```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 05/31/2024 03:39:48 PM
// Design Name:
// Module Name: icicle
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module icicle(
    input clk,
    input [15:0] slugY,
    input [15:0] slugX,
    input [15:0] column,
    input [1:0] rand,
    input flash,
    input btnC,
    input frame,
    output [15:0] icicleY,
    output color,
    output collision,
    output [3:0] fade
    );

    wire fall, iceLD, TZ, T2, T4, underwater, slug, timerStart, timerStartZ, frzn,
fadeIN, spawn, sixteenframes, init;
    wire [15:0] LDvalue, randTime, timer24, sixteentimer;
    counterUD16L iciclePOS (.clk(clk) , .UD(1'b1), .CE(frame & fall), .LD(iceLD),
.Din(LDvalue), .Q(icicleY), .UTC(), .DTC());

    iceState state (.clk(clk), .activate(btnC), .TZ(TZ) , .T2(T2), .T4(T4),
.underwater(underwater), .slug(slug), .startTimer(timerStart), .frzn(frzn),
.fall(fall), .init(init), .fade(fadeIN), .spawn(spawn), .timerStartZ(timerStartZ));
```

```
    counterUD16L timerZ (.clk(clk), .UD(1'b0), .CE(frame & spawn), .LD(timerStartZ),
.Din(randTime), .Q(), .UTC(), .DTC(TZ));
    counterUD16L timer4 (.clk(clk), .UD(1'b1), .CE(frame & (fadeIN | frzn)),
.LD(timerStart) , .Din(16'b0), .Q(timer24), .UTC(), .DTC());

    assign iceLD = (spawn | fall) & frame;
    assign T2 = (timer24 == 16'd120);
    assign T4 = (timer24 == 16'd240);
    assign randTime = 16'd120 & {16{~rand[1]}} & {16{~rand[0]}} | 16'd150 &
{16{~rand[1]}} & {16{rand[0]}} | 16'd180 & {16{rand[1]}} & {16{~rand[0]}} | 16'd210
& {16{rand[1]}} & {16{rand[0]}};
    assign underwater = (icicleY > 16'd358);
    FDRE #(.INIT(1'b0)) choosecolor (.C(clk), .CE(spawn), .R(1'b0), .D(rand[0]),
.Q(color));



    assign slug =
    (icicleY + 40) >= slugY & (icicleY + 40) <= (slugY + 16) & column <= (slugX +
16) & (column + 6) >= slugX //bottom of icicle collides with top of slug
    |  (((slugY + 16 >= icicleY) & (slugY + 16 <= icicleY + 40)) | ((slugY >=
icicleY) & (slugY <= icicleY + 40))) & (slugX <= column + 6) & slugX >= column
//left side of icicle collides with right side slug
    |  (((slugY + 16 >= icicleY) & (slugY + 16 <= icicleY + 40)) | ((slugY >=
icicleY) & (slugY <= icicleY + 40))) & ((slugX + 16) >= column) & slugX + 16 <=
column + 6; //right side of icicle collides with left side of slug

    assign LDvalue = 16'd8 & {16{spawn}} | icicleY + 3 & {16{~spawn}};
    assign collision = slug & ~spawn & ~fadeIN & ~init;
    assign sixteenframes = sixteentimer[5];

    counterUD16L sixteenframe (.clk(clk), .UD(1'b1), .CE(fadeIN & frame),
.LD(sixteenframes & frame | spawn & TZ), .Din(16'b0), .Q(sixteentimer));
    counterUD16L fader (.clk(clk), .UD(1'b1), .CE(sixteenframes & fadeIN & frame),
.LD(frzn & T2 | fall & underwater), .Din(16'b0), .Q(fade));

endmodule
```