

```

1  module draw_game (clk, reset, gameover, frog_x, frog_y, car2, car5, car7, car9, car11, car12
, RedPixels, GrnPixels);
2      input logic clk, reset, gameover;
3      input logic [3:0] frog_x, frog_y;
4      input logic [15:0] car2, car5, car7, car9, car11, car12;
5      output logic [15:0][15:0] RedPixels; // 16x16 array of red LEDs
6      output logic [15:0][15:0] GrnPixels; // 16x16 array of green LEDs
7
8      always_ff @(posedge clk) begin
9
10
11         if (reset) begin
12             RedPixels <= 0;
13             GrnPixels <= 0;
14         end
15
16         RedPixels <= 0;
17         RedPixels[2] <= car2;
18         RedPixels[5] <= car5;
19         RedPixels[7] <= car7;
20         RedPixels[9] <= car9;
21         RedPixels[11] <= car11;
22         RedPixels[12] <= car12;
23
24         // we update the cars no matter what, but only update GrnPixels if the game is still
going // gameover will STAY set to 1 once player loses.
25         if (gameover) begin
26             RedPixels[frog_x][frog_y] <= 1'b1;
27         end
28         else begin
29             GrnPixels <= 0;
30             GrnPixels[frog_x][frog_y] <= 1'b1;
31             RedPixels[frog_x][frog_y] <= 0;
32         end;
33     end
34 endmodule
35
36 module draw_game_testbench();
37     logic clk, reset, gameover;
38     logic [3:0] frog_x, frog_y;
39     logic [15:0] car2, car5, car7, car9, car11, car12;
40     logic [15:0][15:0] RedPixels; // 16x16 array of red LEDs
41     logic [15:0][15:0] GrnPixels; // 16x16 array of green LEDs
42
43     draw_game dut(clk, reset, gameover, frog_x, frog_y, car2, car5, car7, car9, car11, car12,
RedPixels, GrnPixels);
44
45     // Set up the clock.
46     parameter CLOCK_PERIOD=100;
47     initial begin
48         clk <= 0;
49         forever #(CLOCK_PERIOD/2) clk <= ~clk;
50     end
51
52     initial begin
53
54         @(posedge clk);
55         reset <= 1;
56         @(posedge clk);
57         reset <= 0;
58         @(posedge clk);
59         @(posedge clk);
60         // assign default values from start of game
61         @(posedge clk);
62         @(posedge clk);
63         @(posedge clk);
64         @(posedge clk);
65         frog_y <= 8; frog_x <= 0;
66         @(posedge clk);
67         car2 <= 16'b1110000110011111;
68         car5 <= 16'b1011110011111000;
69         car7 <= 16'b0111100111101001;
70         car9 <= 16'b1111100000111011;
71         car11 <= 16'b1100011111100011;
72         car12 <= 16'b1110011111110000;
73         @(posedge clk);

```

```
74      @(posedge clk);
75      @(posedge clk);
76      @(posedge clk);
77      @(posedge clk);
78      @(posedge clk);
79      @(posedge clk);
80      @(posedge clk);
81      @(posedge clk);
82      @(posedge clk);
83      @(posedge clk);
84      @(posedge clk);
85      @(posedge clk);
86      @(posedge clk);
87
88      $stop; // End the simulation.
89  end
90
91  endmodule
92
```