



Capturas del juego

Alumnos: Cárdenas Quiroz Christian, Marquéz Ontiveros Marcelo.

Matricula: 369196.

Grupo: 432.

Profesor: Yépiz Núñez Pedro.

Materia: Programación estructurada.

11/Junio/2023

```

1  #include <allegro.h>
2  #include "inicio.h"
3
4  //using namespace std;
5
6  BITMAP *buffer;//crea las variables bitmap
7  BITMAP *ruleta1;
8  BITMAP *ruleta2;
9  BITMAP *ruleta3;
10 BITMAP *ruleta4;
11 BITMAP *division1;
12 BITMAP *division2;
13 BITMAP *division3;
14 BITMAP *division4;
15 BITMAP *division5;
16 BITMAP *multiplicacion1;
17 BITMAP *multiplicacion2;
18 BITMAP *multiplicacion3;
19 BITMAP *multiplicacion4;
20 BITMAP *multiplicacion5;
21 BITMAP *resta1;
22 BITMAP *resta2;
23 BITMAP *resta3;
24 BITMAP *resta4;
25 BITMAP *resta5;
26 BITMAP *suma1;
27 BITMAP *suma2;
28 BITMAP *suma3;
29 BITMAP *suma4;
30 BITMAP *suma5;
31 BITMAP *correcto;
32 BITMAP *incorrecto;
33 BITMAP *inicio;
34 int x = 0;//osicion x
35 int y = 0;//osicion y
36
37
38 void ruleta(BITMAP *ruleta)// procedimiento para mostrar imagenes de la ruleta
39 {
40     blit(ruleta,screen,0,0,x,y,640,480);//carga la imagen en pantalla
41     rest(30);//tiempo de la imagen en pantalla
42 }
43
44 int random(int max)// funcion randompara generar aleatoriedad
45 {
46     int n;
47     srand(time(NULL));
48     n = rand();
49     return n%max;
50 }
51
52 void respuesta(int correcta)//rocedimiento para validarla respuesta correcta e incorrecta
53 {
54     int val;
55     while(keypressed()) //devuelve el ultimo valor almacenado en el buffer(limpia el buffer)
56     readkey();// funcion lee la tecla presionada desde el teclado
57     val = readkey();
58     // rest(5000);
59
60     if( (val >> 8) == correcta)//lee la letra presionada
61     blit(correcto,screen,0,0,x,y,640,480);//si es igual a la tecla digitada muestra la imagen correcta
62     else
63     blit(incorrecto,screen,0,0,x,y,640,480);// si no es asi muestra el mensaje de incorrecto
64 }
65
66 void pregunta_aleatoria(BITMAP *preg1,BITMAP *preg2,BITMAP *preg3,BITMAP *preg4,BITMAP *preg5)//muestra las i
67 {
68     int ran = (random(5)+1); //se llama la funcion random de 0 a 4 +1
69     if(ran == 1)//si el numero aleatorio es 1
70     {
71         rest(1000);//espera 3 segundos antes de mostrar la imagen
72         blit(preg1,screen,0,0,x,y,640,480);//muestra la imagen segun el parametro de tipo bitmap que se le p
73         respuesta(KEY_C);//llama al procedimiento respuesta paravalidar las respuestas
74     }
75     if(ran == 2)
76     {
77         rest(1000);
78         blit(preg2,screen,0,0,x,y,640,480);
79         respuesta(KEY_A);
80     }
81     if(ran == 3)
82     {
83         rest(1000);
84         blit(preg3,screen,0,0,x,y,640,480);
85     }
86 }

```

```

88     respuesta(KEY_B);
89 }
90 if(ran == 4)
91 {
92     rest(1000);
93     blit(preg4,screen,0,0,x,y,640,480);
94     respuesta(KEY_B);
95 }
96 if(ran == 5)
97 {
98     rest(1000);
99     blit(preg5,screen,0,0,x,y,640,480);
100     respuesta(KEY_A);
101 }
102 }
103 }
104
105 void girar_ruleta()//procedimiento para girar la ruleta muestra las imagenes
106 {
107     if(key[KEY_ENTER])// si se presiona la tecla enter inicia el giro de la ruleta
108     {
109         for(int i=0;i<10;i++)//organiza para mostrar las imagenes
110         {
111             rull(ruleta1);
112             rull(ruleta2);
113             rull(ruleta3);
114             rull(ruleta4);
115         }
116
117         int r = (random(4)+1);//genera un numero aleatorio de 0 a 4+1
118         if(r == 1)// si el numero aleatorio es 1
119         {rull(ruleta1);//muestra la imagen de la ruleta en laprimera posicion
120         pregunta_aleatoria(multiplicacion1,multiplicacion2, multiplicacion3,multiplicacion4, multiplicacion5);
121         }
122         if(r == 2)
123         {rull(ruleta2);
124         pregunta_aleatoria(division1,division2, division3,division4, division5);
125         }
126         if(r == 3)
127         {rull(ruleta3);
128         pregunta_aleatoria(suma1,suma2, suma3,suma4, suma5);
129         }
130         if(r == 4)
131         {rull(ruleta4);
132         pregunta_aleatoria(resta1,resta2, resta3,resta4, resta5);
133         }
134     }
135 }
136 }
137
138 void jugar()//procedimiento para iniciar el juego
139 {
140     blit(inicio,screen, 0,0,x,y,640,480);//muestra la imagen de inicio del juego
141     //rest(500);
142     while(!key[KEY_ESC])//mientras no se presione escape
143     {
144         girar_ruleta();//llama al procedimiento para girar la ruleta
145     }
146 }
147 }
148
149 int main()
150 {
151     inicia_allegro(640, 480);//llama a todos los componentes para inicializar allegro
152     buffer = create_bitmap(640,480);//crea un buffer inicial de fondo blanco
153     clear_to_color(buffer,0xFFFFFF);
154     ruleta1 = load_bitmap("imagenes/imagenes_ruleta/rul1.bmp",NULL);//carga las imagenes al bitmap
155     ruleta2 = load_bitmap("imagenes/imagenes_ruleta/rul2.bmp",NULL);
156     ruleta3 = load_bitmap("imagenes/imagenes_ruleta/rul3.bmp",NULL);
157     ruleta4 = load_bitmap("imagenes/imagenes_ruleta/rul4.bmp",NULL);
158     division1 = load_bitmap("imagenes/divi/divi1.bmp",NULL);
159     division2 = load_bitmap("imagenes/divi/divi2.bmp",NULL);
160     division3 = load_bitmap("imagenes/divi/divi3.bmp",NULL);
161     division4 = load_bitmap("imagenes/divi/divi4.bmp",NULL);
162     division5 = load_bitmap("imagenes/divi/divi5.bmp",NULL);
163     multiplicacion1 = load_bitmap("imagenes/multi/multi1.bmp",NULL);
164     multiplicacion2 = load_bitmap("imagenes/multi/multi2.bmp",NULL);
165     multiplicacion3 = load_bitmap("imagenes/multi/multi3.bmp",NULL);
166     multiplicacion4 = load_bitmap("imagenes/multi/multi4.bmp",NULL);
167     multiplicacion5 = load_bitmap("imagenes/multi/multi5.bmp",NULL);
168     resta1 = load_bitmap("imagenes/resta/resta1.bmp",NULL);
169     resta2 = load_bitmap("imagenes/resta/resta2.bmp",NULL);
170     resta3 = load_bitmap("imagenes/resta/resta3.bmp",NULL);
171     resta4 = load_bitmap("imagenes/resta/resta4.bmp",NULL);
172     resta5 = load_bitmap("imagenes/resta/resta5.bmp",NULL);

```

```

175 suma1 = load_bitmap("imagenes/suma/suma1.bmp",NULL);
176 suma2 = load_bitmap("imagenes/suma/suma2.bmp",NULL);
177 suma3 = load_bitmap("imagenes/suma/suma3.bmp",NULL);
178 suma4 = load_bitmap("imagenes/suma/suma4.bmp",NULL);
179 suma5 = load_bitmap("imagenes/suma/suma5.bmp",NULL);
180 correcto = load_bitmap("imagenes/correcto.bmp",NULL);
181 inicio = load_bitmap("imagenes/logo1.bmp",NULL);
182 incorrecto = load_bitmap("imagenes/incorrecto.bmp",NULL);
183 blit(buffer,screen, 0,0,0,1200,600);
184
185 jugar();//llama al procedimeinto jugar
186
187 //readkey();
188 destroy_bitmap(buffer);//libera la las imagenes de lamemoria
189 destroy_bitmap(ruleta1);
190 destroy_bitmap(ruleta2);
191 destroy_bitmap(ruleta3);
192 destroy_bitmap(ruleta4);
193 destroy_bitmap(division1);
194 destroy_bitmap(division2);
195 destroy_bitmap(division3);
196 destroy_bitmap(division4);
197 destroy_bitmap(division5);
198 destroy_bitmap(multiplicacion1);
199 destroy_bitmap(multiplicacion2);
200 destroy_bitmap(multiplicacion3);
201 destroy_bitmap(multiplicacion4);
202 destroy_bitmap(multiplicacion5);
203 destroy_bitmap(resta1);
204 destroy_bitmap(resta2);
205 destroy_bitmap(resta3);
206 destroy_bitmap(resta4);
207 destroy_bitmap(resta5);
208 destroy_bitmap(suma1);
209 destroy_bitmap(suma2);
210 destroy_bitmap(suma3);
211 destroy_bitmap(suma4);
212 destroy_bitmap(suma5);
213 destroy_bitmap(correcto);
214 destroy_bitmap(incorrecto);
215 destroy_bitmap(inicio);
216
217 return 0;
218 }
219 END_OF_MAIN();
220
221

```

Juego corriendo:





Ruleta Cargando...

$11 * 11 = ?$

a) 121

b) 111

c) 101

