```
1   // Eugene Ngo
2   // 11/18/2022
3   // EE 271
4   // Lab 4
5
6   // normalLight module takes in clk, reset, L, R, NL, NR as inputs and they are all 1-Bit
    and returns
7   // lightOn which is also 1-Bit. This module determines the next state based on the inputs
    and moves the light
8   // to the left if key 3 is pressed and moves the light to the right if key 0 is pressed.
    When the game is reset
9   // these lights are turned off.
10
11  module normalLight(lightOn, clk, reset, L, R, NL, NR);
12
13     output logic lightOn;
14     input logic clk, reset;
15     input logic L, R, NL, NR;
16
17     enum {on, off} ps, ns;
18
19     always_comb begin
20        case(ps)
21           on:     if(~R & L | R & ~L) ns = off;
22                   else ns = on;
23
24           off:    if(NR & L & ~R | NL & R & ~L) ns = on;
25                   else ns = off;
26
27        endcase
28     end
29
30     always_comb begin
31        case(ps)
32           on: lightOn = 1;
33
34           off: lightOn = 0;
35        endcase
36     end
37
38     always_ff @(posedge clk) begin
39        if(reset)
40           ps <= off;
41        else
42           ps <= ns;
43     end
44
45  endmodule
46
47  // normalLight_testbench tests all expected, unexpected, and edgecase behaviors of the
    lights.
48  module normalLight_testbench();
49     logic L, R, NL, NR;
50     logic lightOn;
51     logic clk, reset;
52
53     normalLight dut (.lightOn, .clk, .reset, .L, .R, .NL, .NR);
54
55     //Set up the clock
56     parameter CLOCK_PERIOD = 100;
57     initial begin
58        clk <= 0;
59        forever #(CLOCK_PERIOD / 2)
60        clk <= ~clk;
61     end
62
63     //Set up the inputs to the design. Each line is a clock cycle
64     initial begin
65                                          @(posedge clk);
66        reset <= 1;                       @(posedge clk);
67                                          @(posedge clk);
68        reset <= 0;                       @(posedge clk);
69                                          @(posedge clk);
```

```
70          L <= 1; R <= 0; NL <= 0; NR <= 1;   @(posedge clk);
71                                              @(posedge clk);
72                                              @(posedge clk);
73                                              @(posedge clk);
74                          NL <= 1; NR <= 0;   @(posedge clk);
75                                              @(posedge clk);
76                                              @(posedge clk);
77                                              @(posedge clk);
78          L <= 0; R <= 1;                     @(posedge clk);
79                                              @(posedge clk);
80                                              @(posedge clk);
81                                              @(posedge clk);
82                          NL <= 0; NR <= 1;   @(posedge clk);
83                                              @(posedge clk);
84                                              @(posedge clk);
85                                              @(posedge clk);
86          L <= 1;         NL <= 1; NR <= 0;   @(posedge clk);
87                                              @(posedge clk);
88                                              @(posedge clk);
89                                              @(posedge clk);
90          reset <= 1;                         @(posedge clk);
91                                              @(posedge clk);
92                                              @(posedge clk);
93          L <= 0; R <= 1; NL <= 1; NR <= 0;   @(posedge clk);
94                                              @(posedge clk);
95                                              @(posedge clk);
96          L <= 1; R <= 0; NL <= 0; NR <= 0;   @(posedge clk);
97                                              @(posedge clk);
98                                              @(posedge clk);
99          L <= 0; R <= 1; NL <= 0; NR <= 0;   @(posedge clk);
100                                             @(posedge clk);
101                                             @(posedge clk);
102         $stop;
103     end
104 endmodule
```