**Mary Pickford – CoolPaint++**

**Description:**

CoolPaint++ is a paint-like web application for drawing lines, shapes and other drawing stuff.

**Main Modules:**

main.js – represents the global variables and functions in the app

* ctx – global object containing canvas context
* mouseClicked – bool variable shows if mouse is clicked or not
* keyPressed – bool variable shows if key is pressed or not
* mousePositionX – X position of the mouse
* mousePositionY – Y position of the mouse
* currentTool – object containing the current drawing tool
* tools – array containing all tools created by the app
* makeTool(name, cursor, onMouseDown, onMouseMove, onMouseUp, updateToolSettings) – creates tool and adds it to the tools array
  + name – name of the created tool
  + cursor – link to the tool’s cursor
  + onMouseDown – code which runs on mouse down for the current tool
  + onMouseMove - code which runs on mouse move or the current tool
  + onMouseUp - code which runs on mouse up for the current tool
  + updateToolSettings – function called before using the tool, and after some change

engine.js – represents the logic and the relationship between the tools and the UI

* init(canvas, width, height) – initializes the application
  + canvas – represents canvas html object
  + width – represents width of the canvas
  + height – represents height of the canvas
* changeTool(toolItem) – sets the current tool to the specified tool
  + toolItem – current tool in the menu
* addToolsInTheSidebar() – adds all created tools to the sidebar menu
* updateToolSettings() – calls current tool update settings and update global app settings
* updateCursorColor(color) – shows the current selected color from the pallete next to the cursor
  + color – selected color
* onMouseDown() – calls the current tool mouseDown event (if available)
* onMouseUp() – calls the current tool mouseUp event (if available)
* onMouseMove() – calls the current tool mouseMove event (if available)
* onKeyDown – code evaluated when key is pressed (used for redo and undo)
* clearCurrentLayer() – clears the whole canvas

logger.js – module for redo and undo actions

* logAction() – adds canvas action to an array with actions
* undoAction() – removes the last done action
* redoAction() – adds the last undo action

**Tools:**

bigBrush.js –

brushTool.js – represents brush for free drawing

changeSizing.js –

circleTool.js – represents circle with specified radius

cloudTool.js – represents cloud figure

ellipseTool.js – represents ellipse which is drawn with dragging

eraserTool.js – represents an eraser for deleting drawings

eyeDropperTool.js – represents a tool for picking a color from the canvas

loadImage.js – represents a tool for loading an image from URL

paintBucketTool.js – fills an area of pixels with similar colors( within a given border or the entire canvas)

pickColor.js – extracts pixel data from the current x and y on the canvas

rectTool.js – represents rectangle which is drawn with dragging

savePicture.js – represents an auto download functionality – extracts url from canvas and sets <a> tag href.

sprayCanTool.js – represents a pattern of pixels scattered on random onto the canvas

starTool.js – represents tool for drawing star figures

triangleTool.js – represents tool for drawing triangles

writeTextTool.js – represents tool for writing text on the canvas

**UI Components:**

alphaChanger.js –

colorPickerJqueryUI.js –

colorsSlider.js –

load.js –

sideBar.js –

svgLoader.js – scripts for a SVG loader when running app