PixelBit

A cartoon of a round object with a rainbow and stars

AI-generated content may be incorrect.

Github: https://github.com/codeninja2000/PixelBit

PixelBit is a lightweight, user-friendly image editing application built with Java and JavaFX. It provides essential image manipulation features with an intuitive interface.

Contents

[Features 1](#_Toc205060895)

[Basic Operations 1](#_Toc205060896)

[Image Filters 2](#_Toc205060897)

[Image Manipulation 2](#_Toc205060898)

[Technical Details 2](#_Toc205060899)

[System Requirements 2](#_Toc205060900)

[Project Structure 2](#_Toc205060901)

[Design Patterns 2](#_Toc205060902)

[Getting Started 3](#_Toc205060903)

[Installation 3](#_Toc205060904)

[Basic Usage 3](#_Toc205060905)

[UML Diagrams 4](#_Toc205060906)

[Screenshots 9](#_Toc205060907)

## Features

### Basic Operations

* Load and save images in common formats (JPG, PNG, GIF)
* Undo/Redo functionality for all editing operations
* Reset image to its original state

### Image Filters

* Grayscale conversion
* Sepia tone effect
* Color inversion
* Brightness adjustment
* Contrast control

### Image Manipulation

* Crop functionality
* Image resizing capabilities

## Technical Details

### System Requirements

* Java 24 or higher
* JavaFX runtime

### Project Structure

* **Model**: Contains the core image processing logic (EditableImage, PBModel)
* **View**: Handles the user interface (PBImageView)
* **Controller**: Manages user interactions and updates (PBController)
* **Commands**: Implements Command pattern for operations

### Design Patterns

* Command Pattern: For implementing undo/redo functionality
* MVC Architecture: Separation of concerns between Model, View, and Controller
* Factory Pattern: For filter creation and management

## Getting Started

### Installation

1. Clone the repository:

git clone https://github.com/codeninja2000/PixelBit.git

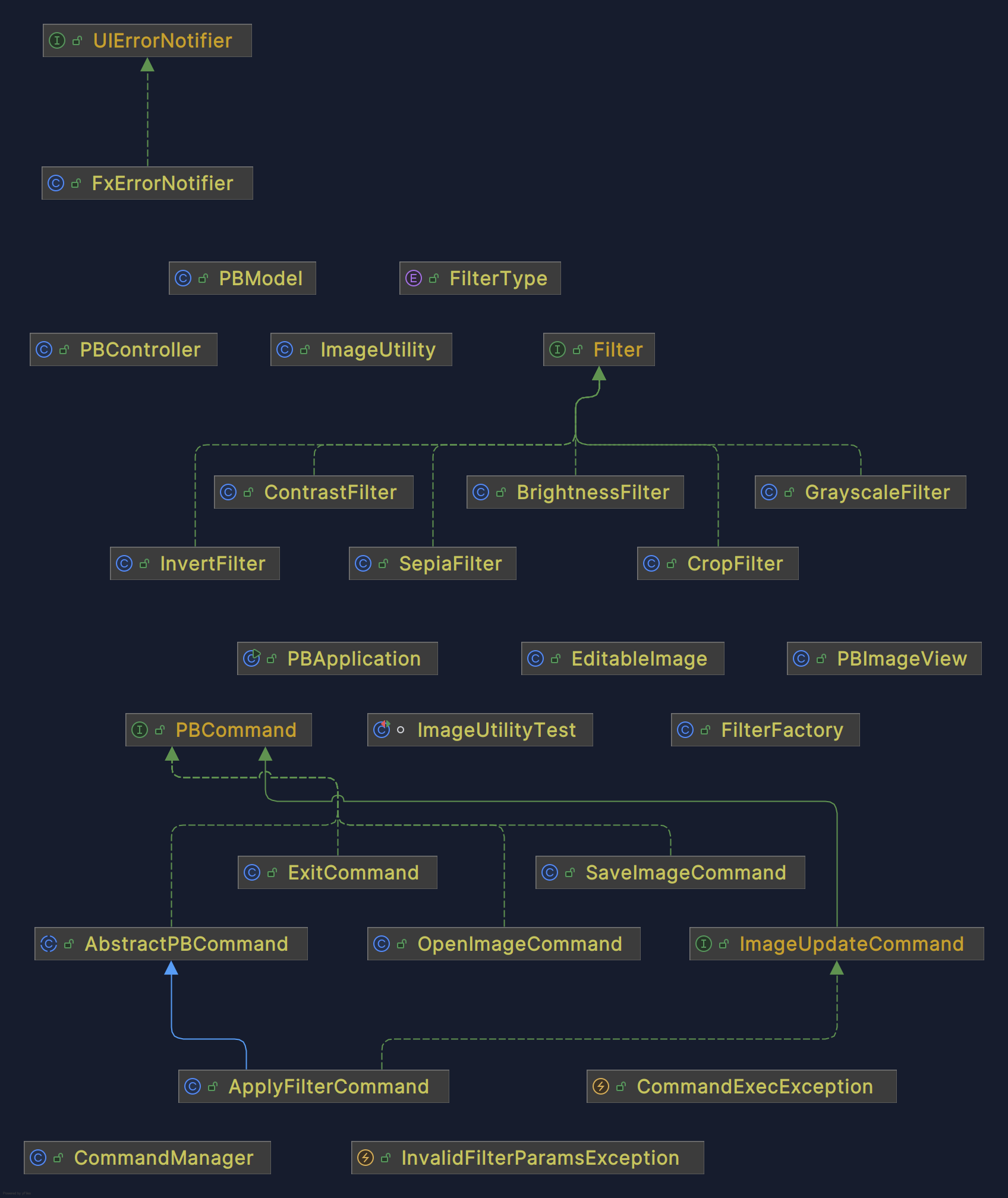
1. Build the project using your preferred IDE or command line tools (e.g., Maven, Gradle).
2. Run the application using the main class PBApplication.

It is recommended to use Intellij IDEA for the three steps above, as it provides excellent support for JavaFX projects and should work out of the box.

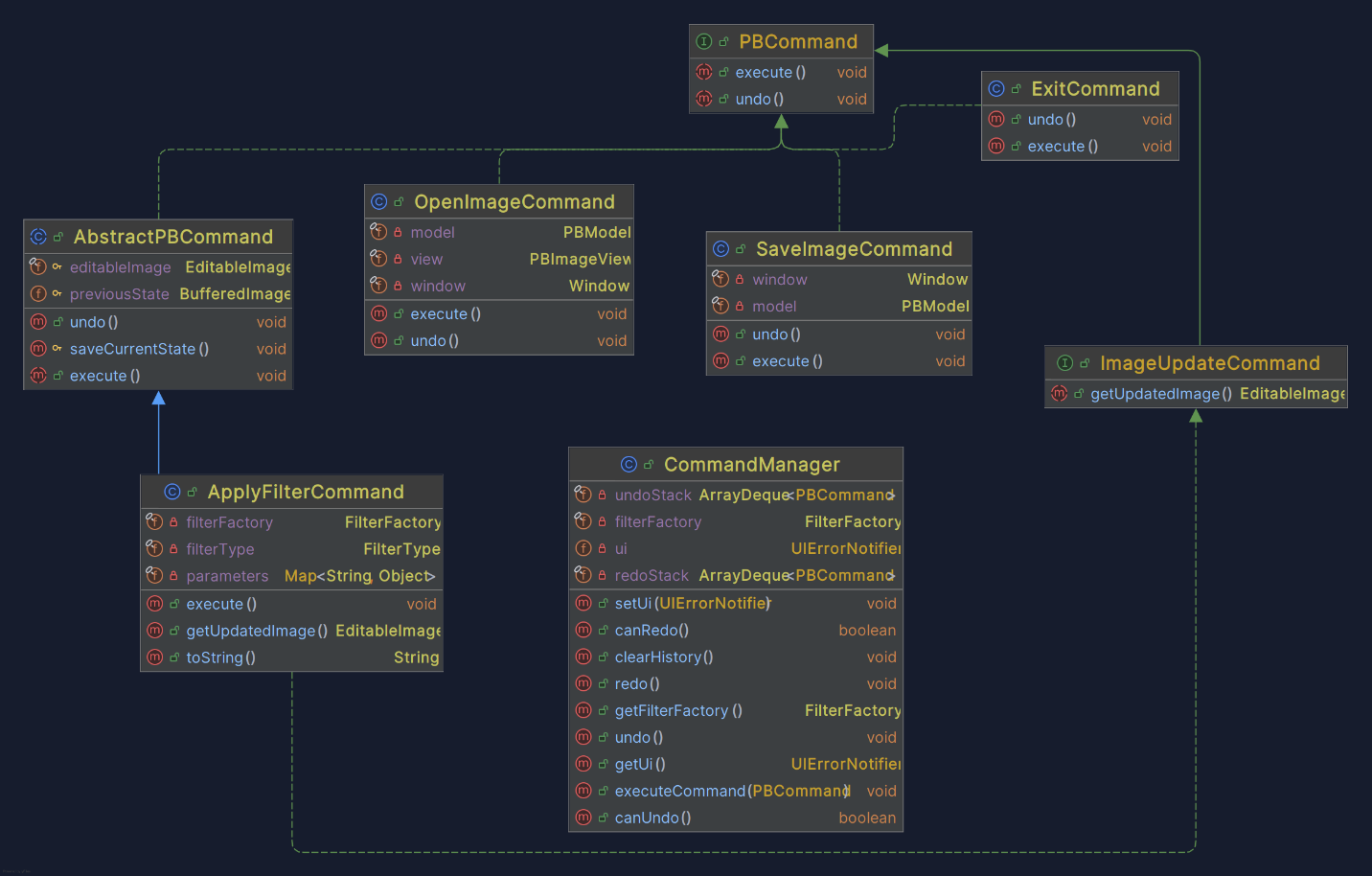
### Basic Usage

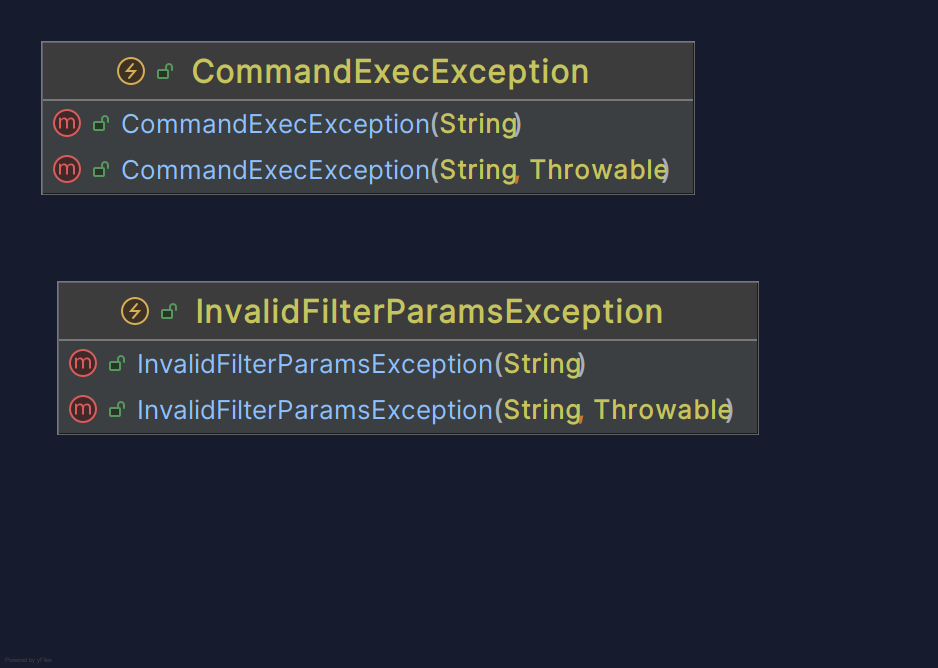
1. **Open an Image**:
   * Use File → Open or the Open button in the toolbar
   * Supported formats: JPG, PNG, GIF
2. **Apply Filters**:
   * Click the filter buttons in the toolbar (Grayscale, Sepia, Invert)
   * Use sliders to adjust brightness and contrast
3. **Crop Image**:
   * Click the Crop button
   * Enter crop dimensions in the dialog
   * Click Apply to crop
4. **Save Changes**:
   * Use File → Save to save your edited image
5. **Undo/Redo**:
   * Use Edit → Undo/Redo or toolbar buttons
   * Reset to original using the Reset button

## UML Diagrams



**Command module:**

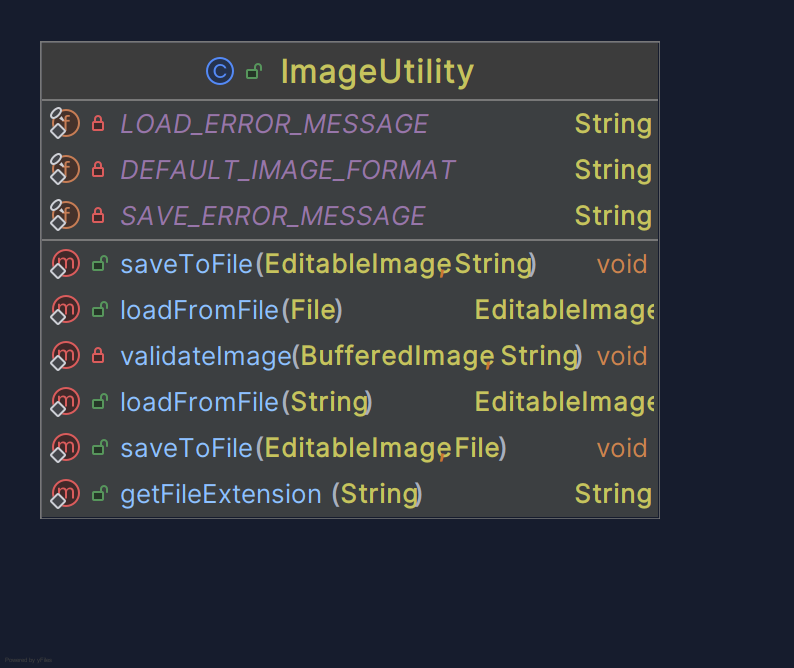


**Exception module:** 

**Model module:**



**Util module:**



**PBController file:**



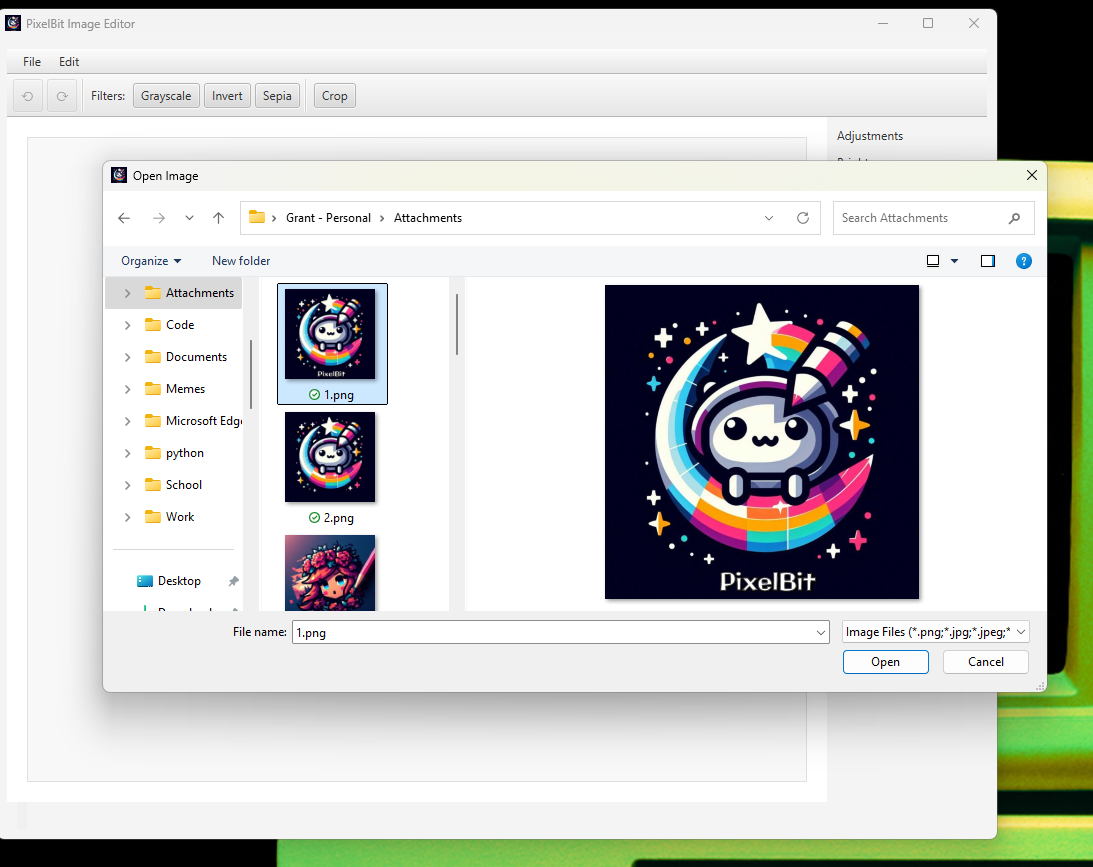
## Screenshots

Loading editor:

A screenshot of a computer

AI-generated content may be incorrect.

Open Image File Chooser:



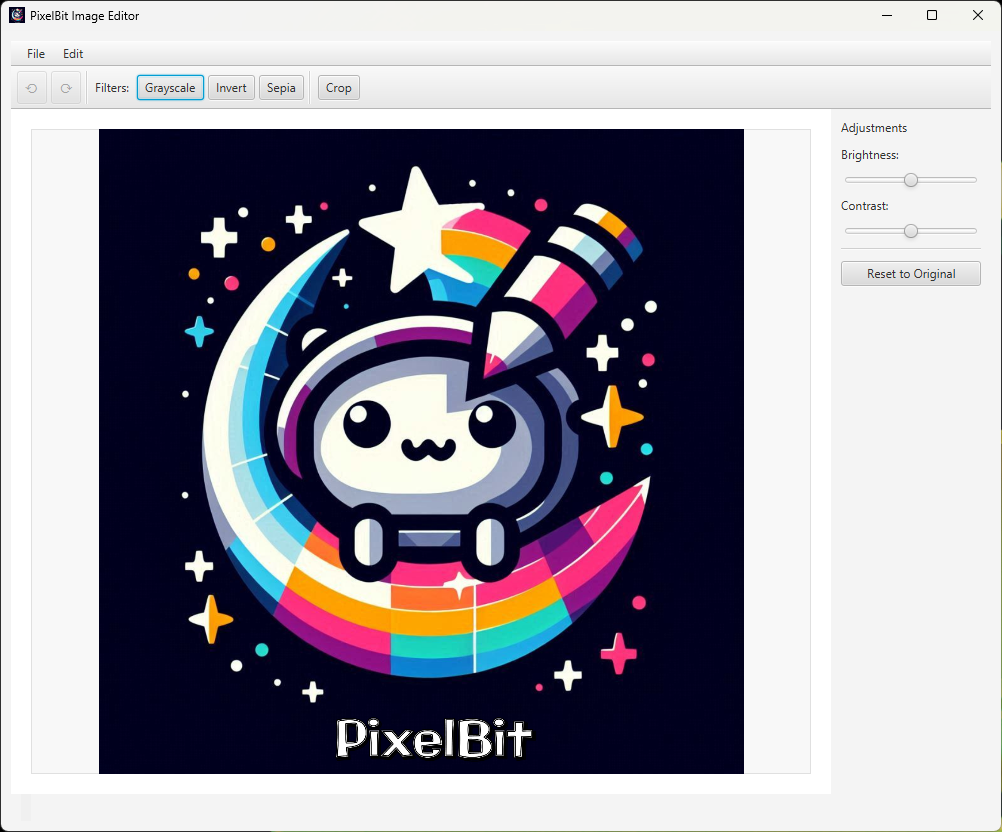
Menu items :

A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

Original Image:



Applying Invert Filter

A screenshot of a computer

AI-generated content may be incorrect.

Applying Grayscale Filter:

A screenshot of a computer

AI-generated content may be incorrect.

Applying Sepia Filter:

A screenshot of a computer

AI-generated content may be incorrect.

Crop Adjustment Panel:

A screenshot of a computer

AI-generated content may be incorrect.

Adjusting Brightness:

A screenshot of a computer

AI-generated content may be incorrect.

Adjusting Contrast:

A screenshot of a computer

AI-generated content may be incorrect.

After cropping using 700 for width and height:

A screenshot of a computer

AI-generated content may be incorrect.