

```
timescale 1ns / 1ps
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 05/27/2024 09:03:48 PM
// Design Name:
// Module Name: gameRGB
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
module gameRGB(
    input btnC,
    input btnU,
    input btnL,
    input btnR,
    input [15:0] sw,
    input clk,
    input frame,
    output [15:0] leds,
    input [15:0] Vpos,
    input [15:0] Hpos,
    output [3:0] red,
    output [3:0] green,
    output [3:0] blue,
    output [9:0] iceHit,
    output [9:0] iceColor
);
    wire flash, frozen;
    wire leftborder, rightborder, topborder, underwater, bottomborder, atrest;
//collision detection
    wire [15:0] slugH, slugV, slugYDin, iceY1, iceY2, iceY3, iceY4, iceY5, iceY6,
iceY7, iceY8, iceY9, iceY10;
    wire [39:0] fades1, fades2, fades3, fades4, fades5, fades6, fades7, fades8,
fades9, fades10;
```

```

wire [7:0] randy;
wire border, iceberg, water, slug, icey1, icey2, icey3, icey4, icey5, icey6,
icey7, icey8, icey9, icey10; //static background elements

assign border = (Vpos < 8) & (Vpos >=0) | (Vpos > 16'd471 & Vpos < 16'd480) |
(Vpos > 16'd7) & (Vpos < 16'd472) & (Hpos >= 16'd0) & (Hpos <= 16'd7) | (Vpos >
16'd7) & (Vpos < 16'd472) & (Hpos >= 16'd631) & (Hpos <= 16'd639);
assign water = ~border & (Vpos >= 16'd359 & Vpos <= 16'd479);
assign iceberg = ~border & ((Vpos >= 16'd287 & Vpos <= 16'd479) & (Hpos >=
16'd279 & Hpos <= 16'd343));
assign slug = ((Hpos >= slugH) & (Hpos <= slugH + 16'd16) & (Vpos >= slugV) &
(Vpos <= slugV + 16'd16)) & (1'b1 & ~bottomborder | flash & bottomborder);
assign icey1 = ((Hpos >= 16'd29) & (Hpos <= 16'd29 + 6) & (Vpos >= iceY1) &
(Vpos <= iceY1 + 40) & ~water & ~iceberg) & (1'b1 & ~iceHit[0] | flash & iceHit[0]);
assign icey2 = ((Hpos >= 16'd87) & (Hpos <= 16'd87 + 6) & (Vpos >= iceY2) &
(Vpos <= iceY2 + 40) & ~water & ~iceberg) & (1'b1 & ~iceHit[1] | flash & iceHit[1]);
assign icey3 = ((Hpos >= 16'd145) & (Hpos <= 16'd145 + 6) & (Vpos >= iceY3) &
(Vpos <= iceY3 + 40) & ~water & ~iceberg) & (1'b1 & ~iceHit[2] | flash & iceHit[2]);
assign icey4 = ((Hpos >= 16'd203) & (Hpos <= 16'd203 + 6) & (Vpos >= iceY4) &
(Vpos <= iceY4 + 40) & ~water & ~iceberg) & (1'b1 & ~iceHit[3] | flash & iceHit[3]);
assign icey5 = ((Hpos >= 16'd261) & (Hpos <= 16'd261 + 6) & (Vpos >= iceY5) &
(Vpos <= iceY5 + 40) & ~water & ~iceberg) & (1'b1 & ~iceHit[4] | flash & iceHit[4]);
assign icey6 = ((Hpos >= 16'd319) & (Hpos <= 16'd319 + 6) & (Vpos >= iceY6) &
(Vpos <= iceY6 + 40) & ~water & ~iceberg) & (1'b1 & ~iceHit[5] | flash & iceHit[5]);
assign icey7 = ((Hpos >= 16'd377) & (Hpos <= 16'd377 + 6) & (Vpos >= iceY7) &
(Vpos <= iceY7 + 40) & ~water & ~iceberg) & (1'b1 & ~iceHit[6] | flash & iceHit[6]);
assign icey8 = ((Hpos >= 16'd435) & (Hpos <= 16'd435 + 6) & (Vpos >= iceY8) &
(Vpos <= iceY8 + 40) & ~water & ~iceberg) & (1'b1 & ~iceHit[7] | flash & iceHit[7]);
assign icey9 = ((Hpos >= 16'd493) & (Hpos <= 16'd493 + 6) & (Vpos >= iceY9) &
(Vpos <= iceY9 + 40) & ~water & ~iceberg) & (1'b1 & ~iceHit[8] | flash & iceHit[8]);
assign icey10 = ((Hpos >= 16'd551) & (Hpos <= 16'd551 + 6) & (Vpos >= iceY10) &
(Vpos <= iceY10 + 40) & ~water & ~iceberg) & (1'b1 & ~iceHit[9] | flash & iceHit[9]);

assign red = {4{border}} | {4{slug}} | {4{icey1}} & fades1[3:0] &
{4{~iceColor[0]}} | {4{icey2}} & fades2[3:0] & {4{~iceColor[1]}} | {4{icey3}} &
fades3[3:0] & {4{~iceColor[2]}} | {4{icey4}} & fades4[3:0] & {4{~iceColor[3]}} |
{4{icey5}} & fades5[3:0] & {4{~iceColor[4]}} | {4{icey6}} & fades6[3:0] &
{4{~iceColor[5]}} | {4{icey7}} & fades7[3:0] & {4{~iceColor[6]}} | {4{icey8}} &
fades8[3:0] & {4{~iceColor[7]}} | {4{icey9}} & fades9[3:0] & {4{~iceColor[8]}} |
{4{icey10}} & fades10[3:0] & {4{~iceColor[9]}} ;

assign green = {iceberg, iceberg, 1'b0, 1'b0} | {4{slug}} | {4{icey1}} &
fades1[3:0] & {4{iceColor[0]}} | {4{icey2}} & fades2[3:0] & {4{iceColor[1]}} |
{4{icey3}} & fades3[3:0] & {4{iceColor[2]}} | {4{icey4}} & fades4[3:0] &
{4{iceColor[3]}} | {4{icey5}} & fades5[3:0] & {4{iceColor[4]}} | {4{icey6}} &
fades6[3:0] & {4{iceColor[5]}} | {4{icey7}} & fades7[3:0] & {4{iceColor[6]}} |

```

```

{4{icey8}} & fades8[3:0] & {4{iceColor[7]}} | {4{icey9}} & fades9[3:0] &
{4{iceColor[8]}} | {4{icey10}} & fades10[3:0] & {4{iceColor[9]}}};

    assign blue = {4{water}} | {4{icey1}} & fades1[3:0] | {4{iceberg}} | {4{icey2}}
& fades2[3:0] | {4{icey3}} & fades3[3:0] | {4{icey4}} & fades4[3:0] | {4{icey5}} &
fades5[3:0] | {4{icey6}} & fades6[3:0] | {4{icey7}} & fades7[3:0] | {4{icey8}} &
fades8[3:0] | {4{icey9}} & fades9[3:0] | {4{icey10}} & fades10[3:0];

    wire slugXUD, slugYUD, slugXCE, slugYCE, slugXLD, slugYLD;

    wire Csynced, Lsynced, Rsynced, Usynced; //sync inputs. might have to change CE
in order to account for different states of slug
    FDRE #(.INIT(1'b0)) SYNCbtnL (.C(clk), .CE(1'b1), .R(1'b0), .D(btnL), .Q(Lsynced)
    FDRE #(.INIT(1'b0)) SYNCbtnR (.C(clk), .CE(1'b1), .R(1'b0), .D(btnR), .Q(Rsynced)
    FDRE #(.INIT(1'b0)) SYNCbtnU (.C(clk), .CE(1'b1), .R(1'b0), .D(btnU), .Q(Usynced)
    FDRE #(.INIT(1'b0)) SYNCbtnC (.C(clk), .CE(1'b1), .R(1'b0), .D(btnC), .Q(Csynced)

    wire stop, timerStart, timerOver;
    FDRE #(.INIT(1'b1)) loadX (.C(clk), .CE(1'b1), .R(1'b0), .D(1'b0), .Q(slugXLD));
    counterUD16L slugX (.clk(clk), .UD(slugXUD), .CE(slugXCE), .LD(slugXLD),
.Din(16'd303), .Q(slugH));
    counterUD16L slugY (.clk(clk), .UD(1'b1), .CE(slugYCE), .LD(slugYLD),
.Din(slugYDin), .Q(slugV));
    assign slugXUD = Rsynced;
    assign slugXCE = ((Lsynced & ~leftborder & ~underwater) | (Rsynced &
~rightborder & ~underwater)) & frame & ~frozen;
    assign slugYCE = (~atrest & ~topborder | topborder & ~Usynced) & frame &
~bottomborder & ~frozen; //will have to add condition for frozen
    assign slugYDin = 16'd271 & {16{slugXLD}} | (slugV - 16'd2) & {16{~slugXLD}} ;
    assign slugYLD = slugXLD | Usynced & frame & ~underwater & ~topborder & ~frozen;

    assign leftborder = slugH <= 7 | (slugH == 16'd343 & slugV > 16'd271); //collide
with left border or right side of iceberg
    assign rightborder = slugH + 16'd16 >= 16'd631 | ((slugH + 16'd16) == 16'd279 &
slugV > 16'd271); //collidge with right border or left side of iceberg
    assign topborder = slugV <= 16'd7;
    assign bottomborder = (slugV + 16'd16) == 16'd471;
    assign underwater = (slugV + 16'd16) >= 16'd359;
    assign atrest = (slugV == 16'd271) & (slugH + 16 > 16'd278) & (slugH < 16'd344);
    wire [15:0] flashOUT;

    counterUD16L flashingstuff(.clk(clk), .UD(1'b1), .CE(frame), .LD(1'b0),
.Din(16'b0), .Q(flashOUT), .UTC(), .DTC());
    assign flash = flashOUT[4];

    slugState goslugs (.clk(clk), .T2(timerOver), .icicle(|iceHit), .frzn(frozen),
.timerStart(timerStart)); //need icicle wire

```

```

    wire [15:0] timerOUT;
    counterUD16L slugStateTimer (.clk(clk), .UD(1'b1), .CE(frame & (frozen)),
.LD(timerStart) , .Din(16'b0), .Q(timerOUT), .UTC(), .DTC());
    assign timerOver = (timerOUT == 16'd120);

    randnumgen random (.clk(clk), .rnd(randy));
    icycle ice1(.clk(clk), .slugY(slugV), .slugX(slugH), .column(16'd29),
.rand(randy[7:6]), .btnC(Csynced & ~underwater), .frame(frame), .flash(flash),
.icycleY(iceY1), .color(iceColor[0]), .collision(iceHit[0]), .fade(fades1[3:0]));
    icycle ice2(.clk(clk), .slugY(slugV), .slugX(slugH), .column(16'd87),
.rand(randy[5:4]), .btnC(Csynced & ~underwater), .frame(frame), .flash(flash),
.icycleY(iceY2), .color(iceColor[1]), .collision(iceHit[1]), .fade(fades2[3:0]));
    icycle ice3(.clk(clk), .slugY(slugV), .slugX(slugH), .column(16'd145),
.rand(randy[3:2]), .btnC(Csynced & ~underwater), .frame(frame), .flash(flash),
.icycleY(iceY3), .color(iceColor[2]), .collision(iceHit[2]), .fade(fades3[3:0]));
    icycle ice4(.clk(clk), .slugY(slugV), .slugX(slugH), .column(16'd203),
.rand(randy[1:0]), .btnC(Csynced & ~underwater), .frame(frame), .flash(flash),
.icycleY(iceY4), .color(iceColor[3]), .collision(iceHit[3]), .fade(fades4[3:0]));
    icycle ice5(.clk(clk), .slugY(slugV), .slugX(slugH), .column(16'd261),
.rand(randy[7:6] ^ randy[1:0]), .btnC(Csynced & ~underwater), .frame(frame),
.flash(flash), .icycleY(iceY5), .color(iceColor[4]), .collision(iceHit[4]),
.fade(fades5[3:0]));
    icycle ice6(.clk(clk), .slugY(slugV), .slugX(slugH), .column(16'd319),
.rand(randy[5:4] ^ randy [3:2]), .btnC(Csynced & ~underwater), .frame(frame),
.flash(flash), .icycleY(iceY6), .color(iceColor[5]), .collision(iceHit[5]),
.fade(fades6[3:0]));
    icycle ice7(.clk(clk), .slugY(slugV), .slugX(slugH), .column(16'd377),
.rand(randy[7:6] ^ randy[3:2]), .btnC(Csynced & ~underwater), .frame(frame),
.flash(flash), .icycleY(iceY7), .color(iceColor[6]), .collision(iceHit[6]),
.fade(fades7[3:0]));
    icycle ice8(.clk(clk), .slugY(slugV), .slugX(slugH), .column(16'd435),
.rand(randy[5:4] ^ randy[1:0]), .btnC(Csynced & ~underwater), .frame(frame),
.flash(flash), .icycleY(iceY8), .color(iceColor[7]), .collision(iceHit[7]),
.fade(fades8[3:0]));
    icycle ice9(.clk(clk), .slugY(slugV), .slugX(slugH), .column(16'd493),
.rand(randy[1:0] ^ randy[3:2]), .btnC(Csynced & ~underwater), .frame(frame),
.flash(flash), .icycleY(iceY9), .color(iceColor[8]), .collision(iceHit[8]),
.fade(fades9[3:0]));
    icycle ice10(.clk(clk), .slugY(slugV), .slugX(slugH), .column(16'd551),
.rand(randy[7:6] ^ randy[5:4]), .btnC(Csynced & ~underwater), .frame(frame),
.flash(flash), .icycleY(iceY10), .color(iceColor[9]), .collision(iceHit[9]),
.fade(fades10[3:0]));

endmodule

```