|  |
| --- |
| **Main**  public static void main(String args[])  private static void setup()  private static void drawEdge()  public static int getColor(int r, int g, int b) |

|  |
| --- |
| **ClickableWindow**  public ClickableWindow(Webcam webcam)  public void mouseClicked(MouseEvent arg0)  public void mouseEntered(MouseEvent arg0)  public void mouseExited(MouseEvent arg0)  public void mousePressed(MouseEvent arg0)  public void mouseReleased(MouseEvent arg0) |

|  |
| --- |
| **Master\_Detector**  public Master\_Detector(byte[] newPixels, int height, int width)  public boolean isEdge(int row, int col)  public void updateImage(byte[] newPixels)  private boolean hasAdjacentEdge(int row, int col) |

|  |
| --- |
| **Detector**  protected Detector(byte[] newPixels, int height, int width)  protected abstract boolean canBeEdge(int row, int col)  protected abstract void convertTo2DInt(byte[] newPixels)  protected int convertToIntColor(byte color) |

|  |
| --- |
| **Motion**  public Motion(byte[] newPixels, int height, int width)  protected boolean canBeEdge(int row, int col)  protected void convertTo2DInt(byte[] newPixels)  private static int getColor(int r, int g, int b)  private double getDistance(int color1, int color2)  private static int getRed(int pixel)  private static int getGreen(int pixel)  private static int getBlue(int pixel) |

|  |
| --- |
| **BWDetector**  protected BWDetector(byte[] newPixels, int height, int width)  protected abstract boolean canBeEdge(int row, int col)  protected void convertTo2DInt(byte[] newPixels)  protected boolean isBorder(int row, int col) |

|  |
| --- |
| **Vertical\_Horizontal**  public Vertical\_Horizontal(byte[] newPixels, int height, int width)  protected boolean canBeEdge(int row, int col)  private boolean isVerticalEdge(int row, int col)  private boolean isHorizontalEdge(int row, int col) |

|  |
| --- |
| **Brightness**  public Brightness(byte[] newPixels, int height, int width)  protected boolean canBeEdge(int row, int col) |

Overview

The class which runs is the Main class, which creates a ClickableWindow to display the webcam footage and a Master\_Detector object to draw the edges around objects. The Master\_Detector is the main detector object which combines the Motion detector, Brightness detector, and Vertical\_Horizontal detector into one advanced detector. The three previously mentioned detectors implement the Detector abstract class and the Brightness and Vertical\_Horizontal detectors implement the BWDetector abstract class because they both use black and white pixel values to detect edges. The Brightness detector uses the row difference and column difference in brightness to detect edges. The Vertical\_Horizontal detects all horizontal and vertical edges. The Motion detector detects the edges by using the change in pixels between frames. All of these detectors have some threshold value to determine whether a pixel is an edge or not. The Master\_Detector combines all the detectors' edge detections and then checks if each edge is adjacent to at least two other edges. The Master\_Detector saves each edge in a 2D boolean array which is accessed by the Main class when it draws the edges on the window.

Design Decision: the Master\_Detector class

Initially, the Blob detection system only featured the Motion detector. The problem with that was that the algorithm detected too much and it was not completely accurate. To solve this problem, I added the Brightness detector, but at this point it was in the same Detector class as the Motion detector. While adding the Brightness filter was surely an improvement, there was still a lot more room for improvement. I finally decided to add the Vertical\_Horizontal detector which improved the detector to the level it's at today. All three detectors were in the single Detector class, which made the code seem very disorganized. At that point it made sense to have each detector as its own object which extends an abstract Detector class. This would ensure that methods which only applied to one detector could be placed in that detector's class, which improved the legibility of the code. The Brightness detector and Vertical\_Horizontal detector ended up sharing a few methods so to avoid redundancy, a BWDetector abstract class was created. The current design of the code makes it much more legible and makes it easier to debug. This design decision also allows a collaborator to add more detectors easily, without disturbing the already made detectors.