



Changes Needed for Viva Stampede Source Code

Thanks for purchasing a license for the Viva Stampede source code. The team here at Bright Newt hopes that your match-three games makes a lot of money in the App Store!

Your license entitles you to distribute the code in binary form only (by submitting your app to Apple for review). The assets spreadsheet shows all of the new assets that you must create, highlighted in blue, including new character art and backgrounds.

You are free to reuse the music, sounds, animations, and menu buttons. All of the sound effects are from the Public Domain.

To achieve success with your app, you should pick a unique theme, title, and color palette, and try to differentiate your app as much as possible from the original. Creating your own app using the Viva Stampede code constitutes your agreement to these terms.

+++++

I. General Instructions for Reskinning the App

1. Create new App ID on developer.apple.com.
Go to Member Center -> Certificates, Identifiers & Profiles -> iOS Apps -> Identifiers.
Create new App ID, corresponding to your project.
2. Create development profile on developer.apple.com.
 - If you want to use iOS Team Provisioning Profile (wildcard development profile with '*' App ID) for development and testing, skip the development profile creation.
 - Go to Member Center -> Certificates, Identifiers & Profiles -> iOS Apps -> Provisioning Profiles.
 - Create new development provisioning profile for your App ID.
 - Download and install it.
3. Create distribution profile on developer.apple.com.
 - Go to Member Center -> Certificates, Identifiers & Profiles -> iOS Apps -> Provisioning Profiles.
 - Create new distribution provisioning profile for your App ID.
 - Download and install it.
4. On itunesconnect.apple.com go to Manage Your Apps. Create new application and configure it properly.

5. On itunesconnect.apple.com select your app and configure in-app purchases.
 - Full (ad-free) version
 - Selectable backgrounds
 - In-game currency packs for 500, 1000, 3000, 7500, 20000, 50000 tokens or coins
6. On itunesconnect.apple.com configure new version of your application and make it ready to upload.
7. Download the project source code and open it in Xcode.
8. Tap the project name in the project tree and change it to one that is relevant to your app's theme. Agree, when Xcode asks to rename corresponding entities.
9. Open Schema Manager (Product -> Scheme -> Manage Schemes). Remove existing scheme. Autocreate the scheme again; its name will be changed to the project name.
10. Open the project info and replace the App ID, Bundle Display name and Bundle version.
11. Select the development and distribution provisioning profiles.
 - Go to Build Settings.
 - Find setting "Code Signing Identity".
 - For Ad-Hoc and Debug options, select your development profile.
 - For Distribution and Release options, select your provisioning profile.
12. Replace the icon files in Graphics folder of the project.
13. Read the third party SDKs (see below).
14. Change the link for rating.
 - Search for "define rateLink" and place the iTunes link for rating this app here (ProjectName-Prefix.pch).
15. Change IAP identifiers.
 - Search for "IAP macros" and place the IAP identifiers here (ProjectName-Prefix.pch).
16. Change the references to the application name.
 - Search for app name (currently - "Viva Stampede") and edit all the corresponding string constants.
 - Test the application and check if all the test messages are appropriate. If no, search them and edit corresponding string constants.
17. Change the references to email addresses.
 - Search for "define supportEmail" and place support email address here (ProjectName-Prefix.pch).
18. Redesign user interface, if needed. The programmer is assumed to be familiar with editing xib files and replacing corresponding images for multiple devices.
19. Perform all the other changes needed.
20. Test the project. If needed, use the TestFlight for remote testing.
21. Archive the project (Product -> Archive).
22. From Organizer, select the archive and distribute it to App Store.

II. Instructions for Adding Third-Party SDKs

1. Download third-party SDKs and put them to the folder "SDKs." For legal reasons, no developer is allowed to distribute these SDKs as a part of licensing source code, so you need to put them back into the app yourself. Some of them may be accessible for downloading after registration (see below). In SDK's folder the following folders should exist:
 - Chartboost
 - Flurry iPhone SDK

- PlayHaven
 - PushWoosh
 - RevMobAds.framework
 - Tapjoy
 - TestFlight (optional)
2. If some link to any SDK is broken (red items in project tree), remove the group of this SDK from project tree and drag-and-drop the SDK folder from Finder to project tree again.
 - Select option "Create groups for any added folders."
 3. Check if the project is built correctly. If not, some SDKs are missing or not added.
 4. After reading some SDK the warning may appear, that old library search path cannot be found. If so, go to Build Settings, edit "Library Search Paths" setting and remove the old library search path, which is not automatically deleted when removing the library from the project tree.
 5. Chartboost - used for fullscreen ads.
 - Register on chartboost.com (if not registered).
 - Set up your app and get App ID and App Signature.
 - In project, search for "cb.appId" and "cb.appSignature" and place the ID and Signature here (AppDelegate.m).
 6. Flurry SDK - used for bottom banner ads.
 - Register on www.flurry.com (if not registered).
 - Set up your app and get App ID.
 - In this app, set up an ad space and get space name.
 - In project, search for "Flurry startSession" and place the ID here (AppDelegate.m).
 - Search for "define flurrySpaceName" and place the space name here (ViewController.m).
 7. PlayHaven - used for presenting popups after level completion, for fullscreen ads and for More Games Widget.
 - Register on www.playhaven.com (if not registered).
 - Set up your app and get App Token and App Secret.
 - Configure placements "level_complete" and "main_menu".
 - Configure More Games Widget for "more_games" placement (<http://help.playhaven.com/customer/portal/articles/585270-how-to-create-a-more-games-widget>).
 - In project, search for "define kPlayHavenID" and "define kPlayHeavenSecret" and place Token and Secret here (ProjectName-Prefix.pch).
 8. PushWoosh - used for sending push notifications.
 - Register on www.pushwoosh.com (if not registered).
 - Set up your app and get App Code.
 - In project, search for "Pushwoosh_APPID" and place the App Code here (ProjectName-Info.plist).
 - Configure your push certificate and export it to PushWoosh (<http://www.pushwoosh.com/programming-push-notification/ios/ios-configuration-guide/>). This step may be automated for Premium PushWoosh account. For testing, configure sandbox gateway; for release, configure production gateway.
 9. RevMob Ads - used for fullscreen ads.
 - Register on www.revmob.com (if not registered).
 - Set up your app and get App ID.
 - In project, search for "RevMobAds startSessionWithAppID" and place the ID here (AppDelegate.m).
 10. TapJoy - used for collecting statistics.

- Register on advertisers.tapjoy.com (if not registered).
- Set up your app and get App ID and Secret Key.
- In project, search for "TapjoyConnect requestTapjoyConnect" and place the ID and Secret Key here (AppDelegate.m).
- TestFlight (optional) - used for remote testing and sending debug logs.
- In project, search for "define USE_TESTFLIGHT" (in ProjectName-Prefix.pch). If you do not want to use TestFlight, comment this line (in) and skip other steps.
- Register on testflightapp.com (if not registered).
- Set up your app and get App Token.
- In project, search for "TestFlight takeOff" and place the Token here (AppDelegate.h).
- For test builds, uncomment the definition of USE_TESTFLIGHT macro. Otherwise, comment it. Do not forget to comment it before building the release version.