

Nombre: Miguel Alejandro Santiago Pérez

No. de Matrícula.: zap362

Materia: Fundamentos de la Programación

Grupo: Dev1

Turno: Matutino

Carrera: Desarrollo de Software Interactivo Y Videojuegos

Tema: SwitchCase

No: R.6

Fecha propuesta: 12/04/2021

Fecha de Entrega: 18/04/2021

Escuela: Instituto de Estudios Universitarios Amerike

Plantel Zapopan

Calle: Calle Montemorelos

No: 3503

Colonia: Rinconada de la Calma

C.P.: 45080

Teléfono: 3336326100

Ciudad: Zapopan



Firma del alumno (a)

Firma de revisión fecha

Qué se evalúa:	10 pts.	7 pts.	4pts.	Pts.
Entrega electrónica	Es en tiempo y forma al iniciar la clase. (1 pts.)	Después de 30 minutos de iniciada la clase. (.7 pts.)	Al minuto 40. (Posteriormente ya no se reciben) (.4pts.)	
Del formato.	Cumple con todos los elementos solicitados. (1 pts.)	No cumple con dos elementos solicitados. (.7 pts.)	No cumple con tres o más elementos solicitados. (.4pts.)	
La ortografía.	Tiene dos errores ortográficos. (1 pts.)	Tiene de tres a cuatro errores ortográficos. (.7 pts.)	Tiene cinco o más errores ortográficos. (.4pts.)	
Del tema y objetivo.	La teoría y ejemplos corresponden al tema tratado. (1 pts.)	La teoría o ejemplos no corresponden al tema tratado. (.7 pts.)	La teoría y ejemplos no corresponden al tema tratado. (.4pts.)	
El programa y los cálculos.	Los parámetros y componentes corresponden al 100% de lo planeado. (1 pts.)	El programa arroja un error o componente no corresponden al 100% de lo planeado. (7 pts.)	El programa arroja dos errores o componentes no corresponden al 100% de lo calculado. (.4pts.)	
Diagramas.	Los diagramas a bloques, de flujo y esquemáticos son acorde al de la práctica y siguen una secuencia lógica. (1 pts.)	Los diagramas a bloques, o de flujo o esquemáticos no son acorde al de la práctica y o no siguen una secuencia lógica. (.7 pts.)	Los diagramas a bloques, de flujo y esquemáticos no son acorde al de la práctica y o no siguen una secuencia lógica. (.4pts.)	
La tabla de valores.	Los valores calculados y medidos presentan una desviación máxima del 10%. (1 pts.)	Los valores calculados y medidos presentan una desviación máxima del 15%. (.7 pts.)	Los valores calculados y medidos presentan una desviación máxima del 20%. (.4pts.)	
Las observaciones y conclusiones.	Son específicas y congruentes con la práctica. (1 pts.)	Las observaciones o conclusiones son específicas y congruentes con la práctica. (.7 pts.)	Las observaciones y las conclusiones no son específicas y congruentes con la práctica. (.4pts.)	
Bibliografía.	Es acorde al (los) tema (s) tratado (s) y está completa (1 pts.)	Es acorde a algún (os) tema (s) tratado (s), le falta algún elemento que la conforman (.7 pts.)	No es acorde al (los) tema (s) tratado (s), le faltan 2 elementos que la conforma (.4pts.)	
Fuentes de consulta.	Es acorde al (los) tema (s) tratado (s) (1 pts.)	Es acorde a algún (os) tema (s) tratado (s) (.7 pts.)	Es acorde a algún (los) tema (s) tratado (s) (.4pts.)	

Índice El objetivo es crear un simulador de batallas Pokémon

Teoría Dependiendo del Pokémon que el usuario busque usar se seleccionara un ataque, el daño generado por ese ataque será random y se repetirá hasta que la vida de uno de los 2 Pokémon llegue a 0

Cálculos

```
int dano = rand() % (Pik.danoMaxImpactrueno - Pik.danoMinImpactrueno + 1) + Pik.danoMinImpactrueno;
```

En el caso arriba el Impactrueno tiene un daño máximo de 30 y mínimo de 10, en este caso primero se restaran entre ambos para obtener un 20 para que según la esquema de la función rand luego volver a sumar el daño mínimo para tener 30 como total

```
PokemonBattles_Sim (Global Scope) main()
1 // PokemonBattles_Sim.cpp : This file contains the 'main' function. Program execution begins and ends there.
2 //
3
4 #include <iostream>
5 #include <stdlib.h>
6 #include <time.h>
7 #include <windows.h>
8 #include <dos.h>
9
10 using namespace std;
11
12 //Equipo Pokemon jugador
13 class Pikachu
14 {
15 public:
16     string nombre = "Pikachu";
17     int vidaMaxima = 100;
18     int vida = vidaMaxima;
19     int danoMaxImpactrueno = 30;
20     int danoMinImpactrueno = 10;
21     int danoMaxAtaqueRapido = 20;
22     int danoMinAtaqueRapido = 5;
23     int danoMaxVoltioCruel = 50;
24     int danoMinVoltioCruel = 15;
25     int danoMaxSurf = 35;
26     int danoMinSurf = 10;
27 };
28
29 class Charmander
30 {
31 public:
32     string nombre = "Charmander";
33     int vidaMaxima = 100;
34     int vida = vidaMaxima;
35     int danoMaxPuncDinamico = 40;
36     int danoMinPuncDinamico = 15;
37     int danoMaxAscuas = 15;
38     int danoMinAscuas = 5;
39     int danoMaxNitroCarga = 60;
40     int danoMinNitroCarga = 30;
41     int danoMaxColmilloIgneio = 50;
42     int danoMinColmilloIgneio = 20;
43 }
```

PokemonBattles_Sim (Global Scope)

```
46 class Bulbasaur
47 {
48 public:
49     string nombre = "Bulbasaur";
50     int vidaMaxima = 100;
51     int vida = vidaMaxima;
52     int danoMaxLatigoCepa = 30;
53     int danoMinLatigoCepa = 10;
54     int danoMaxHojaAfilada = 40;
55     int danoMinHojaAfilada = 20;
56     int danoMaxBombaGermen = 60;
57     int danoMinBombaGermen = 30;
58 };
59
60 class Squirtle
61 {
62 public:
63     string nombre = "Squirtle";
64     int vidaMaxima = 100;
65     int vida = vidaMaxima;
66     int danoMaxBurbuja = 30;
67     int danoMinBurbuja = 5;
68     int danoMaxPistolaAgua = 20;
69     int danoMinPistolaAgua = 10;
70     int danoMaxMordisco = 40;
71     int danoMinMordisco = 20;
72     int danoMaxHidrobomba = 60;
73     int danoMinHidroBomba = 30;
74 };
75 //Equipo Pokemon rival
76 class Tauros { ... };
77
78
85
86 int main()
87 {
88     int rival = 0;
89     int pokemon = 0;
90     int ataque = 0;
91     int vida = 100;
92     srand(time(NULL));
93
94     // Equipo Pokemon del usuario
95     Pikachu Pik;
96     Charmander Charm;
```

```
PokemonBattles_Sim (Global)
99
100 // Equipo Pokemon del rival
101 Tauros Tau;
102
103 Pik.nombre = "Pikachu";
104 Tau.nombre = "Tauros";
105
106
107 cout << "Entrenador guay Alan te ha retado a un combate " << endl;
108 cout << "¿Que te gustaria hacer?" << endl;
109 cout << "1. Pelear" << endl;
110 cout << "2. Huir" << endl;
111 cin >> rival;
112
113 Sleep(600);
114 system("cls");
115
116 switch (rival)
117 {
118 case 1:
119 {
120 cout << "Entrenador guay Alan ha usado Tauros" << endl;
121 cout << "¿Cual Pokemon de tu equipo te gustaria usar?" << endl;
122 cout << "1. Pikachu" << endl;
123 cout << "2. Charmander" << endl;
124 cout << "3. Bulbasaur" << endl;
125 cout << "4. Squirtle" << endl;
126 cin >> pokemon;
127
128
129 switch (pokemon)
130 {
131 //Pikachu
132 case 1:
133 {
134 cout << "Pikachu ha entrado al combate" << endl;
135 cout << "Que ataque te gustaria usar?" << endl;
136 cout << "1. Impactrueno" << endl;
137 cout << "2. Ataque Rapido" << endl;
138 cout << "3. Voltio Cruel" << endl;
139 cout << "4. Surf" << endl;
140 cin >> ataque;
141
142 //Ataques Pikachu
143 switch (ataque)
144 {
145 //Impactrueno
146 case 1:
147 {
148
149 system("cls");
150 do
151 {
152 cout << "Pikachu ha usado impactrueno" << endl;
```

```
PokemonBattles_Sim (Global Scope)
153 // danoMax = 10 "dano minimo = 30
154 int dano = rand() % (Pik.danoMaxImpactrueno - Pik.danoMinImpactrueno + 1) + Pik.danoMinImpactrueno;
155 int danoTau = rand() % (Tau.danoMax - Tau.danoMin + 1) + Tau.danoMin;
156 Tau.vida -= dano;
157 Pik.vida -= danoTau;
158 if (dano > 30)
159 {
160     cout << "El ataque fue super efectivo Tauros ha recibido " << dano << " de dano" << endl;
161     cout << "La salud de Tauros es " << Tau.vida << endl;
162 }
163 else
164 {
165     cout << "Tauros ha recibido " << dano << " de dano" << endl;
166     cout << "La salud de Tauros es " << Tau.vida << endl;
167 }
168 Sleep(100);
169 cout << "Tauros ha contratacado haciendo " << danoTau << " de dano" << endl;
170 cout << "La salud de Pikachu es " << Pik.vida << endl;
171 } while ((Tau.vida >= 0) && (Pik.vida >= 0));
172 if (Tau.vida <= 0)
173 {
174     cout << "Tauros se ha debilitado" << endl;
175     cout << "Has ganado, Pikachu tiene 40 mas de exp" << endl;
176 }
177 else if (Pik.vida <= 0)
178 {
179     cout << "Pikachu, se ha debilitado" << endl;
180     cout << "Has salido corriendo al centro Pokemon y has perdido todas tus pokebolas" << endl;
181 }
182 break;
183 }
184 //Ataque Rapido
```

Observaciones Usando datos en una pagina de seguidores de Pokémon se puso ataques fuertes y ataques débiles, sin embargo el balanceo del random permitía que cualquiera de los 2 Pokémon en combate ganara sin dar ventaja a ninguno de los 2, el ejemplo se muestra como se programo el Impactrueno de Pikachu

Conclusiones Es entretenido tirar una moneda al aire para saber quien ganara pues los resultados son completamente random

Fuentes de consulta

- https://www.pokexperto.net/index2.php?seccion=nds/nationaldex/movimientos_nivel&pk=1