System design document for Paint++

Anthony Tao, Hampus Ekberg, Henrik Tao, August Slveland

28/09-2018
1.0

1 Introduction

General info. What is this? What does it describe?

This is a system design document which describes the system and structure of Paint++.

1.1 Definitions, acronyms, and abbreviations

Some definitions etc. probably same as in RAD MVC - Model-View-Controller

2 System architecture

Paint++ is a raster graphics editor, which is a software program that allows the user to create art and edit images. The overall structure of the program follows the MVC pattern and only requires one computer running Windows as its operative system. The MVC pattern is a software architecture pattern which divides the system into three main components, model, view and controller. The model is responsible for logic and data. The model does not have dependencies with the other components which are not a part of the model. The view is responsible for the graphics of the software, it updates the graphics using logic and data from the model component. The controller is responsible for processing events, such as user interactions, which can result in changes in both the model and view. Since it is a software application it starts by running the paint++.exe file and stops by exiting or closing down the application window.

2.1 Subsystem decomposition

The model of the system contains tool classes and interfaces connected to the tools. The view The controller

2.2 Model

2.3 View

As above, and continue for all components.

2.4 Controller

3 Persistent data management

How does the application store data (handle resources, icons, images, audio, ...). When? How? URLs, pathes, ... data formats... naming..

4 Access control and security

Different roles using the application (admin, user, ...)? How is this handled?

5 References