

Sprite Editor Existing Market

SpriteEditor by Nerdz - <https://github.com/mahowa/Nerdz.git>

(market research performed by Thu Ha)

Summary of some features: The main window includes a dedicated area for drawing sprites, also provides various buttons for manipulating sprites like rotating along the x-axis, y-axis and rotating 45-degrees. A slider is also included to adjust the speed of sprite animation. It also uses QFileDialog to save files, create new file and load file. Some outstanding features are the use of QTimer for smooth transition, a color palette to select and display current color used in the sprite, and also the support of various transformations such as rotating, erasing or resetting states.

There are some advantages like customizable colors, real-time updates so that users can see their sprite animations in real time and with the help of QT widgets make the editor user-friendly with button and sliders for easy control.

In terms of disadvantages, I think the use of QGraphicsScene and frequent updates with QTimer could lead to inefficient memory usage, potentially slowing down performance when working with multiple or complex scenes.

2D Sprite Editor by Unity

(market research performed by Phuc Hoang)

2D Sprite Editor is integrated into Unity, making development with 2D sprites or sprite sheets easier. One advantage of the 2D Sprite Editor is slicing into multiple parts of a sprite sheet, which allows it to create easier animations and helps in better asset management. It also allows for setting a custom pivot point and editing the bounds of individual sprites, which offers finer control over positioning and animation behavior. The Sprite Editor allows the automatic slicing of uniform grids and manual slicing for more elaborate sprites, adding flexibility depending on the asset's requirements. One serious drawback is that, although Sprite Editor is great for a simple and middle-level task, it needs advanced editing of sprites or pixel art, and detailed work may be required with some other specialized software, such as Aseprite or Photoshop.

BobSprite - <https://bobsprite.com/editor>

(market research performed by Trenton Stratton)

BobSprite has a variety of tools, but some of the key features include a browser-based interface, extensive layer management- to include layer opacity adjustment, great user documentation, and an intuitive color selection interface.

Some advantages are that there are quite a few features available for users and the documentation (which is easily accessed) to support the users. Being browser-based, allows functionality to users without them having to download software. All this provides great flexibility to users with a low barrier of entry. Disadvantages are that the UI feels 15 years old. The buttons and text are slightly hazy, the interface lacks the layout and color scheme that would give it a modern feel. However, most glaring is the layout. It feels cluttered and unintuitive and you can't figure out what a tool is by hovering the mouse over it. Another disadvantage is that it is because it is browser-based, it could have some compatibility issues for users who don't use Google Chrome. Additionally, being browser-based requires additional cost and upkeep over time to keep the hosting domain.

Sprite Editor by AJNaito - <https://github.com/AJNaito/CS3505-Sprite-Editor>

(market research performed by Chanphone Visathip)

Overall, this project does what it is supposed to do. However, there are some advantages and disadvantages of some features in this project. I really like the color picker of this project. When I click “color picker”, the system shows me a built-in color picker of the computer which is really useful. I think Qt have a built-in function that can call this color picker. It includes everything from the eyedropper tool to color sliders. Also, the paint bucket tool is very useful. There are couple things that are drawbacks for this project. First of all, the canvas would be much improved if it had the png checkerboard type background. The grid is white and the canvas is white which presents some challenges in distinguishing the canvas from the rest of the UI. The preview of the current frame, I think maybe have some kind of page that show on the side of different layers frame. So, the users have an idea of what they have.