Multimedia Lab (Finals) - Jaish Khan

> You can also prepare for Viva from this

Tools Used in Labs

1. **Lab 1**
    1. *MS Powerpoint* → Used for creating non-linear, interactive kiosks.
2. **Lab 2** + **Lab 3**
    1. *Wick Editor* → A free, open-source tool for creating games, animations, and interactive content.
3. **Lab 4**
    1. *Aseprite/LibreSprite* → Glyph editors for designing pixel-based fonts and logos.
    2. *MS Paint* → Another option for designing pixel-based fonts and logos.
    3. *Pixel Font Generator* → A web-based tool (https://yal.cc/r/20/pixelfont/) used for converting traced pixel glyphs into TTF fonts.
    4. *FontStruct* → An alternative font creation tool (https://github.com/FontStruct/fonthx).
4. **Lab 5**
    1. *GIMP (GNU Image Manipulation Program)* → A free image editing program used for designing icons and logos.
    2. *Adobe Photoshop* → Mentioned as a comparable tool to GIMP, but not specifically used in the labs.
5. **Lab 6**
    1. *HTML5 Canvas* → Used for drawing lines and sprites.
    2. *SVG (Scalable Vector Graphics)* → Used for creating vector-based graphics in HTML5.

Lab 1: Create a Non-Linear, Interactive PowerPoint Kiosk

> This lab introduces the creation of interactive multimedia using *MS PowerPoint*.

Exercise

Create a kiosk about yourself **non-linear, interactive kiosk** using Powerpoint.

Lab 2: Create a Non-Linear, Interactive Wick Editor Project

> This lab focuses on creating interactive multimedia using *Wick Editor*.

It allows users to:

- Upload images or draw shapes and lines.
- Copy and paste images to create animation frames.
- Use the "Tween" feature for smooth transitions.
- Use the "Onion Skin" feature to see previous frame objects.
- Add interactive elements like sliders and checkboxes.
- Add or write JavaScript code.
- Export projects as animated GIFs or MP4 videos.

> ✎ **Exercise**
>
> Create a kiosk about yourself similar to Lab 1, but using **Wick Editor** and exporting it as an interactive HTML file.

Lab 3: Create a Simple Animation using Wick Editor

> This lab focuses on creating simple animations using *Wick Editor*.

Animation is making something come alive visually. Animation techniques covered:

1. *Cel animation* ⟶ Using a series of progressively different graphics for each frame of the animation. Starts with **keyframes** (first and last frames of an action).
   - *Tweening* → Creating in-between frames to depict the action between keyframes.
2. *Kinematics* ⟶ The study of movement and motion of structures with joints.
   - *Inverse kinematics* → Linking objects together, defining their relationships and limits, and then dragging parts while the computer calculates the result.

> ✎ **Exercise**
>
> Students have to:
>
> - Define **keyframes** in animation programs and video compression algorithms.
> - Explain the concept of **tweening**.
> - Create an animation of a rocket launch using **Wick Editor**.
> - Animate their name using various effects.

- Include a Play button to restart the animation.

Lab 4: Designing Pixel-Based Fonts and Logos

> This lab introduces designing **pixel-based fonts** using **glyph editors** (Aseprite or LibreSprite) and **pixel font generators**.

- **Bitmap images** ⟶ Stored as a series of dots called *pixels*. Each pixel is a small square assigned a color and Can be edited by changing individual pixel colors.
- **Vector images** ⟶ Use mathematical formulas to draw lines and curves. Created from geometric shapes like circles and polygons and Edited by manipulating the lines and curves.
  - *Advantages over bitmap images*: Smaller file size, More scalable without losing quality.

Typography concepts

1. *Character* → A general term for letters, numbers, punctuation marks, and other typographic symbols.
2. *Glyph* → A visual representation of a character.
3. *Ligature* → A single glyph that represents two or more characters.
4. *Typeface* → A collection of glyphs.
5. *Font* → A process or tool for selecting and positioning glyphs from a typeface.
6. *Typesetting* → The process of turning characters into a visual presentation of glyphs.
7. *Kerning* → Adjusting the spacing between adjacent glyphs.

Font Generation ⟶ Trace pixel glyphs for the alphabet using tools like MS Paint or Aseprite/LibreSprite. Use a pixel font generator (e.g., https://yal.cc/r/20/pixelfont/) to convert the traced glyphs into TTF fonts.

Logo creation ⟶ Similar to pixel font creation, use MS Paint or Aseprite/LibreSprite to design pixel-based logos.

Exercise

Students have to:

- Explain when to use bitmap or vector images.
- Define "glyph" and "font."
- Create a pixel-based font for the first five letters of the English alphabet and the first five letters of the Urdu or Sindhi alphabet.

- Design a pixel-based logo for the Department of Information Technology and Software Engineering.

Lab 5: Graphic Design: Designing an Icon in GIMP/Photoshop

> This lab introduces using graphic design tools like **GIMP** (GNU Image Manipulation Program) to design logos and icons.

GIMP ⟶ A free image editing program with features similar to Photoshop.

Image Resolution ⟶ The number of pixels an image contains, usually measured in dots per inch (dpi) or dots per centimeter (dpcm). Affects image quality - higher resolution means better quality but larger file size. Doubling the resolution quadruples the file size.

Color Images
* *24-bit color images* → Each pixel is represented by three bytes (RGB), supporting 16,777,216 possible colors. Can be stored as 32-bit images, with the extra byte for each pixel storing an alpha value for special effects like transparency.
* *Color Look Tables* → Used in 8-bit color images to save space. Instead of storing color information for each pixel, the image stores an index that references a table containing 3-byte color values.

Popular file formats

- **GIF:** Graphical Interchange Format.
  - Limited to 8-bit (256) colors.
  - Supports interlacing for gradual image display.
- **JPEG:** Joint Photographic Experts Group.
  - Takes advantage of the limitations of the human vision system to achieve high compression rates.

1. **Images for web** ⟶ Pixel dimensions are more important than resolution. Images with the same pixel dimensions will display at the same size on the web, regardless of resolution.
2. **Changing image resolution in GIMP** ⟶ Go to *Image > Print Size*. Enter the desired resolution in the *X and Y Resolution* fields. Note that changing the resolution will change the document's width and height, but not the number of pixels.
3. **Resolution and printing** ⟶ Higher resolution generally results in better print quality. Find a resolution that balances desired print size and quality.

Exercise

Students have to:

- Calculate the storage space required for a 600×200 pixel image at 24-bit color depth.
- Convert a simple image from RGB to indexed color and create a color lookup table.
- Explain the difference between GIF and JPEG file formats.
- Design a logo for the Department of Information Technology and Software Engineering using GIMP or Photoshop.
- Create an icon for the Multimedia System and Design subject using GIMP or Photoshop.
- Perform image resolution changing operations.

Lab 6: Graphics: Drawing Lines and Sprites in HTML5 using CANVAS

> This lab covers drawing lines and using sprite images in **HTML5 Canvas**.

Image Sprite → A collection of images combined into a single image. This reduces the number of server requests and saves bandwidth.

Graphics APIs in HTML

- **2D Graphics** → *Canvas* uses JavaScript to draw graphics on the fly. *SVG* defines vector-based graphics in XML format.
- **3D Graphics** → *WebGL* (can also be used for 2D) with Canvas.

Feature	Canvas	SVG
Resolution	Dependent	Independent
Event handlers	No support	Support
Text rendering	Poor	Good
Image saving	Can save as PNG or JPG	N/A
Suitability	Graphic-intensive games	Applications with large rendering areas (e.g., Google Maps)

- **Drawing Lines:**
  - Explains the concept of line width and how to draw crisp lines on canvas.
  - Provides JavaScript code to draw a grid of lines.

Exercise

Students have to:

- Draw rectangles, circles, and lines using both Canvas and SVG elements.
- Use a sprite image in Canvas to:
  - Draw the sprite image.
  - Create a simple animated sprite.
  - Make the sprite follow the mouse on click.