Melody's Escape - Modding Guide

Melody's Escape supports 3 types of elements that can be modded:

- **character skins**
- **dynamic hairstyles**
- **color schemes** (menus and in-game).

The game comes pre-installed with a *sample* for each mod type. Those samples can be found inside the */Mods/* sub-folder of the game's installation folder.

Each new *mod* should be a new sub-folder (located inside the parent */Mods/* folder), with all relevant files placed inside this folder. The name of that folder will become the name of your mod.

A "mod" can be a **combination of any of the 3 types**, you are not required to provide all 3 at the same time, nor do you need to create 3 different folders for each type if you wish to provide a similar themed skin/hairstyle/color-scheme.
You can for instance create a mod with a specific character skin and haircut and distribute those under the same name/folder. You cannot however have multiple of the same type (like character skins) under the same mod.

A **mod editor** ("*ModStudio.exe*") is provided within the installation folder of Melody's Escape (for Windows only, due to technology constraints), but it is not required to create and upload mods, as you can manually make the modifications on textures and XML files. It is a big time saver though, as it allows you to visually check the changes in real-time and reload textures on the fly.

Each mod found in the */Mods/* sub-folder is loaded when the game is launched, so each modifications will require a restart of the game. If a mod has been successfully loaded, it will become available in the "Customize Appearance" menu.

Steam Workshop mods must be uploaded via the game in the "*Customize Appearance*" menu. Once in that menu, click on [*Workshop Upload*] in the top right of the menu's dialog box.
You can also **update** your *Workshop* mods using the same interface if you have made any changes (like adding a new thumbnail or changing a texture).

I. Character Skin

Melody's **character appearance** can modified using a simple sprite-sheet (.**png** texture) and an optional XML definition file used to describe sprites' boundaries and pivot points.

Files:

- **character_skin.png** (*required* - the texture of the character skin)
- **skin_sprites_areas.xml** (*optional* - the XML definition of sprites areas and pivot)

The game uses a skeletal animation system, so Melody is drawn from a real-time animated composition of multiple parts (*bones*) with each bones' position and rotation animated dynamically.

You can change the appearance of Melody by modifying the sprites and their visual offsets, but you cannot modify the overall *proportions* of the character (so you can't mod a tiny character for instance, or create entirely new animations).

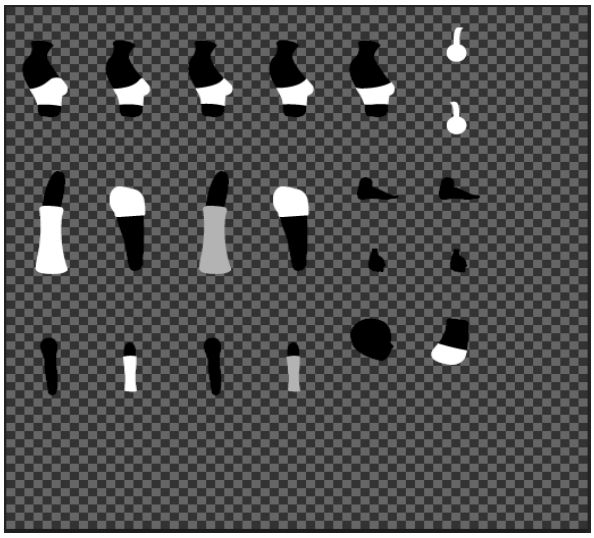

Illustration 1: The default sprite-sheet of Melody

Illustration 2: The composite result

The **optional** XML file contains all the coordinates, size and pivot coordinates of each specific sprites. If no XML file is provided in your mod, the game will use the default coordinates from the original Melody skin.

This XML file allows you to completely change the order and position in which sprites are located in the sprite-sheet, and even have multiple sprites refer to the same texture area (for instance, to have all 5 torsos variations point to the same sprite, if you don't wanna draw each variations).

You can manually edit the sample XML provided with the *_Sample Skin* mod. Each area key is using the following format:

```
<Area Key="torso_A" X="146" Y="2" W="70" H="142" PX="35" PY="98" />
```

Where *Key* is the name of the sprite, *X* and *Y* are the coordinates (starting from the top left of the texture image), *W* and *H* are the size of the sprite area, and *PX* and *PY* are the coordinates of the pivot point, relative to the top left of the sprite's area (X and Y)

The pivot acts as an offset, which allows you to shift a sprite around if you need more space than

the default values allows (for instance, to add a backpack to the torso sprite).

There is a body part sprite which points to an empty space in the default Melody skin: “**belt**”. This **belt** element is drawn **above the legs** and is tied to the position of the "*pelvis*" element. You can use it for instance to draw a skirt above the legs in your custom skins.

ModStudio.exe allows you to easily set each parts areas, and to reload the skin texture on demand, which is quite useful to avoid having to restart the game manually each time when testing your skin.

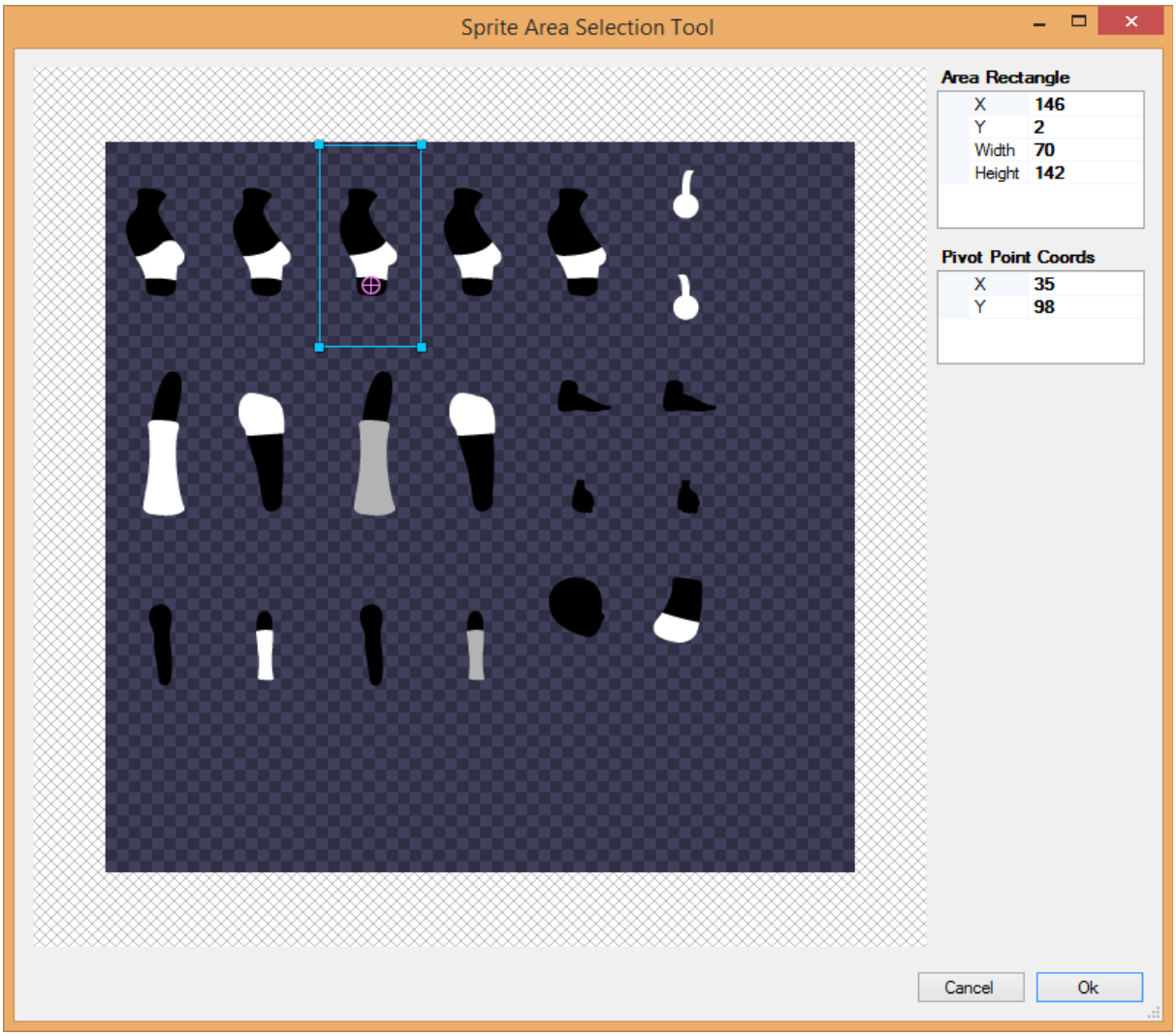

Illustration 3: ModStudio's sprite area selection tool

II. Dynamic Haircut

Melody's hairstyle is a collection of multiple static sprites animated and deformed in real-time using a physics simulation, in a manner similar to a "cloth simulation".

This particular mod type is the most complicated of the 3, so the use of *ModStudio.exe* is greatly recommended if you are on Windows.

As with the character skin, each separate hair-lock is selected from a sprite-sheet (.**png** texture)

Files:

- **hairstyle.png** (*optional* - the texture with hair-locks. Will use the default one is absent)
- **hairstyle.xml** (*required* - the XML definition of the hairstyle)

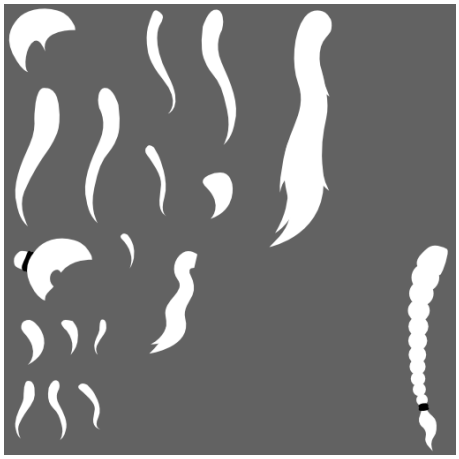

Illustration 4: The default sprite-sheet for hair-locks

A hairstyle is composed of 2 elements (*HairStyle Definitions*): *Front* (drawn **above** the head) and *Back* (drawn **below** the head)

Each HairStyleDefinition can have a static (non-animated) sprite, for the base look of the hairstyle, and multiple optional *hair locks*, which each have their own sprite and behavior settings.

Here are the specifics settings for each elements of a complete hairstyle:

1) Hairstyle

- **HideHeadphones**: set to **true** to hide the headphones sprites on the **character skin** (useful for hairstyles covering them, like a helmet. Default to **false**.

2) HairStyleDefinition (Front and Back)

- **Offset**: the offset/coordinates of this HairStyle Definition compared to the top-left of its buffer rendering area
- **Gravity**: the gravity applied to each child Hair Locks (you can control both the horizontal and vertical axis)
- **StaticSpriteArea**: the sprite-sheet area of the "base" sprite. This can be used for instance to have a hair-colored scalp or an helmet. Leave values at 0 to not show any sprite.

3) HairLock

- **Name**: the name of the lock, only used for easy mod design purposes.
- **TotalNodes**: the number of (invisible) verlet nodes to simulate hair movement for that particular lock. Ranges from 2 to 20 (lower numbers means more sharp angles, high number means smoother angle but more wobbly effect). Please note that the first few nodes are always static, to ensure that the base of a lock remains in place correctly.
- **DirectionAngle**: the angle (in degrees) from which the lock is oriented towards (allowing for effects like spikes pointing upwards)
- **ImpulseFactors**: the factor by which to multiply impulses from Melody's movements and wind, ranging from 0 (static lock) to 1 (full force effect). For best effect, it's better to have very long locks around 1, and small locks in the front around 0.1 (so that they don't wiggle all over the place). This is one of the **most important variables** to play with in order to recreate the best hair movement effect, so don't hesitate to experiment with this value.
- **SpriteArea**: the bounding rectangle selecting the desired sprite on the sprite-sheet. It can *sometimes* be beneficial to include more blank space on one side, or try different box sizes until the lock movement feels just right.
- **GravityOverride**: an optional field allowing you to override the parent's gravity for this specific lock

You can add as many locks as you wish, but for performances reasons, both Front and Back definitions each have a limit of **1280 polygons** for rendering the hair-locks, after which any additional lock won't be displayed anymore (this poly count is displayed in Mod Studio)

Please note that the hair animation is rendered on a size-limited graphical *virtual buffer*, and for this reason the allocated size of the animation is restricted. While there is plenty space for most haircuts to not be affected, some very long hair-locks (like a knee-length hair-lock) can be be clamped visually if they overflow on the limits. If this happens to you, I would suggest to reduce the size of the lock until it doesn't go out of bounds anymore.

This buffer width and height cannot be expanded, as it would impact performances on lower end computers. Make sure to try the hairstyle with each movement animations to ensure that it always stays in bounds.

Illustration 5: This is about the max height a hairstyle can reasonably reach without going out of bounds once animated. The blue grid represents the graphical buffer (separated in 2 parts)

III. Color Scheme

Almost every element in Melody's Escape can have its color customized, from the menus to the obstacles.

File:

- **colors.txt** (*required* – a text file with the color values of each elements)

The colors are in the RGB (red, green, blue) format, and the order of the elements in the text file cannot be modified.

Here are the elements which can be colored:

- **Obstacles** (4 types + the "Direction" one from Intense difficulty)
- **Intensity** backgrounds (4 types, walking, jogging, running and flying)
- **Ground color**
- **Outline color** (of light orbs)
- **Obstacle activation color** (when an obstacle is successfully validated)
- **Obstacle failed color** (when an obstacle is failed)
- **Obstacle grace color** (when an obstacle is auto-validated due to the "*grace*" mechanic in Relaxing difficulty)
- **Interface colors** (background, text, secondary text, menus)
- **HUD color** (the text in game, like Score and Multiplier)

And finally you can add a description to the color scheme, which will be shown when selecting it in the *Customize Appearance* menu.

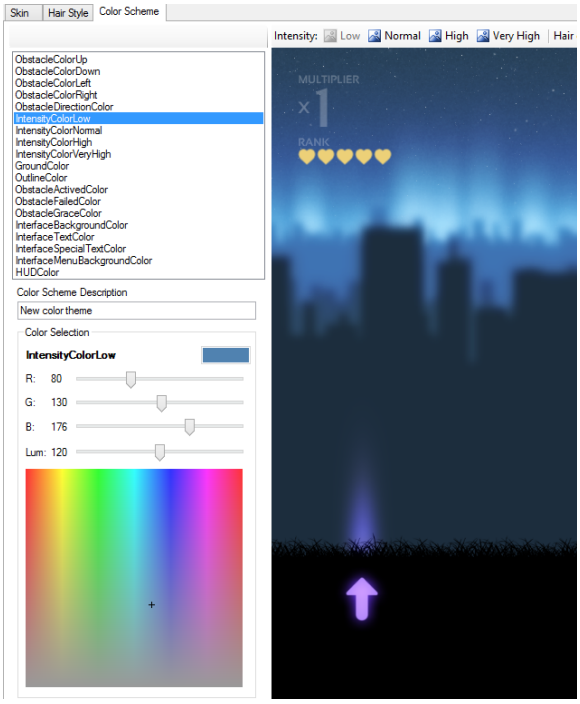

Illustration 6: ModStudio's color picker tools and live preview

IV. Appendix

Workshop Thumbnails

If you are planning to upload your mod on the *Steam Workshop*, you ideally need to place a JPG image called “**preview.jpg**” inside your mod's folder.
The thumbnail size is **360x360px**, and a sample one is provided in each of the samples mods folders.

Please note that *Mod Studio* **does not** generate thumbnails and you will still need to create one manually.

If you forgot to add a thumbnail before uploading your mod, you can always add the file in your mod folder and **update** the mod via the *Workshop Upload* menu.