

Planet Shooter

created: 10/14/2014

latest update: 10/18/2014

by: Hybera LLC

↓

email: axelsyrex@gmail.com (<mailto:axelsyrex@gmail.com>)

Copyright Hybera LLC 2014
made with the [Documenter v2.0](http://rxa.li/documenter)
(<http://rxa.li/documenter>)

Table of Contents

[Start](#)

[Introduction](#)

[Requirements](#)

[Installation](#)

[Game Properties](#)

Introduction

Thank you for purchasing Planet Shooter game.

This documentation should help you to set up a fully functional bubble shooter game with all of the features required to deploy a successful applicaiton.

If you need further assistance, please do not hesitate to contact me in ChupaMobile!

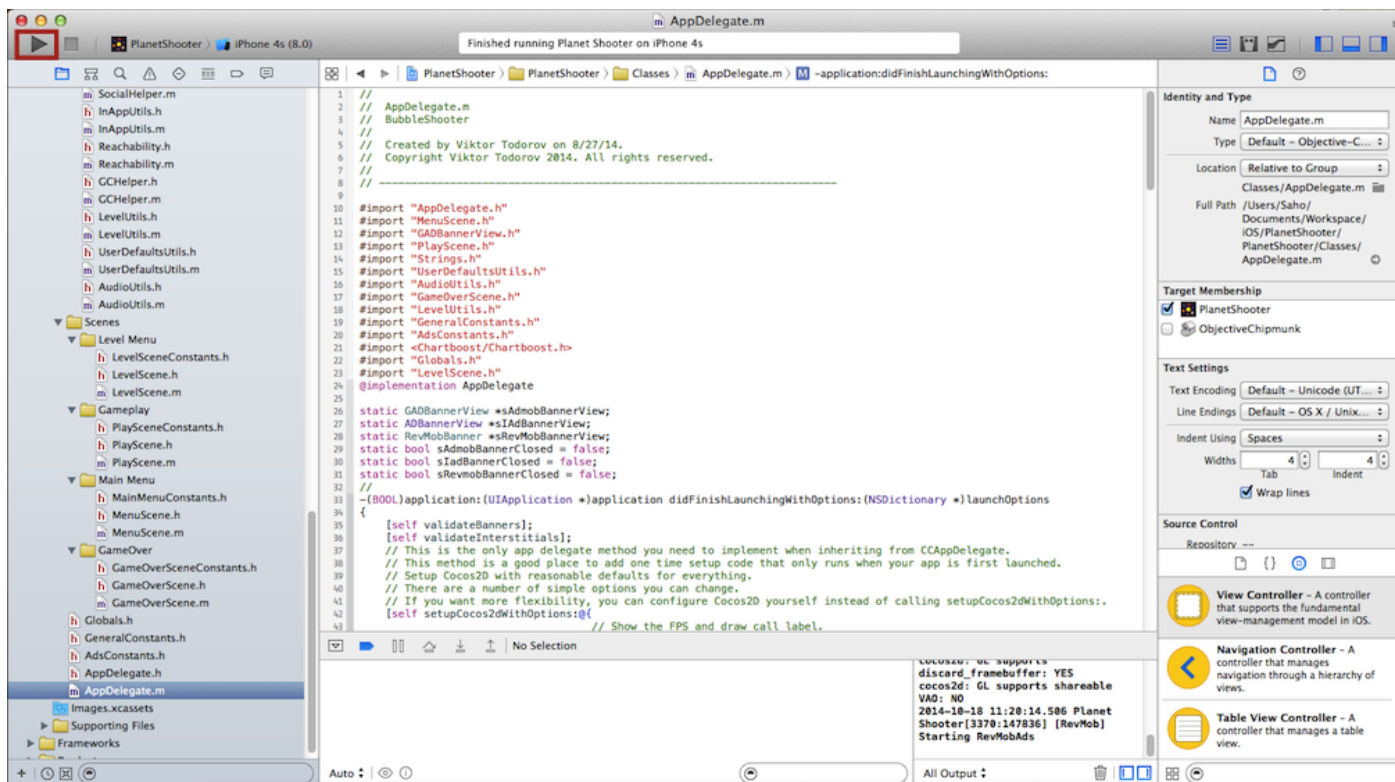
Requirements

In order to run and test the app you need:

- a machine with OS X Mavericks
- minimum iOS version required to run the app is iOS 6
- XCode version 6 or later

Installation

1. Unzip the archive that you have downloaded from ChupaMobile
2. Open "Planet Shooter" folder
3. Open **PlanetShooter.xcodeproj** file
4. Press the play button in the top left corner to run the game.

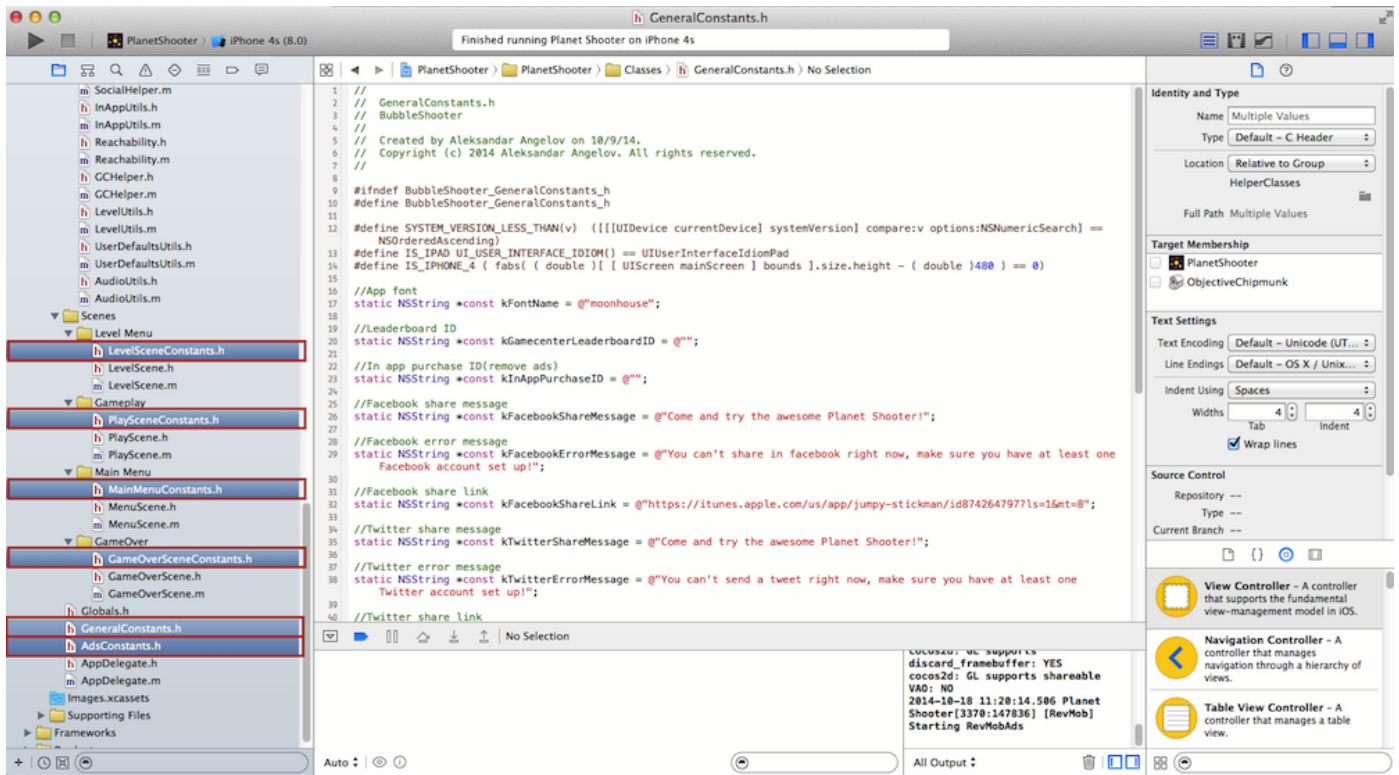


Game Properties

Constants

There are 5 header files that contain constants for different properties:

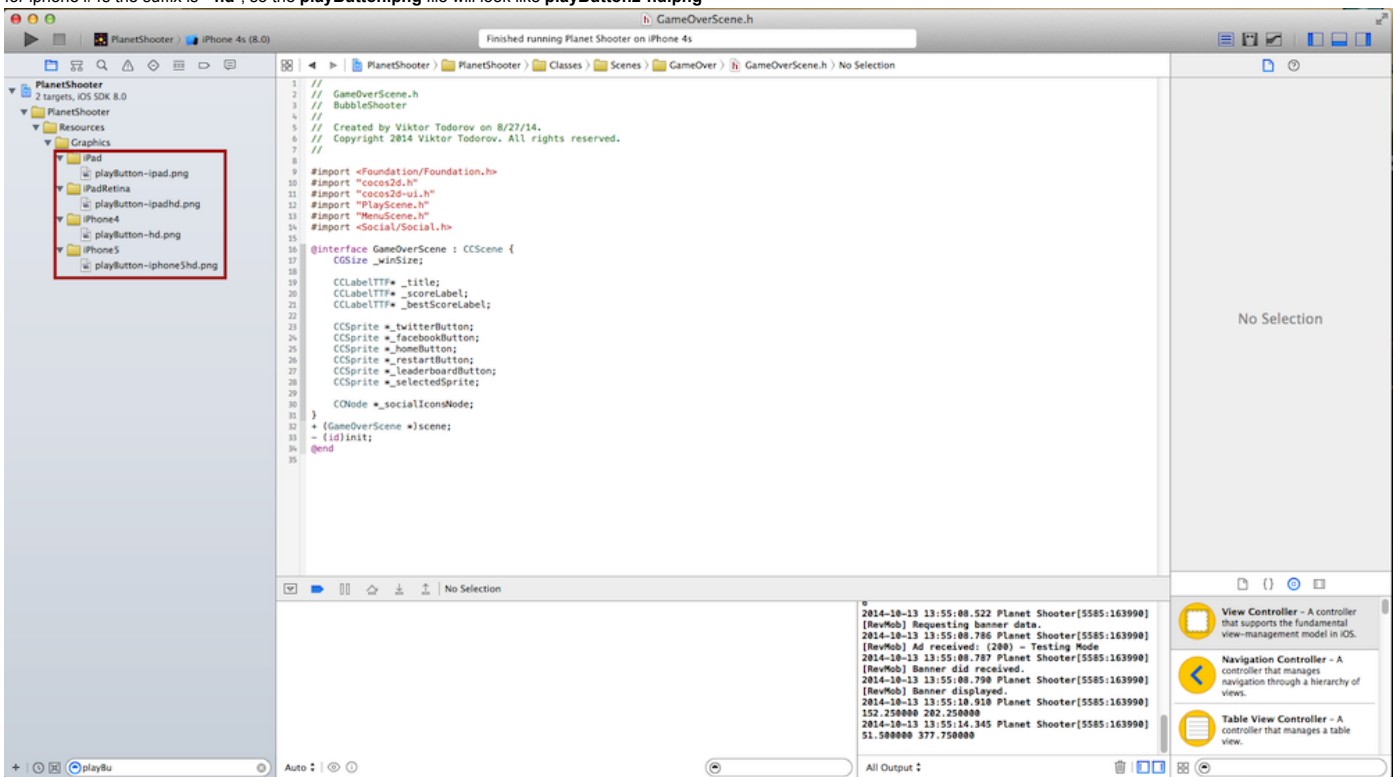
- **PlaySceneConstants.h** contains constants responsible for the elements(images,buttons,...) in the Play Scene view. They include properties such as image names,position offsets,animation durations, bubble fly speed and others.
- **MainMenuConstants.h** offers the same set of features as **PlaySceneConstants.h** does, but for the Main Menu Scene View.
- **GameOverSceneConstants.h** offers the same set of features as **PlaySceneConstants.h** does, but for the Game Over Scene View.
- **LevelSceneConstants.h** offers the same set of features as **PlaySceneConstants.h** does, but for the Level Scene View.
- **GeneralConstants.h** contains constants that store data about the app font,gamecenter leaderboard id, in app purchase id, rate app id, audio file names,etc.
- **AdsConstants.h** contains constants that store data about the ad banner or interstitial id.



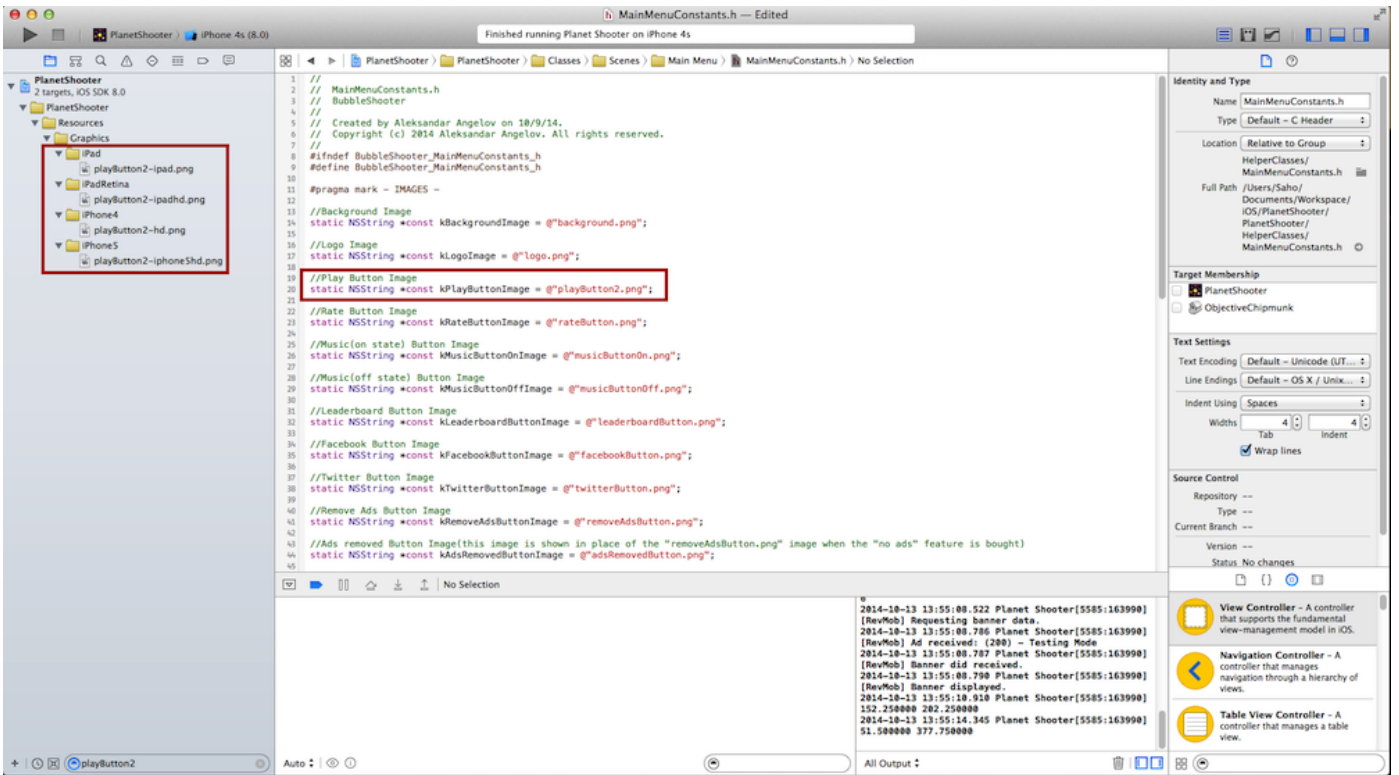
Change the graphics

In order to change an image you have to take the following steps:

1. Provide an image with all the necessary resolutions for each device. For example if you have an image file called **playButton2.png**, you need to import 4 different files in the project for that image. Those 4 images should be resized to a dimension that is suitable for the device they are used in. Each file has a different suffix that corresponds for each device:
 - for ipad retina the suffix is "-ipadhd", so the **playButton.png** file will look like **playButton2-ipadhd.png**
 - for ipad the suffix is "-ipad", so the **playButton.png** file will look like **playButton2-ipad.png**
 - for iphone5/5s/6/6+ the suffix is "-iphone5hd", so the **playButton.png** file will look like **playButton2-iphone5hd.png**
 - for iphone4/4s the suffix is "-hd", so the **playButton.png** file will look like **playButton2-hd.png**



2. After you import the 4 images required for each device, you need to find the **NSString** constant that holds the name of the image that you want to change and rename to the one that you have just added. In our example we are changing the play button in the Main Menu, so open **MainMenuConstants.h** and rename the **kPlayButtonImage** constant value to **"playButton2.png"**.

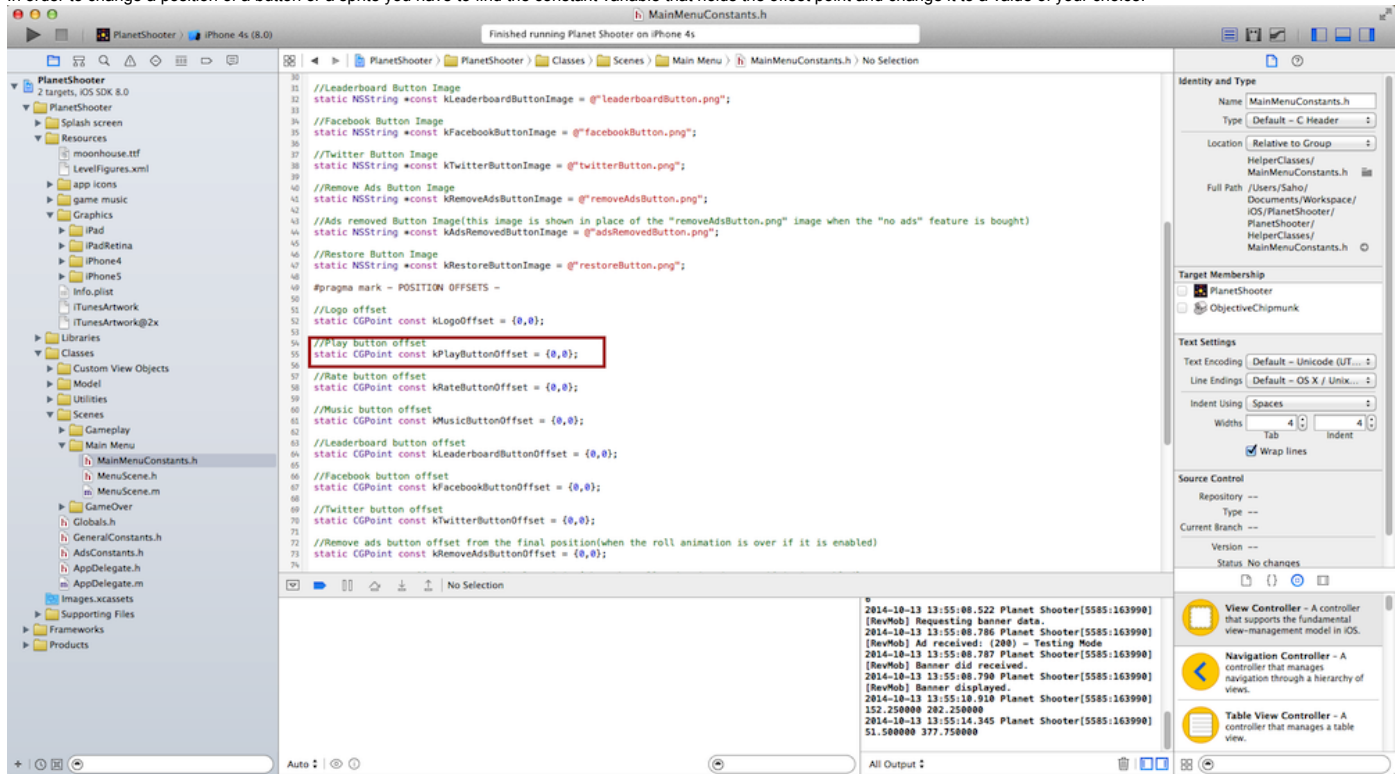


3. You can now run the app and see the modification.

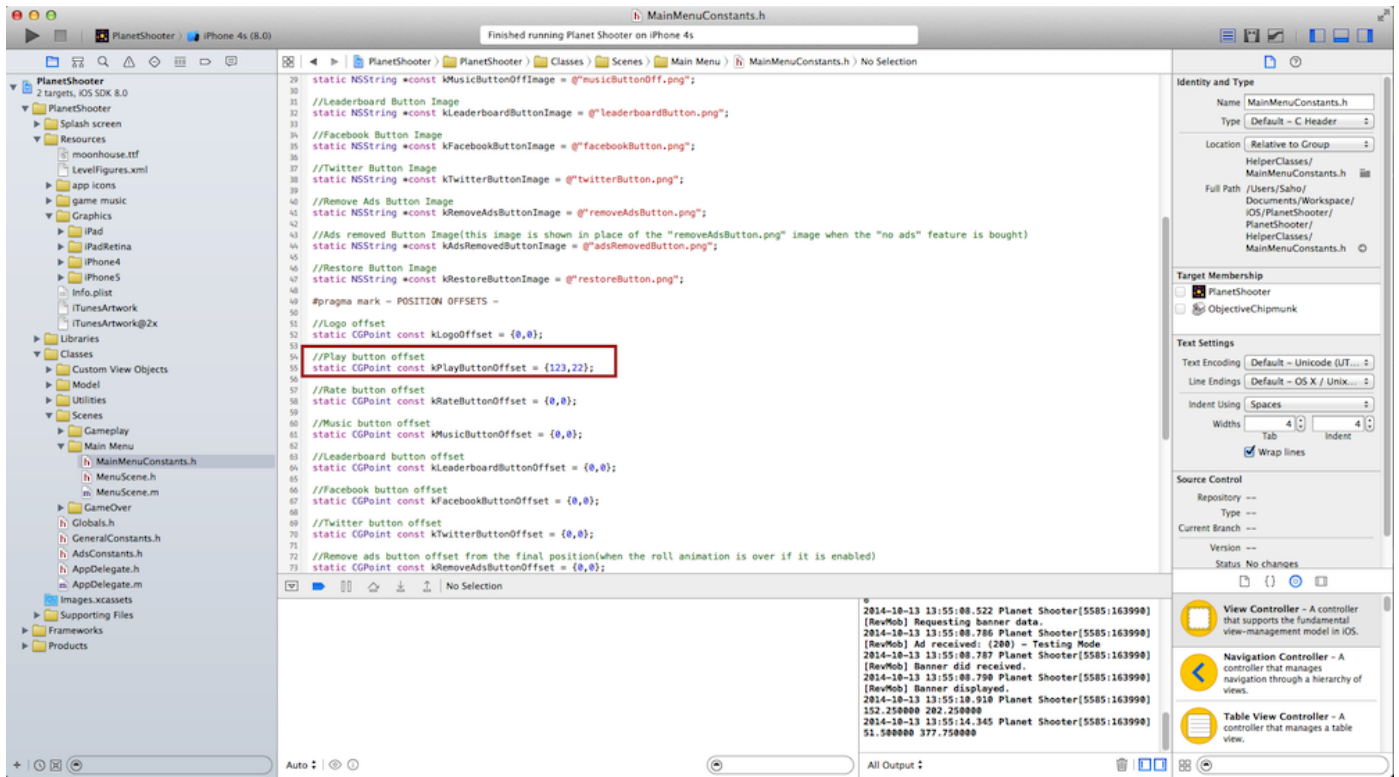
In order to not show a button or a sprite just set the constant variable that holds the name to the image it uses to nil. For example if you do not want the play button in Main Menu to be visible, set the **kPlayButtonImage** constant in **MainMenuConstants.h** to nil.

Change the positions

In order to change a position of a button or a sprite you have to find the constant variable that holds the offset point and change it to a value of your choice.



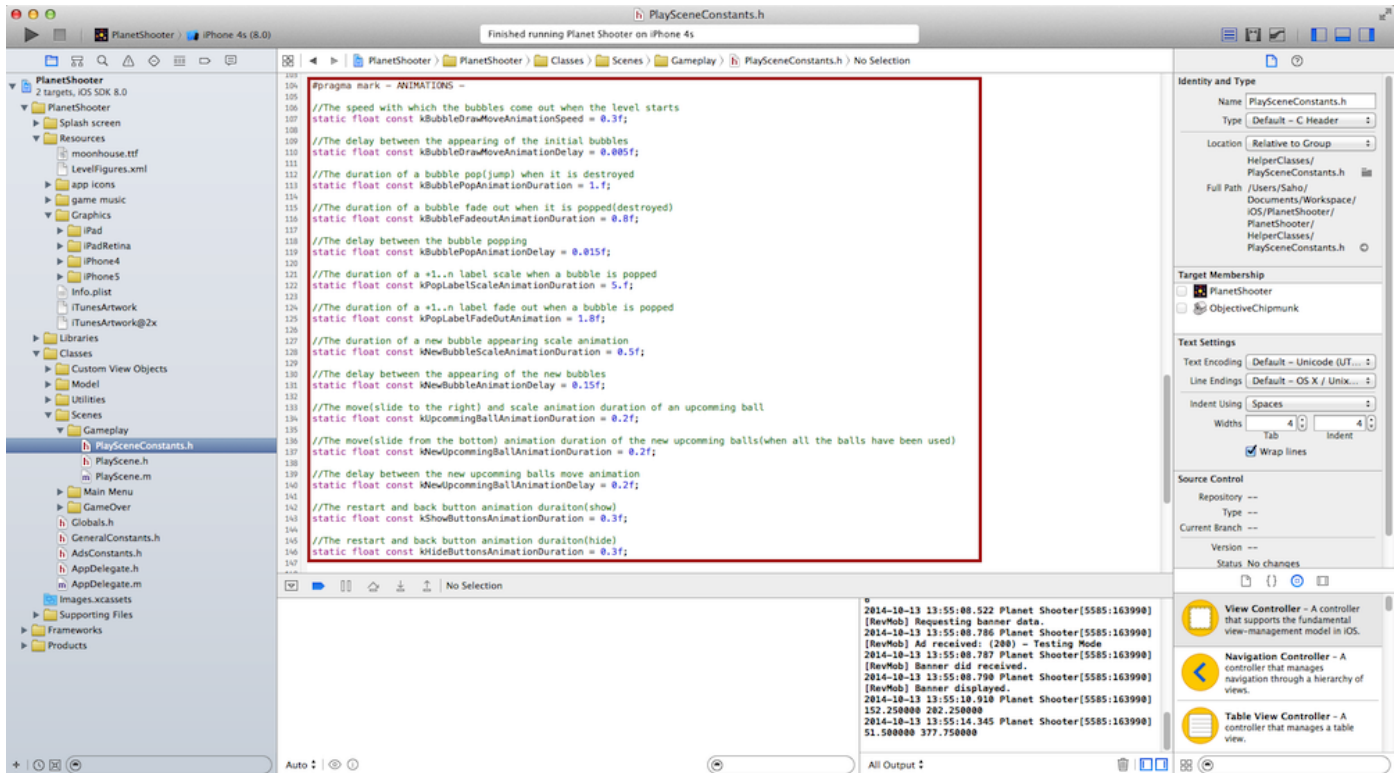
For example if you want to change the position of the play button in Main Menu Scene, change the **kPlayButtonOffset** value to a position of your choice.



NOTE that the change to the offset constants is an offset from the original position and sometime it may affect some other elements, if they are dependend to the position of the changed element.

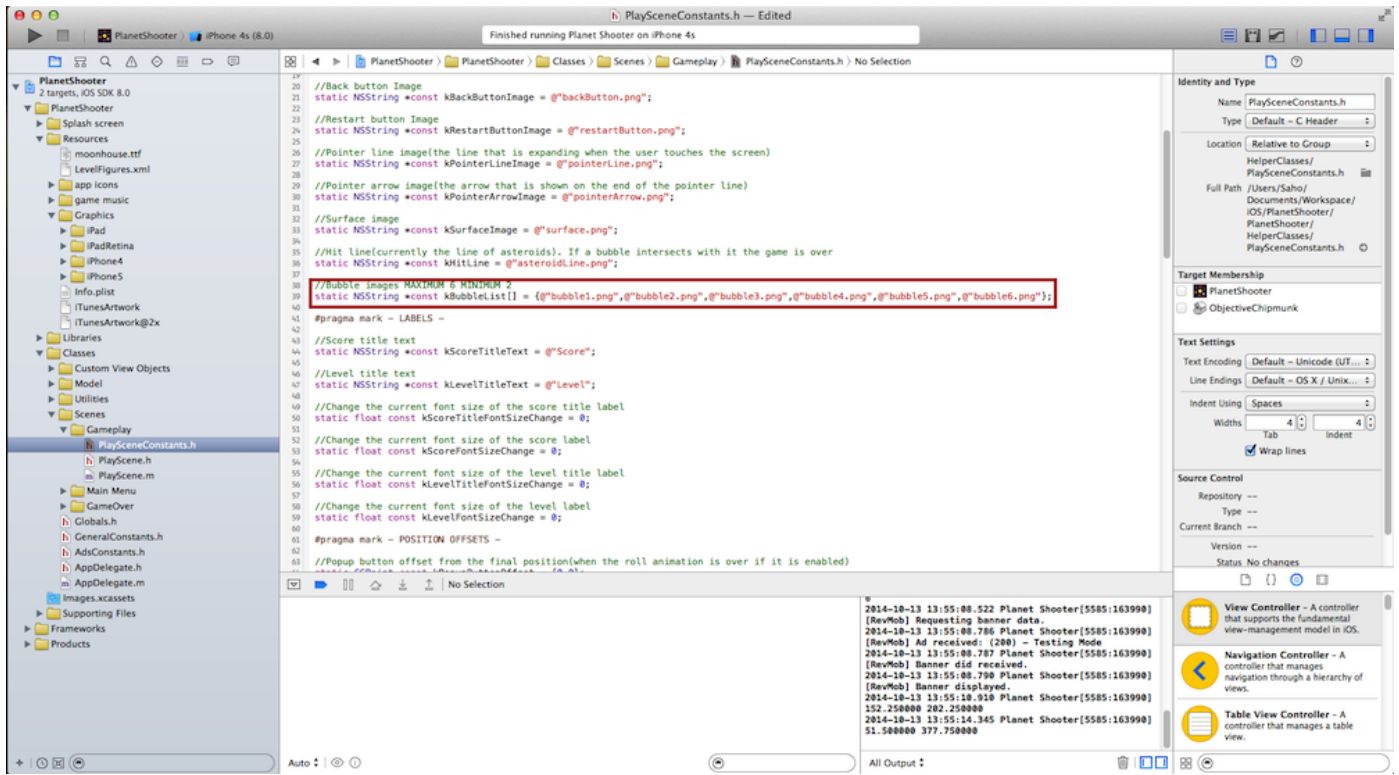
Modify the animations

In the constants file responsible for the play view you can find a section called Animations that contains data about the animation properties such as delay or durations. There you can change the duration of the bubble popup animation, or the rolling(show/hide) speed of the popup button and many others. Inspect the section to get a better overview.

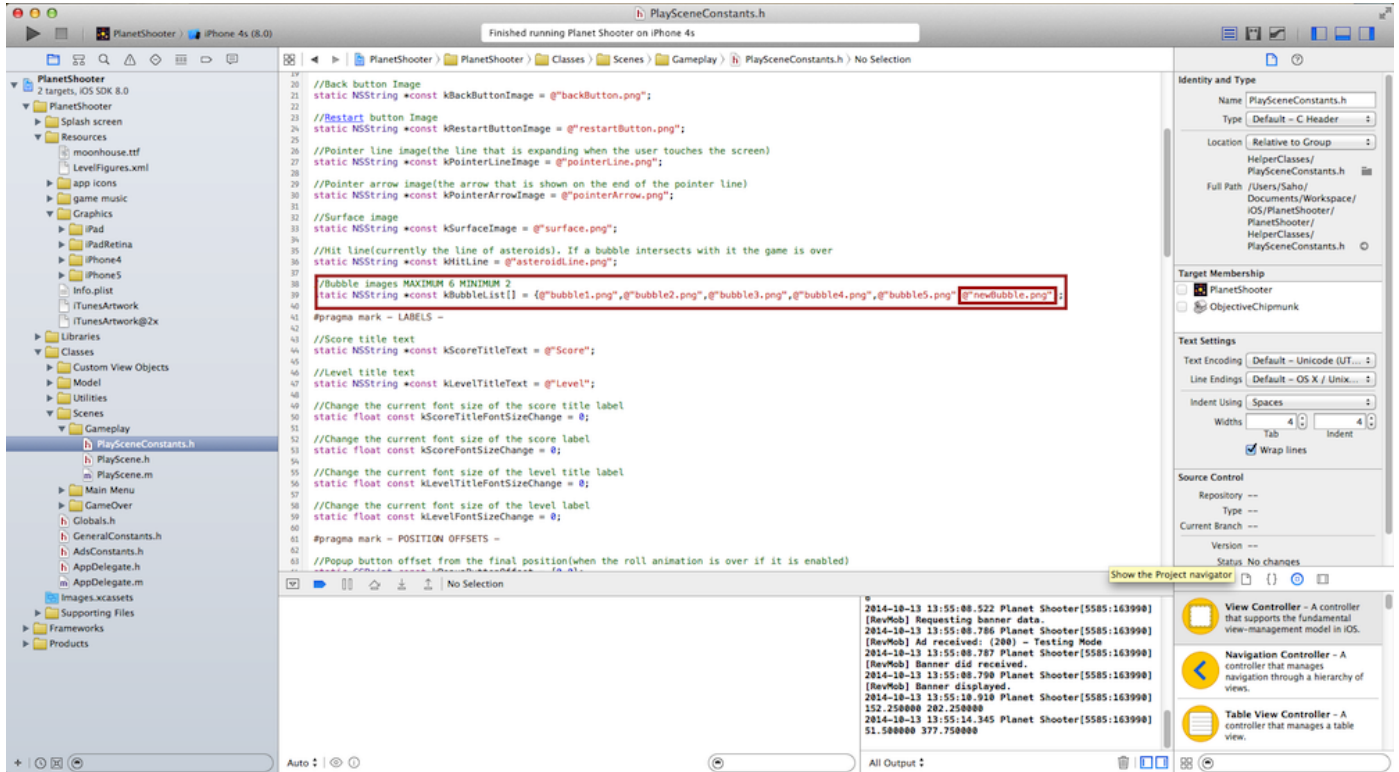


Modify the bubble images

The bubble images may be modified in PlaySceneConstants.h file. There you will find a constant array called kBubbleList that holds all 6 bubble image names.



To change a bubble image you can repeat the procedure in the "Change the graphics" section.



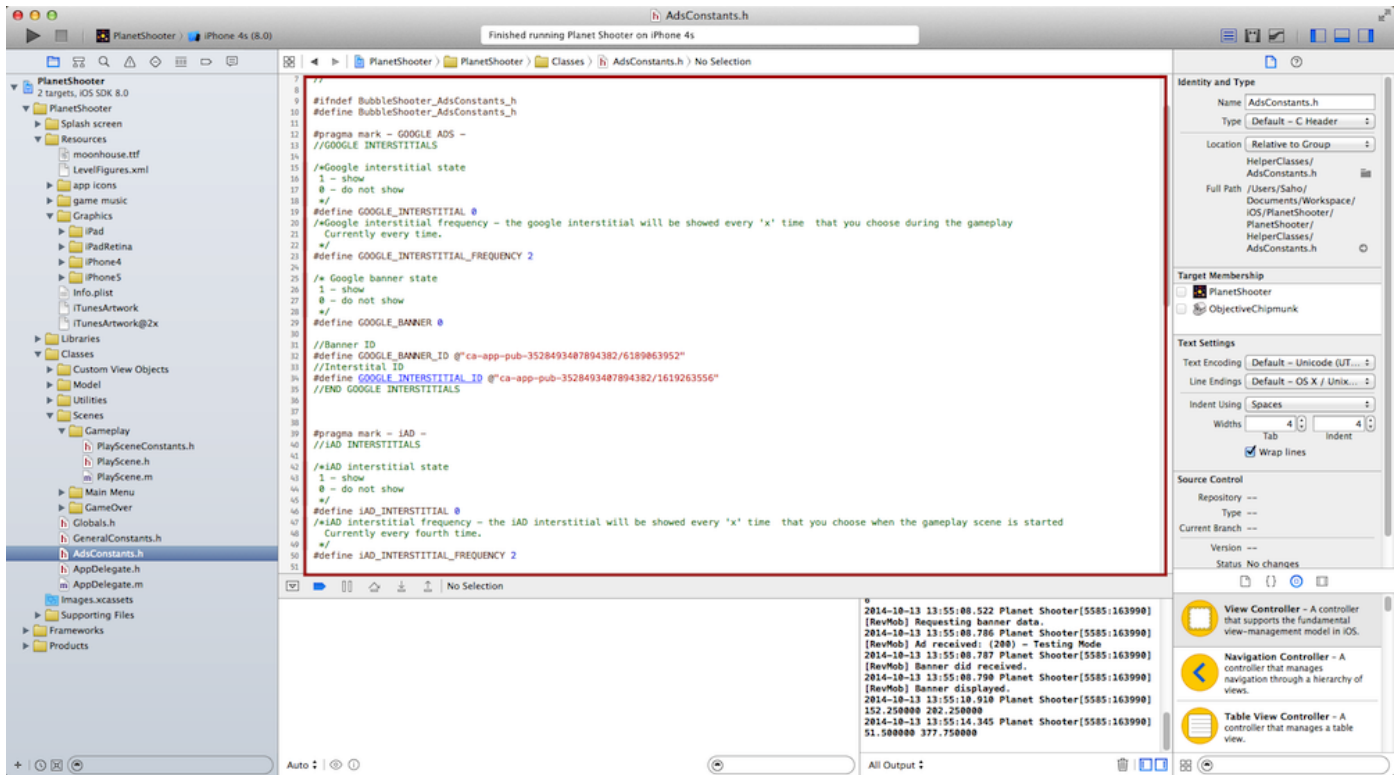
NOTE THAT THE MAXIMUM NUMBER OF BUBBLES SUPPORTED IS 6 AND THE MINIMUM IS 2!

ADS

There are 4 different types of ads supported - Google ads, Revmob, iAds, Chartboost. You can enable banners and interstitials in the **AdsConstats.h** file.

GOOGLE ADS

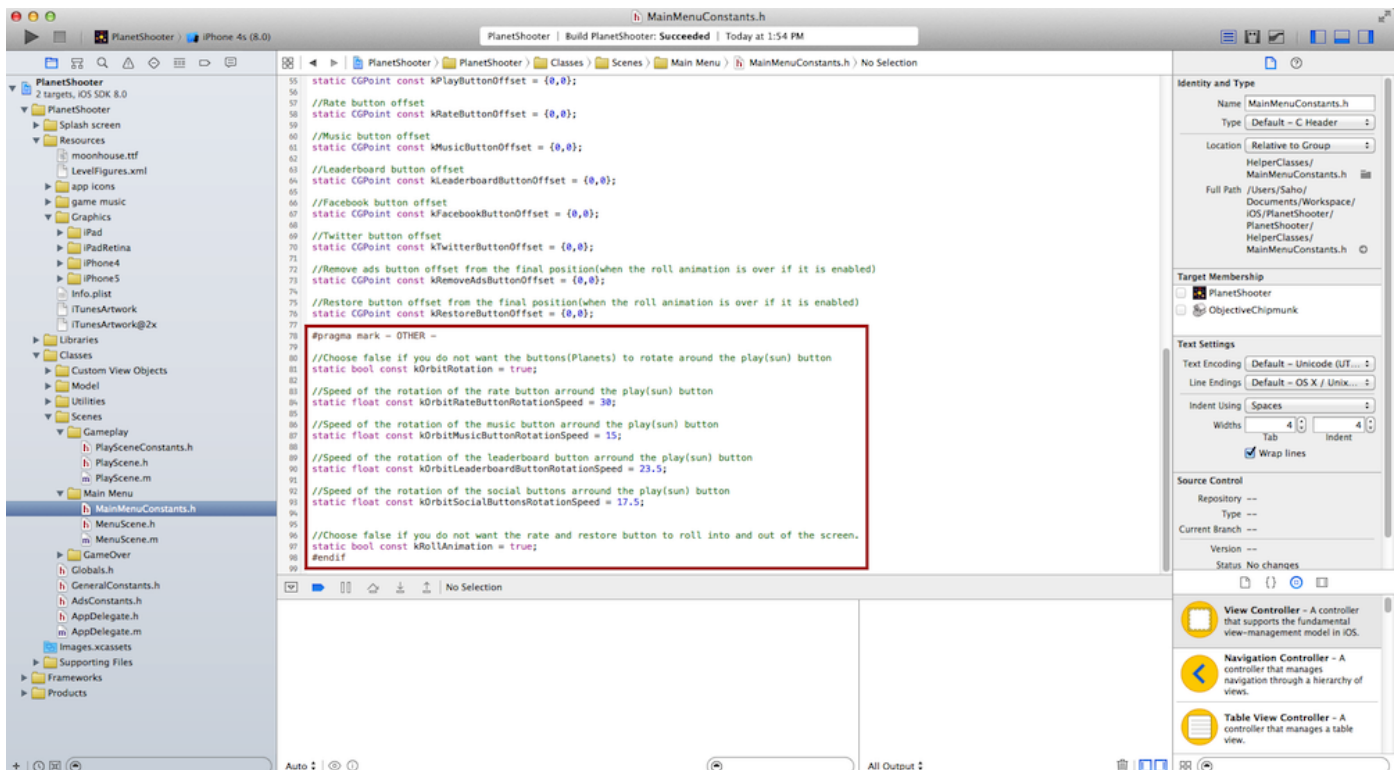
- To enable the google banner just set the **GOOGLE_BANNER** macro to 1 and the **GOOGLE_BANNER_ID** to the banner id provided by google.
- To enable the google interstitials just set the **GOOGLE_INTERSTITIAL** macro to 1 and the **GOOGLE_INTERSTITIAL_ID** to interstitial id provided by google.
- The interstitial is shown only in the Play Scene view. You can set the frequency of the interstitial popup by changing the value of **GOOGLE_INTERSTITIAL_FREQUENCY** e.g. if you choose 1, the interstitial will be shown every time the play scene has been started, if you choose 2 every second time and so on.



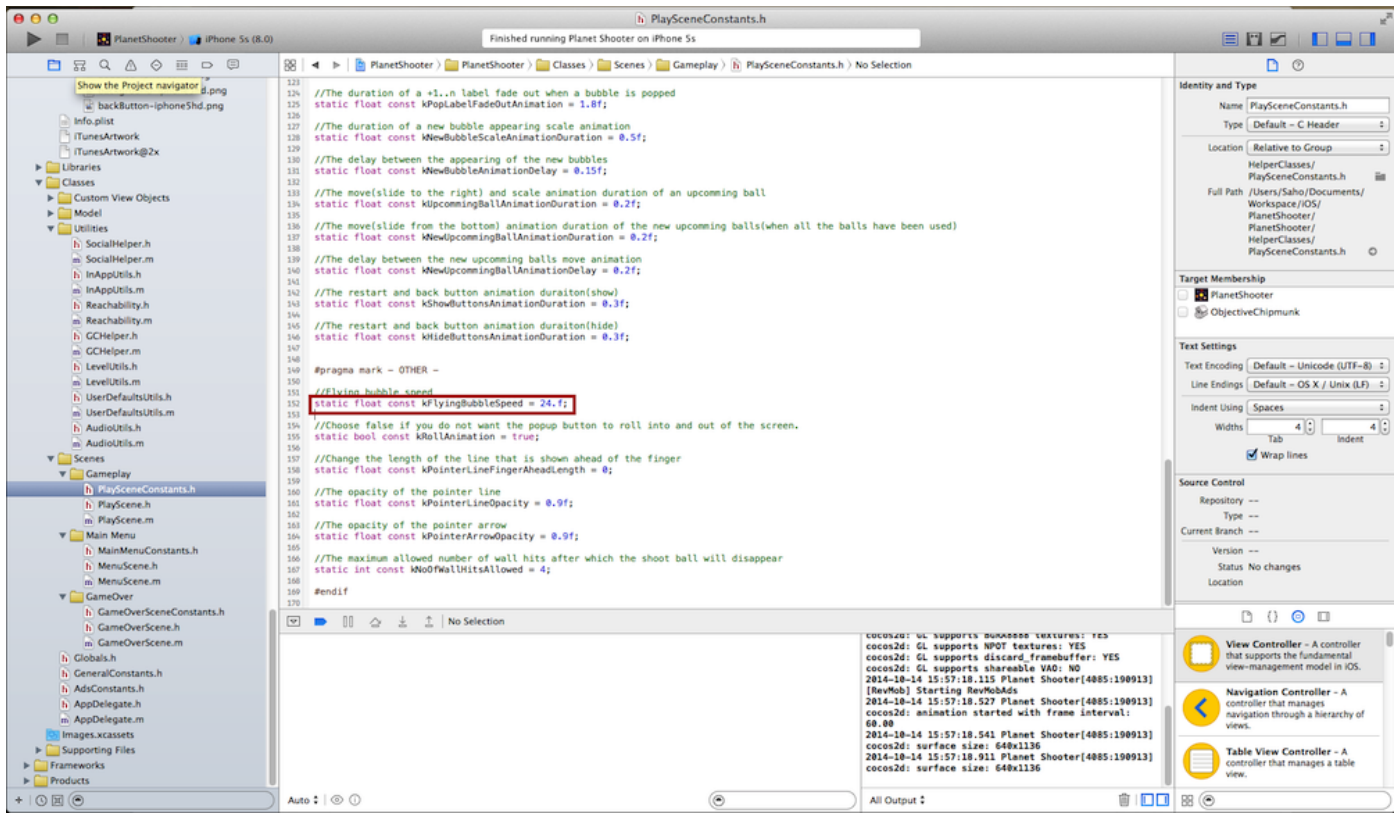
NOTE THAT YOU CAN PICK ONLY ONE BANNER AND ONLY ONE INTERSTITIAL PLATFORM. YOU CAN MIX THEM(GOOGLE BANNER WITH REVMOB INTERSTITIAL or CHARTBOOST INTERSTITIAL WITH iAD BANNER or etc), BUT YOU CAN NOT USE GOOGLE BANNER AND REVMOB BANNER SIMULTANIOUSLY, THE SAME RULE APPLIES TO INTERSTITIALS!

Other

In each of the 3 header files that contains constants data for the views you will find a section called OTHER. This section contains constants that are responsible for different set of features, for example you can stop the buttons rotation around the sun in the Main Menu view and you can modify the flying speed of the shoot ball. Inspect the files to get a better overview.



To change the speed of the flying bubble open **PlaySceneConstants.h** and find the constant **kFlyingBubbleSpeed** and change its value to a number of your choice.



Video

Watch the [video tutorial](https://www.youtube.com/watch?v=qcZIX2vd_xY&list=UUHfey3wd73KGk1Mk-LeGogq) about the game properties if something is still unclear to you.

Levels

Brief Explanation

The levels used in game are loaded from a xml document, but do not worry, it is a simple structured document used to hold data about the bubble figure.

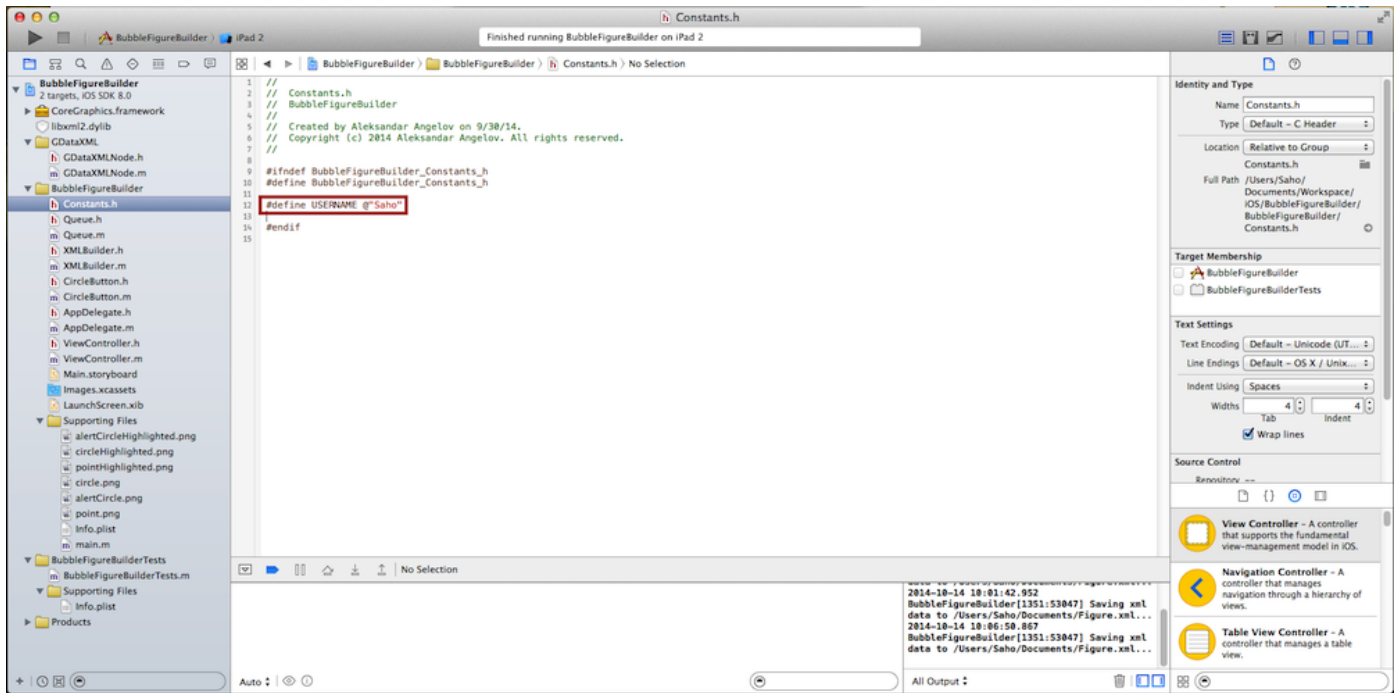
The level hierarchy goes from the top of the document to the bottom, which means that the most top figure is level 1, the next one is level 2 and so on. If you have 4 levels loaded and if the user has completed all of them, then the levels starts to repeat, which means that level 5 will be level 1, level 6 will be level 2 and so on.

The scoring in the game works as follows - for every bubble popped you receive 1 point in level 1, in level 2 you receive 2 points, in level 3 you receive 3 points and so on. There are no combos currently, but can be added later.

Create and import level

If this is your first run of the level builder project, take the following steps:

1. Unzip the **BubbleFigureBuilder.zip** file
2. Open the **BubbleFigureBuilder.xcodeproj**
3. Open **Constants.h** file and change the **USERNAME** macro to your OS X username. If you do not know your username open the terminal and type **"whoami"** and press return and you will get the username. Currently the username is **"Saho"**.



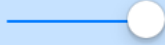
!!!NOTE!!! DO NOT EDIT ANYTHING IN THE STORYBOARD!!!

To create a level take the following steps:

1. Open the bubble figure builder xcode project if you did not do it yet, choose iPad 2 simulator and press the play button in the top left corner to run the builder.
2. You will see a field filled with black dots.

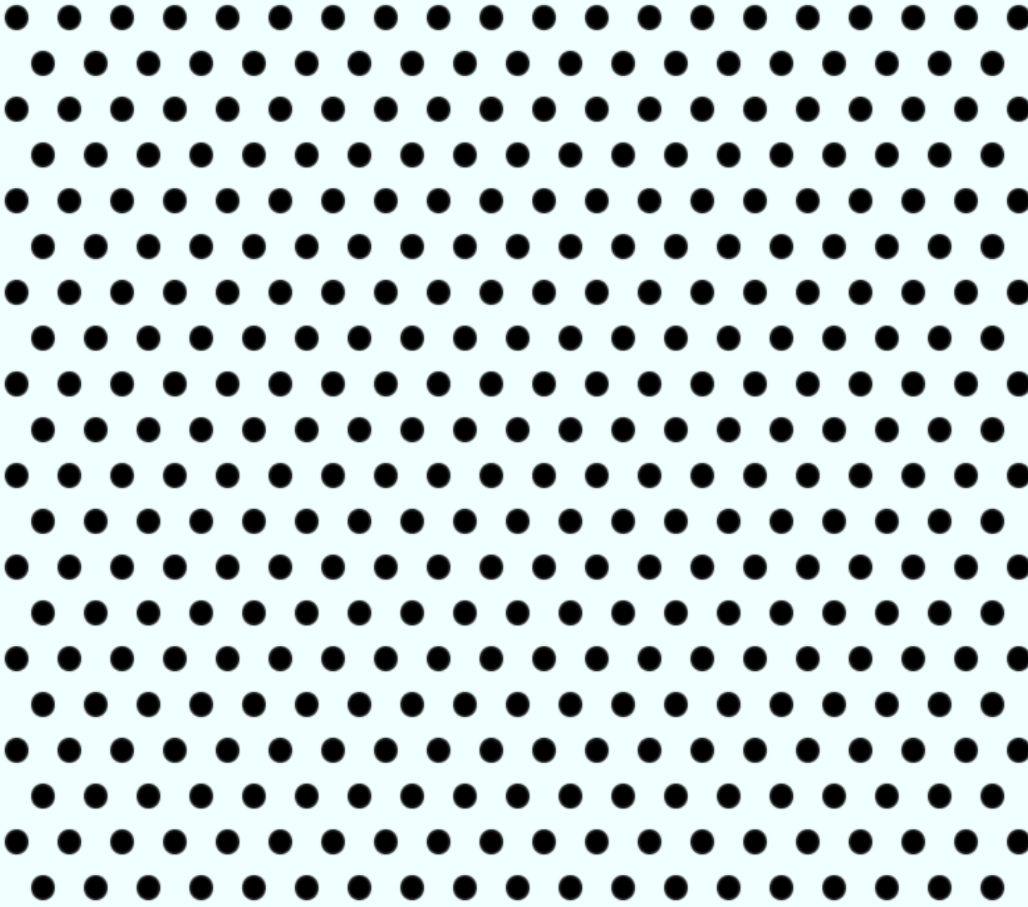
GENERATE

Different bubbles



RESET

6



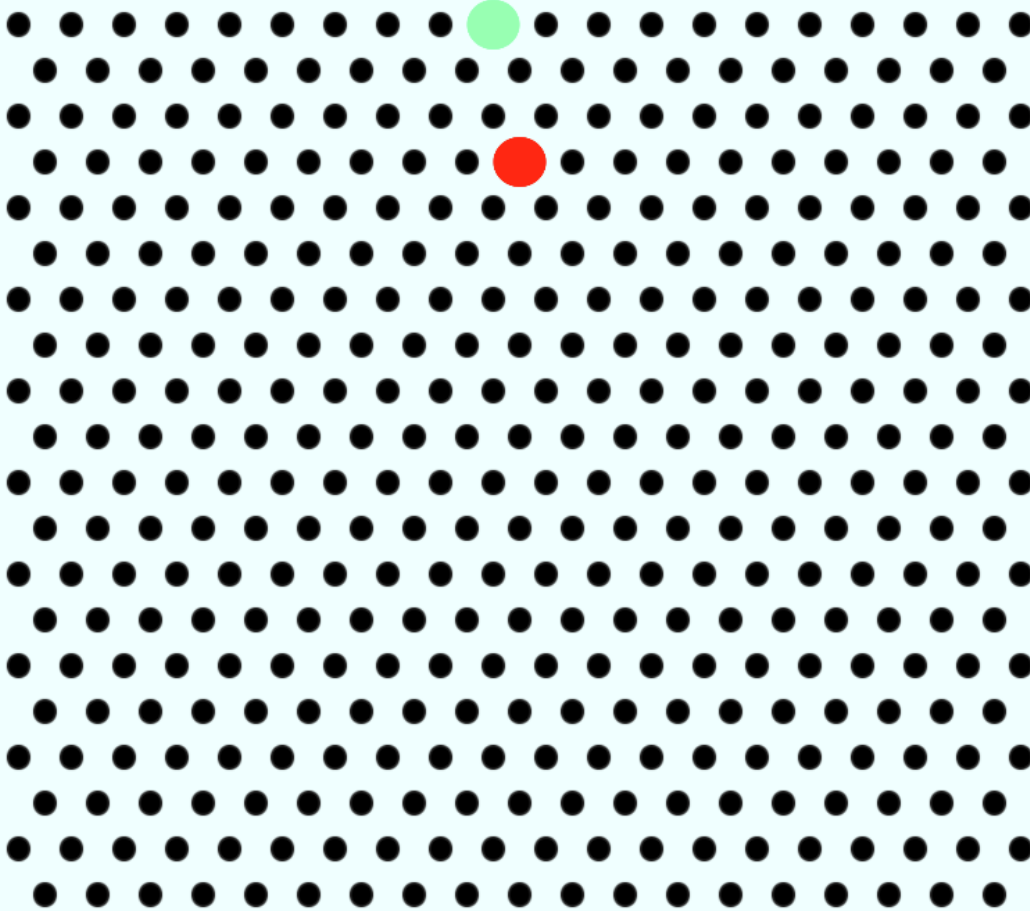
Click on some of the black dots and you will see a green or red circle, this is a bubble that will be shown in the game.

GENERATE

Different bubbles

RESET

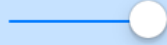
6



Firstly some explanation about the purpose of the red and green color circle- in order to make the game more realistic, there has to be some base(anchor) line of bubbles to which all the other bubbles have to be connected. In this game that happens to be the most top line of black dots. When you pick a black dot that has no connection to a base circle(bubble), then a red circle will appear instead of green.

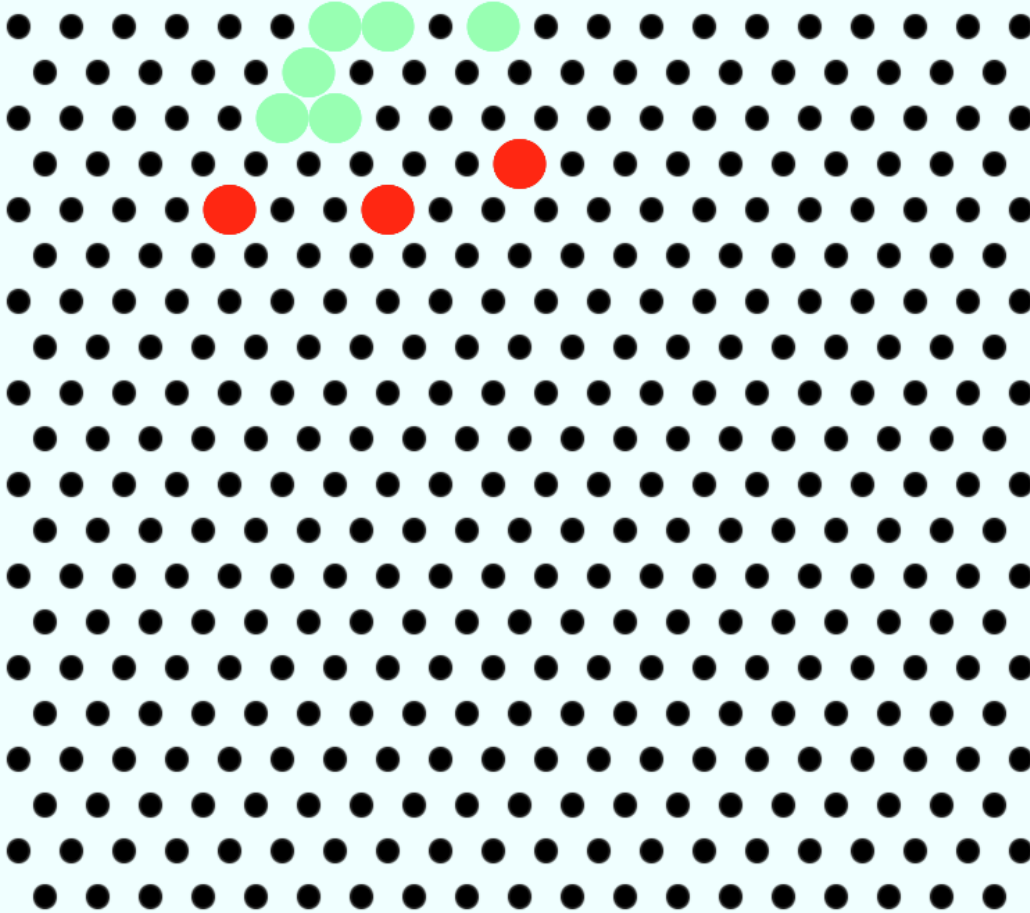
GENERATE

Different bubbles



RESET

6



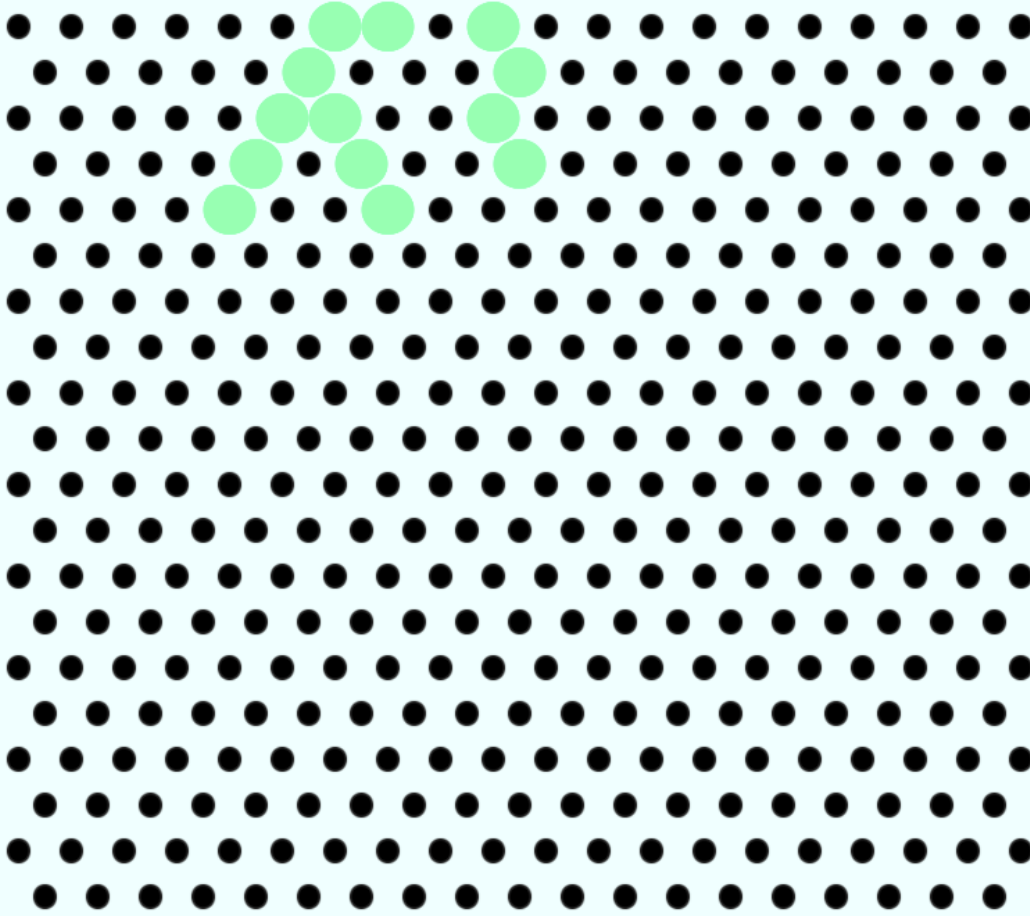
This is a warning that there are circles(bubbles) that are floating in space without anchor and the level cannot be generated until it is fixed. To fix it just connect the red bubbles to the base and they will change their color to green.

GENERATE

Different bubbles

RESET

6

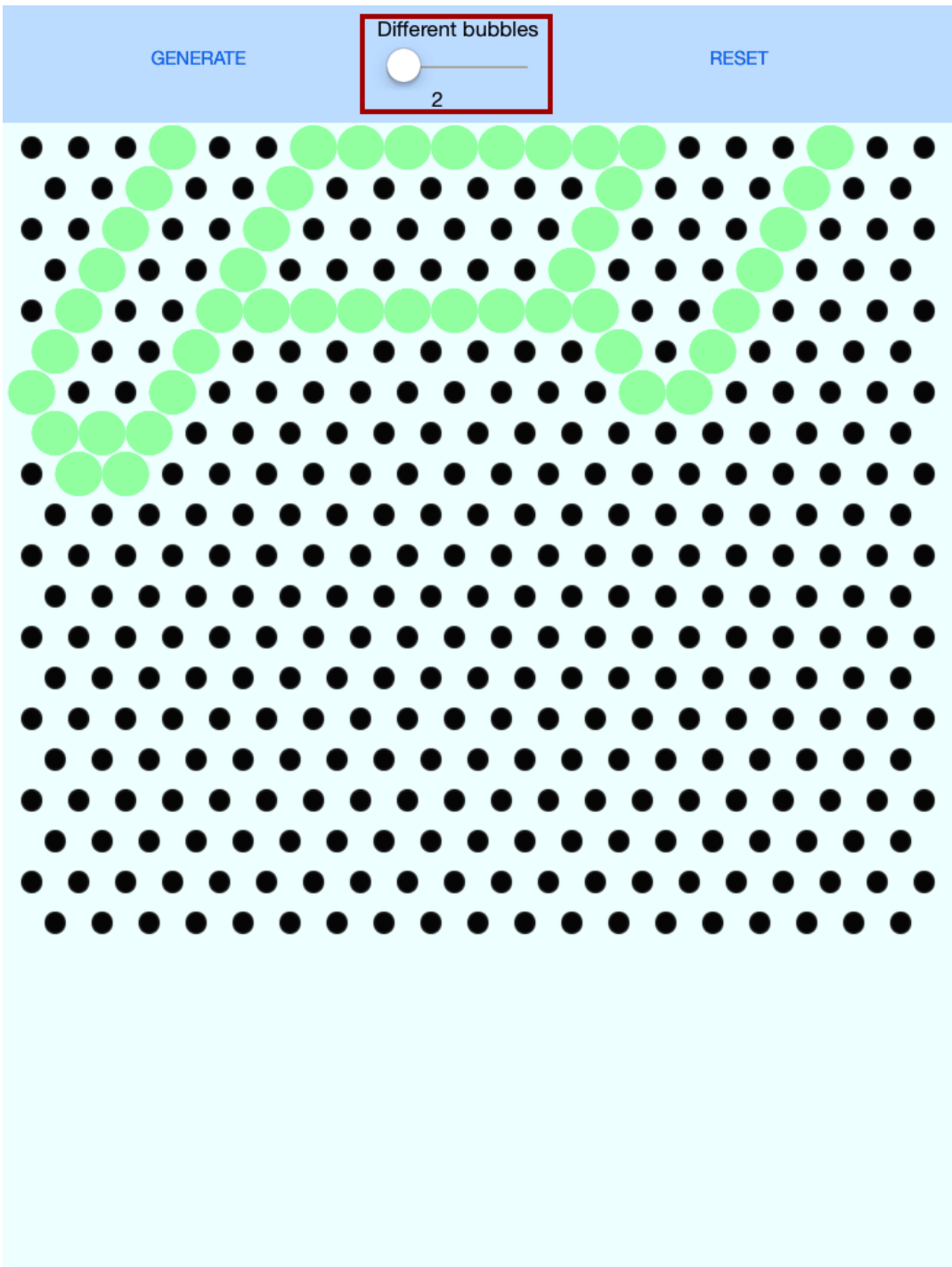


3. Feel free to make what figure you like.

GENERATE Different bubbles 6 RESET

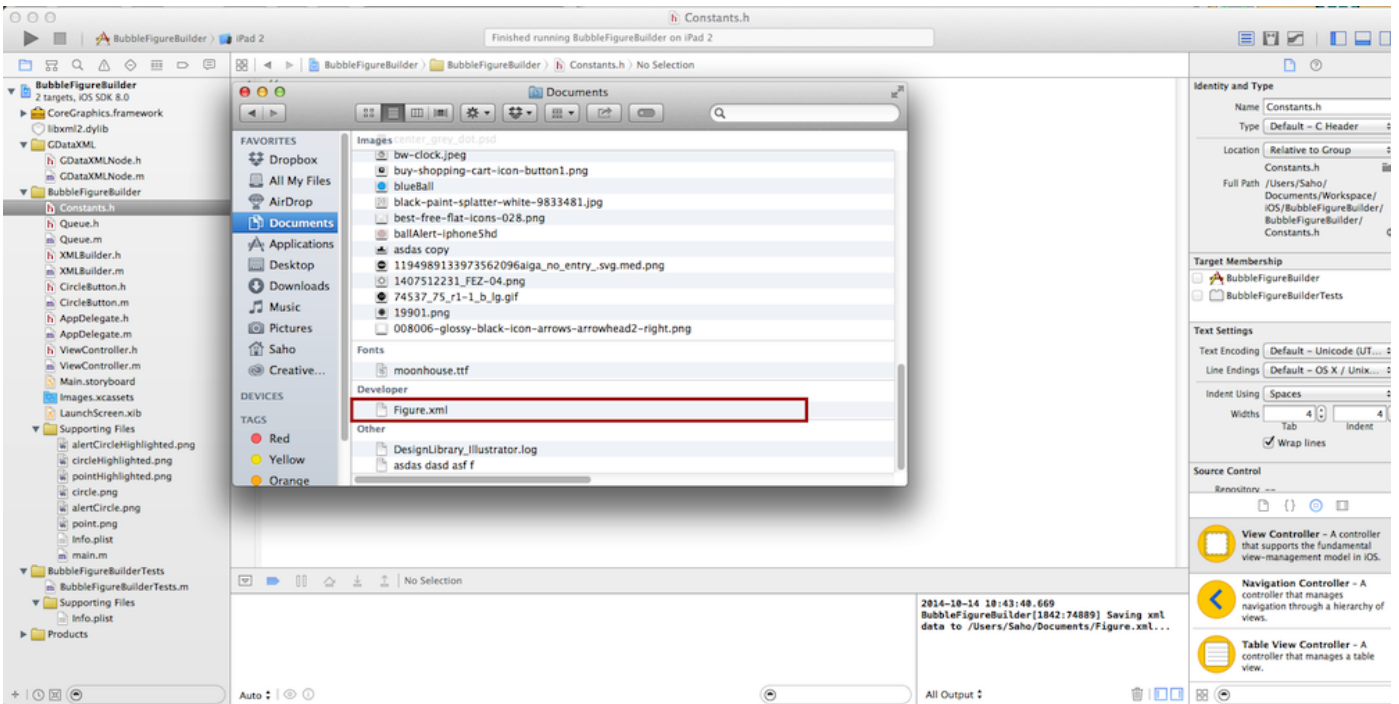
The image shows a 20x20 grid of circles. The top 6 rows contain a mix of black and green circles, representing 'different bubbles'. The remaining 14 rows contain only black circles. Above the grid is a control bar with 'GENERATE' and 'RESET' buttons, a slider labeled 'Different bubbles' set to 6, and a white circle on the slider.

4. After you are done and all of the circles are valid(green), then you can choose the number of different bubbles from the slider.

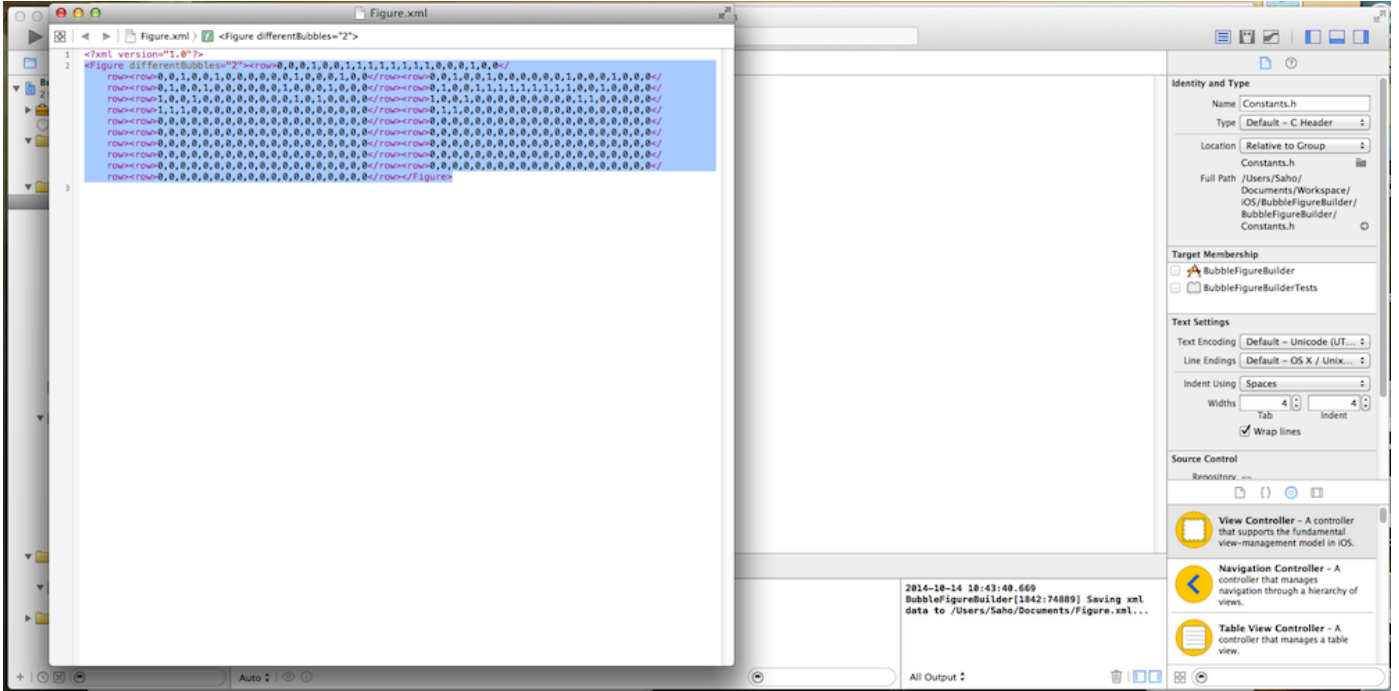


The meaning of this value indicates the number of different bubble types that will be used in the level. For example if you have 6 types(red,green,yellow,white,gray,orange) of bubbles and if you choose 2, then in the game you will see only red and gray bubbles, or green and orange, or white and yellow or any combination as long as they are 2, and this makes the level easier for the user(it is a good idea to use this approach in the first levels). For a bigger challenge choose a bigger value for the different bubbles. Currently the maximum is 6 and the minimum is 2.

5. To generate the level you just created press the generate button, or if you do not like the result and want to start from the beginning press the reset button.
6. Go to the documents folder and find the file you just created, it should be called **Figure.xml**.

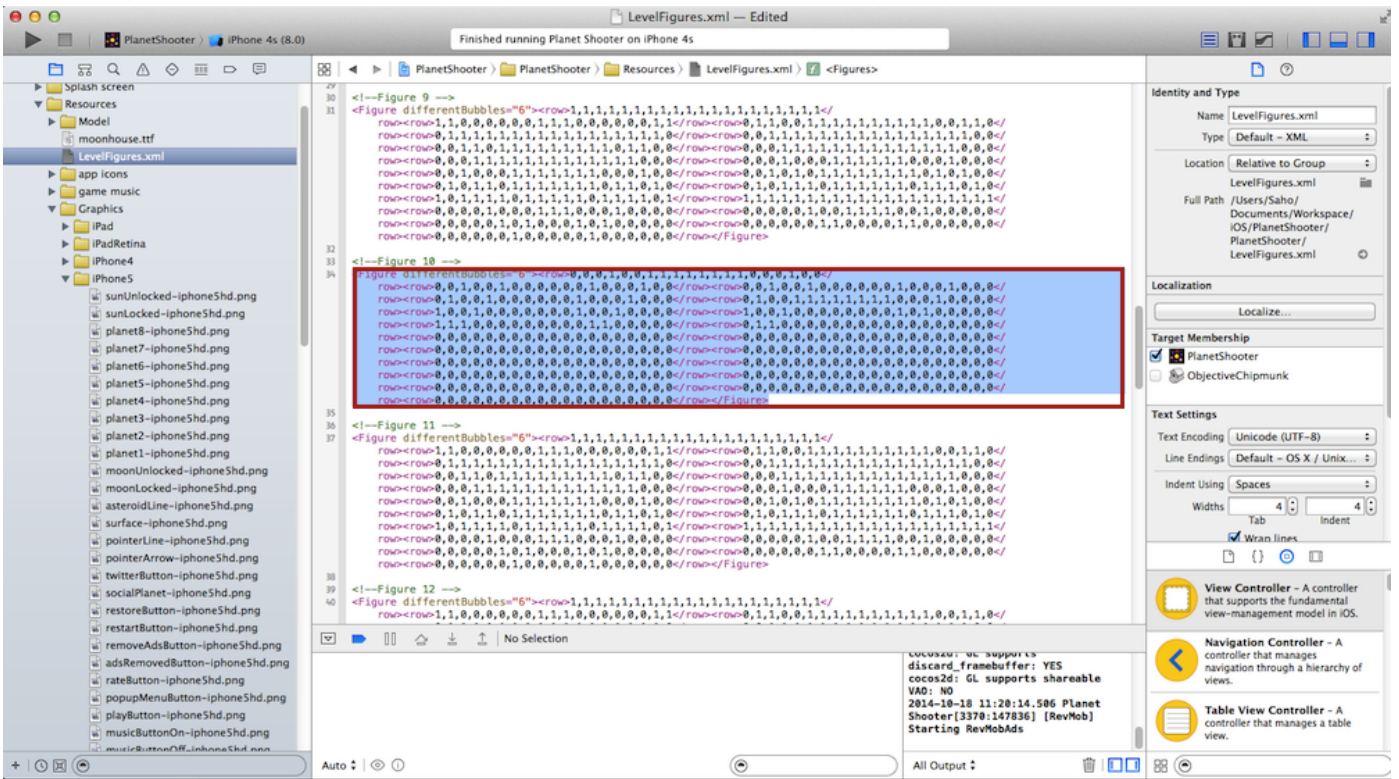


7. Open it and copy the content below the `<?xml version="1.0"?>` line.

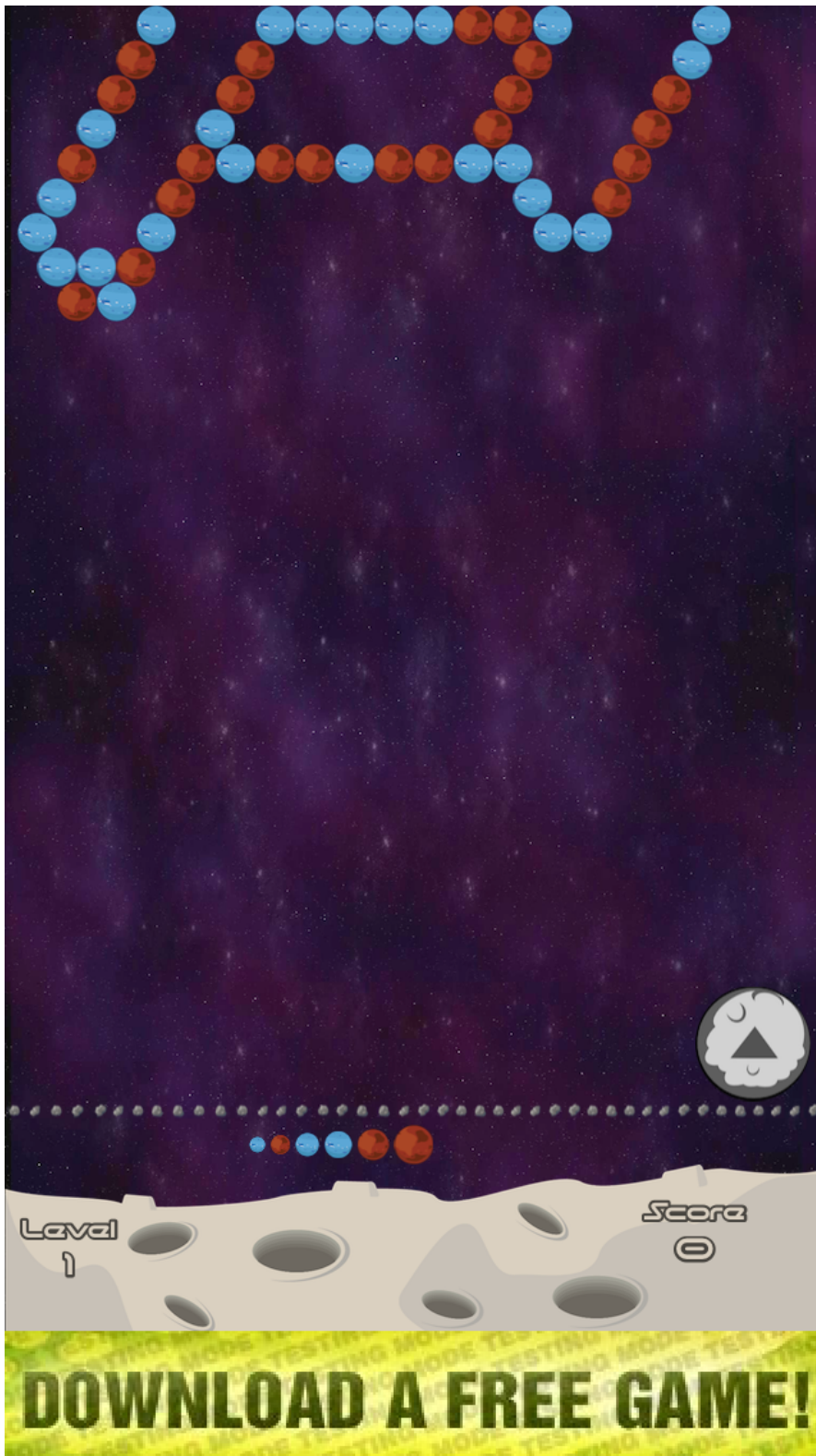


8. Open the Planet Shooter xcode project and find the LevelFigures.xml file.

9. Replace some of the existing figures content between the `<Figures>` and `</Figures>` tag, but be careful to not change some of the other figures (Here I replaced the figure for level 10 with the one I just made).



10. Run the game and test the figure that you just created.



Repeat the process if you want to make new figure.

Video

Watch the [video tutorial](https://www.youtube.com/watch?v=9JEkSOARIO0&feature=youtu.be) for the figure builder if something is still unclear to you.

Conclusion

Feel free to make suggestions for improvements and report bugs to my email or in ChupaMobile. I plan to port the game to the newest **cocos2d** version with **Sprite Builder**, so it will be easier for the less experienced users to modify the interface.