3D Pirate Run documentation

raizensoft

Apr 16, 2024

Contents:

CHAPTER 1

Getting Started

1.1 What's inside

- **production**
  Game production ready files
- **dist**
  Compiled and minified Javascript and CSS resources
- **libs**
  External Javascript libraries used in development.
- **src**
  Main javascript source files.
- **test**
  Test folder including data used in development.
- **Gruntfile.js**
  Grunt build file.
- **package.json**
  node.js project file.
- **npm-shrinkwrap.json**
  node.js package dependencies file.
- **docs**
  Documentation includes html and pdf format

1.2 Use production ready files

The *production* folder contains everything you need to deploy the game to different mediums. The files are compressed and optimized to be embedded in website or integrated in a Cordova app.

1.3 Folder Structure

```
assets/
  graphics/
  sounds/
  text/
css/
  fonts/
  pru3d.min.css
  app.css
js/
  pru3d.min.js
data/
index.html
```

- *assets:* Contain all game graphics, sounds and text resources
- *css:* Main stylesheet folder
  - *fonts:* Custom icon fonts folder
  - *pru3d.min.css:* Minimized game stylesheet
  - *app.css:* Generic and non-application specific stylesheet
- *js:*
  - *pru3d.min.js:* Main minimized Javascript file
- *data:* Game additional data
- *index.html:* Main html file

CHAPTER 2

Embedding

You can easily embed the game in any web page using iframe

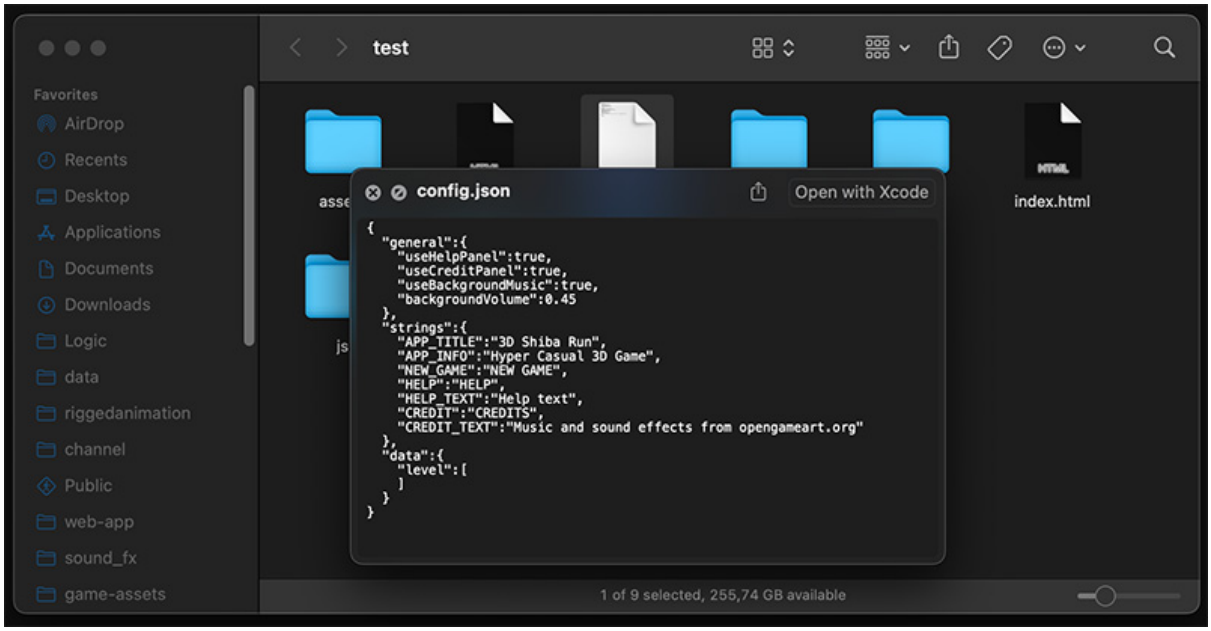
```
<iframe src="index.html" widht="640" height="960"></iframe>
```

See *iframe.html* in *production* folder for a complete example

CHAPTER 3

Customization

You can customize the game by editing *config.json* (production/ test folder) or insert javascript code into intializing code in *index.html*



3.1 config.json

- General parameters

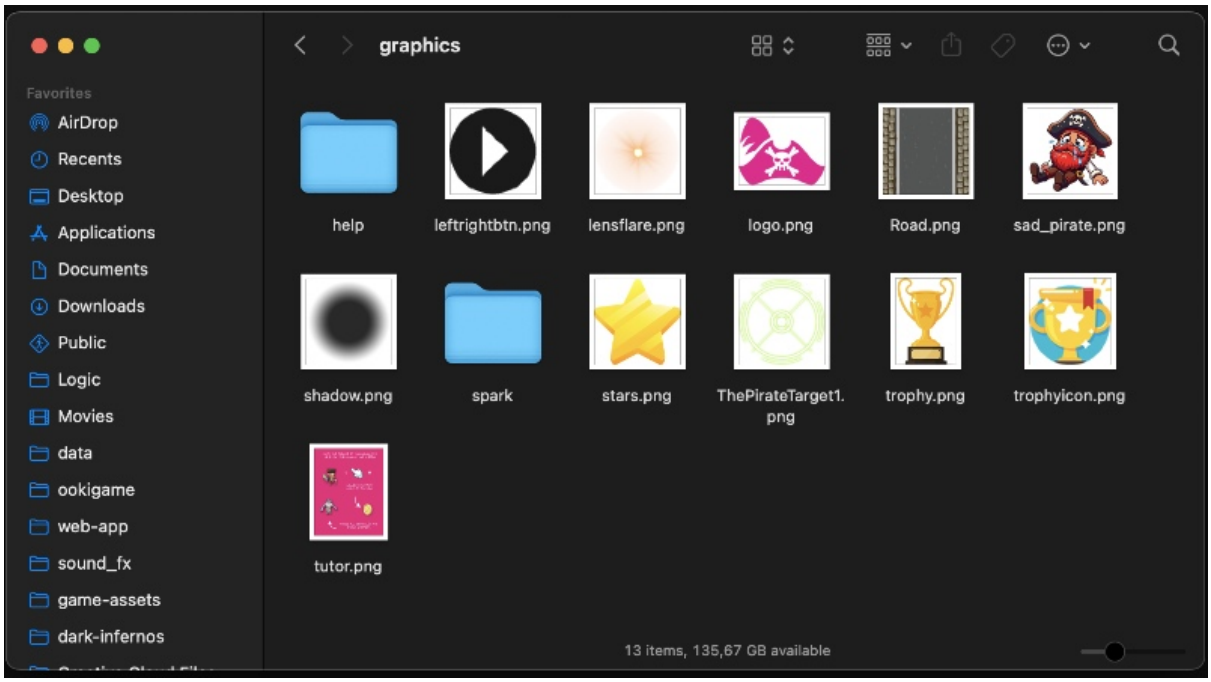
Name	Type	Default	Description
useHelpPanel	boolean	true	Enable or disable help panel
useCreditPanel	boolean	true	Enable or disable credit panel
useBackground-Music	boolean	true	Enable or disable background music
backgroundVol-ume	number	0.45	Adjust the background volume

- **String resources for changing text interfaces:**
  APP_TITLE, APP_INFO, NEW_GAME, HELP, HELP_TEXT, SETTING, CREDIT, CREDIT_TEXT

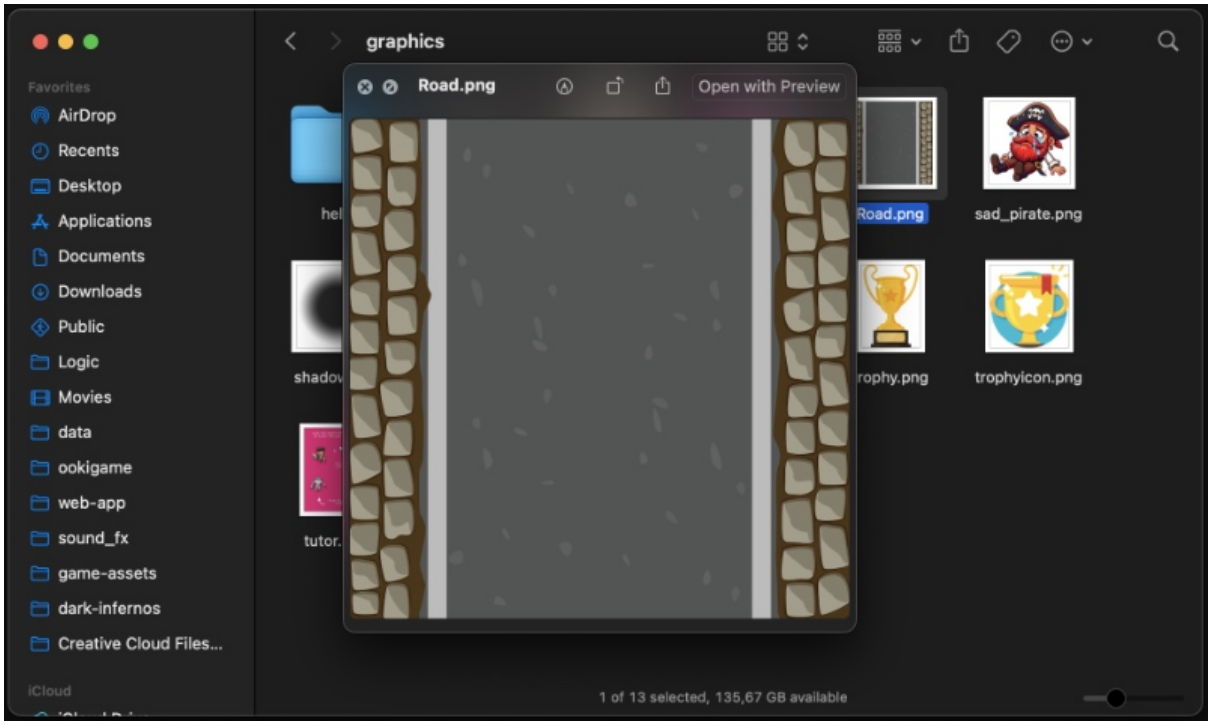
3.2 Skinning

This section will explain the process of skinning the game.

- Navigate to “test/assets/graphics” folder and make changes to these assets.
- After that, run “grunt production” to create the new distribution.

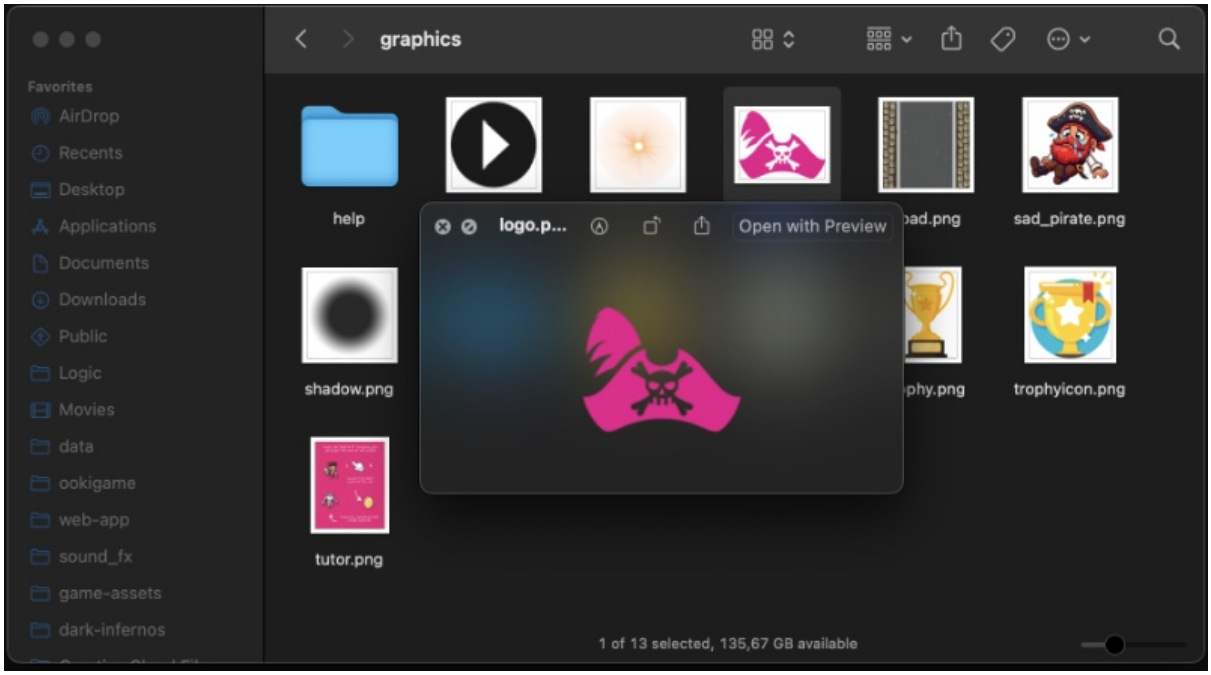


- Replace “Road.png” to change the grass texture

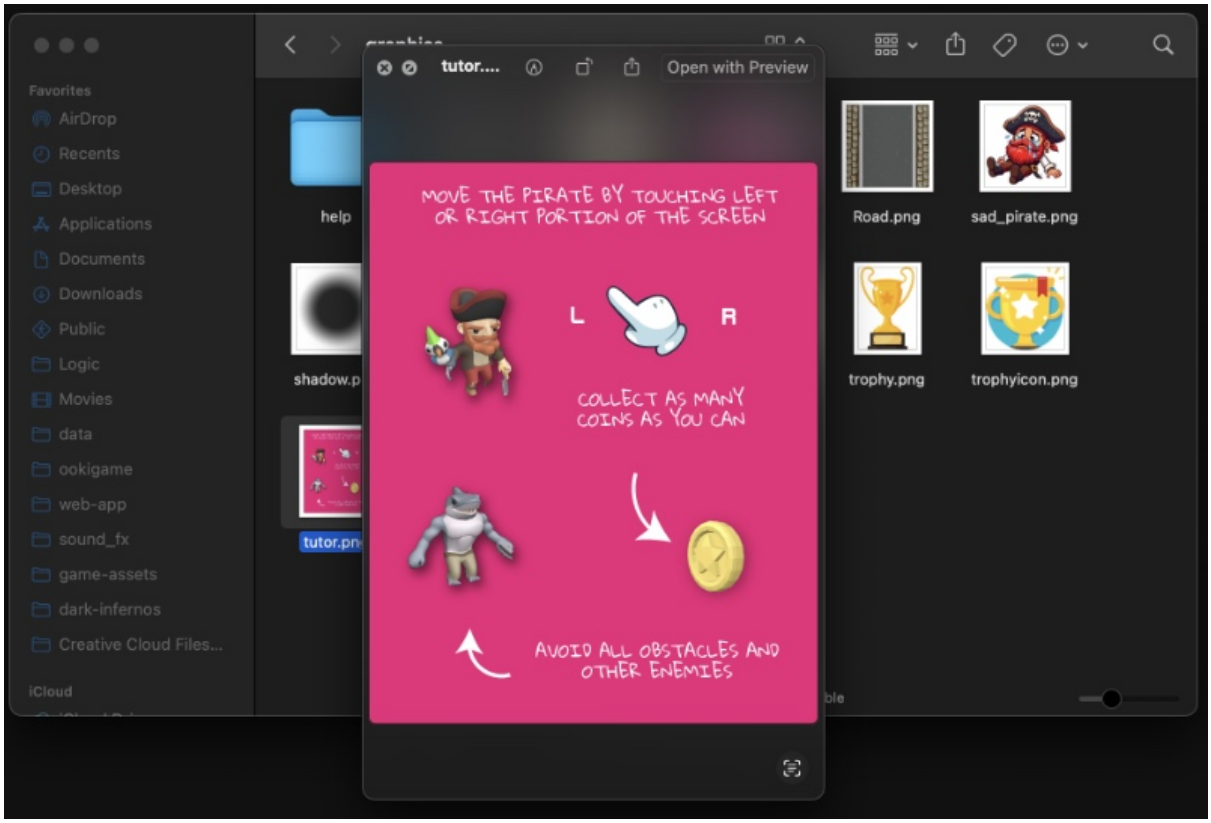


- Replace “sad_pirate.png” to change the game over screen shiba image

- Replace “logo.png” to change the game logo.



- Replace “tutor.png” to change the intro tutorial.



3.3 Javascript options

- You also have more options to customize the game using the javascript object settings.

```
// Init default options
this.defaultOptions = {
  blockSize:16,
  sceneColor:0x20294f,
  fitFactor:1,
  ambientLight:0xffffff,
  verticalShift:-20,
  shearAngle:25,
  lightMovingSpeed:3,
  carSpeed:140,
  enemySpeed:180,
  maxHealth:3
};
```

3.4 You Pirate speed and other enemies speed

- Input the value of *pirateSpeed* and *enemySpeed* to change these entities speed

```
var el = document.querySelector('.rs-pru3d');
var pru3d = new PirateRun(el, {
  pirateSpede:80,
  enemySpeed:200
});
```

3.5 Max health

- Input the value of *maxHealth* to change your car maximum health, default is 3

```
var el = document.querySelector('.rs-pru3d');
var pru3d = new PirateRun(el, {
  maxHealth:3
});
```

CHAPTER 4

Assets

You can reskin the game by replacing many assets in different folders using the same asset names:

```
assets/
  graphics/
  sounds/
  text/
```

- *graphics:* Core game graphics, many can be replaced like logo, padding pattern, board textures...
- *sounds:* Game sound and music effects
- *text:* Help file text content

CHAPTER 5

Cordova

Cordova is a mobile development framework that enables developers creates game and app using familiar web technologies. Packaging the game source code to be used in cordova app is really simple.

5.1 Installation

See Cordova - Get Started to install cordova into your system

- Create new Cordova game with a namespace:

```
cordova create MyGame com.mycompany.game
```

- Add “android” platform to deploy the app in android, and “browser” platform to test in browser

```
cordova platform add browser
cordova platform add android
```

- Copy the entire content of *production* folder into *www* folder and overwrite all the files.
- Plug in your android device and test the game using:

```
cordova run android
```

CHAPTER 6

Development

Using the provided build tools, you can quickly setup the development environment and start customizing the game very easily.

6.1 Setup environment

- Install nodejs
- Run Window command line program (or Terminal app in Linux/Unix OS)
- Change current directory to the game folder
- Type “npm install”
- Once finished, type “grunt” to launch a local http server in the background and start developing
- Test the gallery by go to “http://localhost:4040/test/”

6.2 Asynchronous Module

Modular programming has many advantages over one monolothic code base. All the app components are designed as individual classes wrapped in AMD module and then loaded asynchronously using require.js. This approach ensures the separation of concerns between software components and help implementing future features much easier.

6.3 Entry point

In the main html file, setup an entry Javascript file which configures requirejs and bootstraps the game:

- index.html

```
<script src="js/require.js" data-main="js/entry.js"></script>
```

- entry.js

```
1  // Setup baseUrl for source folder and library paths
2  requirejs.config({
3    baseUrl:"../src/",
4    paths:{
5      libs:"../libs/"
6    }
7  });
8
9  require(['rs/pru3d/PirateRun', 'libs/domReady'],
10
11   function(PirateRun, domReady) {
12
13     "use strict";
14
15     domReady(function() {
16
17       var el = document.querySelector('.rs-pru3d');
18       var pru3d = new PirateRun(el);
19       window.pru3d = pru3d;
20     });
21   });
```

See require.js for more detail about developing with AMD module

CHAPTER 7

Build Tasks

The following build tasks are available for development and production. In the terminal, type `grunt {taskname}` to execute the task

- **default:**
  Default Grunt tasks, this will launch “http-server” and “watch” tasks
- **dist**
  Create distrubitions, including mifnied CSS and Javascript files
- **production**
  Game production ready files
- **http-server**
  Launch a http server in the background for local testing
- **compass**
  Compile SCSS files to CSS files
- **pug**
  Compile pug templates to html files
- **watch**
  Watch for changes in sass and pug folders and auto compile them
- **cssmin**
  Minified application css file, exported in *dist* folder (bp3d.min.css and bp3d.light.min.css)
- **clean**
  Various targets to cleanup the project before rebuilding
- **copy**
  Copy assets, css and js files from *test* folder to build examples and dist files

- **requirejs:**
  Build, compile and minify *bp3d.min.js*, exported in *dist* folder
- **compress:**
  Compressed distributed minified script *bp3d.min.js* into GZIP format, produce *bp3d.min.gz.js*

CHAPTER 8

WordPress Integration

You can easily insert the game into WordPress post or page using iframe tag

8.1 Upload the game to WordPress

- Inside download package, rename “production” folder to “pirate-run”
- Upload “pirate-run” to WordPress “uploads” folder. The path of “uploads” is “your_site_root/wp-content/uploads”
- Check that you can launch the game by going to “yourwebsite.com/wp-content/uploads/pirate-run/index.html”

8.2 Insert the game into WordPress

- Create a new WordPress post
- Switch to Code Editor by triggering the three dot icon in the top bar and choose Editor - Code editor
- Add some title for the post
- Add iframe tag to embed the game in the post body

```
1  <iframe src="yourwebsite.com/wp-content/uploads/pirate-run/index.html"␣
↪width="1000" height"800"></iframe>
```

- Change the game width and height by modifying those values in iframe tag
- View the post with the game you just embed