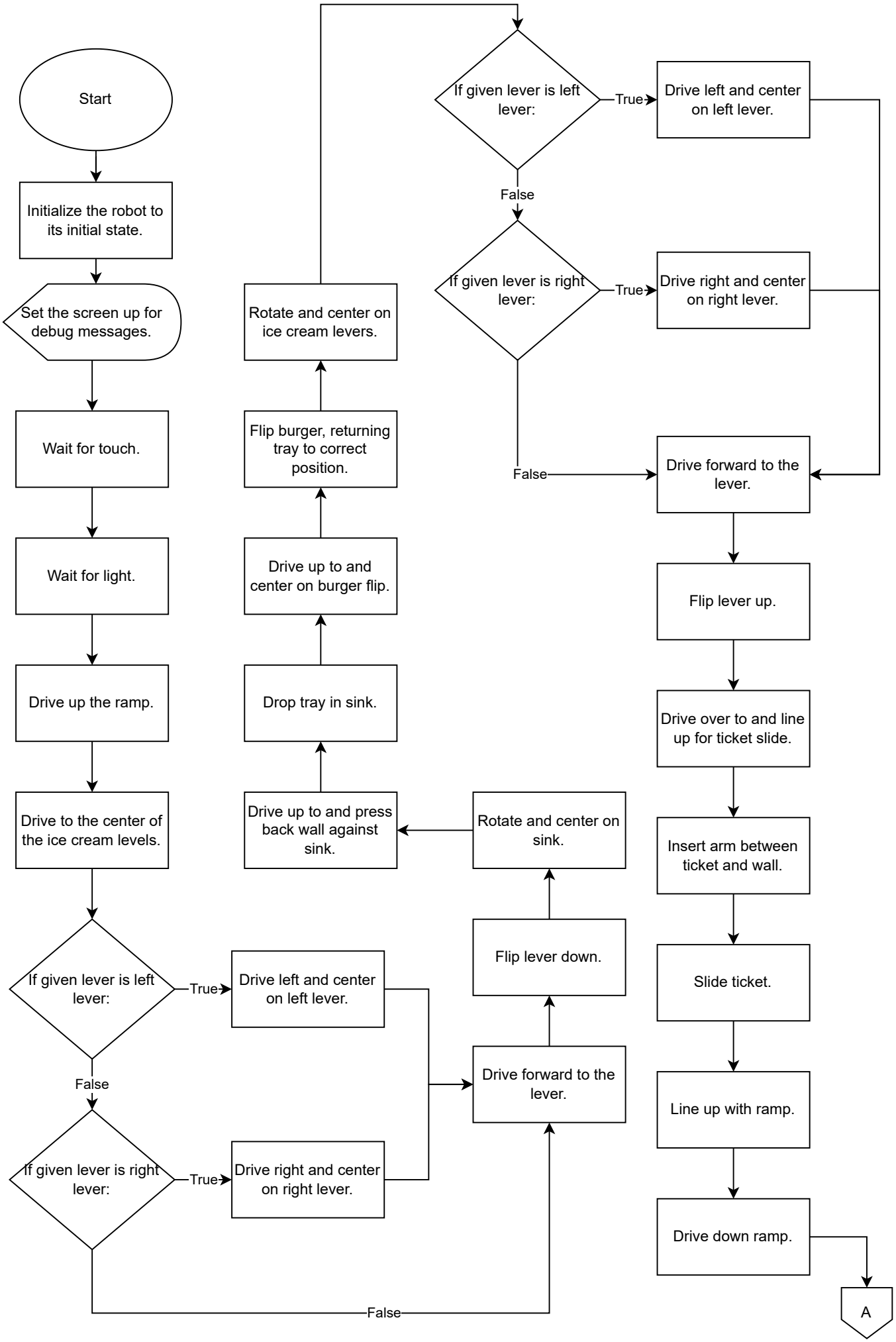
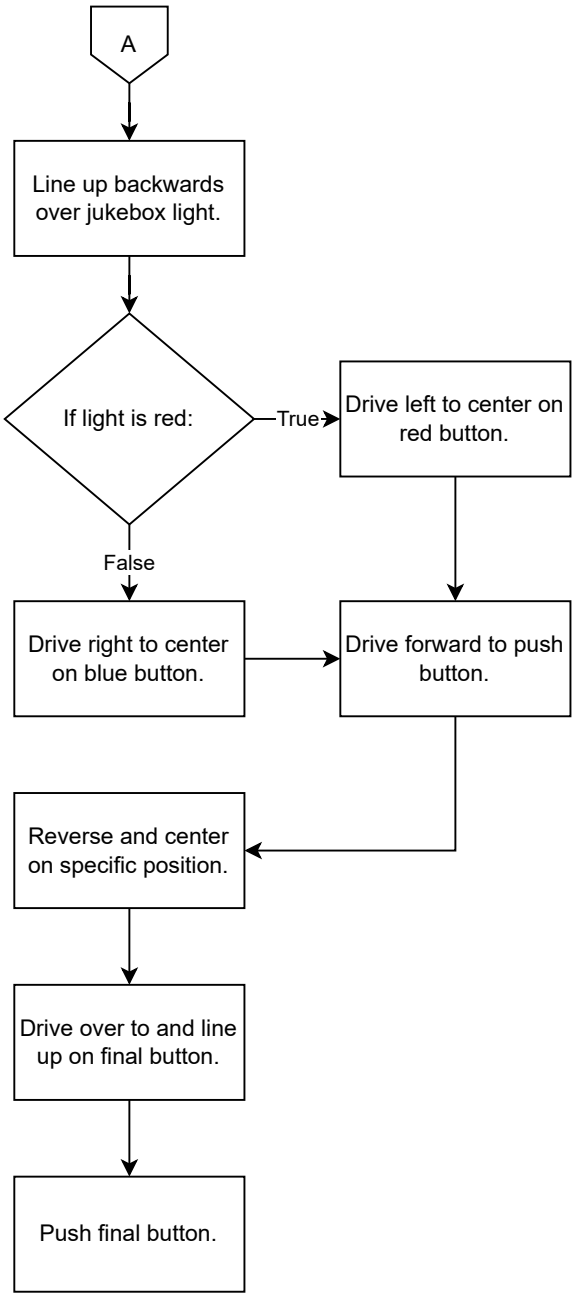
E6: Alex Felderean, Lauren Hawkinson, Grace Jiang, Daniel Rihm
Clingan 12:40





Logic

Code is written to be as modular as possible, with nearly every process being a method call; each call starts and ends at specific states to allow the logic or behavior of given methods to be changed completely, so long as they still start and end in the same state.

First the robot waits for the light, then it drives up the ramp and flips the lever so that the ice cream timer is running while it does other tasks. Then it drops the tray in the sink and flips the burger before flipping the ice cream lever back up. It then does the ticket slider since it has easier access on the upper level. Then it returns to the lower level to do the only remaining task: the jukebox. Then the robot ends the run by pushing the final button.

E6: Alex Felderean, Lauren Hawkinson, Grace Jiang, Daniel Rihm

Clingan 12:40

Function: Robot::moveUnbounded

Description:

```
/**
 * @brief Moves the robot in the specified direction at the specified speed.
 * Requires 0 < speed < 100 AND 0 < angle < 360.
 *
 * @param angle The angle at which the robot moves at in degrees.
 * @param speed The speed at which the robot should move as a percent
 * of the max motor speed.
 */
```

Pseudocode:

```
function moveUnbounded(var angle, var speed):
   var motorFS = speed * cos(MOTOR_FRONT_ANGLE * PI / 180.0 - angle * PI / 180.0)
   var motorLS = speed * cos(MOTOR_LEFT_ANGLE * PI / 180.0 - angle * PI / 180.0)
   var motorRS = speed * cos(MOTOR_RIGHT_ANGLE * PI / 180.0 - angle * PI / 180.0)

   setMotorPercent(motorFS, motorLS, motorRS)
END moveUnbounded
```

And `setMotorPercent` is formally defined as:

```
/**
 * @brief Sets all motors' percents to the specified amounts.
 * Requires 0 < [all parameters] < 100.
 *
 * @param fSpeed The front motor's speed.
 * @param lSpeed The left motor's speed.
 * @param rSpeed The right motor's speed.
 */
```

And `cos` is formally defined as:

```
/**
 * @brief Computes cosine.
 *
 * @param x Any angle in radians.
 * @returns The cosine of x.
 */
```