**AlgoSurg: Markerless Tracking for Surgical Navigation**

We start by creating a small directory for saving our synthetic dataset. `dataset\_dir` is for the actual values, while `masks\_dir` is for the corresponding masked images. We define an image size and background colour as follows: The notation (255, 255, 255) denotes the colour white on the RGB scale.

In the provided code we generate a synthetic dataset comprising 10 images. Each image contains randomly positioned shapes such as rectangles, circles, or triangles, drawn in varying sizes and colours on a blank canvas. The code also creates corresponding binary masks for each image based on the shapes drawn, saving both images and masks in separate directories. We can use this dataset to train a simple CNN model.

This code iterates over each synthetic image and its corresponding mask, loads them using PIL, and displays them side by side using Matplotlib subplots. The synthetic images are displayed in colour, while the masks are displayed in grayscale for clarity. Finally, `plt.tight\_layout()` ensures that the subplots are neatly arranged, and `plt.show()` displays the figure containing the images and masks.

The provided code builds and trains a convolutional neural network (CNN) for image segmentation using synthetic images and masks. The model architecture consists of an encoder-decoder structure with convolutional layers for feature extraction and up sampling layers for image reconstruction. The network comprises encoder and decoder sections. The encoder extracts feature with convolutional layers and max-pooling. The decoder reconstructs masks using up-sampling and concatenation layers. A sigmoid-activated convolutional layer generates binary masks for segmentation. The model is trained for 10 epochs using the Adam optimizer and binary cross entropy loss. It achieves segmentation by outputting binary masks through a sigmoid activation function. The trained model is saved for future use.

Test Loss: 0.0165

Test Accuracy: 0.995.

Below are some of the masked images from the validation dataset.

