Report On

# Digital sketchpad

Submitted in partial fulfillment of the requirements of the Course project in

Semester III of Second Year Computer Engineering

By

Rabbani Ansari (Roll No.04)

Soham Bandivdekar (Roll No.08)

Aryan Bhatare (Roll No.16)

Supervisor

Prof. Sneha Mhatre

Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering



(2023-24)

Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

CERTIFICATE

This is to certify that the project entitled “Digital sketchpad” is a bonafide work of Rabbani Ansari (Roll No.04), Soham Bandivdekar (Roll No.08),Aryan Bhatare (Roll No.16) submitted to the University of Mumbai in partial fulfillment of the requirement for the Course project in semester III of Second Year Computer Engineering.

Supervisor

Prof. Sneha Mhatre

Dr Megha Trivedi Dr. H.V. Vankudre

Head of Department Principal

**Abstract:-**

Digital sketch pads have revolutionized the way artists and designers express their creativity in the modern era. This paper delves into the multifaceted realm of digital sketch pads, examining their impact on the creative process, artistic exploration, and technological advancement. Through an exploration of various digital sketch pad software and hardware options, this abstract investigates how these tools empower artists to transcend traditional boundaries, experiment with new techniques, and unlock innovative avenues of expression.

Drawing upon insights from both practitioners and researchers, this abstract highlights the versatility and accessibility of digital sketch pads, providing a comprehensive overview of their features, capabilities, and potential applications. Moreover, it explores the integration of digital sketch pads into educational settings, fostering collaborative learning environments and facilitating skill development across diverse disciplines.

Furthermore, this abstract examines the symbiotic relationship between digital sketch pads and emerging technologies such as artificial intelligence and augmented reality, envisioning new possibilities for artistic creation and interactive experiences. By embracing digital sketch pads as dynamic platforms for creative exploration, individuals from all backgrounds can harness the power of technology to unleash their imagination and shape the future of visual expression.

In conclusion, this abstract underscores the transformative impact of digital sketch pads on the creative landscape, emphasizing their role as catalysts for innovation, collaboration, and artistic evolution in the digital age.

Top of Form

**Index**

**Content:- Pg. No**

**1 Introduction**

1.1 Introduction

1.2 Problem Statement

**2 Proposed System**

2.1 Block diagram , its description and working [ER diagram]

2.2 Module Description

2.3 Brief description of software & hardware used and its programming

2.4 Code

**3 Results and conclusion**

**4 References**

**Introduction:-**

In the ever-evolving landscape of digital artistry, the digital sketch pad stands as a beacon of innovation and creativity. From its humble beginnings as rudimentary drawing software to its current status as a sophisticated tool used by professionals and enthusiasts alike, the digital sketch pad has undergone a remarkable transformation. This introduction sets out to explore the evolution, significance, and impact of digital sketch pads on the realm of visual expression.

Digital sketch pads, also known as graphics tablets, pen displays, or drawing software, represent a convergence of traditional artistry and modern technology. These versatile tools provide artists with a digital canvas upon which they can unleash their imagination, experiment with diverse techniques, and bring their creative visions to life. Unlike traditional mediums, digital sketch pads offer a myriad of features and capabilities, including undo functions, layering options, and a vast array of digital brushes and colors, enabling artists to explore and refine their craft with unprecedented freedom and precision.

The emergence of digital sketch pads has democratized the creative process, making art more accessible to individuals from all walks of life. Whether a seasoned professional seeking new avenues of expression or an aspiring artist exploring their talents, digital sketch pads offer a user-friendly interface and intuitive controls that cater to a wide range of skill levels and artistic styles. Moreover, the versatility of digital sketch pads extends beyond traditional artistry, finding applications in fields such as animation, graphic design, architecture, and more, where precision, efficiency, and creative flexibility are paramount.

Problem statement:-

1. Interface Complexity: Many digital sketchpad software interfaces remain complex and unintuitive, posing significant barriers to entry for beginners and discouraging users from exploring the full range of features and capabilities. Simplifying user interfaces and providing comprehensive tutorials could help mitigate this issue.
2. Cost and Accessibility: High initial costs associated with purchasing digital sketchpad hardware and software licenses limit access to these tools, particularly for individuals with limited financial resources or those residing in economically disadvantaged regions. Increasing affordability through price reductions, subsidies, or alternative financing options could broaden access to digital sketchpads.
3. Learning Curve: The steep learning curve associated with mastering digital sketchpad techniques and workflows can deter aspiring artists from fully embracing these tools. Developing user-friendly educational resources, such as online tutorials, workshops, and interactive guides, could facilitate skill acquisition and promote greater confidence and proficiency among users.
4. Integration with Traditional Methods: Despite their advantages, digital sketchpads often face challenges in seamlessly integrating with traditional artistic techniques and workflows. Enhancing compatibility with traditional mediums and providing tools for easy transition between digital and physical formats could encourage broader adoption among artists who value the versatility of both approaches.
5. Technical Limitations: Technical issues such as input lag, compatibility issues, and hardware/software malfunctions can disrupt the creative process and compromise the quality of artwork produced using digital sketchpads. Continual refinement of hardware and software technology, alongside responsive customer support and troubleshooting resources, is essential to address these challenges and enhance user satisfaction.

**Block diagram , its description and working [ER diagram]:-**

Display Device

Display Interface

Input processing unit

Digital sketchpad

**Module Description:-**

Module Purpose: The Digital Sketchpad Interface module serves as the bridge between the user and the digital sketchpad system, facilitating intuitive interaction and efficient utilization of the digital sketching tools. It encompasses various functionalities to ensure seamless communication between the user's input devices and the digital sketchpad software, enhancing the overall user experience and productivity.

Key Features and Functionalities:

1. User Input Handling:

Capturing and processing user input from input devices such as styluses, graphic pens, or touchscreens.

Interpretation of input gestures, strokes, and pressure sensitivity to accurately reflect the user's artistic intent.

1. Interface Design and Navigation:

Designing an intuitive and user-friendly interface layout for easy navigation and accessibility of sketching tools and functions.

Implementation of menus, toolbars, and customizable workspace options to accommodate diverse user preferences and workflows.

1. Tool Selection and Customization:

Providing a comprehensive array of digital drawing tools, including brushes, pencils, erasers, and color palettes.

Enabling users to customize tool settings, such as brush size, opacity, and pressure sensitivity, to suit their specific artistic style and requirements.

1. Canvas Manipulation:

Handling canvas navigation, zooming, panning, and rotation functionalities to facilitate precise positioning and manipulation of artwork.

Support for multiple layers, allowing users to organize and manage complex compositions effectively.

1. Real-time Feedback and Preview:

Providing real-time visual feedback and preview of strokes, effects, and modifications to enable immediate assessment and adjustment.

Integration of dynamic visual indicators and tooltips to enhance user guidance and feedback during sketching sessions.

1. File Management and Export:

Implementation of file management features for saving, loading, and organizing digital sketches and projects.

Support for exporting artwork in various file formats, including JPEG, PNG, PSD, and SVG, to facilitate sharing and collaboration with others.

**Brief description of software & hardware used and its programming:-**

**Hardware Components:-**

1. Graphics Tablet or Pen Display: The primary input device used for creating digital artwork. It typically consists of a pressure-sensitive surface that simulates drawing on paper and comes with a stylus or pen for precise input.
2. Stylus or Pen: A handheld input device used for drawing, painting, and navigating within the digital sketchpad software. It often features pressure sensitivity and customizable buttons for accessing drawing tools and functions.
3. Display Device: The screen or monitor where the digital artwork is visualized and manipulated. This could be a computer monitor, a standalone pen display, or a tablet device with touchscreen capabilities.
4. Computer System: The underlying hardware infrastructure that powers the digital sketchpad software. This includes the central processing unit (CPU), graphics processing unit (GPU), random access memory (RAM), and storage drives necessary for running the software and storing artwork files.

**Software Components:-**

1. Python: Python is favored for its simplicity, readability, and extensive library support, making it suitable for rapid prototyping and development of digital sketchpad software. Python frameworks like PyQt or Tkinter enable the creation of GUI-based applications with ease, while libraries like OpenCV and PIL provide powerful image processing capabilities.
2. Digital Sketchpad Software: The core application or software program used for creating, editing, and managing digital artwork. It provides a virtual canvas where users can draw, paint, and manipulate images using various tools and features.
3. Drawing Tools: A set of digital brushes, pencils, erasers, and other artistic tools available within the sketchpad software. These tools emulate traditional art mediums and allow users to create diverse styles and effects.
4. User Interface: The graphical interface through which users interact with the digital sketchpad software. It includes menus, toolbars, palettes, and customizable workspace layouts designed to facilitate intuitive navigation and efficient access to drawing tools and functions.

Code:-

import tkinter as tk # Importing the tkinter library as tk

from tkinter import colorchooser, filedialog, messagebox # Importing specific modules from tkinter

class DrawingApp:

def \_\_init\_\_(self, master):

self.master = master

self.master.title("Digital Sketchpad") # Setting the title of the main window

self.master.geometry("800x600") # Setting the initial size of the main window

self.brush\_color = "black" # Default brush color

self.eraser\_color = "white" # Default eraser color

self.brush\_size = 2 # Default brush size

self.opacity = 1.0 # Default opacity

self.setup\_gui() # Setting up the graphical user interface

self.setup\_canvas() # Setting up the drawing canvas

def setup\_gui(self):

# Creating buttons and scales for various functionalities and arranging them within the main window

self.color\_button = tk.Button(self.master, text="Brush Color", command=self.choose\_brush\_color)

self.color\_button.pack(side=tk.LEFT, padx=2)

self.eraser\_button = tk.Button(self.master, text="Eraser", command=self.use\_eraser)

self.eraser\_button.pack(side=tk.LEFT, padx=5)

self.brush\_button = tk.Button(self.master, text="Brush", command=self.use\_brush)

self.brush\_button.pack(side=tk.LEFT, padx=5)

self.clear\_button = tk.Button(self.master, text="Clear Canvas", command=self.clear\_canvas)

self.clear\_button.pack(side=tk.LEFT, padx=5)

self.save\_button = tk.Button(self.master, text="Save Drawing", command=self.save\_drawing)

self.save\_button.pack(side=tk.LEFT, padx=5)

self.size\_label = tk.Label(self.master, text="Brush Size:")

self.size\_label.pack(side=tk.LEFT, padx=5)

self.size\_scale = tk.Scale(self.master, from\_=1, to=20, orient=tk.HORIZONTAL, command=self.change\_brush\_size)

self.size\_scale.set(self.brush\_size)

self.size\_scale.pack(side=tk.LEFT, padx=5)

self.opacity\_label = tk.Label(self.master, text="Opacity:")

self.opacity\_label.pack(side=tk.LEFT, padx=5)

self.opacity\_scale = tk.Scale(self.master, from\_=0.1, to=1.0, resolution=0.1, orient=tk.HORIZONTAL, command=self.change\_opacity)

self.opacity\_scale.set(self.opacity)

self.opacity\_scale.pack(side=tk.LEFT, padx=5)

def setup\_canvas(self):

# Creating a canvas widget for drawing and binding mouse motion event for drawing functionality

self.canvas = tk.Canvas(self.master, bg="white", width=800, height=600)

self.canvas.pack(expand=tk.YES, fill=tk.BOTH)

self.canvas.bind("<B1-Motion>", self.draw)

def draw(self, event):

# Drawing function triggered by mouse motion event

x1, y1 = (event.x - self.brush\_size), (event.y - self.brush\_size)

x2, y2 = (event.x + self.brush\_size), (event.y + self.brush\_size)

self.canvas.create\_oval(x1, y1, x2, y2, fill=self.brush\_color, outline=self.brush\_color, width=int(self.brush\_size\*self.opacity))

def choose\_brush\_color(self):

# Function to select brush color using color chooser dialog

self.brush\_color = colorchooser.askcolor(color=self.brush\_color)[1]

def use\_eraser(self):

# Function to switch to eraser mode (set brush color to eraser color)

self.brush\_color = self.eraser\_color

def use\_brush(self):

# Function to select brush color using color chooser dialog

self.brush\_color = colorchooser.askcolor(color=self.brush\_color)[1]

def change\_brush\_size(self, val):

# Function to change brush size based on scale value

self.brush\_size = int(val)

def change\_opacity(self, val):

# Function to change opacity based on scale value

self.opacity = float(val)

def clear\_canvas(self):

# Function to clear the canvas

self.canvas.delete("all")

def save\_drawing(self):

# Function to save the drawing as an image file

filename = filedialog.asksaveasfilename(defaultextension=".png", filetypes=[("PNG files", "\*.png")])

if filename:

self.canvas.postscript(file=filename + ".eps") # Saving as EPS (Encapsulated PostScript) for better quality

messagebox.showinfo("Drawing Saved", "Your drawing has been saved successfully.")

def main():

root = tk.Tk()

app = DrawingApp(root)

root.mainloop()

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Result:-**

****

**Conclusion:-**

The digital sketchpad stands as a testament to the fusion of artistry and technology, offering artists, designers, and enthusiasts a versatile canvas for unleashing their creativity in the digital realm. Throughout this exploration, we have witnessed the evolution, significance, and impact of digital sketchpads on the landscape of visual expression.

From its humble beginnings as rudimentary drawing software to its current status as sophisticated hardware devices and software applications, the digital sketchpad has transformed the way we create, edit, and share digital artwork. Its intuitive interfaces, diverse toolsets, and real-time feedback mechanisms have empowered users to transcend traditional boundaries, experiment with new techniques, and bring their imaginative visions to life with unprecedented precision and efficiency.

**Refrence:-**

1. Digital Sketching: Tools and Techniques for Product Designers" by James Self: This book offers insights into digital sketching techniques specifically tailored for product designers, covering topics such as sketching software, stylus techniques, and workflow optimization.
2. Digital Painting Techniques: Practical Techniques of Digital Art Masters" edited by 3DTotal Team: While focusing more broadly on digital painting techniques, this book includes sections on digital sketching, offering tips and tutorials from professional digital artists.
3. Sketching User Experiences: The Workbook" by Saul Greenberg, Sheelagh Carpendale, Nicolai Marquardt, and Bill Buxton: Although not solely dedicated to digital sketchpads, this workbook provides valuable insights into the role of sketching in the design process, including digital sketching tools and methodologies.
4. Online Resources:Websites of digital sketchpad software providers such as Adobe Photoshop, Procreate, Autodesk SketchBook, and Clip Studio Paint often offer tutorials, user guides, and forums for users to learn and share tips.