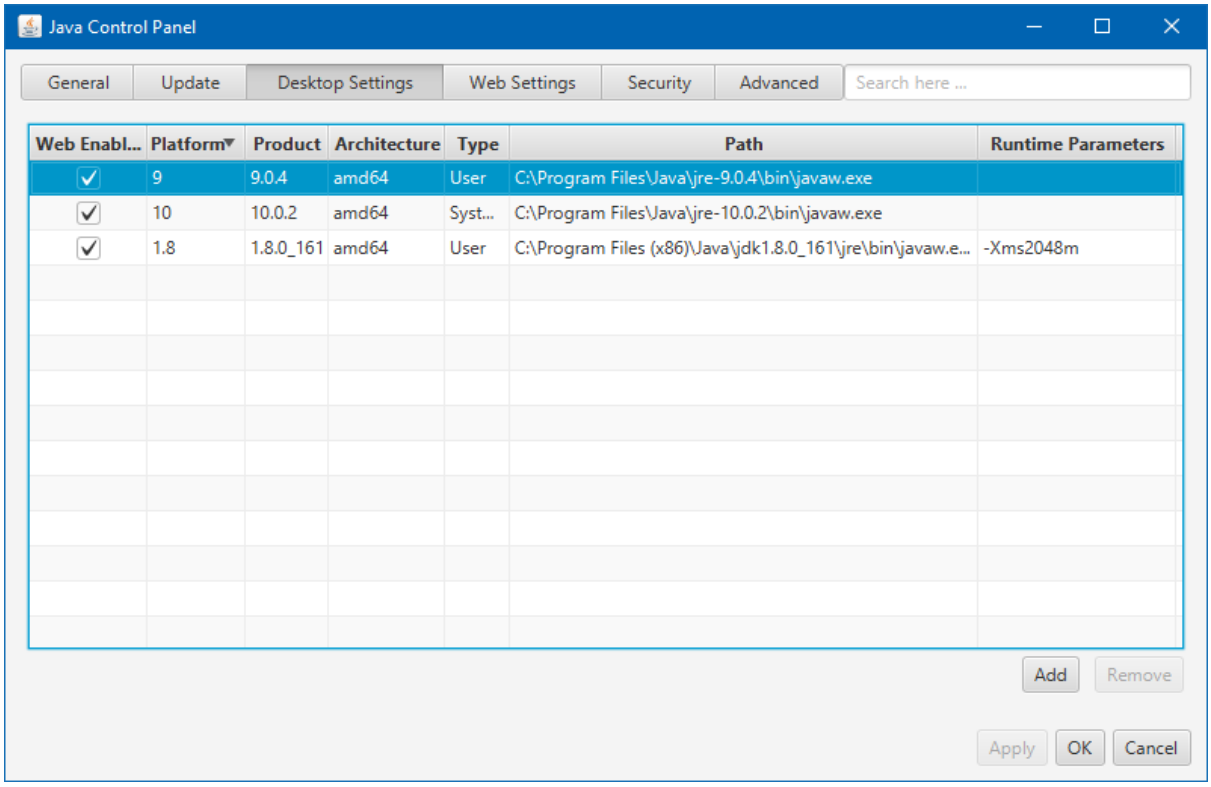
4. Installing Java SDK (JDK)

Run the installer and follow it. Install to default location. It will ask you to install JRE (runtime component) so install that too. You should leave it all default *C:/ProgramFiles/Java* path.

After you've installed it, you can open *Configure Java*. In Windows 10 just write in search *Java* and it should appear like on the image to the right.



Once you've opened the window, you will see all versions of *Java* installed and their configurations under desktop settings. Under advanced you can turn on debugging and other useful features, that you won't need for building your games (at least I don't use them). Shown below. If you don't have *Java CP*, you can open *Java* in *Control panel* and check it out there.



Now all you have to do is create proper environmental variables in your PC for Java to work correctly. How do you do it? Why do we need JAVA_PATH variable? Our *Gradle* builder will build our game using Java and some downloaded *Cordova* stuff, that's why. So here are the steps needed to set up your JAVA_PATH var.:

a) Open *Control Panel*

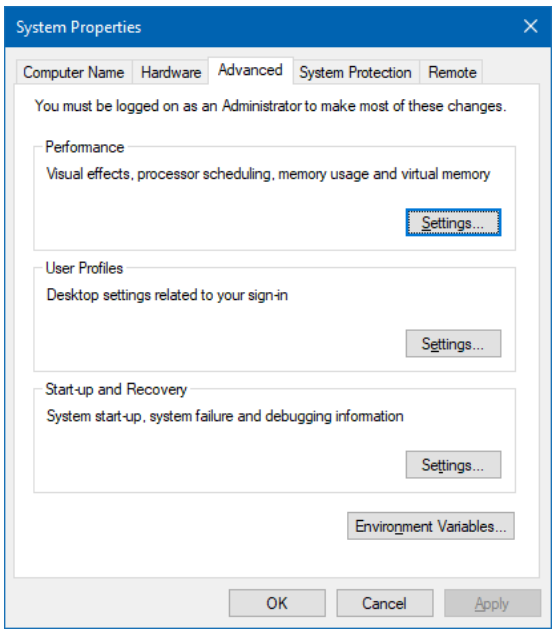
b) Open *System*

c) Open *Advanced system settings*

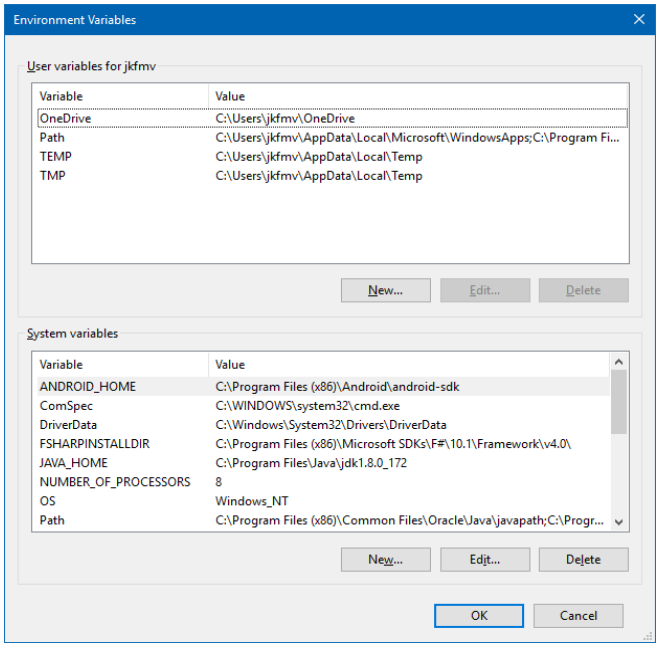
d) Open tab *Advanced*

e) Short way to get here for Win10 users, write "*Environmental variables*" into search

f) You should be looking at this:

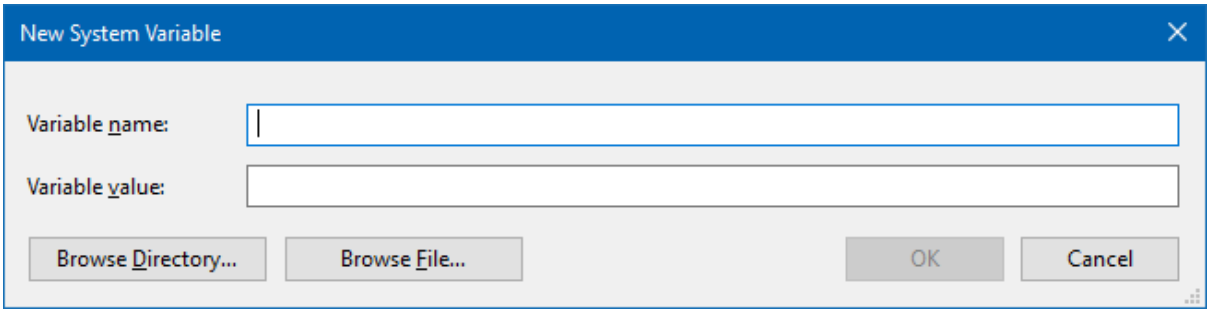


g) Click button *Environmental variables*



h) Now as you may notice in *System variables*, you have to find a variable named JAVA_HOME. If it doesn't exist click *new*.

i) In this window:



j) Write down the variable name: JAVA_HOME (Make sure it's UPPERCASE)

k) Under value, browse to your *Java* installation JDK folder (directory). Make sure it's JDK not JRE folder.

l) It's usually always in C:/ProgramFiles/Java/JDKxxxxx (select that folder with **latest 1.8 version**) - it has to be 1.8 version or our builder won't work!

m) If you want to check if you did right, take a look at the upper picture and see how my JAVA_HOME is set.

n) Now since we are already here we can set ANDROID_HOME variable too. Set it to C:\Program files (x86)\Android\android-sdk (if you have that folder, and you should after installing Visual Studio and Android SDKs)

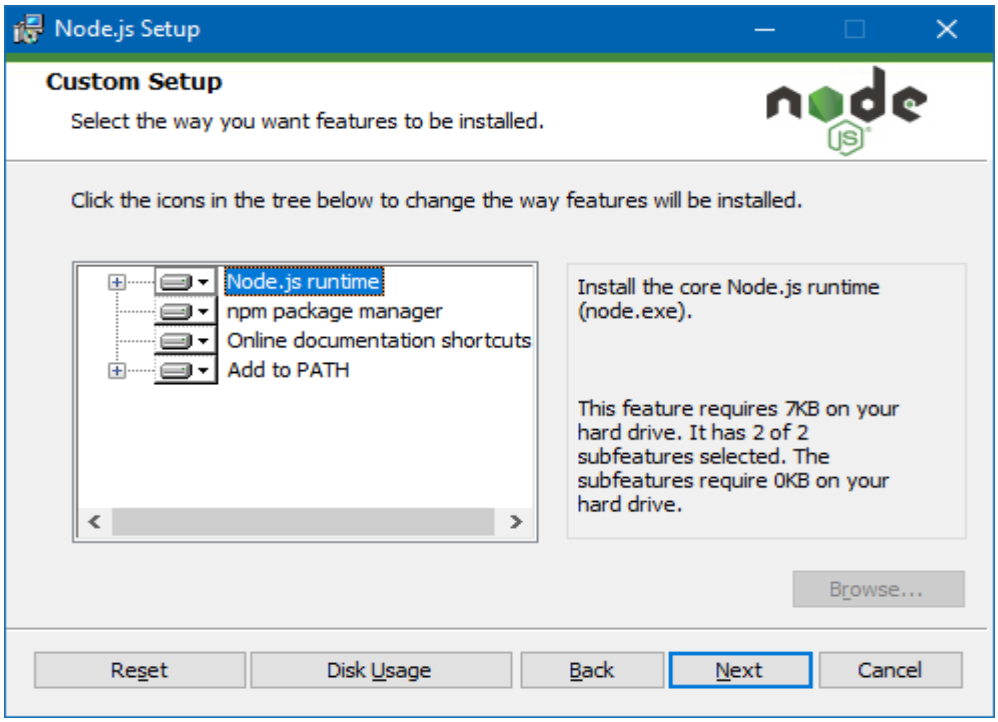
o) Once that is installed, click on the system variable named *Path* and click *Edit*

p) In Windows 10 you will see each path in a table where you can add new, but in Windows 7 it's all just one line separated with ';'. Check if you have these paths:
%ANDROID_HOME%\tools
%ANDROID_HOME%\platform-tools
C:\Program Files\nodejs\ (check after installing NodeJS, next chapter)
C:\Gradle\gradle-4.9\bin (check after installing Gradle, 2nd next chapter)
C:\Program Files\Git\cmd (check after installing Git, 3rd next chapter)

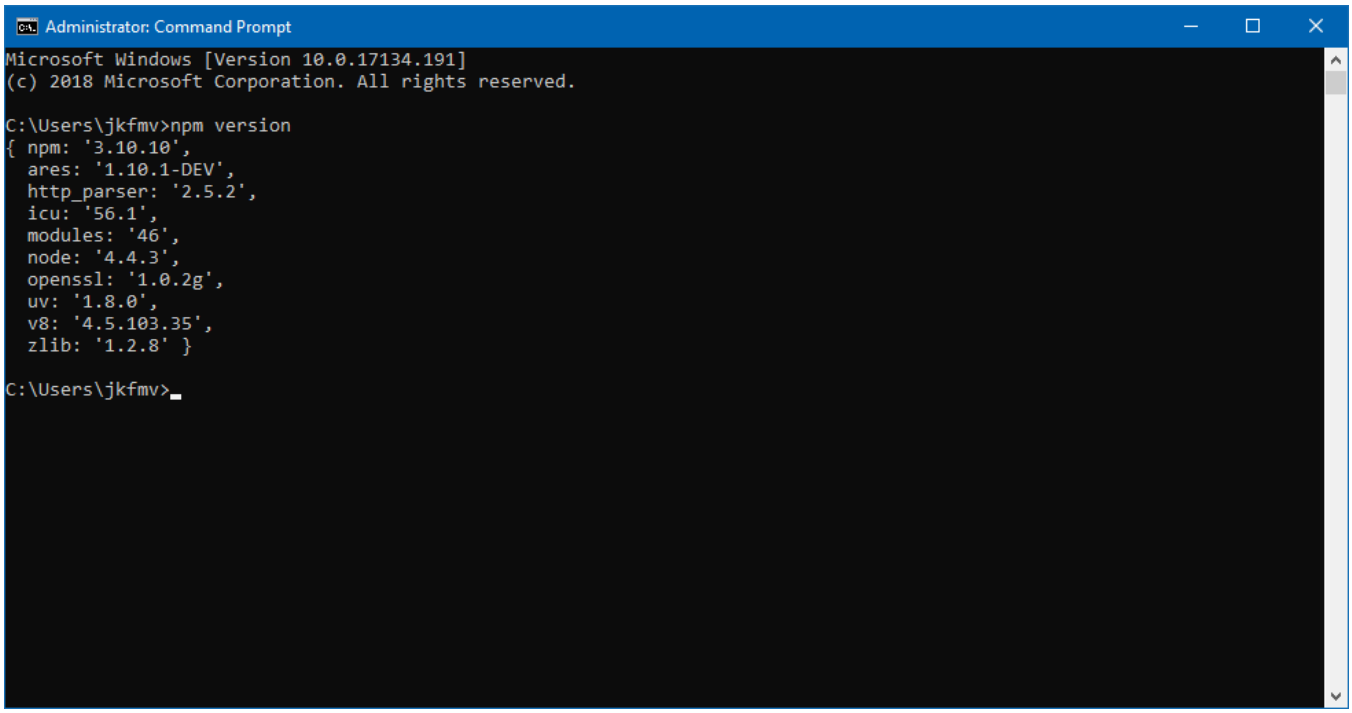
And that's about it. Your Java and Android-SDKs should be set up. Let's go install final tools for working with Cordova.

5. Installing NodeJS

Basically run the installer you've got from NodeJS website (link above). Keep the default installation path. Make sure all components are selected on install window. Like this screen:



NPM package manager is what we really need. Once it's installed, go check if the environmental variable is set (in chapter 4.). Run command prompt and write *npm version*. You should see something like this:



6. Installing Gradle and Git

For Gradle installation I hope you have downloaded the binary only file.

a) Create folder *Gradle* in C:/

b) Extract *gradle.zip* to C:/Gradle

c) Go to chapter 4 and check the environmental variable for *Gradle* if it appeared, if not:

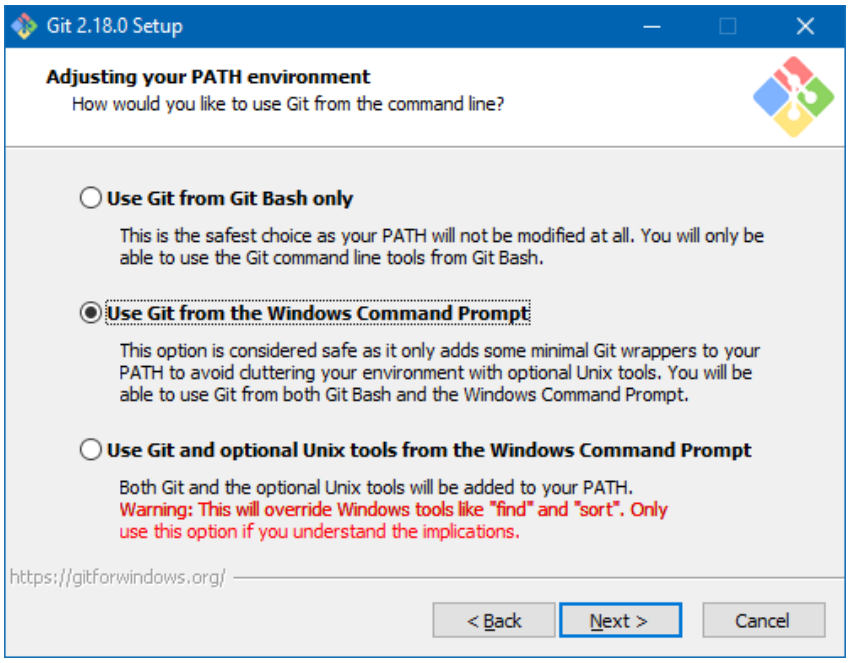
   1) You will have to add another one to *Path* variable

   2) Add this folder – C:\Gradle\gradle_4.9\bin

   3) Make sure you pick the correct folder!

d) Open CMD and run *gradle –v* command. You should get some Gradle info. If it doesn't work, you've done something wrong.

Now download Git. Run the installer. Follow the default settings, but make sure you select this one:



Once it's done, return to chapter 4 and the *Git* environmental variable. It should appear by itself in *Path*, but if it isn't there update accordingly. It should be something like C:\Program Files\Git\cmd.

Now open a CMD and write *git version*, and hit *enter*. If git version appears, you have set it up correctly.

If all this passed okay that means you've set up your *Path*, JAVA_HOME and ANDROID_HOME variables correctly, and installed everything as it should be.

7. Building and testing your game with Cordova

7.1. Setting up Cordova

So you've basically installed everything and you're ready to go. Well not yet. You have to install Cordova for your PC. Best method is to install it globally, by doing this:

a) Open a command prompt
b) Execute this command: *npm install -g* *cordova@8.0.0*

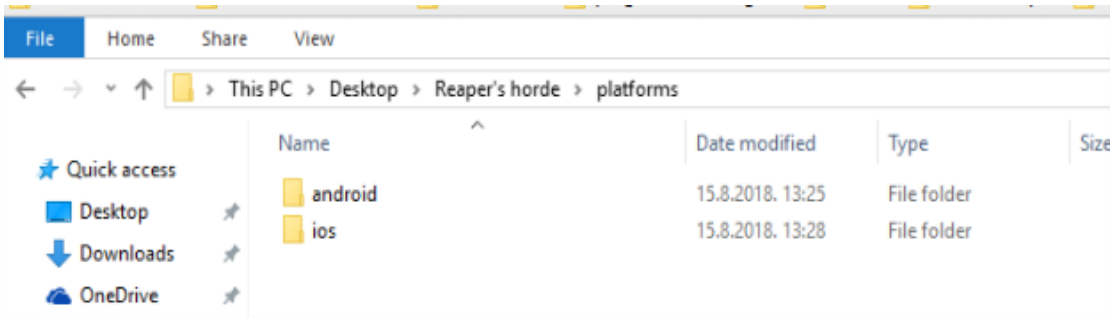
After you've done this *npm* will install the Cordova runtime version 8.0.0 to your PC. IT will be global because of the flag *–g*. You can install Cordova separately different versions into any folder, but I would advise you to use global, latest version. Before you install 8.0.0 you can always check on Cordova website what the latest version is.

7.2. Construct 2/3 games

Since Construct 2 and 3 can already export your game as a Cordova project, we will use that for our start point. So basically here's what you do:

a) Export Construct game as a Cordova project
b) Open CMD (command prompt)
c) Position yourself to the folder of the game, example if you've exported the game to desktop, you would end up writing a command like "*cd C:\Users\jkfmv\Desktop\Reaper's Horde*" (Reaper's horde is the name of the folder that Construct generated while exporting the data)
d) Checking inside this folder there should be a '*www'* folder and a *config.xml* and *config.json* file. You can remove *config.json*. If this structure does not exist, something went wrong with export and you should try again.
e) Now in command prompt you can run these commands:
    1) *Cordova platform ls* – will list all platforms installed and those ready to install
    2) *Cordova platform add xxxx@x.y.z* which will add a platform you want your game to build to. For example *Cordova platform add android.*
    3) If you want a specific version then you can specify the x.y.z. parameter, for example: *Cordova platform add android@7.1.0*

4) Running these commands will automatically prepare the data in config.xml and will create a folder named *platforms* which will have a folder for each added platform, for example for ios and android:



f) Each of these platforms now has its own setup of folders and copy of the www folder (which is our assets (game) we are compiling)

g) In Reaper's Horde folder, the main folder, a folder '*plugins'* appeared as well. After adding a platform, our *npm* and *git* automatically pulled the plugins from the config.xml file and they are global to our projects.

h) Now you can run command *Cordova build* which will build everything, or specify a platform you want to build for, for example *Cordova build android*.

i) If everything is fine, you should see a build succeeded in the output in the command prompt window. This made you a *DEBUG APK* or an IOS project for *XCode*. You can follow the path to the generated APK seen in the command prompt.

   - Note: IOS will give you a project folder, no APKs there. You can find that project opening the folder *Plaftorms/ios*. Basically copy the whole IOS folder and open *ProjectName.Xcodeproj* folder, and inside it you will have a *pbxproj* file which runs then on *XCode* on a Mac. You can then build the game in *XCode* and play it on simulator / iphone.

j) If you want to build an release APK, you just have to do: *Cordova build android -release*

7.3. Non Construct 2/3 games (or apps)

So basically you've written an app in JS and that app works in your browser. You want to create a Cordova app out of it and run it anywhere (mobiles, net etc...)

Well most of the build steps are the same in last chapter, but we will cover here how to create an empty Cordova project and fill it with your app, and basically then you follow the 7.2. Chapter from point E.

a) Create an empty folder on your Desktop and give it some name
b) Open Command prompt in there (in the folder)
c) Then write and execute: *Cordova create testApp com.app.apptest*
    - You can change the '*testApp*' and '*com.app.apptest*' to something you like. This is just an example, use your own names. And yes they must be of format x.y.z.
d) After that write: *cd testApp* -> this will move you to the folder of the app.
e) Now take your app with everything you have (basically copy paste your app) to the *www* folder in *testApp* folder.
    - Make sure your app starts from *index.html* or you will have to play with *config.xml* to get it to work.
    - If your app has a *config.xml* for Cordova, or plugins I suggest you copy the parts to the generated config.xml in *testApp folder*. If you have trouble with this, contact me and I will help you.
f) That's basically it.
g) Now follow the chapter 7.2. from point E – add platform, build and you will get your app in an APK.

8. Advanced techniques and help

8.1. IPhone X fix

So basically a lot of Construct 2 users have come to this problem. They've exported the game, built the game, but on IPhone X it looks weird. It doesn't cover the full screen nor is it working as intended. No messing with *config.xml* and anything like that is going to help this problem, but there is a solution. Removing the black bars, white bars and getting the game over the full screen is what we will show you how to do here. So let's start:

A) Besides this PDF file there is an *imagesForIPhoneX* zip that you can download since you will need it.

B) Extract those images / zip into your APP prior to building process (adding platforms)

- They should be extracted into the 'www' folder that Construct 2/3 creates

C) This is only, and ONLY for IPhone builds. Remove those images when doing Android build since they will increase your size, since Android allows APKs up to 100MB and IPhone allows up to 4GB.

D) Open config.xml of your app. Add this:

```
<platform name="ios">
    <splash src="www/Default@2x~iphone~anyany.png" />
    <splash src="www/Default@2x~iphone~comany.png" />
    <splash src="www/Default@2x~iphone~comcom.png" />
    <splash src="www/Default@3x~iphone~anyany.png" />
    <splash src="www/Default@3x~iphone~anycom.png" />
    <splash src="www/Default@3x~iphone~comany.png" />
    <splash src="www/Default@2x~ipad~anyany.png" />
    <splash src="www/Default@2x~ipad~comany.png" />
  </platform>
```

E) Open your *index.html* in *www* folder. Find this line:

```
<!-- Allow fullscreen mode on iOS devices. (These are Apple specific meta tags.) -->
<meta name="viewport" content="width=device-width, height=device-height, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0, user-scalable=no, minimal-ui" />
```

F) Add this to the line at the end: *,viewport-fit=cover*

- The line has *name=viewport*, and the rest of the line might not be as shown on the upper picture, but the *viewport-fit* is probably missing. So don't worry if your line is not exactly the same as this image above.

G) After that you should see some CSS code lower in the same index.html file. Find the code for html,body, add marked lines in:

```
html, body {
    background: #000;
    color: #fff;
    overflow: hidden;
    touch-action: none;
    -ms-touch-action: none;
    padding-top: constant(safe-area-inset-top);
    padding-top: env(safe-area-inset-top);
}
```

H) Now you can add platform IOS as described in chapter 7. After that is done, you can do one final step to fix this problem:

I) Open the *platforms/ios* folder, go to *ProjectName* folder, you will find a file named *ProjectName-Info.plist*. Open it with Notepad++

J) Add these lines before the line *</dict>:*

```
<Key>UILaunchStoryboardName</key>
<String>CDVLaunchScreen</string>
```

K) Now you can run the app on XCode and see it it works on IPhone X

So what did we basically do?

We set the viewport of our game / app to cover max of the area on screen. This should remove the black bars we get on the IPhone X. We've added some splash screen images of the resolution of IPhone X's screen, because they somehow create white bars during launch and keep the app/game shrink. And finally we set the top padding for a few pixels (safe area) down, since the status bar of IPhone X when gone makes the app go up and then this piece looks bad.

That's about it for the IPhone X fix. IF you didn't manage to fix it, or partially fixed it, pm me on Discord and we'll go through it together, link down below.

I haven't talked about signing, aligning and other additional steps to finish the games APK, so that will be eventually updated in this guide.

If you have further questions, or trouble setting this up and running it, please contact me on our Construct 2 Discord: https://discord.gg/dZDU7Re