`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 2019/05/27 21:16:25

// Design Name:

// Module Name: final

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module final(

input clk,

input c1,

input c2,

input c3,

output reg hs, //行頻率

output reg vs, //場頻率

output reg [3:0]R, //RGB數值，其中red,green各三bit，blue兩bit

output reg [3:0]G,

output reg [3:0]B,

output reg [2:0]lifeled = 3'b111,

output reg [0:6]led,

output reg [7:0]e,

output reg ss = 0,

input left,

input right,

input reset

);

reg [30:0]q = 0;

reg [9:0]v = 400; //主角初始verticle中心位置

reg [9:0]h = 280; //主角初始horizontal中心位置

reg [9:0]v1=0; //第一顆隕石vertical中心位置

reg [9:0]h1=27; //第二顆隕石horizontal中心位置

reg [9:0]v2=0; //第二隻怪物初始verticle中心位置

reg [9:0]h2=532; //第二隻怪物初始horizontal中心位置

reg [9:0]v3=0; //第三隻怪物初始verticle中心位置

reg [9:0]h3=31; //第三隻怪物初始horizontal中心位置

reg [9:0]v4=0; //第四隻怪物初始verticle中心位置

reg [9:0]h4=530; //第四隻怪物初始horizontal中心位置

reg [9:0]v5=0; //第五隻怪物初始verticle中心位置

reg [9:0]h5=270; //第五隻怪物初始horizontal中心位置

reg [9:0]sv1=0; //第一顆隕石vertical中心位置

reg [9:0]sh1=27; //第二顆隕石horizontal中心位置

reg [9:0]sv2=0; //第二隻怪物初始verticle中心位置

reg [9:0]sh2=532; //第二隻怪物初始horizontal中心位置

reg [9:0]sv3=0; //第三隻怪物初始verticle中心位置

reg [9:0]sh3=31; //第三隻怪物初始horizontal中心位置

reg [9:0]sv4=0; //第四隻怪物初始verticle中心位置

reg [9:0]sh4=530; //第四隻怪物初始horizontal中心位置

reg [9:0]sv5=0; //第五隻怪物初始verticle中心位置

reg [9:0]sh5=270; //第五隻怪物初始horizontal中心位置

reg [9:0]sv6=0; //第五隻怪物初始verticle中心位置

reg [9:0]sh6=270; //第五隻怪物初始horizontal中心位置

reg [4:0]num[54:0];

reg [4:0] fall = 5'b00000;

reg [3:0]bi;

reg [2:0]pow = 0;

reg [3:0]one;

reg [3:0]two;

reg [3:0]three;

reg [3:0]four;

reg [3:0]five;

reg [3:0]six;

reg [3:0]seven;

reg [3:0]eight;

reg [3:0]aone = 0;

reg [3:0]atwo= 0;

reg [3:0]athree= 0;

reg [3:0]afour= 0;

reg [3:0]afive= 0;

reg [3:0]asix= 0;

reg [3:0]aseven= 0;

reg [3:0]aeight= 0;

reg [3:0]bone= 0;

reg [3:0]btwo= 0;

reg [3:0]bthree= 0;

reg [3:0]bfour= 0;

reg [3:0]bfive= 0;

reg [3:0]bsix= 0;

reg [3:0]bseven= 0;

reg [3:0]beight= 0;

reg [3:0]cone= 0;

reg [3:0]ctwo= 0;

reg [3:0]cthree= 0;

reg [3:0]cfour= 0;

reg [3:0]cfive= 0;

reg [3:0]csix= 0;

reg [3:0]cseven= 0;

reg [3:0]ceight= 0;

reg [3:0]done= 0;

reg [3:0]dtwo= 0;

reg [3:0]dthree= 0;

reg [3:0]dfour= 0;

reg [3:0]dfive= 0;

reg [3:0]dsix= 0;

reg [3:0]dseven= 0;

reg [3:0]deight= 0;

reg [3:0]eone= 0;

reg [3:0]etwo= 0;

reg [3:0]ethree= 0;

reg [3:0]efour= 0;

reg [3:0]efive= 0;

reg [3:0]esix= 0;

reg [3:0]eseven= 0;

reg [3:0]eeight= 0;

reg [25:0]bill[4:0];

reg over = 0;

reg [2:0]k = 0;

reg [2:0]o = 0;

reg [2:0]sta;

reg [7:0]i = 0;

integer j[7:0];

initial begin

for (i = 0;i < 8; i = i + 1)

j[i] <= i;

end

initial begin

one = 10;

two = 10;three = 10;

four = 10;

five = 10;

six = 10;

seven = 10;

eight = 10;

end

initial begin

bill[4] = 0;

bill[3] = 0;

bill[2] = 0;

bill[1] = 0;

bill[0] = 0;

end

//螢幕顯示code

parameter ActiveLines = 640; //行的有效顯示區

parameter ActiveFrames = 480; //場的有效顯示區

parameter LinePixels = 800; //行的真正寬度(144+640+16=800)

parameter FramePixels = 521; //場的真正長度(35+480+10=525，這邊只用到521)

parameter HorizontalPulse = 96; //水平脈衝波長

parameter HorizontalFront = 16; //行前方像素寬度

parameter VerticalPulse = 2; //垂直脈衝波長

parameter VericalFront = 10; //前方向像素長度

reg [9:0] Hcnt; //水平counter

reg [9:0] Vcnt; //垂直counter

reg clk\_50M = 0; //50MHz分頻頻率

reg clk\_25M = 0; //25 MHz分頻頻率

always@(posedge(clk))

begin

clk\_50M <= ~clk\_50M;

q<=q+1;

end

always@(posedge(clk\_50M))

begin

clk\_25M <= ~clk\_25M;

end

always@(posedge(clk\_25M))

begin

if( Hcnt == LinePixels -1 ) //行掃描到最右邊，Hcnt歸零，Vcnt+1

begin

Hcnt <= 0;

if( Vcnt == FramePixels -1 ) //若場也掃描到最右角，Vcnt歸零

Vcnt <=0;

else

Vcnt <= Vcnt + 1;

end

else

Hcnt <= Hcnt + 1; //其餘時 Hcnt+1

//下方:控制行頻率hs

if( Hcnt == ActiveLines - 1 + HorizontalFront)

hs <= 1'b0;

else if( Hcnt == ActiveLines - 1 + HorizontalFront +HorizontalPulse )

hs <= 1'b1;

//下方:控制場頻率 vs

if( Vcnt == ActiveFrames - 1 + VericalFront )

vs <= 1'b0;

else if( Vcnt == ActiveFrames - 1 + VericalFront + VerticalPulse )

vs <= 1'b1;

end

always@( posedge q[12])

begin

e = ~(2\*\*pow);

pow <= pow + 1;

if(pow>7)

begin

pow <= 0;

end

end

//background

always@(\*) begin

num[0] <= 1;

num[1] <= 11;

num[2] <= aone;

num[3] <= atwo;

num[4] <= athree;

num[5] <= afour;

num[6] <= afive;

num[7] <= asix;

num[8] <= aseven;

num[9] <= aeight;

num[10] <= 10;

num[11] <= 2;

num[12] <= 11;

num[13] <= bone;

num[14] <= btwo;

num[15] <= bthree;

num[16] <= bfour;

num[17] <= bfive;

num[18] <= bsix;

num[19] <= bseven;

num[20] <= beight;

num[21] <=10;

num[22] <= 3;

num[23] <= 11;

num[24] <= cone;

num[25] <= ctwo;

num[26] <= cthree;

num[27] <= cfour;

num[28] <= cfive;

num[29] <= csix;

num[30] <= cseven;

num[31] <= ceight;

num[32] <=10;

num[33] <= 4;

num[34] <= 11;

num[35] <= done;

num[36] <= dtwo;

num[37] <= dthree;

num[38] <= dfour;

num[39] <= dfive;

num[40] <= dsix;

num[41] <= dseven;

num[42] <= deight;

num[43] <=10;

num[44] <= 5;

num[45] <= 11;

num[46] <= eone;

num[47] <= etwo;

num[48] <= ethree;

num[49] <= efour;

num[50] <= efive;

num[51] <= esix;

num[52] <= eseven;

num[53] <= eeight;

num[54] <= 10;

end

reg cnt = 1;

always@( Vcnt||Hcnt) begin

if (over == 0 && cnt == 1) begin

if(Vcnt<= 22|| Hcnt<=30 || Hcnt>=600 || Vcnt>=465 ) begin //邊界

R = 0;

G = 0;

B = 0;//edge

end

else if ((Hcnt>=h && Hcnt<=h+2 && Vcnt>=v && Vcnt<=v+27)||(Hcnt>=h+2 && Hcnt<=h+4 && Vcnt>=v+8 && Vcnt<=v+27)||

(Hcnt>=h-2 && Hcnt<=h && Vcnt>=v && Vcnt<=v+27)|| (Hcnt>=h-4 && Hcnt<=h-2 && Vcnt>=v+8 && Vcnt<=v+27)||

(Hcnt>=h+5 && Hcnt<=h+7 && Vcnt>=v+17 && Vcnt<=v+27)|| (Hcnt>=h+7 && Hcnt<=h+9 && Vcnt>=v+19 && Vcnt<=v+27)||

(Hcnt>=h+9 && Hcnt<=h+11 && Vcnt<=v+27 && Vcnt>=v+21)||

(Hcnt>=h+11 && Hcnt<=h+13 && Vcnt<=v+27 && Vcnt>=v+23)||(Hcnt>=h-7 && Hcnt<=h-5 && Vcnt<=v+27 && Vcnt>=v+17)||

(Hcnt>=h-9 && Hcnt<=h-7 && Vcnt<=v+27 && Vcnt>=v+19) || (Hcnt>=h-11 && Hcnt<=h-9 && Vcnt<=v+27 && Vcnt>=v+21)||

(Hcnt>=h-13 && Hcnt<=h-11 && Vcnt>=v+23 && Vcnt<=v+27)||(Hcnt>=h+7 && Hcnt<=h+8 && Vcnt<=v+27 && Vcnt>=v+19) ||

(Hcnt>=h && Hcnt<=h+12 && Vcnt<=v+28 && Vcnt>=v+27) ||(Hcnt>=h && Hcnt<=h+10 && Vcnt<=v+29 && Vcnt>=v+28) ||(Hcnt>=h && Hcnt<=h+8 && Vcnt<=v+30 && Vcnt>=v+29) ||

(Hcnt>=h && Hcnt<=h+6 && Vcnt<=v+31 && Vcnt>=v+30) ||(Hcnt>=h && Hcnt<=h+4 && Vcnt<=v+33 && Vcnt>=v+32) ||(Hcnt>=h && Hcnt<=h+2 && Vcnt<=v+34 && Vcnt>=v+33) ||

(Hcnt>=h-12 && Hcnt<=h && Vcnt<=v+28 && Vcnt>=v+27) ||(Hcnt>=h-10 && Hcnt<=h && Vcnt<=v+29 && Vcnt>=v+28) ||(Hcnt>=h-8 && Hcnt<=h && Vcnt<=v+30 && Vcnt>=v+29) ||

(Hcnt>=h-6 && Hcnt<=h && Vcnt<=v+31 && Vcnt>=v+30) ||(Hcnt>=h-4 && Hcnt<=h && Vcnt<=v+32 && Vcnt>=v+31) ||(Hcnt>=h-2 && Hcnt<=h && Vcnt<=v+33 && Vcnt>=v+32)

)begin

if (c1 == 1) begin

R = 15;

G = 0;

B = 0;

end

else if (c2 == 1) begin

R = 0;

G = 10;

B = 0;

end

else begin

R = 6;

G = 6;

B = 6;

end

end

else if((Hcnt>=h1-5 && Hcnt<=h1+5 && Vcnt>=v1-13 && Vcnt<=v1-11) || (Hcnt>=h1-9 && Hcnt<=h1+9 && Vcnt>=v1-11 && Vcnt<=v1-9) || (Hcnt>=h1-11 && Hcnt<=h1+11 && Vcnt>v1-9 && Vcnt<=v1-7) ||

(Hcnt>=h1-13 && Hcnt<=h1+13 && Vcnt>=v1-7 && Vcnt<=v1-5) || (Hcnt>=h1-13 && Hcnt<=h1+13 && Vcnt>=v1-5 && Vcnt<=v1-3) || (Hcnt>=h1-13 && Hcnt<=h1+13 && Vcnt>=v1-3 && Vcnt<=v1-1) ||

(Hcnt>=h1-15 && Hcnt<=h1+15 && Vcnt>=v1-1 && Vcnt<=v1+1) || (Hcnt>=h1-15 && Hcnt<=h1+15 && Vcnt>=v1+1 && Vcnt<=v1+3) || (Hcnt>=h1-15 && Hcnt<=h1+15 && Vcnt>=v1+3 && Vcnt<=v1+5) ||

(Hcnt>=h1-15 && Hcnt<=h1+15 && Vcnt>=v1+5 && Vcnt<=v1+7) || (Hcnt>=h1-5 && Hcnt<=h1+5 && Vcnt>=v1+17 && Vcnt<=v1+19) || (Hcnt>=h1-9 && Hcnt<=h1+9 && Vcnt>=v1+15 && Vcnt<=v1+17) ||

(Hcnt>=h1-11 && Hcnt<=h1+11 && Vcnt>v1+13 && Vcnt<=v1+15) || (Hcnt>=h1-13 && Hcnt<=h1+13 && Vcnt>=v1+11 && Vcnt<=v1+13) || (Hcnt>=h1-13 && Hcnt<=h1+13 && Vcnt>=v1+9 && Vcnt<=v1+11) ||

(Hcnt>=h1-13 && Hcnt<=h1+13 && Vcnt>=v1+7 && Vcnt<=v1+9) )begin //第一隻怪物紫色

R=0;

G=0;

B=15;

end

else if((Hcnt>=h2-5 && Hcnt<=h2+5 && Vcnt>=v2-13 && Vcnt<=v2-11) || (Hcnt>=h2-9 && Hcnt<=h2+9 && Vcnt>=v2-11 && Vcnt<=v2-9) || (Hcnt>=h2-11 && Hcnt<=h2+11 && Vcnt>v2-9 && Vcnt<=v2-7) ||

(Hcnt>=h2-13 && Hcnt<=h2+13 && Vcnt>=v2-7 && Vcnt<=v2-5) || (Hcnt>=h2-13 && Hcnt<=h2+13 && Vcnt>=v2-5 && Vcnt<=v2-3) || (Hcnt>=h2-13 && Hcnt<=h2+13 && Vcnt>=v2-3 && Vcnt<=v2-1) ||

(Hcnt>=h2-15 && Hcnt<=h2+15 && Vcnt>=v2-1 && Vcnt<=v2+1) || (Hcnt>=h2-15 && Hcnt<=h2+15 && Vcnt>=v2+1 && Vcnt<=v2+3) || (Hcnt>=h2-15 && Hcnt<=h2+15 && Vcnt>=v2+3 && Vcnt<=v2+5) ||

(Hcnt>=h2-15 && Hcnt<=h2+15 && Vcnt>=v2+5 && Vcnt<=v2+7) || (Hcnt>=h2-5 && Hcnt<=h2+5 && Vcnt>=v2+17 && Vcnt<=v2+19) || (Hcnt>=h2-9 && Hcnt<=h2+9 && Vcnt>=v2+15 && Vcnt<=v2+17) ||

(Hcnt>=h2-11 && Hcnt<=h2+11 && Vcnt>v2+13 && Vcnt<=v2+15) || (Hcnt>=h2-13 && Hcnt<=h2+13 && Vcnt>=v2+11 && Vcnt<=v2+13) || (Hcnt>=h2-13 && Hcnt<=h2+13 && Vcnt>=v2+9 && Vcnt<=v2+11) ||

(Hcnt>=h2-13 && Hcnt<=h2+13 && Vcnt>=v2+7 && Vcnt<=v2+9) )begin //第一隻怪物紫色

R=7;

G=3;

B=15;

end

else if((Hcnt>=h3-5 && Hcnt<=h3+5 && Vcnt>=v3-13 && Vcnt<=v3-11) || (Hcnt>=h3-9 && Hcnt<=h3+9 && Vcnt>=v3-11 && Vcnt<=v3-9) || (Hcnt>=h3-11 && Hcnt<=h3+11 && Vcnt>v3-9 && Vcnt<=v3-7) ||

(Hcnt>=h3-13 && Hcnt<=h3+13 && Vcnt>=v3-7 && Vcnt<=v3-5) || (Hcnt>=h3-13 && Hcnt<=h3+13 && Vcnt>=v3-5 && Vcnt<=v3-3) || (Hcnt>=h3-13 && Hcnt<=h3+13 && Vcnt>=v3-3 && Vcnt<=v3-1) ||

(Hcnt>=h3-15 && Hcnt<=h3+15 && Vcnt>=v3-1 && Vcnt<=v3+1) || (Hcnt>=h3-15 && Hcnt<=h3+15 && Vcnt>=v3+1 && Vcnt<=v3+3) || (Hcnt>=h3-15 && Hcnt<=h3+15 && Vcnt>=v3+3 && Vcnt<=v3+5) ||

(Hcnt>=h3-15 && Hcnt<=h3+15 && Vcnt>=v3+5 && Vcnt<=v3+7) || (Hcnt>=h3-5 && Hcnt<=h3+5 && Vcnt>=v3+17 && Vcnt<=v3+19) || (Hcnt>=h3-9 && Hcnt<=h3+9 && Vcnt>=v3+15 && Vcnt<=v3+17) ||

(Hcnt>=h3-11 && Hcnt<=h3+11 && Vcnt>v3+13 && Vcnt<=v3+15) || (Hcnt>=h3-13 && Hcnt<=h3+13 && Vcnt>=v3+11 && Vcnt<=v3+13) || (Hcnt>=h3-13 && Hcnt<=h3+13 && Vcnt>=v3+9 && Vcnt<=v3+11) ||

(Hcnt>=h3-13 && Hcnt<=h3+13 && Vcnt>=v3+7 && Vcnt<=v3+9))begin //第二隻怪物橘色

R=15;

G=7;

B=0;

end

else if((Hcnt>=h4-5 && Hcnt<=h4+5 && Vcnt>=v4-13 && Vcnt<=v4-11) || (Hcnt>=h4-9 && Hcnt<=h4+9 && Vcnt>=v4-11 && Vcnt<=v4-9) || (Hcnt>=h4-11 && Hcnt<=h4+11 && Vcnt>v4-9 && Vcnt<=v4-7) ||

(Hcnt>=h4-13 && Hcnt<=h4+13 && Vcnt>=v4-7 && Vcnt<=v4-5) || (Hcnt>=h4-13 && Hcnt<=h4+13 && Vcnt>=v4-5 && Vcnt<=v4-3) || (Hcnt>=h4-13 && Hcnt<=h4+13 && Vcnt>=v4-3 && Vcnt<=v4-1) ||

(Hcnt>=h4-15 && Hcnt<=h4+15 && Vcnt>=v4-1 && Vcnt<=v4+1) || (Hcnt>=h4-15 && Hcnt<=h4+15 && Vcnt>=v4+1 && Vcnt<=v4+3) || (Hcnt>=h4-15 && Hcnt<=h4+15 && Vcnt>=v4+3 && Vcnt<=v4+5) ||

(Hcnt>=h4-15 && Hcnt<=h4+15 && Vcnt>=v4+5 && Vcnt<=v4+7) || (Hcnt>=h4-5 && Hcnt<=h4+5 && Vcnt>=v4+17 && Vcnt<=v4+19) || (Hcnt>=h4-9 && Hcnt<=h4+9 && Vcnt>=v4+15 && Vcnt<=v4+17) ||

(Hcnt>=h4-11 && Hcnt<=h4+11 && Vcnt>v4+13 && Vcnt<=v4+15) || (Hcnt>=h4-13 && Hcnt<=h4+13 && Vcnt>=v4+11 && Vcnt<=v4+13) || (Hcnt>=h4-13 && Hcnt<=h4+13 && Vcnt>=v4+9 && Vcnt<=v4+11) ||

(Hcnt>=h4-13 && Hcnt<=h4+13 && Vcnt>=v4+7 && Vcnt<=v4+9) )begin //第三隻怪物玫瑰色

R=15;

G=8;

B=8;

end

else if((Hcnt>=h5-5 && Hcnt<=h5+5 && Vcnt>=v5-13 && Vcnt<=v5-11) || (Hcnt>=h5-9 && Hcnt<=h5+9 && Vcnt>=v5-11 && Vcnt<=v5-9) || (Hcnt>=h5-11 && Hcnt<=h5+11 && Vcnt>v5-9 && Vcnt<=v5-7) ||

(Hcnt>=h5-13 && Hcnt<=h5+13 && Vcnt>=v5-7 && Vcnt<=v5-5) || (Hcnt>=h5-13 && Hcnt<=h5+13 && Vcnt>=v5-5 && Vcnt<=v5-3) || (Hcnt>=h5-13 && Hcnt<=h5+13 && Vcnt>=v5-3 && Vcnt<=v5-1) ||

(Hcnt>=h5-15 && Hcnt<=h5+15 && Vcnt>=v5-1 && Vcnt<=v5+1) || (Hcnt>=h5-15 && Hcnt<=h5+15 && Vcnt>=v5+1 && Vcnt<=v5+3) || (Hcnt>=h5-15 && Hcnt<=h5+15 && Vcnt>=v5+3 && Vcnt<=v5+5) ||

(Hcnt>=h5-15 && Hcnt<=h5+15 && Vcnt>=v5+5 && Vcnt<=v5+7) || (Hcnt>=h5-5 && Hcnt<=h5+5 && Vcnt>=v5+17 && Vcnt<=v5+19) || (Hcnt>=h5-9 && Hcnt<=h5+9 && Vcnt>=v5+15 && Vcnt<=v5+17) ||

(Hcnt>=h5-11 && Hcnt<=h5+11 && Vcnt>v5+13 && Vcnt<=v5+15) || (Hcnt>=h5-13 && Hcnt<=h5+13 && Vcnt>=v5+11 && Vcnt<=v5+13) || (Hcnt>=h5-13 && Hcnt<=h5+13 && Vcnt>=v5+9 && Vcnt<=v5+11) ||

(Hcnt>=h5-13 && Hcnt<=h5+13 && Vcnt>=v5+7 && Vcnt<=v5+9) )begin //第四隻怪物藍色

R=0;

G=7;

B=15;

end

else if (Hcnt >=sh1 && Hcnt <= sh1 + 1 && Vcnt >=sv1 && Vcnt <= sv1+1) begin

R = 15;

G = 15;

B = 0;

end

else if (Hcnt >=sh2 && Hcnt <= sh2 + 1 && Vcnt >=sv2 && Vcnt <= sv2+1) begin

R = 15;

G = 15;

B = 0;

end

else if (Hcnt >=sh3 && Hcnt <= sh3 + 1 && Vcnt >=sv3 && Vcnt <= sv3+1) begin

R = 15;

G = 15;

B = 0;

end

else if (Hcnt >=sh4 && Hcnt <= sh4 + 1 && Vcnt >=sv4 && Vcnt <= sv4+1) begin

R = 15;

G = 15;

B = 0;

end

else if (Hcnt >=sh5 && Hcnt <= sh5 + 1 && Vcnt >=sv5 && Vcnt <= sv5+1) begin

R = 15;

G = 15;

B = 0;

end

else if (Hcnt >=sh6 && Hcnt <= sh6 + 1 && Vcnt >=sv6 && Vcnt <= sv6+1) begin

R = 15;

G = 15;

B = 0;

end

else if (Vcnt>=22 && Vcnt<=465 && Hcnt>=30 && Hcnt<=600) begin //background

R = 1;

G = 1;

B = 3;

end

end

else if (over == 1) begin

if(Vcnt<= 22|| Hcnt<=30 || Hcnt>=600 || Vcnt>=465 ) begin //邊界

R = 0;

G = 0;

B = 0;//edge

end

else if (Vcnt>=22 && Vcnt<=465 && Hcnt>=30 && Hcnt<=600) begin //background

R = 1;

G = 1;

B = 3;

end

end

end

//命

reg stop =0;

reg[8:0]ttt = 0;

reg r;

always@(posedge q[15]) begin

if ((h-h1<=15||h1-h<=15)&&(v-v1<=15||v1-v<=15)||(h-h2<=15||h2-h<=15)&&(v-v2<=15||v2-v<=15)||(h-h3<=15||h3-h<=15)&&(v-v3<=15||v3-v<=15)||

(h-h4<=15||h4-h<=15)&&(v-v4<=15||v4-v<=15)||(h-h5<=15||h5-h<=15)&&(v-v5<=15||v5-v<=15))

begin

if (cnt == 1) begin

if (lifeled[2] == 1) begin

r = reset;

lifeled[2] = 0;

stop = 1;

end

else if (lifeled[1] == 1 && stop == 0) begin

lifeled[1] = 0;

stop = 1;

end

else if (lifeled[0] == 1 && stop == 0) begin

lifeled[0] = 0;

over =1;

cnt = 0;

stop = 1;

end

end

end

if (stop == 1) begin

if (ttt > 90) begin

stop = 0;

ttt = 0;

end

else begin ttt = ttt + 1; end

end

if (cnt == 0) begin

if (reset != r)begin cnt = 1;

over = 0;

ss = 0;

lifeled = 3'b111;

end

end

end

//分數

reg [25:0]score = 0;

reg ab = 0;

reg[25:0]zz = 0;

always@(posedge q[20]) begin

if (over == 0) begin

ab = 1;

score = score + 1;

if (score <10)

eight = score;

else if (score<100) begin

seven = score/10;

eight = score - 10\*seven;

end

else if (score<1000) begin

six = score/100;

seven = (score - 100\*six)/10;

eight = (score - 100\*six - 10\*seven);

end

else if (score<10000) begin

five = score/1000;

six = (score - 1000\*five)/100;

seven = (score - 1000\*five - 100\*six)/10;

eight = (score - 1000\*five - 100\*six - 10\*seven);

end

else if (score<100000) begin

four = score/10000;

five = (score - 10000\*four)/1000;

six = (score - 10000\*four - 1000\*five)/100;

seven = (score - 10000\*four - 1000\*five - 100\*six)/10;

eight = (score - 10000\*four - 1000\*five - 100\*six - 10\*seven);

end

else if (score<1000000) begin

three= score/100000;

four = (score - 100000\*three)/10000;

five = (score - 100000\*three - 10000\*four)/1000;

six = (score - 100000\*three- 10000\*four - 1000\*five)/100;

seven = (score - 100000\*three - 10000\* four - 1000\* five - 100\*six)/10;

eight = (score - 100000\*three - 10000\* four - 1000\* five - 100\*six - 10\*seven);

end

else if (score<10000000) begin

two= score/1000000;

three = (score - 1000000\*two)/100000;

four = (score - 1000000\*two - 100000\*three)/10000;

five= (score - 1000000\*two- 100000\*three - 10000\*four)/1000;

six = (score - 1000000\*two - 100000\* three - 10000\* four - 1000\*five)/100;

seven = (score - 1000000\*two - 100000\* three - 10000\* four - 1000\*five - 100\*six)/10;

eight = (score - 1000000\*two - 100000\* three - 10000\* four - 1000\*five - 100\*six - 10\*seven);

end

else if (score<100000000) begin

one= score/10000000;

two = (score - 10000000\*one)/1000000;

three = (score - 10000000\*one - 1000000\*two)/100000;

four = (score - 10000000\*one- 1000000\*two - 100000\*three)/10000;

five= (score - 10000000\*one - 1000000\* two - 100000\*three - 10000\*four)/1000;

six = (score - 10000000\*one - 1000000\* two - 100000\*three- 10000\*four - 1000\*five)/100;

seven = (score - 10000000\*one - 1000000\* two - 100000\* three - 10000\*four - 1000\*five- 100\*six)/10;

eight = score - 100000000\* one - 1000000\* two - 100000\* three - 10000\*four - 1000\*five- 100\*six - 10\*seven;

end

end

if(over==1)

begin

if (ab == 1) begin

ab = 0;// 決定new名次

zz = score;

score = 0;

if (bill[0]<zz) begin

sta = 0; end

else if (bill[1]<zz) begin

sta = 1; end

else if (bill[2]<zz) begin

sta = 2; end

else if (bill[3]<zz) begin

sta = 3; end

else if (bill[4]<zz) begin

sta = 4; end

case (sta)

0: begin //1st

bill[4] = bill[3];

bill[3] = bill[2];

bill[2] = bill[1];

bill[1] = bill[0];

bill[0] = zz;

eone= done;

etwo=dtwo;

ethree=dthree;

efour= dfour;

efive=dfive;

esix=dsix;

eseven=dseven;

eeight=deight;

done= cone;

dtwo=ctwo;

dthree=cthree;

dfour= cfour;

dfive=cfive;

dsix=csix;

dseven=cseven;

deight=ceight;

cone= bone;

ctwo=btwo;

cthree=bthree;

cfour= bfour;

cfive=bfive;

csix=bsix;

cseven=bseven;

ceight=beight;

bone= aone;

btwo=atwo;

bthree=athree;

bfour= afour;

bfive=afive;

bsix=asix;

bseven=aseven;

beight=aeight;

aone= one;

atwo=two;

athree=three;

afour= four;

afive=five;

asix=six;

aseven=seven;

aeight=eight;

ab = 0;

end

1: begin //2nd

bill[4] = bill[3];

bill[3] = bill[2];

bill[2] = bill[1];

bill[1] = zz;

eone= done;

etwo=dtwo;

ethree=dthree;

efour= dfour;

efive=dfive;

esix=dsix;

eseven=dseven;

eeight=deight;

done= cone;

dtwo=ctwo;

dthree=cthree;

dfour= cfour;

dfive=cfive;

dsix=csix;

dseven=cseven;

deight=ceight;

cone= bone;

ctwo=btwo;

cthree=bthree;

cfour= bfour;

cfive=bfive;

csix=bsix;

cseven=bseven;

ceight=beight;

bone= one;

btwo=two;

bthree=three;

bfour= four;

bfive=five;

bsix=six;

bseven=seven;

beight=eight;

ab = 0;

end

2: begin //3rd

bill[4] = bill[3];

bill[3] = bill[2];

bill[2] = zz;

eone= done;

etwo=dtwo;

ethree=dthree;

efour= dfour;

efive=dfive;

esix=dsix;

eseven=dseven;

eeight=deight;

done= cone;

dtwo=ctwo;

dthree=cthree;

dfour= cfour;

dfive=cfive;

dsix=csix;

dseven=cseven;

deight=ceight;

cone= one;

ctwo=two;

cthree=three;

cfour= four;

cfive=five;

csix=six;

cseven=seven;

ceight=eight;

ab = 0;

end

3: begin //4th

bill[4] = bill[3];

bill[3] = zz;

eone= done;

etwo=dtwo;

ethree=dthree;

efour= dfour;

efive=dfive;

esix=dsix;

eseven=dseven;

eeight=deight;

done= one;

dtwo=two;

dthree=three;

dfour= four;

dfive=five;

dsix=six;

dseven=seven;

deight=eight;

cone= one;

ab = 0;

end

4: begin//5th

bill[4] = zz;

eone= one;

etwo=two;

ethree=three;

efour= four;

efive=five;

esix=six;

eseven=seven;

eeight=eight;

ab = 0;

end

endcase

one = 10;

two = 10;

three = 10;

four = 10;

five = 10;

six = 10;

seven = 10;

eight = 10;

end

end

end

//主角移動

always@(posedge q[17]) begin

if ((left == 1) && (h >30)) //'向左

h = h - 1;

else if (right == 1 && h<600) //向右

h = h +1;

end

//隕石移動

reg [9:0] fast = 1;

reg [9:0]move;

reg [9:0]smove;

reg [9:0]ran;

reg [4:0]a = 0;

reg [4:0]en = 0;

reg [7:0]tt = 0;

reg [4:0]t = 0;

reg [4:0]p;

reg [5:0] star = 0;

always@(posedge q[14]) begin

ran = ran +1;

if (ran > 640)

ran = 0;

end

always@(posedge q[13]) begin

a = 0;

// if (cnt == 1) begin

// if (score > 5000) fast = fast \* 2;

// else if (score > 20000) fast = fast \* 2;

// else if (score > 50000) fast = fast \* 2;

// move = 2400/fast;

// smove = move/5;

// end

if (tt >150)

tt = 0;

else if (tt == 150) begin

case (ran%5)

0: begin if (en[4] == 0) begin h1 = ran; fall[4] = 1; en[4] = 1;end end

1: begin if (en[3] == 0) begin h2 = ran; fall[3] = 1; en[3] = 1;end end

2: begin if (en[2] == 0) begin h3 = ran; fall[2] = 1; en[2] = 1;end end

3: begin if (en[1] == 0) begin h4 = ran; fall[1] = 1; en[1] = 1;end end

4: begin if (en[0] == 0) begin h5 = ran; fall[0] = 1; en[0] = 1;end end

endcase

tt = tt + 1;

end

else tt = tt + 1;

if (t > 30)

t = 0;

else if (t == 30) begin

case ( ran %6)

0: begin if (star[0] == 0)begin star[0] = 1; sh1 = ran;end end

1: begin if (star[1] == 0)begin star[1] = 1; sh2 = ran;end end

2: begin if (star[2] == 0)begin star[2] = 1; sh3 = ran;end end

3: begin if (star[3] == 0)begin star[3] = 1; sh4 = ran;end end

4: begin if (star[4] == 0)begin star[4] = 1; sh5 = ran;end end

5: begin if (star[5] == 0)begin star[5] = 1; sh6 = ran;end end

endcase

t = t + 1;

end

else t = t + 1;

if (star[0] == 1) begin

sv1 = sv1 + 1;

if (sv1>465) begin star[0] = 0; sv1 = 0;end

end

if (star[1] == 1) begin

sv2 = sv2 + 1;

if (sv2>465) begin star[1] = 0; sv2 = 0;end

end

if (star[2] == 1) begin

sv3 = sv3 + 1;

if (sv3>465) begin star[2] = 0; sv3= 0;end

end

if (star[3] == 1) begin

sv4 = sv4 + 1;

if (sv4>465) begin star[3] = 0; sv4 = 0;end

end

if (star[4] == 1) begin

sv5 = sv5 + 1;

if (sv5>465) begin star[4] = 0; sv5 = 0;end

end

if (star[5] == 1) begin

sv6 = sv6 + 1;

if (sv6>465) begin star[5] = 0; sv6 = 0;end

end

if (fall[4] == 1) begin

v1 = v1 + 1;

if (v1%4 == 0) begin

if (h1<600 && a[4] == 0)

begin

h1 = h1 + 1;

end

else begin

a[4] = 1;

h1 = h1 - 5;

end

end

if (v1 > 465) begin en[4] = 0; fall[4] = 0;v1 = 0; end

end

if (fall[3] == 1) begin

v2 = v2 + 1;

if (v2%1== 0) begin

if (h2<600 && a[3] == 0) begin

h2 = h2 + 1;

end

else begin

a[3] = 1;

h2 = h2 - 5;

ss = 1;

end

end

end

if (v2 > 465) begin en[3] = 0; fall[3] = 0;v2 = 0; end

if(fall[2] == 1) begin

v3 = v3 + 1;

if (v3%3== 0) begin

if (h3<600 && a[2] == 0) begin

h3 = h3 + 1;

end

else begin

a[2] = 1;

h3 = h3 - 5;

ss = 1;

end

end

end

if (v3 > 465) begin en[2] = 0; fall[2] = 0;v3 = 0; end

if(fall[1] == 1) begin

v4 = v4 + 1;

if (v4%6== 0) begin

if (h4<600 && a[1] == 0) begin

h4 = h4 - 1;

end

else begin

a[1] = 1;

h4 = h4 + 5;

ss = 1;

end

end

if (v4 > 465) begin en[1] = 0; fall[1] = 0;v4 = 0; end

end

if(fall[0] == 1) begin

v5 = v5 + 1;

if (v5%2== 0) begin

if (h5<600 && a[0] == 0) begin

h5 = h5 - 1;

end

else begin

a[0] = 1;

h5 = h5 + 5;

ss = 1;

end

end

if (v5 > 465) begin en[0] = 0; fall[0] = 0;v5 = 0; end

end

if (cnt == 0) begin

fast = 1;

end

end

always @(posedge q[23]) begin //跑馬燈

for (i = 0; i < 8; i = i + 1)

begin

j[i] <= j[i] + 1;

if (j[i] >54)

j[i] <= 0;

end

end

always @(\*)

begin

if (over == 0) begin

if(e == 8'b11111110)

bi <= eight;

if (e == 8'b11111101)

bi <= seven;

if (e == 8'b11111011)

bi <= six;

if (e == 8'b11110111)

bi <= five;

if (e == 8'b11101111)

bi <= four;

if (e == 8'b11011111)

bi <= three;

if (e == 8'b10111111)

bi <= two;

if (e == 8'b01111111)

bi <= one;

end

else begin

if (e == 8'b11111110)

bi <= num[j[7]];

if (e== 8'b11111101)

bi <= num[j[6]];

if (e == 8'b11111011)

bi <= num[j[5]];

if (e == 8'b11110111)

bi <= num[j[4]];

if (e == 8'b11101111)

bi <= num[j[3]];

if (e == 8'b11011111)

bi <= num[j[2]];

if (e == 8'b10111111)

bi <= num[j[1]];

if (e == 8'b01111111)

bi <= num[j[0]];

end

end

always @(\*)

begin

if (bi == 0)

led=7'b0000001;

if(bi == 1)

led=7'b1001111;

if(bi == 2)

led=7'b0010010;

if(bi== 3)

led=7'b0000110;

if(bi== 4)

led=7'b1001100;

if(bi== 5)

led=7'b0100100;

if(bi== 6)

led=7'b1100000;

if(bi== 7)

led=7'b0001111;

if(bi== 8)

led=7'b0000000;

if(bi== 9)

led=7'b0001100;

if(bi==10)

led = 7'b1111111;

if(bi==11)

led = 7'b1111110;

end

endmodule