

## 0.1 Synopsis

The shop window of choice for this project was the (in)famous döner shop in the heart of Enschede: Famly (see the picture below). From this window, I imagined a game where you could go inside and shave your own döner, which rarely happens in real life. So, that's quite literally what I built: A sequence that zooms in on an image of the Famly order window, followed by a simple rendition of its interior; The döner cone, a shaver, and a character that might work at Famly.

## 0.2 Usage and interactions

The interactions facilitated by this program are the following:

- After a short explanation of the rules, the keys *w* or *m* can be used to select either of the WiiMote or mouse modes, respectively. If *w* is pressed, a short instruction to set up a wiimote with the program will follow.
- Having zoomed in, a timer will start counting down, and the player can initiate his döner-shaving by moving the mouse or the WiiMote. If the player shaves up fast enough, pieces of döner will spawn and fall off the cone, along with which the score will go up. However, if the player doesn't shave *smoothly enough*, or shaves *too fast*, darker, burnt pieces will spawn that subtract from the score.
- When the timer runs out, a box with "GAME OVER" will appear, and the final score will continue to display. The player can then press *r* to reset the game to the rule explanation above.
- While either of the mouse buttons is held, the shaver will visually vibrate. If the WiiMote is used, pressing A will cause the shaver to vibrate, and pressing B (the trigger) will cause the shaver to vibrate *along with* rumble in the WiiMote. Especially this last case has a very nice effect; It feels like you're holding the actual shaver.

## 0.3 Architecture

In this program, the introduced classes and their methods are:

- FamlyFinal (the main class), containing: *setup()*, *draw()*, *keyPressed()*, *mouseMoved()*, *mousePressed()*, *mouseReleased()* and *mouseDragged()*.
- Doner, containing: *display()*.
- DonerMan, containing *display()*, *makeBlinkTrue()* and *makeBlinkFalse()*.
- DonerPiece, containing *display()* and *changePos()*.
- MenuScreen, containing *display()* and *displayInstructions()*.
- Scoreboard, containing *display()*, *scorePlusUpdate()*, *scoreMinusUpdate()*, *runGameCountDown()* and *resetScoreboard()*.
- Shaver, containing *display()*, *changePos()*, *velocityVector()*, *wiiRollHeadingStepAnticlockwise()*, *wiiRollHeadingStepClockwise()*, *wiiRollHeadingStepCentre()* and *vibrate()*.
- ShopFront2, containing *display()* and *hideShopFront()*.

## 0.4 Call-graph of the project

Behold below the call-graph of this project. keyPressed until mouseDragged aren't connected, since they only influence the state of multiple global boolean variables.

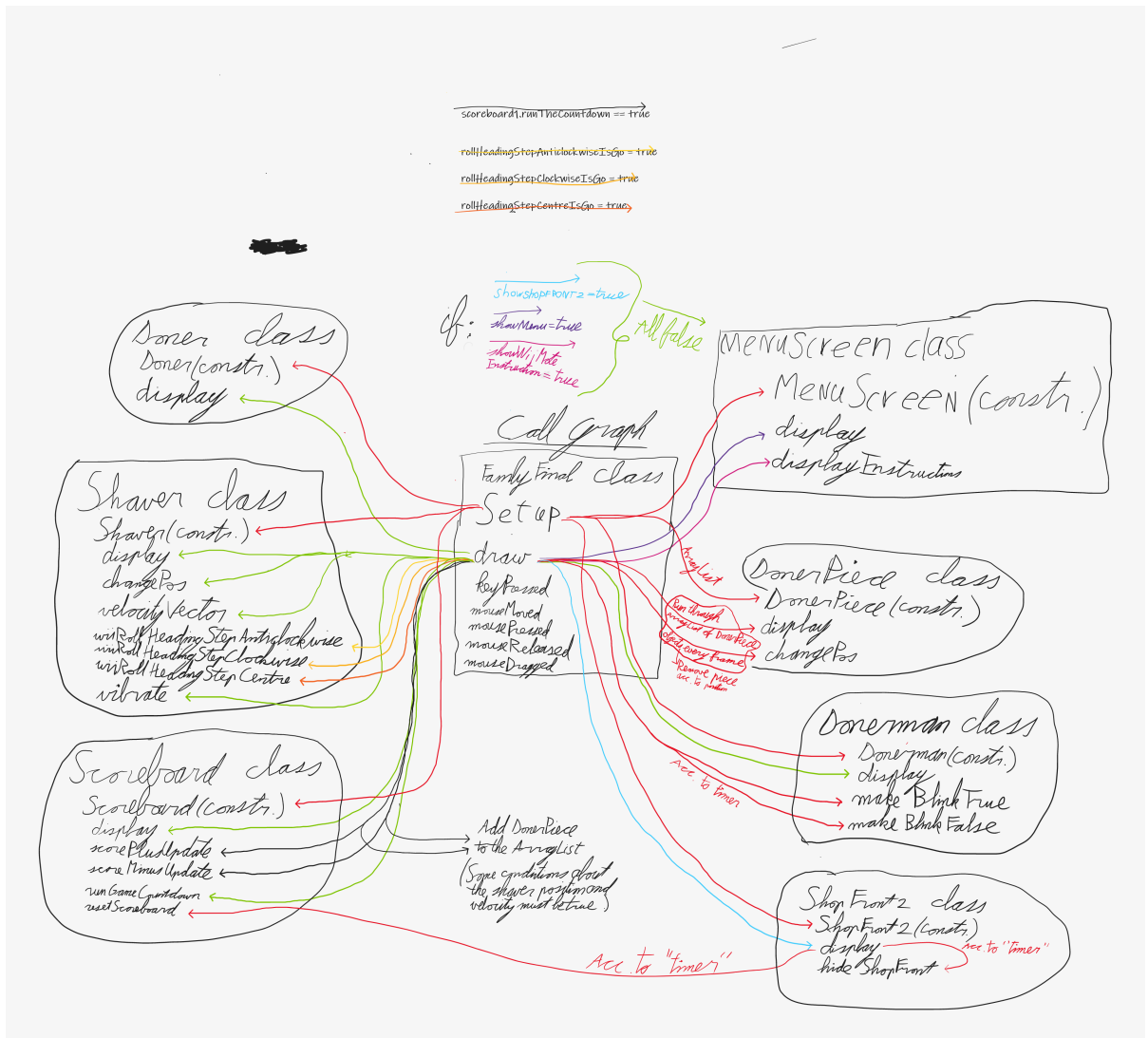


Figure 1: A call-graph of the project. Note that different-coloured lines convey different conditionals.