

BUCCANEERS!

MODDING GUIDE

DOCUMENT VERSION 1.0



Contents

Introduction 3
- What Can I Mod? 3
- What Can't I Mod? 3
- What Tools do I Need? 3

Mod Structure 4
- Mod Folders 4
- JSON Files 4
  - Variable Types 5
- The ModInfo File 5

Flags 7
- Flag JSON Files 7
- Flag Textures 8

Sails 9
- Sail JSON Files 9
- Sail Textures 9
- Tattered Sails 10

Hull Paint Schemes 11
- Hull Paint JSON Files 11
- Hull Paint Textures 12
  - Cannons 12
  - Patterns 12

Cabin Paint Schemes 13
- Cabin Paint JSON Files 13
- Cabin Paint Textures 14
  - Floor 14
  - Cannons 14

Sharing your Mods 15

Troubleshooting 15

Support 15

Changelog 16

Appendices 17
Appendix 1: Cannon Paint Guide 17
Appendix 2: Hull Pattern Guide 17
Appendix 3: Hull Paint Guide 18

Introduction

Welcome aboard, ye scallywags! This guide will tell you all you need to know to embark on your modding journey for *Buccaneers!*, with examples and tips to help you along the way. You'll be showing off your newly customised fleet in no time!

Before you begin, here are some important points to be aware of:

What Can I Mod?

In the current version of *Buccaneers!*, you can only create cosmetic mods. These have no effect on gameplay and are purely for customising your ships to look the way you want. The following content can be added as mods:

- ✓ Flags
- ✓ Sails
- ✓ Hull paint schemes
- ✓ Cabin paint schemes

What Can't I Mod?

Currently, more complex mods such as custom models or those that affect gameplay are not supported, such as:

- × Ship models
- × Ship types or variants
- × Ship upgrades
- × Cannons
- × Weapons
- × Factions
- × Quests
- × Environments
- × Translations

What Tools do I Need?

Modding for *Buccaneers!* has been designed to be as easy as possible without requiring any specialist software or proprietary file formats. All you need is the following:

- A text editor that can edit JSON files (such as Notepad or Notepad++)
- An image editor that can edit and export PNG files (such as Paint, GIMP, Photoshop or similar)

You will also find some template files, example mods and the latest version of this document in this folder inside the game's install directory:

```
Buccaneers_Data\StreamingAssets\Modding~
```

In the next section, you'll learn how mods are structured for the game, along with more details about JSON files and the information you can edit.



Mod Structure

All mods for this game consist of their own unique folder with images and text files that tell the game what the content is and how to interpret it. This section outlines the folder structure, file types and conventions used to create mods.

Mod Folders

The game looks for mods inside this folder:

```
Documents\Buccaneers\Mods
```

There, you will find sub-folders for each mod currently installed. You will need to create a new folder for each mod you wish to make, though a mod can contain as much content as you wish. When naming your mod folder, it is recommended to use an author's name as a prefix to avoid potential naming conflicts with other mods. Here is an example for a mod we will call **"Buried Treasure"**, whose main contributor uses the alias "Captain Salvador":

```
Documents\Buccaneers\Mods\CaptainSalvador_BuriedTreasure
```

You can place all your assets in this folder and the game will automatically recognise them if they follow the naming conventions and use the file formats specified in the following sections. An example list of assets in a mod with a custom flag and paint scheme might look like this:

```
CaptainSalvador_BuriedTreasure
   -  ModInfo_BuriedTreasure.json
   -  Flag_TreasureChest.json
   -  Flag_TreasureChest.png
   -  HullPaint_PirateTreasure.json
   -  HullPaint_PirateTreasure.png
```

JSON Files

Text files in the mods use the JSON format, which the game can read and convert into usable data. These can be created and edited by most text editors, and in most cases, you can simply copy an existing file from another mod (such as the examples shipped with the game) to use as a template.

To help the game distinguish between different types of mod content, **each JSON file must have a specific prefix in its file name**. For example, the name of a custom flag's JSON file must start with the word "Flag". The prefixes to use for each type of content are listed in their respective sections later in this guide.

Variable Types

In each JSON file, you will come across several types of variables that you can modify, which are outlined in the table below.

Variable Type	Example	Description
Text	`"version": "Beta 1"`	Plain text that can be any combination of characters. Rich text formatting is also supported.
Localised text	`"modName": { "translations": [ "en|English name", "ru|Russian name", "zh-Hans|Chinese name" ] }`	A list of text entries for multiple languages. Each entry must start with a locale code* and a "|" separator. If an entry is left blank, the English one is used as a fallback.
Colour	`"patternColour": "#D4D4D4"`	A colour with RGB values expressed in hexadecimal format. You can use an image editor or online tool to get this value for the colour you want.
Integer	`"textureAtlasID": 0`	A whole number without a decimal point.
Rarity	`"rarity": 2`	An integer that determines the item's rarity as follows: 0 (Common), 1 (Rare), 2 (Exotic), 3 (Legendary), 4 (Faction)
Float	`"patternGlossiness": 0.5`	A number with a decimal point.
Boolean	`"unlockedByDefault": false`	A binary value that can be "true" or "false".

*Please note the game currently only supports English (en), Russian (ru) and Simplified Chinese (zh-Hans). Other locale codes will not work, and the game will fall back to the English entry.

The ModInfo File

Every mod needs some basic information to tell the user what it does and who made it, which is displayed in the Mods menu in the game. The ModInfo JSON file is a container for this information, and uses the following structure:

FILE NAME – Must begin with "ModInfo".

- **Mod name** – The name displayed in the Mods menu.
- **Version** – A version number using whatever format you like (e.g. 1.0 or Beta 1).
- **Contributor names** – Names of one or more people who made content for this mod.
- **Mod description** – A brief description of the mod's content.

Here is an example for our “Buried Treasure” mod, made by two people:

ModInfo_BuriedTreasure.json

```
{
	"modName": {
		"translations": [
		"en|Buried Treasure",
		"ru|",
		"zh-Hans|"
	  ]
	},
	"version": "1.0",
	"contributorNames": [
		"Captain Salvador",
		"Captain Thorne"
	],
	"modDescription": {
		"translations": [
		"en|A fine collection of customisations worthy of a true
		pirate.",
		"ru|",
		"zh-Hans|"
	  ]
	}
}
```

In the following sections, each type of mod content will be explained with full instructions and examples to help you get started.

Flags

Custom flags are the simplest type of mod for *Buccaneers!* which allow you to add your own designs as usable **ensigns** for your ships. These designs can be used regardless of faction, and they will always be flown from the stern of your ships.

Please note that you cannot replace the **national** flags (flown from the top of the mainmast), as these are required for faction identification at sea.

To create a custom flag, you need a JSON file and a texture, which are explained in the following sections.

Flag JSON Files

You can create as many new flag designs as you like, but each individual design requires its own JSON file. These files use the following structure:

FILE NAME – Must start with "Flag".

- **Flag texture** – The file name of the texture containing this flag design. Note that you do **not** need to include the file extension (.png).
- **Texture atlas ID** – If you use a texture atlas containing multiple flag designs, this integer tells the game which grid cell to use (starting at 0 in the bottom left and counting up for each column and row). Leave this at 0 if you are using an individual texture.
- **Grid size** – The number of columns (X) and rows (Y) in the texture atlas, if applicable. Leave both values at 1 if using an individual texture.
- **Grid extents** – A float between 0 and 1 indicating how much of the full texture's width (X) and height (Y) is taken up by the atlas, if applicable. Leave both values at 1.0 if using an individual texture.

Here is an example using an individual texture for a flag depicting a treasure chest. Note that flag designs do not have display names in the game, so you don't need to name each flag.

Flag_TreasureChest.json

```
{
    "flagTexture": "Flag_TreasureChest",
    "textureAtlasID": 0,
    "gridSize": {
        "x": 1,
        "y": 1
    },
    "gridExtents": {
        "x": 1.0,
        "y": 1.0
    }
}
```

Flag Textures

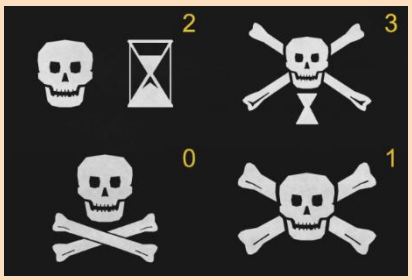
All flags in the game use a 3:2 size ratio, so flag textures should keep to this ratio to avoid stretching. The recommended resolution is **768 x 512 pixels per flag**. You can make larger textures if you wish, but you will see diminishing returns unless running the game at 4K resolution or higher.

If you want to create multiple flags for a single mod, you have two options: either create individual textures for each flag or pack them into a single texture atlas. Examples of each are shown below:

Individual texture (768 x 512)



Texture atlas (1536 x 1024) with a 2x2 grid, showing atlas IDs in the corners.





Flag textures in the base game use an overlay to give them a cloth-like appearance. You can find this overlay texture under "Modding~\Templates" in the game's install folder and add it to your own designs to make them look consistent with the base game designs. When using an image editing application that supports multiple layers, simply add the overlay texture as the top layer above your designs.

If you wish to create pirate flags that match the colours of the base game designs, you can use the following hex codes:

Background: #1A1A1A	Foreground: #E7E7E7

When you are done, **you must export your textures as PNG files**. The game will not recognise other file formats.

Sails

A custom sail design is a great way to make your ship stand out from a distance, and creating one is very straightforward. Like flags, you only need to create one texture and set up a JSON file as described below.

Sail JSON Files

Each sail JSON file uses the following structure:

FILE NAME – Must start with "Sail".

- **Colour name** – The display name of the sail colour shown in the ship customisation menu.
- **Rarity** – The rarity of the sail colour from 0 (Common) to 3 (Legendary). This affects the purchase price, with higher rarities costing more.
- **Unlocked by default** – If you want this sail colour to be available for purchase, set this to "false". If you want it unlocked and ready to use straight away, set this to "true".
- **Sail texture** – The file name of the texture used for this sail colour (without the .png extension).
- **Translucent colour** – The colour value used when the sun shines through the sails. Usually this should be a brighter version of the primary colour featured on your sail design. For example, if your sails are dark blue, set this value to a lighter blue.

Sail_PirateGold.json

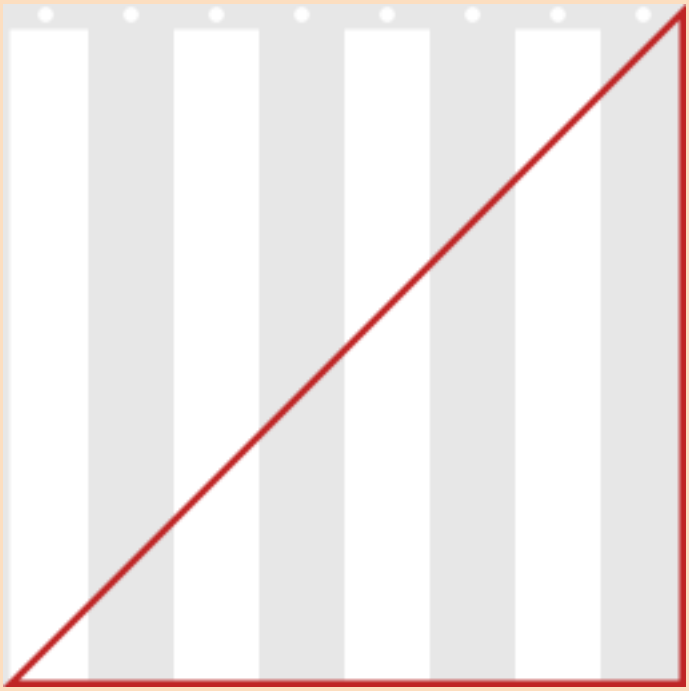
```
{
    "colourName": {
            "translations": [
            "en|Pirate Gold",
            "ru|",
            "zh-Hans|"
        ]
    },
    "rarity": 1,
    "unlockedByDefault": false,
    "sailTexture": "Sail_PirateGold",
    "translucentColour": "#FFC000"
}
```

Sail Textures

All sails on a ship share a single square texture which can be any size (the default is **2048 x 2048 pixels**). With the current system, this means **individual sails cannot be styled differently** from each other.

If you add any patterns or intricate designs, be aware that these may display incorrectly on triangular sails which use the lower right half of the texture outlined in red on the template below.

Sail texture template showing triangular sail coverage in red



Tattered Sails

If you want your sails to appear tattered with tears and holes in them, you can do this by making parts of your texture transparent (typically using the alpha channel). However, as the game applies its own cut-outs to show sail damage in combat, this may clash with your design or lead to unexpected results. You may want to test a few iterations to find a design that works for you.

Hull Paint Schemes

A ship's paint scheme gives it so much personality, and a custom design gives you a chance to fine-tune your ship's appearance from top to bottom.

These are more complex than sails and flags, but the same principles apply when setting up your mod. The biggest difference is there are more rules to be aware of when making custom textures which are explained later in this section.

Hull Paint JSON Files

Each hull paint JSON file uses the following structure:

FILE NAME – Must start with "HullPaint".

- **Paint scheme name** – The display name of the paint scheme used in the ship customisation menu.
- **Rarity** – The rarity of the paint scheme.
- **Unlocked by default** – If you want this paint scheme to be available for purchase, set this to "false". If you want it unlocked and ready to use straight away, set this to "true".
- **Main texture** – The texture that determines the colours of each part of the ship.
- **Ship name colour** – The colour of the name text displayed on the ship's stern.
- **Predefined cannon texture** – The name of an existing cannon texture included with the base game. Leave this blank if you want to use a custom texture.
- **Custom cannon texture** – The name of a custom texture you created for the cannons.
- **Show pattern** – If you want to show a decorative pattern texture on part of the hull, set this to "true". Otherwise, set it to "false".
- **Predefined pattern texture** – The name of an existing pattern texture included with the base game. Leave this blank if you want to use a custom texture.
- **Custom pattern texture** – The name of a custom texture you created for the hull pattern.
- **Pattern colour** – The colour applied to the pattern texture.
- **Pattern metallic** – If you want the pattern to behave like a metallic surface, set this to 1. Otherwise, set this to 0.
- **Pattern glossiness** – A decimal value between 0 and 1 that determines how smooth the pattern surface is. The default is 0.5.

Here is an example for a hull paint scheme named “Pirate Treasure”:

HullPaint_PirateTreasure.json

```
{
    "paintSchemeName": {
            "translations": [
            "en|Pirate Treasure",
            "ru|",
            "zh-Hans|"
        ]
    },
    "rarity": 3,
    "unlockedByDefault": false,
    "mainTexture": "HullPaint_PirateTreasure",
    "shipNameColour": "#E3CE5C",
    "predefinedCannonTexture": "Cannon3",
    "customCannonTexture": "",
    "showPattern": true,
    "predefinedPatternTexture": "ShipPattern1",
    "customPatternTexture": "",
    "patternColour": "#E3CE5C",
    "patternMetallic": 1.0,
    "patternGlossiness": 0.7
}
```

Hull Paint Textures

Ship hulls are painted using a grid of colour swatches that represent different parts such as the deck, masts and various outer stripes. This makes them very easy to repaint with whatever colour combinations you wish.

See Appendix 3 for a full guide showing how each colour swatch maps to different parts of a ship.

Cannons

Hull paint schemes also affect cannons, and these have their own texture with a different grid of colours. You can either use one of the built-in cannon colours listed below or create your own texture using the guide in Appendix 1.

Cannon1 **Cannon2** **Cannon3** **Cannon4** **Cannon5** **Cannon6**



Patterns

You can also choose to apply a decorative pattern to specific parts of the outer hull, like the “Royal Blue” and “Spanish Red” paint schemes in the base game. You can either use that texture or create your own, but you must keep your design within the same boundaries for it to tile correctly on the ship. See the guide in Appendix 2 for details.

Cabin Paint Schemes

Your cabin is your own personal space on your ship, so you can truly make it unique to you with a custom design. These are less complex than hull paint schemes but follow a similar set of rules for creating custom textures.

Cabin Paint JSON Files

Each cabin paint JSON file uses the following structure:

FILE NAME – Must start with "CabinPaint".

- **Paint scheme name** – The display name of the cabin paint used in the cabin repainting menu.
- **Rarity** – The rarity of the cabin paint.
- **Unlocked by default** – If you want this cabin paint to be available for purchase, set this to "false". If you want it unlocked and ready to use straight away, set this to "true".
- **Cabin texture** – The texture that determines the colours of each part of the cabin.
- **Predefined floor texture** – The name of an existing floor texture included with the base game. Leave this blank if you want to use a custom texture.
- **Custom floor texture** – The name of a custom texture you created for the floor.
- **Predefined cannon texture** – The name of an existing cannon texture included with the base game. Leave this blank if you want to use a custom texture.
- **Custom cannon texture** – The name of a custom texture you created for the cannons.

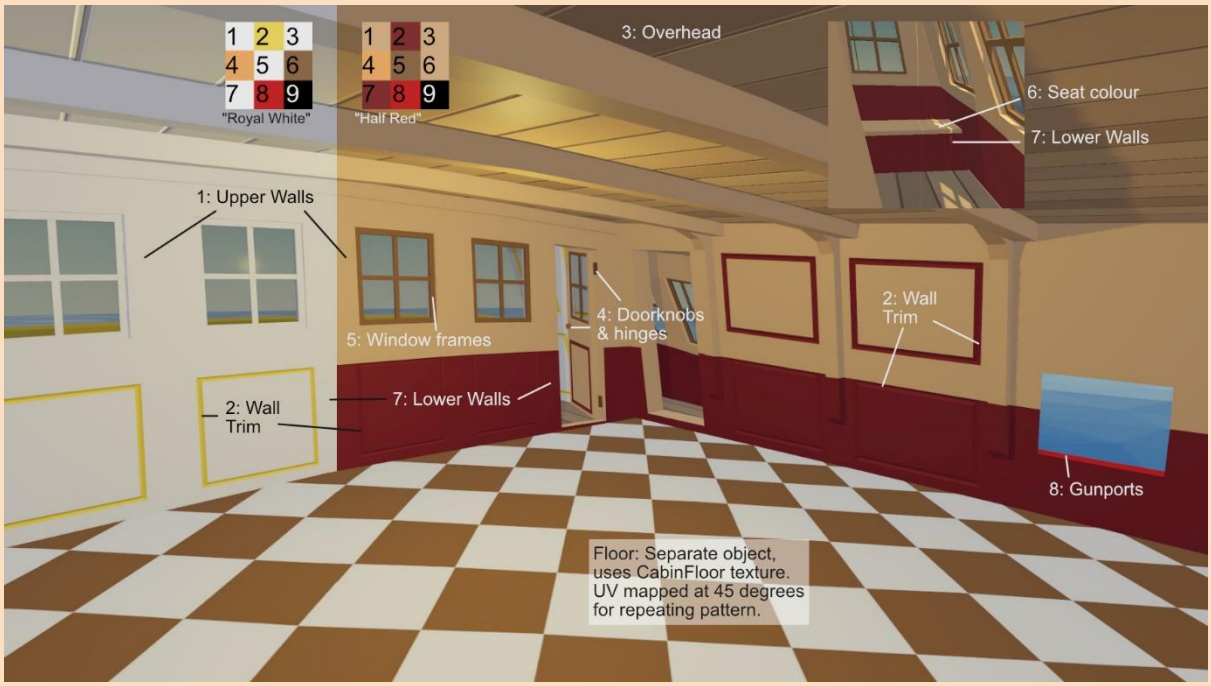
Here is an example for a cabin paint scheme named "Pirate Gold":

CabinPaint_PirateGold.json

```
{
    "paintSchemeName": {
            "translations": [
            "en|Pirate Gold",
            "ru|",
            "zh-Hans|"
        ]
    },
    "rarity": 3,
    "unlockedByDefault": false,
    "cabinTexture": "CabinPaint_PirateTreasure",
    "predefinedFloorTexture": "CabinFloor4",
    "customFloorTexture": "",
    "predefinedCannonTexture": "Cannon3",
    "customCannonTexture": ""
}
```

Cabin Paint Textures

Cabins use a similar grid of colour swatches to ship hulls, allowing you to paint specific parts separately. See the guide below for the texture mapping.



Floor

The floor is textured separately from the walls, and you can use a built-in floor texture listed below or create your own. This texture is tiled diagonally across the entire floor, so make sure your design is seamless at all four edges.

CabinFloor1	CabinFloor2	CabinFloor3	CabinFloor4	CabinFloor5	CabinFloor6	CabinFloor7
						

Cannons

Some cabins also have cannons inside which use the same style of textures described in the Hull Paint Schemes section, so you can either use a built-in colour or create one specifically to match your cabin design.

Sharing your Mods

Once your mod is complete, you're ready to share it with other players! The easiest way to do this is to pack your mod into a ZIP file so you can upload it to a file sharing service of your choice.

In Windows File Explorer, you can quickly create a ZIP file by right-clicking your mod folder and selecting "Compress to Zip file". You can also use other tools such as 7-Zip for more control over this process, but your mod is likely to be small enough that standard ZIP compression will suffice.

Troubleshooting

If you try to load a mod in the game and run into issues, the easiest way to find out what's wrong is to check your log file. This gets generated each time you run the game and can be found here:

```
%userprofile%\AppData\LocalLow\Skyward Digital\Buccaneers\Player.log
```

Open the log file in any text editor and search for "[MODS]" – this will show you all the information about mods that the game tries to load. If something goes wrong such as a texture failing to load, you will see a warning which lists the name of the mod file and the reason it didn't load correctly.

Support

If you have any questions about anything in this guide or need help with modding, feel free to join the Skyward Digital Discord server for advice and support.



Join the Discord here!

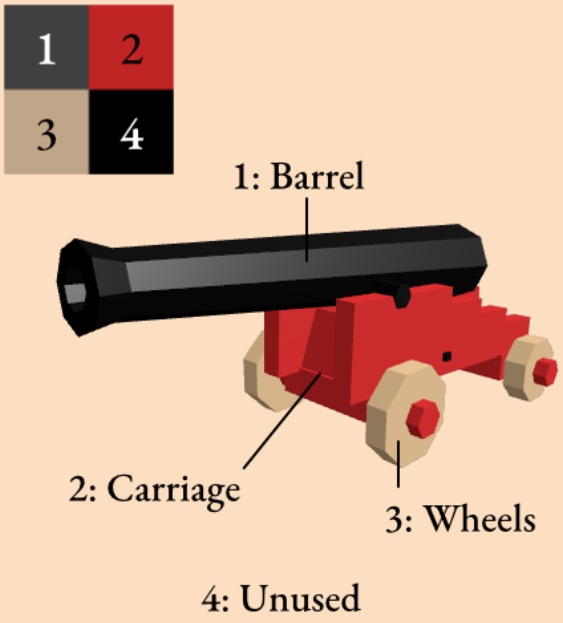
Changelog

Version 1.0 – Applies to Buccaneers! version 1.1.0 and later.

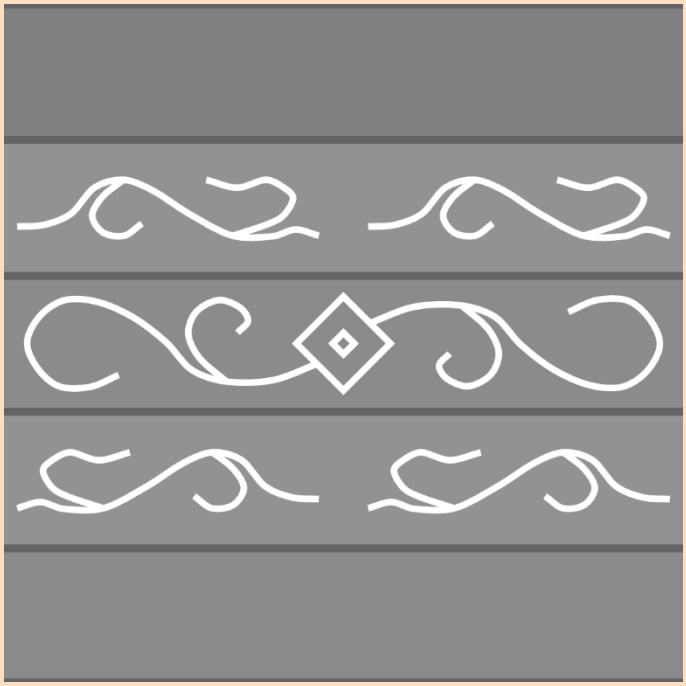
- Initial modding framework release.

Appendices

Appendix 1: Cannon Paint Guide



Appendix 2: Hull Pattern Guide



Background: Greyscale planking texture. Do not include this with the exported pattern.

Foreground: Greyscale pattern texture. Export this with a transparent background as a PNG file.

For best results, keep your design within the middle three planks. Using the top or bottom planks may have unexpected results.

Do not use colours for this texture as it will be tinted by the paint scheme in-game.

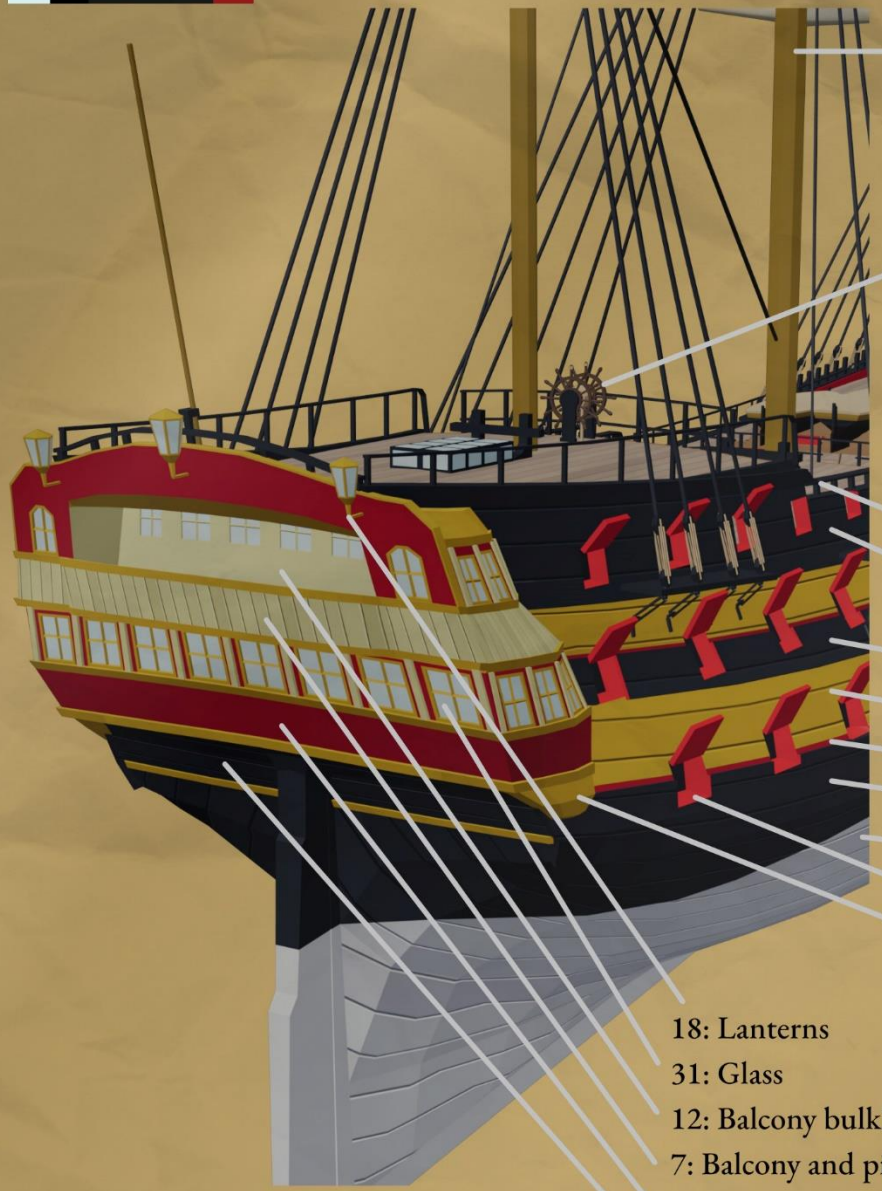
Appendix 3: Hull Paint Guide

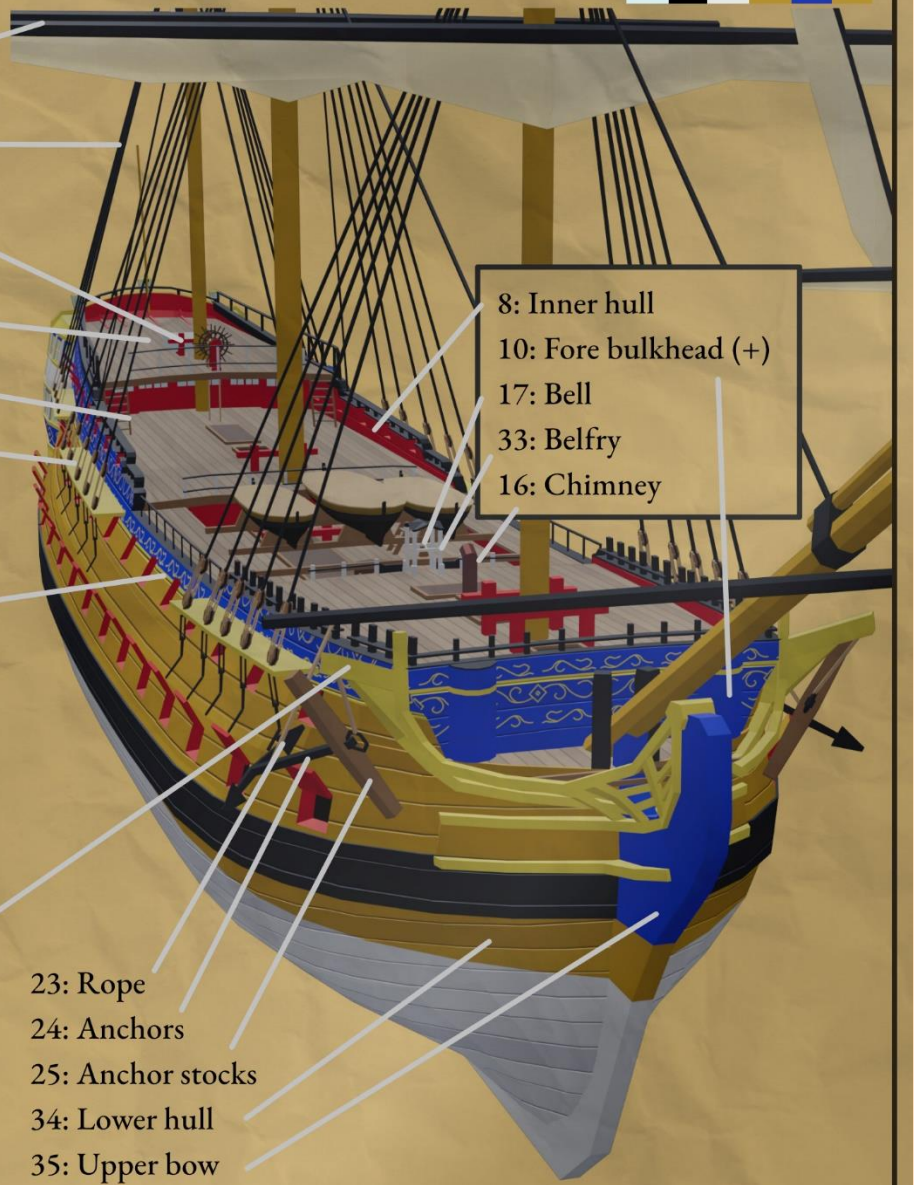




"Spanish Ochre"

"Royal Blue"





19: Masts
20: Yards
22: Standing rigging

26: Helm and bitts
27: Wheel
28: Deck
29: Capstans
30: Hatches and ladders
21: Channels

14: Upper rails
4: Upper hull
13: Mid rails
3: Upper wale (+)
1: Main hull stripes
36: Accent stripes
2: Main wale
5: Underwater hull
9: Inner gunports
15: Bow and stern rails

8: Inner hull
10: Fore bulkhead (+)
17: Bell
33: Belfry
16: Chimney

18: Lanterns
31: Glass
12: Balcony bulkhead (+)
7: Balcony and pillars
11: Stern galleries
6: Stern counter

23: Rope
24: Anchors
25: Anchor stocks
34: Lower hull
35: Upper bow

32: Black gunport filler

© Skyward Digital Limited 2024. All rights reserved.