

PocketGammon

Creating Skins

Version 1.1 as of March 3, 2005



This document and the PocketGammon software are protected under international law. No part of it may be reproduced in any way without the written permission of jens regge edv-beratung, Karlsruhe, Germany.

Copyright © 2004, 2005 jens regge edv-beratung. All rights reserved.

Introduction

PocketGammon gives you the ability to create your own “skins”, that is, the playing board, the dice and everything else that makes up the visual representation of a game. The skins are stored on the mobile device in the subdirectory `Skins` under the directory where PocketGammon is installed, usually `\Program Files\Games\PocketGammon` (English version) or `\Programme\Spiele\PocketGammon` (German version).

Distribution of your Skins

If you have created (and tested!) a nicely looking skin and want to share it with others, we will be happy to make it available for download on our web site, of course maintaining your copyright.

Please note that we cannot accept skins for sale, only skins that are free to use for anybody can be placed for download. Before sending us the skins, please send an e-mail first telling us about your intention to support@regge-edv.de.

We reserve the right to refuse skins for any reason, especially if the skin shows pictures or symbols that are prohibited by law in Germany.

Questions and Bug reports

Please send all questions and bug reports to support@regge-edv.de.

Files that make up a Skin

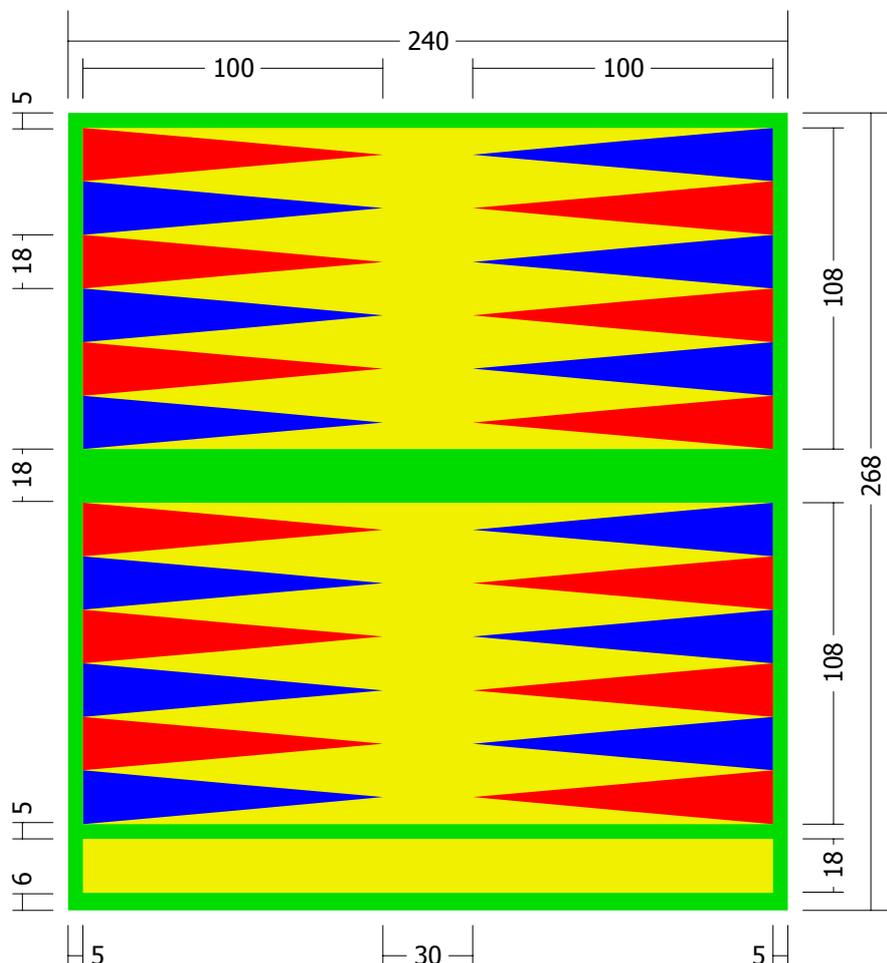
Each skin is stored in a separate directory under `Skins` and is made up of the following files:

<i>File</i>	<i>Purpose</i>
<code>board.bmp</code>	The playing board.
<code>dice.bmp</code>	The 12 dice faces (6 for each color).
<code>double.bmp</code>	The 6 faces of the double cube.
<code>off.bmp</code>	The two vertical stones for the "off".
<code>pmask.bmp</code>	The mask image for the points and the bar.
<code>stones.bmp</code>	The stones, 3 for each color.
<code>thumb.bmp</code>	A "thumbnail" image for the configuration dialog (optional).
<code>info.txt</code>	A text file containing copyright and color information.

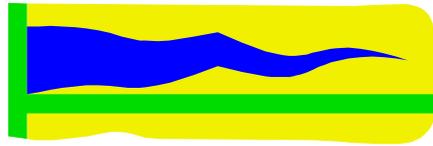
The directory name of the skin is the name displayed in the configuration dialog of PocketGammon.

Board Dimensions

The dimensions of a PocketGammon board are fixed in the current version. It must be 240 pixels in width and 268 pixels in height. PocketGammon will not load bitmaps that do not have these dimensions. The image below shows the dimensions (in pixels) of the different parts of a board:



Note that the *dimensions* are fixed, but the *shape* of the points (nor the shape of the dice and stones) are not. For instance, you could create a board with points like the one shown below, as long as the enclosing rectangle is not larger than 100 by 18 pixels.



Furthermore, all points on each side of the board must have the same shape, an explanation for this is given in the section about the `pmask.bmp` file. To make the board look smoother, you probably want to design it in your favorite graphics program at a larger size (for instance, at 1200 by 1340 pixels) and then rescale it using *anti-aliasing*. This is perfectly ok, just make sure your *point mask* `pmask.bmp` matches to points on the board.

The Point Mask

PocketGammon flashes stones and points in a different color when the computer makes a move. Because it cannot guess where a point is and what it is shaped like, you must tell it. This is done with the point mask bitmap `pmask.bmp`. An example for a point mask for triangle shaped points is shown below. Note that picture is shown with a gray background to show the dimensions of the bitmap more clearly:



The point mask is exactly 300 pixels wide and 18 pixels high and consist of three parts, each making up 100 pixels: The leftmost 100 pixels are used to render the flashing points on the left side of the board, the middle 100 pixels are used to render the points on the right side and the last part is used to make the “off” (the lower part of the board where the stones that have been played off are stored).

The mask image can contain not only black and white pixels but also gray values, as the enlargement below shows:



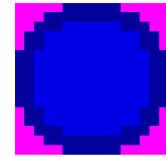
Whenever PocketGammon needs to flash a point, it overlays the board with the mask image at the appropriate position. If a pixel in the mask image is black, the corresponding board pixel will remain unchanged. If it is white, the corresponding board pixel will be highlighted completely. If it is gray, the resulting pixel will be a blend of the original board pixel and a highlighted pixel, depending on the gray value.

Note that actually only the red color channel value of the mask image is used, the green and blue values are not taken into account. For a grayscale mask, this does not make a difference.

Stones & Dices

Stones, dices and the double cube are grouped in four bitmaps, each bitmap containing all stones, all dice faces, all double cube faces and all vertical “off”-stones. Unlike the board, these should not be anti-aliased because they are rendered a different board

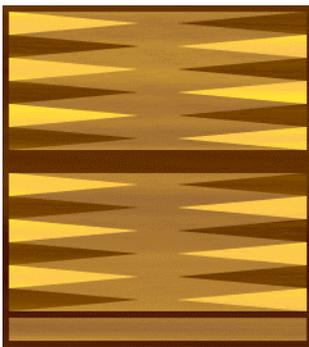
positions. For instance, a round stone with a green background might look nice on green points, but will look rather ugly on red ones. To solve this, a mask is integrated in these bitmaps: all pixels having the color magenta (with RGB-values 255, 0, 255) will not be rendered on the board. Unlike the point mask, there are no “gray values” for this mask: a pixel that does *not exactly* have this color value *will be rendered*. The consequence is that you cannot display “pure” magenta in stones and dices. If you want to, use a slightly different color value. The picture to the right shows a blue stone from the “Red & Blue on Gray” skin:



Bitmap Layouts & Color Depth

The table below shows the layouts of each of the bitmaps that makes up a complete skin, chosen from the “Teak” skin. The *Color Depth* is the type that the bitmap must have, other values might give strange results when being rendered. “24 Bits” stands for true color bitmaps, where each pixel is made up of three bytes (the red, green and blue values), whereas “8 Bits” stands for a palette bitmap where each pixel is represented by one byte and is mapped to a palette entry. All bitmaps must be in “Uncompressed Windows Bitmap” format, RLE-compressed bitmaps will not work properly.

board.bmp



Purpose:

The playing board.

Layout:

240 by 268 pixels, 8 Bits

dice.bmp



Purpose:

Contains the dice faces for both players, the first six are for player “white”, the second six are for player “black”. Each face is 18 pixels wide, the faces line up seamlessly.

Layout:

216 by 18 pixels, 8 Bits

double.bmp



Purpose:

Contains the faces for the double cube. Each face is 16 pixels wide, the faces line up seamlessly.

Layout:

96 by 16 pixels, 24 Bits

stones.bmp



Purpose:

Contains the stone images for both players, the first three are for player "white", the second three are for player "black". Each image is 16 pixels wide, the images line up seamlessly.

Layout:

96 by 16 pixels, 24 Bits

off.bmp



Purpose:

Contains the "off" images for both players, the first image is for player "white", the second is for player "black". Each image is 4 pixels wide, the images line up seamlessly.

Layout:

8 by 16 pixels, 24 Bits

pmask.bmp



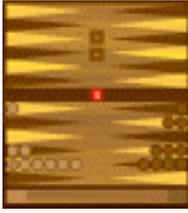
Purpose:

Contains the point mask image as described above. The first 100 pixels are used for the points on the left side of the board, the second 100 pixels are for the points on the right side and the last 100 pixels are for the "off". The images line up seamlessly.

Layout:

300 by 18 pixels, 24 Bits

thumb.bmp



Purpose:

Contains a thumbnail image that is displayed in the configuration dialog when a skin is selected. If a skin does not contain a thumbnail image, the board image is scaled down and displayed.

Layout:

70 by 78 pixels, 8 Bits

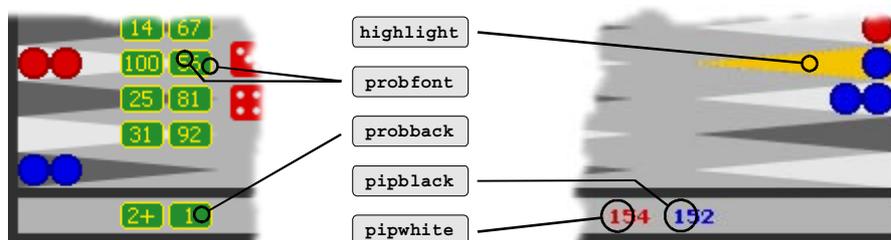
The “info.txt” File

This is a regular text file containing information about the colors to be used for various non-bitmap items as well as copyright information. They have the form *key=value*, valid elements are shown in the table below:

Key	Type	Description
highlight	Color Code	Default color used for highlighting flashing stones and for drawing the movement arrows.
probfont	Color Code	Color used for the probability numbers and the border.
proback	Color Code	Color used for the probability background.
pipwhite	Color Code	Color used for the “white” player PIP count.
pipblack	Color Code	Color used for the “black” player PIP count.
info	Text	Copyright information, displayed in configuration dialog

Color codes are in the form *RRGGBB*, where each of the three groups stands for a color channel value in hexadecimal numbers. For instance, *FFFFFF* represents white (all color channels are set to *FF* hexadecimal, which is 255 in decimal). Likewise, *DC143C* represents a color like “crimson”, with the RGB values 220, 20 and 60.

The pictures below show the items for the “Red & Blue on Gray” skin.



The *info* item is a descriptive text (up to 80 characters) that is displayed next to the board thumbnail in the configuration dialog. You can place your copyright note here. To place a newline character in the text, write a *\n* in the *info* string.