FINAL PROJECT

REPORT 2

Tri Bui

828135536

COMPE271

Professor Kenneth Arnold

February 20, 2023

1. Initial Design and Description

A math related video game with X-axis moving character, the other objects move down on Y-axis with value that have math ( + x / - with a number) to win the game get those objects to get the biggest number to kill an random number amount of enemies.

1. How it work

- Switching to another way of printing mayeb. Using an Array as the map for the game and using Rand to make math equation

- Rand generated number of enemies that gradually increase overtime

1. Function of the code

- draw() : draw the outline of the game screen

- setup() : setting up the starting array

- enemies() : to generate the number of enimies

- mathchoice() : making math equation and choice

- logic() : movement for game

- kbhit() : implement of kbhit from conio.h on Windows because the library is not available on Linux

1. Progress:

- Finished draw()

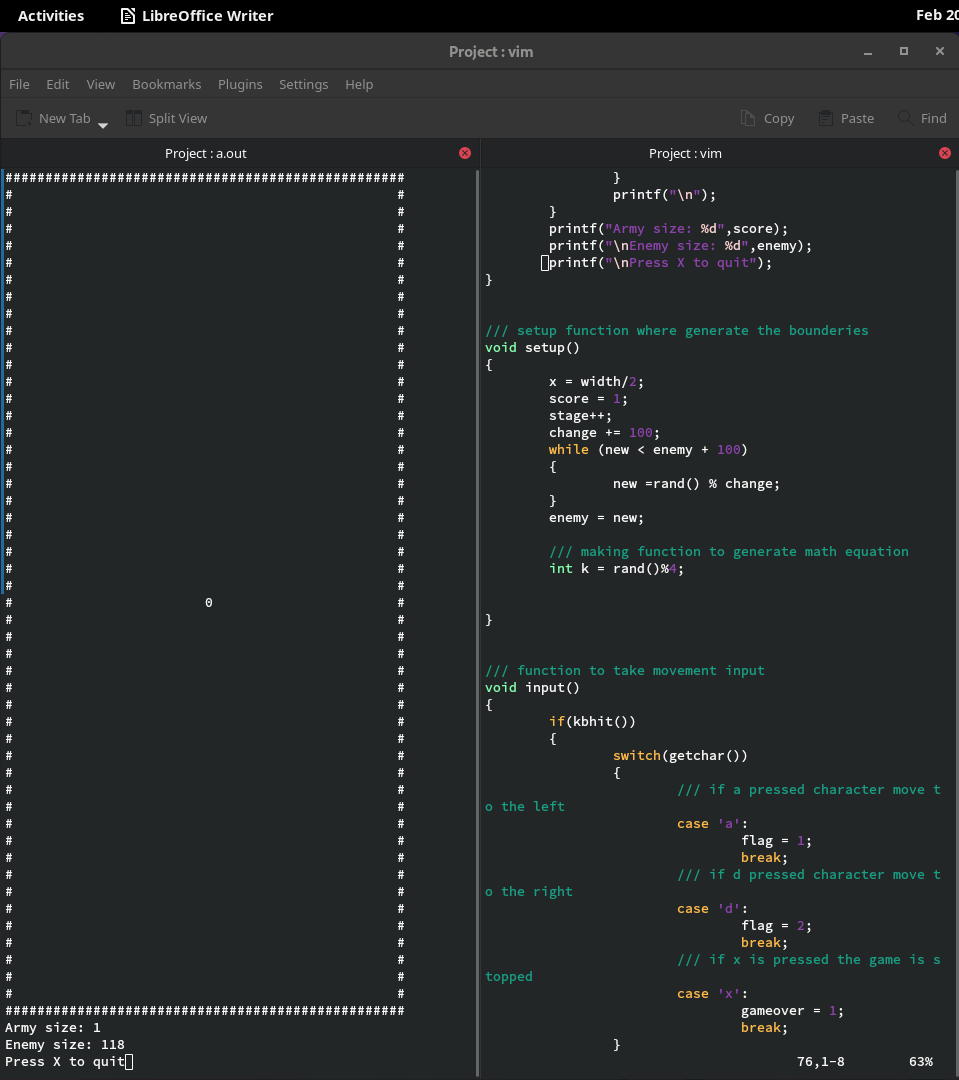
- kbhit() working but need some tweaks

- Generator for enemy is working

- input() and logic() currently not working need some adjustment

- Need to make the math function work

1. Screenshot:



1. Code:

/// Name: Tri Bui

1. /// RedID: 82813556
2. /// Class: Compe271
3. /// Professor: Kenneth Arnold
4. /// Project name: game project for class final
5. /// Project type: C
6. /////////////////
7. #include <stdio.h>
8. #include <stdlib.h>
9. #include <termios.h>
10. #include <unistd.h>
11. #include <fcntl.h>
12. #include <curses.h>
13. #include <time.h>
14. /// setting up parameter for the game screen
15. int score, gameover;
16. int height = 50;
17. int width = 50;
18. int x,flag;
19. int enemy,new;
20. int stage = 0;
21. int change = 50;
22. /// kbhit from conio.h but implement using ncurses.h created with help online on making it
23. int kbhit(void)
24. {
25. struct termios oldt, newt;
26. int ch;
27. int oldf;
28. tcgetattr(STDIN\_FILENO, &oldt);
29. newt = oldt;
30. newt.c\_lflag &= ~(ICANON | ECHO);
31. tcsetattr(STDIN\_FILENO, F\_GETFL, 0);
32. fcntl(STDIN\_FILENO, F\_SETFL, oldf | O\_NONBLOCK);
33. ch = getchar();
34. tcsetattr(STDIN\_FILENO, TCSANOW, &oldt);
35. fcntl(STDIN\_FILENO, F\_SETFL, oldf);
36. if(ch != EOF)
37. {
38. ungetc(ch, stdin);
39. return 1;
40. }
41. return 0;
42. }
43. /// Function to draw the area of game
44. void draw()
45. {
46. system("clear");
47. for(int i = 0; i < height; i++)
48. {
49. for(int j = 0; j < width; j++)
50. {
51. if (i == 0 || i == width - 1 || j == 0 || j == height - 1)
52. {
53. printf("#");
54. }
55. else
56. {
57. if ( i == x && j == height/2)
58. printf("0");
59. else
60. printf(" ");
61. }
62. }
63. printf("\n");
64. }
65. printf("Army size: %d",score);
66. printf("\nEnemy size: %d",enemy);
67. printf("\nPress X to quit");
68. }
69. /// setup function where generate the bounderies
70. void setup()
71. {
72. x = width/2;
73. score = 1;
74. stage++;
75. change += 100;
76. while (new < enemy + 100)
77. {
78. new =rand() % change;
79. }
80. enemy = new;
81. /// making function to generate math equation
82. int k = rand()%4;
83. }
84. /// function to take movement input
85. void input()
86. {
87. if(kbhit())
88. {
89. switch(getchar())
90. {
91. /// if a pressed character move to the left
92. case 'a':
93. flag = 1;
94. break;
95. /// if d pressed character move to the right
96. case 'd':
97. flag = 2;
98. break;
99. /// if x is pressed the game is stopped
100. case 'x':
101. gameover = 1;
102. break;
103. }
104. }
105. }
106. /// function for the logic of the movement
107. void logic()
108. {
109. sleep(0.01);
110. while(x < width && x > 0)
111. {
112. switch(flag)
113. {
114. case 1:
115. x--;
116. break;
117. case 2:
118. x++;
119. break;
120. default:
121. break;
122. }
123. }
125. }
126. /// main function
127. void main()
128. {
129. srand(time(NULL));
130. setup();
131. while (!gameover)
132. {
133. draw();
134. input();
135. logic();
136. }
138. }