Code Road: Developing a Moving Car in C++

Ella Norienne C. Dacapio

Cherry Lee H. Jimenez

In Partial Fulfillment of the Requirements

for the CS121 Computer Programming 2

Department of Computer Science

College of Information Technology and Computing

University of Science and Technology of Southern Philippines

Cagayan de Oro City, 9000

June 2023

TABLE OF CONTENTS

CHAPTER I

INTRODUCTION

1.1 Background of the Project

In computer programming, the graphics header file is graphics.h, which provides multiple functions to draw shapes using coordinates like rectangles, ellipses, circles, and lines. By using C++ language-based programming, a moving car is drawn with the use of lines and circles. This type of software simulates a 2D car movement that allows the user to change its speed depending on the user input, resulting from the input-output function.

1.2 Statement of the Problem

The goal is to provide an interactive experience by giving people control over the vehicle's speed. Achieving the desired functionality will require addressing several challenges:

1. Designing a visually captivating animation of a car and the background is challenging. It necessitates proficiency in real-time animation updates and familiarity with graphics libraries.
2. Recording and processing human input to automatically adjust the car's speed can be complex. The software must swiftly recognize user input and promptly update the program.
3. The car's position on the screen must be dynamically updated to create a realistic motion simulation. This means constantly computing its new position based on the chosen speed to generate a smooth animation and updating the screen accordingly.

We aim to address these challenges by providing an effective solution that combines graphics, user engagement, and dynamic animation. With this application, users can control the car's speed, engage in an interactive experience, and learn the basics of C++ programming.

1.3 Objective of the Project

The Moving Car Program, developed using C++, provides students with an engaging and interactive learning opportunity. The program supports students in comprehending graphics programming and user interface design by simulating a car's movement and allowing users to control its speed through input.

This project has the benefit of being adaptable to catering to the specific requirements of each student. Individuals with more experience can add intricate components to the program, while novices can start with the fundamentals and advance gradually. This method guarantees that students can study at their own pace and level of understanding.

1.4 Scope and Limitations

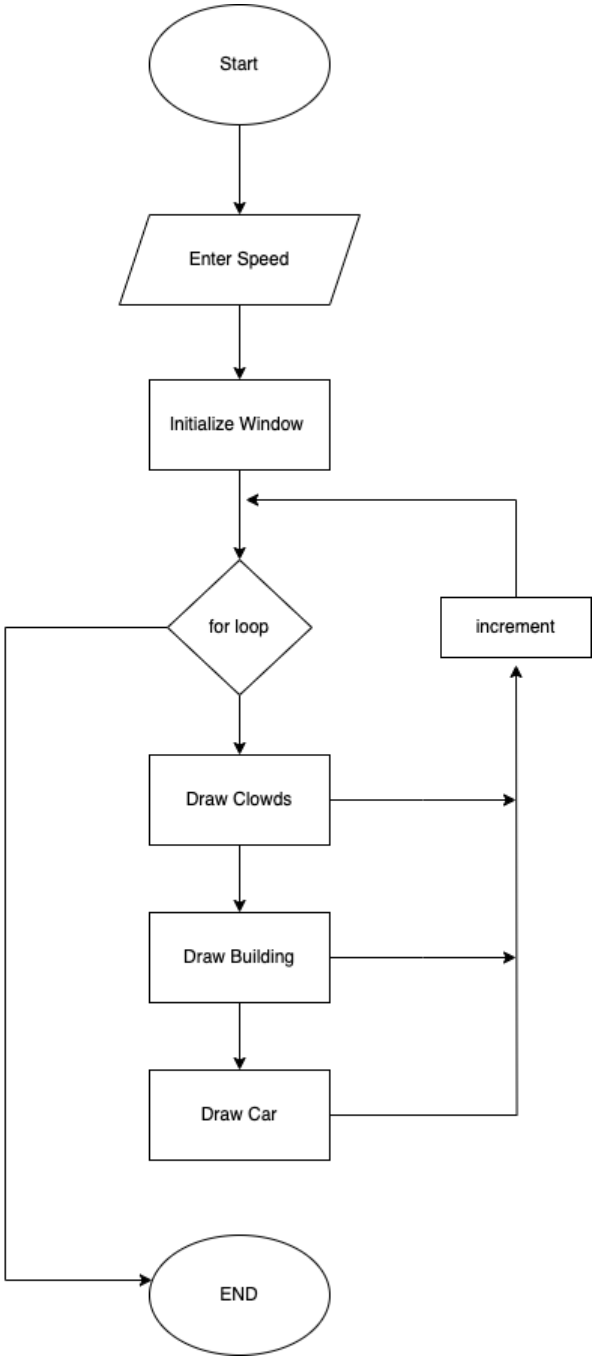
This project only intends to demonstrate and illustrate a simulation of a moving car in graphics programming using C++ as a programming language. To identify how to create a moving car using C++ coding language and determine how it works and functions.

Locale of the Study

This study takes place in the respective houses of each researcher through online platforms such as Facebook and Messenger and during the face-to-face laboratory discussion in the second semester of the school year 2022-2023, starting on April 23, 2023.

CHAPTER II

FLOWCHART



CHAPTER III

SOURCE CODE

```
#include <graphics.h>
#include <iostream>
using namespace std;

int main(){
    int page = 1;
    int sp;

    cout << "INPUT SPEED: " << endl;
    cin >> sp;

    initwindow(1000,800);

    int i = 0, j,maxx;
    maxx = getmaxx();
    for(int i = 0; i < maxx; i ++){
        setactivepage(page);
        setvisualpage(1-page);
        cleardevice();
        int cloudc = 7;
        setcolor(7);
        setfillstyle(SOLID_FILL,7);

        ///clouds
        rectangle(0 +i*2, 20, 50 +i*2, 30);
        floodfill(1 + i*2, 21, cloudc);
        rectangle(25 + i*2, 30, 75 +i*2, 40);
        floodfill(26 + i*2, 31, cloudc);
        rectangle(-90 + i*2, 10, -145 +i*2, 28);
        floodfill(-91 + i*2, 11, cloudc);
        rectangle(-1000 + i*4, 50, -1060 + i*4, 65);
        floodfill(-1001 + i*4, 51, cloudc);
        rectangle(-800 + i*3, 40, -850 + i*3, 55);
        floodfill(-801 + i*3, 41, cloudc);
        rectangle(-500 + i*2, 25, -530 + i*2, 40);
        floodfill(-501, 26, cloudc);
        rectangle(-140 +i*3, 65, -180 +i*3, 80);
        floodfill(-141 + i*3, 66, cloudc);
        rectangle(900+i*2, 30, 980 + i*2, 60);
        floodfill(901 + i*2, 31, cloudc);
        rectangle(750 +i*2, 10, 810 + i*2, 25);
        floodfill(751 + i*2, 11, cloudc);
```

```
43          rectangle(500 + i*2, 20, 560 +i*2, 45);
44          floodfill(501 + i*2, 21, cloudc);
45          rectangle(250 + i*3, 40, 330 + i*3, 60);
46          floodfill(251 + i*3, 41, cloudc);
47          rectangle(-570 + i*3, 60, -620 + i*3, 80);
48          floodfill(-571 + i*3, 61, cloudc);
49          rectangle(-700 +i*2, 70, -760 +i*2, 90);
50          floodfill(-701 +i*2, 71, cloudc);
51          rectangle(195 +i*4, 90, 255 +i*4, 115);
52          floodfill(196 + i*4, 91, cloudc);
53          rectangle(-400 +i*2, 10, -500 +i*2, 25);
54          floodfill(-401 +i*2, 11, cloudc);
55          rectangle(-1050 + i*3, 30, -1070 + i*3, 50);
56          floodfill(-1051, 31, cloudc);
57          rectangle(-1100 + i*2, 35, -1144 + i*2, 50);
58          floodfill(-1101, 36, cloudc);
59          rectangle(-1105 + i*3, 5, -1132 + i*3, 16);
60          floodfill(-1106, 6, cloudc);
61
62          delay(15);
63
64          // building y size length
65          int Bysize = 440;
66
67          //backbuildings
68          int backg = 1;
69          setcolor(backg);
70          setfillstyle(SOLID_FILL, backg);
71          rectangle(0, 245, maxx, Bysize);
72          floodfill(1, 246, backg);
73          rectangle(0, 220, 35, Bysize);
74          floodfill(1, 221, backg);
75          rectangle(35, 190, 90, Bysize);
76          floodfill(36, 191, backg);
77          rectangle(100, 220, 140, Bysize);
78          floodfill(101, 221, backg);
79          rectangle(140, 200, 170, Bysize);
80          floodfill(141, 201, backg);
81          rectangle(185, 220, 240, Bysize);
82          floodfill(186, 221, backg);
83          rectangle(240, 200, 300, Bysize);
84          floodfill(241, 201, backg);
```

```
85          rectangle(320, 225, 370, Bysize);
86          floodfill(321, 226, backg);
87          rectangle(380, 210, 450, Bysize);
88          floodfill(381, 211, backg);
89          rectangle(450, 220, 610, Bysize);
90          floodfill(451, 221, backg);
91          rectangle(500, 195, 550, 220);
92          floodfill(501, 196 , backg);
93          rectangle(600, 210, 650, Bysize);
94          floodfill(601, 211, backg);
95          rectangle(700, 205, 770, Bysize);
96          floodfill(701, 206, backg);
97          rectangle(770, 225, 830, Bysize);
98          floodfill(771, 226, backg);
99          rectangle(840, 200, 890, Bysize);
100         floodfill(841, 201, backg);
101         rectangle(895, 225, 940, Bysize);
102         floodfill(896, 226, backg);
103         rectangle(940, 210, 1000, Bysize);
104         floodfill(941, 211, backg);
105
106         // middleground building
107         int mbackg = 9;
108         setcolor(mbackg);
109         setfillstyle(SOLID_FILL, mbackg);
110         rectangle(0, 300, maxx, Bysize);
111         floodfill(1, 301, mbackg);
112         rectangle(0, 260, 20, Bysize);
113         floodfill(1, 261, mbackg);
114         rectangle(20, 250, 45, Bysize);
115         floodfill(21, 251, mbackg);
116         rectangle(55, 270, 90, Bysize);
117         floodfill(56, 271, mbackg);
118         rectangle(90, 265, 125, Bysize);
119         floodfill(91, 266, mbackg);
120         rectangle(125, 285, 180, Bysize);
121         floodfill(126, 286, mbackg);
122         rectangle(200, 265, 240, Bysize);
123         floodfill(201, 266, mbackg);
124         rectangle(255, 265, 280, Bysize);
125         floodfill(256, 266, mbackg);
126         rectangle(280, 250, 335, Bysize);
```

```
127         floodfill(281, 251, mbackg);
128         rectangle(350, 275, 420, Bysize);
129         floodfill(351, 276, mbackg);
130         rectangle(435, 265, 460, Bysize);
131         floodfill(436, 266, mbackg);
132         rectangle(460, 245, 480, Bysize);
133         floodfill(461, 246, mbackg);
134         rectangle(490, 280, 560, Bysize);
135         floodfill(491, 281, mbackg);
136         rectangle(580, 270, 630, Bysize);
137         floodfill(581, 271, mbackg);
138         rectangle(630, 250, 680, Bysize);
139         floodfill(631, 251, mbackg);
140         rectangle(680, 280, 715, Bysize);
141         floodfill(681, 281, mbackg);
142         rectangle(725, 270, 755, Bysize);
143         floodfill(726, 271, mbackg);
144         rectangle(755, 255, 800, Bysize);
145         floodfill(756, 256, mbackg);
146         rectangle(815, 270, 850, Bysize);
147         floodfill(816, 271, mbackg);
148         rectangle(850, 280, 890, Bysize);
149         floodfill(851, 281, mbackg);
150         rectangle(900, 255, 930, Bysize);
151         floodfill(901, 256, mbackg);
152         rectangle(930, 270, 960, Bysize);
153         floodfill(931, 271, mbackg);
154         rectangle(975, 260, maxx, Bysize);
155         floodfill(976, 261, mbackg);
156
157         //foreground
158         int fbgc = 3;
159         setcolor(fbgc);
160         setfillstyle(SOLID_FILL, fbgc);
161
162         rectangle(0, 351, maxx, Bysize);
163         floodfill(1, 352, fbgc);
164         rectangle(0, 315, 20, Bysize);
165         floodfill(1, 316, fbgc);
166         rectangle(30, 325, 100, Bysize);
167         floodfill(31, 326, fbgc);
168         rectangle(110, 305, 170, Bysize);
```

```
169         floodfill(111, 306, fbgc);
170         rectangle(185, 315, 250, Bysize);
171         floodfill(186, 316, fbgc);
172         rectangle(250, 300, 310, Bysize);
173         floodfill(251, 301, fbgc);
174         rectangle(325, 310, 380, Bysize);
175         floodfill(326, 311, fbgc);
176         rectangle(390, 320, 445, Bysize);
177         floodfill(391, 321, fbgc);
178         rectangle(450, 305, 510, Bysize);
179         floodfill(451, 306, fbgc);
180         rectangle(515, 315, 560, Bysize);
181         floodfill(516, 316, fbgc);
182         rectangle(570, 325, 619, Bysize);
183         floodfill(571, 326, fbgc);
184         rectangle(619, 295, 643, Bysize);
185         floodfill(620, 296, fbgc);
186         rectangle(650, 298, 688, Bysize);
187         floodfill(651, 299, fbgc);
188         rectangle(710, 337, 743, Bysize);
189         floodfill(711, 338, fbgc);
190         rectangle(743, 300, 788, Bysize);
191         floodfill(744, 301, fbgc);
192         rectangle(788, 315, 820, Bysize);
193         floodfill(789, 316, fbgc);
194         rectangle(833, 285, 849, Bysize);
195         floodfill(834, 286, fbgc);
196         rectangle(849, 305, 878, Bysize);
197         floodfill(850, 306,fbgc);
198         rectangle(895, 330, 920, Bysize);
199         floodfill(896, 331, fbgc);
200         rectangle(920, 315, 950, Bysize);
201         floodfill(921, 316, fbgc);
202         rectangle(960, 340, 1000, Bysize);
203         floodfill(961, 341, fbgc);
204
205         // details for the background
206         int col = 0;
207         setcolor(col);
208         setfillstyle(SOLID_FILL, col);
209
210         rectangle(0, 390, 20, Bysize);
```

```
211         floodfill(1, 391, col);
212         rectangle(20, 420, 43, Bysize);
213         floodfill(21, 421, col);
214         rectangle(43, 400, 77, Bysize);
215         floodfill(44, 401, col);
216         rectangle(77, 407, 95, Bysize);
217         floodfill(78, 408, col);
218         rectangle(95, 415, 128, Bysize);
219         floodfill(96, 416, col);
220         rectangle(128, 398, 155, Bysize);
221         floodfill(129, 399, col);
222         rectangle(155, 417, 165, Bysize);
223         floodfill(156, 418, col);
224         rectangle(165, 430, 182, Bysize);
225         floodfill(166, 431, col);
226         rectangle(369, 438, 402, Bysize);
227         floodfill(370, 439, col);
228         rectangle(402, 419, 435, Bysize);
229         floodfill(403, 420, col);
230         rectangle(435, 430, 455, Bysize);
231         floodfill(436, 431, col);
232         rectangle(455, 402, 490, Bysize);
233         floodfill(456, 403, col);
234         rectangle(490, 380, 500, Bysize);
235         floodfill(491, 381, col);
236         rectangle(500, 425, 530, Bysize);
237         floodfill(501, 426, col);
238         rectangle(530, 413, 555, Bysize);
239         floodfill(531, 414, col);
240         rectangle(555, 390, 594, Bysize);
241         floodfill(556, 391, col);
242         rectangle(594, 429, 609, Bysize);
243         floodfill(595, 430, col);
244         rectangle(609, 415, 634, Bysize);
245         floodfill(610, 416, col);
246         rectangle(634, 442, 655, Bysize);
247         floodfill(635, 443, col);
248         rectangle(795, 433, 824, Bysize);
249         floodfill(796, 434, col);
250         rectangle(824, 420, 852, Bysize);
251         floodfill(825, 421, col);
252         rectangle(852, 398, 872, Bysize);
```

```
253         floodfill(853, 399, col);
254         rectangle(872, 395, 906, Bysize);
255         floodfill(873, 396, col);
256         rectangle(906, 418, 934, Bysize);
257         floodfill(907, 419, col);
258         rectangle(934, 410, 957, Bysize);
259         floodfill(935, 411, col);
260         rectangle(957, 400, 980, Bysize);
261         floodfill(958, 401, col);
262         rectangle(980, 411, 990, Bysize);
263         floodfill(981, 412, col);
264         rectangle(990, 390, maxx, Bysize);
265         floodfill(991, 391, col);
266
267         // first wheel
268         int carcolor = 8;
269         setcolor(carcolor);
270         setfillstyle(SOLID_FILL, carcolor);
271         pieslice(1 + i*sp, 570, 0 - i*sp, 90 - i*sp, 30);
272         pieslice(1 + i*sp, 570, 180 - i*sp, 270 - i*sp, 30);
273         circle(1 + i*sp , 570, 30);
274         arc(1 + i*sp, 570, 0, 180, 35);
275
276         // second wheel
277         pieslice(250 + i*sp, 570, 0 - i*sp, 90 - i*sp, 30);
278         pieslice(250 + i*sp, 570, 180 - i*sp, 270 - i*sp, 30);
279         circle(250 + i*sp, 570, 30);
280         arc(250 + i*sp, 570, 0, 180, 35);
281
282         // car
283         line(36 + i*sp, 570, 215 + i*sp, 570);  // center
284         line(-76 + i*sp, 570, -33 + i*sp, 570); // back
285         line(-76 + i*sp, 520, -76 + i*sp, 570); // back height
286         line(335 + i*sp, 570, 285 + i*sp, 570); // front
287         line(336 + i*sp, 570, 336 + i*sp, 535); // front height
288         line(336 + i*sp, 535, 240 + i*sp, 510); // front slant
289         line(240 + i*sp, 510, 195 + i*sp, 455); // front window
290         line(-8 + i*sp, 455, 195 + i*sp, 455); // ceiling
291         line(-8 + i*sp, 455, -33 + i*sp, 500); // back slant
292         line(-76 + i*sp, 520, -33 + i*sp, 500); // trunk
```

```
293
294             // windows
295             line(95 + i*sp, 464, 95 + i*sp, 510); // middle front line
296             line(95 + i*sp, 510, 230 + i*sp, 510); // lower front line
297             line(230 + i*sp, 510, 195 + i*sp, 464); // front line slant
298             line(95 + i*sp, 464, 195 + i*sp, 464); // front line middle
299             line(89 + i*sp, 464, 89 + i*sp, 510); // middle back line
300             line(-30 + i*sp, 510, 89 + i*sp, 510); // lower back line
301             line(-6 + i*sp, 464, -32 + i*sp, 510); // lower back slant
302             line(-5 + i*sp, 464, 89 + i*sp, 464); // back line middle
303             line(0, 445, maxx, 445);
304
305             setcolor(WHITE);
306             setfillstyle(SOLID_FILL, WHITE);
307             line(0, 600, maxx, 600); // road
308             // text
309             outtextxy(450, 620, "THE MOVING CAR");
310
311             page = 1-page;
312         }
313
314         getch();
315         closegraph();
316         return 0;
317
318 }
```

CHAPTER IV

SAMPLE USER INTERFACE