/\* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*

\* CREOQODE 2048 - EYES \*

\* \*

\* This program is written by the Creoqode Team on 02/01/2017. \*

\* Eyes is an animation for Creoqode 2048. \*

\* This program can be modified and reused for general purposes. \*

\* Libraries used for this video game belongs to Adafruit Industries. \*

\* \*

\* Please visit www.creoqode.com/qode-share to find out more! \*

\* \*

\* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*/

#include <Adafruit\_GFX.h> // Core graphics library

#include <RGBmatrixPanel.h> // Hardware-specific library

#define CLK 11

#define LAT 10

#define OE 9

#define A 12

#define B 13

#define C 14

#define D 15

RGBmatrixPanel creoqode(A, B, C, D, CLK, LAT, OE, false, 64);

////////////////////////////////////////////////////////////////////////////////////////////// Eye Illustrations

const char eyes\_1[] = { // Illustration 1

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,2,2,2,2,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,1,1,2,2,2,2,2,2,2,2,1,1,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,1,1,3,2,2,2,2,2,2,2,2,2,2,3,1,1,0,0,0,0,0,0,0,0,

0,0,0,0,1,1,3,3,2,2,2,2,0,0,0,0,2,2,2,2,3,3,1,1,0,0,0,0,0,0,

0,0,0,1,3,3,3,3,2,2,2,0,0,0,0,0,0,2,2,2,3,3,3,3,1,0,0,0,0,0,

0,0,1,3,3,3,3,3,2,2,2,0,0,0,0,0,0,2,2,2,3,3,3,3,3,1,0,0,0,0,

0,0,1,3,3,3,3,3,2,2,2,0,0,0,0,0,0,2,2,2,3,3,3,3,3,1,0,0,0,0,

0,0,0,1,3,3,3,3,2,2,2,2,0,0,0,0,2,2,2,2,3,3,3,3,1,0,0,0,0,0,

0,0,0,0,1,3,3,3,3,2,2,2,2,2,2,2,2,2,2,3,3,3,3,1,0,0,0,0,0,0,

0,0,0,0,0,1,1,3,3,3,2,2,2,2,2,2,2,2,3,3,3,1,1,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,1,1,1,3,2,2,2,2,2,2,3,1,1,1,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

};

const char eyes\_2[] = { // Illustration 2

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,1,2,2,2,2,2,2,2,2,1,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,1,2,2,2,2,0,0,0,0,2,2,2,2,1,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,1,3,2,2,2,0,0,0,0,0,0,2,2,2,3,1,0,0,0,0,0,0,0,0,

0,0,0,1,1,3,3,3,2,2,2,0,0,0,0,0,0,2,2,2,3,3,3,1,1,0,0,0,0,0,

0,0,1,3,3,3,3,3,2,2,2,0,0,0,0,0,0,2,2,2,3,3,3,3,3,1,0,0,0,0,

0,0,0,0,1,3,3,3,2,2,2,2,0,0,0,0,2,2,2,2,3,3,3,1,0,0,0,0,0,0,

0,0,0,0,0,1,1,1,1,2,2,2,2,2,2,2,2,2,2,1,1,1,1,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

};

const char eyes\_3[] = { // Illustration 3

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,1,1,1,2,2,0,0,0,0,0,0,2,2,1,1,1,0,0,0,0,0,0,0,0,

0,0,0,1,1,1,3,3,2,2,2,0,0,0,0,0,0,2,2,2,3,3,1,1,1,0,0,0,0,0,

0,0,1,1,1,1,3,3,2,2,2,0,0,0,0,0,0,2,2,2,3,3,1,1,1,1,0,0,0,0,

0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

};

const char eyes\_4[] = { // Illustration 4

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,

0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

};

unsigned long curtime = millis(); //////////////////////////////////////////////////////////// Time variable in milliseconds

////////////////////////////////////////////////////////////////////////////////////////////// Time intervals in milliseconds

unsigned long eyes\_nextTime = 1000;

unsigned long eyes\_nextTime1 = 500;

unsigned long eyes\_nextTime2 = 1500;

unsigned long eyes\_interval = 1000;

unsigned long eyes\_interval1 = 3000;

unsigned long eyes\_interval2 = 3100;

unsigned long eyes\_interval3 = 3200;

unsigned long eyes\_interval4 = 4600;

unsigned long eyes\_interval5 = 4700;

unsigned long eyes\_interval6 = 4800;

void setup() {

creoqode.begin();

}

void loop() {

unsigned long curtime = millis();

////////////////////////////////////////////////////////////////////////////////////////////// Drawing the Eye Illustration 1

if (curtime - eyes\_nextTime1 < eyes\_interval1){

for(int v = 0; v<20; v++){

for(int t = 0; t<30; t++){

if(eyes\_1[v\*30+t] == 1){

creoqode.drawPixel(2+t, v, creoqode.Color333(1,1,1)); // Left Eye

creoqode.drawPixel(2+32+t, v, creoqode.Color333(1,1,1)); // Right Eye

}

else if (eyes\_1[v\*30+t] == 2){

creoqode.drawPixel(2+t, v, creoqode.Color333(0,1,6));

creoqode.drawPixel(2+32+t, v, creoqode.Color333(0,1,6));

}

else if (eyes\_1[v\*30+t] == 3){

creoqode.drawPixel(2+t, v, creoqode.Color333(7,7,7));

creoqode.drawPixel(2+32+t, v, creoqode.Color333(7,7,7));

}

else {

creoqode.drawPixel(2+t, v, creoqode.Color333(0,0,0));

creoqode.drawPixel(2+32+t, v, creoqode.Color333(0,0,0));

}

}

}

}

////////////////////////////////////////////////////////////////////////////////////////////// Drawing the Eye Illustration 2

if (curtime - eyes\_nextTime1 >= eyes\_interval1 && curtime - eyes\_nextTime1 < eyes\_interval2){

for(int v = 0; v<20; v++){

for(int t = 0; t<30; t++){

if(eyes\_2[v\*30+t] == 1){

creoqode.drawPixel(2+t, v, creoqode.Color333(1,1,1));

creoqode.drawPixel(2+32+t, v, creoqode.Color333(1,1,1));

}

else if (eyes\_2[v\*30+t] == 2){

creoqode.drawPixel(2+t, v, creoqode.Color333(0,1,6));

creoqode.drawPixel(2+32+t, v, creoqode.Color333(0,1,6));

}

else if (eyes\_2[v\*30+t] == 3){

creoqode.drawPixel(2+t, v, creoqode.Color333(7,7,7));

creoqode.drawPixel(2+32+t, v, creoqode.Color333(7,7,7));

}

else {

creoqode.drawPixel(2+t, v, creoqode.Color333(0,0,0));

creoqode.drawPixel(2+32+t, v, creoqode.Color333(0,0,0));

}

}

}

}

////////////////////////////////////////////////////////////////////////////////////////////// Drawing the Eye Illustration 3

if (curtime - eyes\_nextTime1 >= eyes\_interval2 && curtime - eyes\_nextTime1 < eyes\_interval3){

for(int v = 0; v<20; v++){

for(int t = 0; t<30; t++){

if(eyes\_3[v\*30+t] == 1){

creoqode.drawPixel(2+t, v, creoqode.Color333(1,1,1));

creoqode.drawPixel(2+32+t, v, creoqode.Color333(1,1,1));

}

else if (eyes\_3[v\*30+t] == 2){

creoqode.drawPixel(2+t, v, creoqode.Color333(0,1,6));

creoqode.drawPixel(2+32+t, v, creoqode.Color333(0,1,6));

}

else if (eyes\_3[v\*30+t] == 3){

creoqode.drawPixel(2+t, v, creoqode.Color333(7,7,7));

creoqode.drawPixel(2+32+t, v, creoqode.Color333(7,7,7));

}

else {

creoqode.drawPixel(2+t, v, creoqode.Color333(0,0,0));

creoqode.drawPixel(2+32+t, v, creoqode.Color333(0,0,0));

}

}

}

}

////////////////////////////////////////////////////////////////////////////////////////////// Drawing the Eye Illustration 4

if (curtime - eyes\_nextTime1 >= eyes\_interval3 && curtime - eyes\_nextTime1 < eyes\_interval4){

for(int v = 0; v<20; v++){

for(int t = 0; t<30; t++){

if(eyes\_4[v\*30+t] == 1){

creoqode.drawPixel(2+t, v, creoqode.Color333(1,1,1));

creoqode.drawPixel(2+32+t, v, creoqode.Color333(1,1,1));

}

else if (eyes\_4[v\*30+t] == 2){

creoqode.drawPixel(2+t, v, creoqode.Color333(0,1,6));

creoqode.drawPixel(2+32+t, v, creoqode.Color333(0,1,6));

}

else if (eyes\_4[v\*30+t] == 3){

creoqode.drawPixel(2+t, v, creoqode.Color333(7,7,7));

creoqode.drawPixel(2+32+t, v, creoqode.Color333(7,7,7));

}

else {

creoqode.drawPixel(2+t, v, creoqode.Color333(0,0,0));

creoqode.drawPixel(2+32+t, v, creoqode.Color333(0,0,0));

}

}

}

}

////////////////////////////////////////////////////////////////////////////////////////////// Drawing the Eye Illustration 3

if (curtime - eyes\_nextTime1 >= eyes\_interval4 && curtime - eyes\_nextTime1 < eyes\_interval5){

for(int v = 0; v<20; v++){

for(int t = 0; t<30; t++){

if(eyes\_3[v\*30+t] == 1){

creoqode.drawPixel(2+t, v, creoqode.Color333(1,1,1));

creoqode.drawPixel(2+32+t, v, creoqode.Color333(1,1,1));

}

else if (eyes\_3[v\*30+t] == 2){

creoqode.drawPixel(2+t, v, creoqode.Color333(0,1,6));

creoqode.drawPixel(2+32+t, v, creoqode.Color333(0,1,6));

}

else if (eyes\_3[v\*30+t] == 3){

creoqode.drawPixel(2+t, v, creoqode.Color333(7,7,7));

creoqode.drawPixel(2+32+t, v, creoqode.Color333(7,7,7));

}

else {

creoqode.drawPixel(2+t, v, creoqode.Color333(0,0,0));

creoqode.drawPixel(2+32+t, v, creoqode.Color333(0,0,0));

}

}

}

}

////////////////////////////////////////////////////////////////////////////////////////////// Drawing the Eye Illustration 2

if (curtime - eyes\_nextTime1 >= eyes\_interval5 && curtime - eyes\_nextTime1 < eyes\_interval6){

for(int v = 0; v<20; v++){

for(int t = 0; t<30; t++){

if(eyes\_2[v\*30+t] == 1){

creoqode.drawPixel(2+t, v, creoqode.Color333(1,1,1));

creoqode.drawPixel(2+32+t, v, creoqode.Color333(1,1,1));

}

else if (eyes\_2[v\*30+t] == 2){

creoqode.drawPixel(2+t, v, creoqode.Color333(0,1,6));

creoqode.drawPixel(2+32+t, v, creoqode.Color333(0,1,6));

}

else if (eyes\_2[v\*30+t] == 3){

creoqode.drawPixel(2+t, v, creoqode.Color333(7,7,7));

creoqode.drawPixel(2+32+t, v, creoqode.Color333(7,7,7));

}

else {

creoqode.drawPixel(2+t, v, creoqode.Color333(0,0,0));

creoqode.drawPixel(2+32+t, v, creoqode.Color333(0,0,0));

}

}

}

}

////////////////////////////////////////////////////////////////////////////////////////////// Resetting the time variable to make the loop continue

if (curtime - eyes\_nextTime1 >= eyes\_interval6){

eyes\_nextTime1 = curtime;

}

}