

Nombre: Angel Jacinto _____ No. de Matrícula.: __ZAP
534 _____

Materia: _Fundamentos de programación _____ Grupo: DSI 1 SEMESTRE Turno: M_____

Carrera: Ingeniería en desarrollo de software interactivos y videojuegos _____

Tema: proyecto 1 No: R.1

Fecha propuesta: sabado 4 de junio de 2022 Fecha de Entrega: miercoles 8 de junio de 2022

Escuela: Amerike Plantel GDL

Calle: _Montemorelos_____ No: 3503 Colonia: Rinconada de la Calma. _____C.P.: 45080_____

Teléfono: 3336326100

Ciudad: Guadalajara



Angel Jacinto

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.)	

```
#include <iostream>
using namespace std;
```

```
struct Player
{
    string name;
    int health;
    int attack;
    int defense;
    int vest;
    int Power_up;
};

void main()
{
    srand(time(nullptr));
    Player amber;
    amber.name = "amber the turd ";
    amber.health = 100;
    amber.attack = rand() % 10+1;
    amber.defense = rand() % -10-1;
    amber.vest = rand() % 10+1;
    amber.Power_up = rand() % 70+1;
    int amber_the_turd_total =(amber.attack + amber.defense + amber.vest + amber.Power_up);
    amber.health = (amber.health - amber_the_turd_total);
```

```
Player Jonny_Deep;
Jonny_Deep.name = "Jonny Deep";
Jonny_Deep.health = 100;
Jonny_Deep.attack = rand() % 10+1;
Jonny_Deep.defense = rand() % -10-1;
Jonny_Deep.vest = rand() % 10 + 1;
```

Nombre:
Práctica:

No. R-1
Página 2

```
Jonny_Deep.Power_up = rand() % 70+1;
int Jonny_total = (Jonny_Deep.attack + Jonny_Deep.defense + Jonny_Deep.vest +
Jonny_Deep.Power_up);
Jonny_Deep.health = (Jonny_Deep.health - Jonny_total);
```

```
Player camile;
camile.name = "Camile Vazques";
camile.health = 100;
camile.attack = rand() % 10+1;
camile.defense = rand() % -10-1;
camile.vest = rand() % 10 + 1;
camile.Power_up = rand() % 70+1;
int camile_total = (camile.attack + camile.defense + camile.vest + camile.Power_up);
camile.health = (camile.health - camile_total);
```

```
Player judge;
judge.name = "Judge";
judge.health = 100;
judge.attack = rand() % 10+1;
judge.defense = rand() % 10-1;
judge.vest = rand() % 10 + 1;
judge.Power_up = rand() % 70+1;
int judge_total = (judge.attack + judge.defense + judge.vest + judge.Power_up);
judge.health = (judge.health - judge_total);
```

```
cout << "---PLAYERS---: " << endl << endl;
cout << amber.name << endl;
cout << Jonny_Deep.name << endl;
cout << camile.name << endl;
cout << judge.name << endl;
cout << endl << endl;
cout << "---SCORE BOARD---" << endl << endl;
cout << amber.name << " : " << amber_the_turd_total << endl;
cout << Jonny_Deep.name << " : " << Jonny_total << endl;
cout << camile.name << " : " << camile_total << endl;
cout << judge.name << " : " << judge_total << endl;
cout << endl << endl;
```

```
cout << "---RESULTS---" << endl << endl;
// ATTACKS//
if (amber.health < Jonny_Deep.health && amber.health < camile.health && amber.health <
judge.health)
{
cout << amber.name << ": [WIN]" << endl;
cout << Jonny_Deep.name << ": [lose]" << endl;
cout << camile.name << ": [lose]" << endl;
```

```
cout << judge.name << ": [lose]" << endl;
}
if (Jonny_Deep.health < amber.health && Jonny_Deep.health < camile.health &&
Jonny_Deep.health < judge.health)
{
cout << Jonny_Deep.name << ": [WIN]" << endl;
cout << amber.name << ": [lose]" << endl;
cout << camile.name << ": [lose]" << endl;
cout << judge.name << ": [lose]" << endl;
}
if (camile.health < amber.health && camile.health < Jonny_Deep.health && camile.health <
judge.health)
{
cout << camile.name << ": [WIN]" << endl;
cout << amber.name << ": [lose]" << endl;
cout << Jonny_Deep.name << ": [lose]" << endl;
cout << judge.name << ": [lose]" << endl;
}
if (judge.health < amber.health && judge.health < camile.health && judge.health <
Jonny_Deep.health)
{
cout << judge.name << ": [WIN]" << endl;
cout << amber.name << ": [lose]" << endl;
cout << camile.name << ": [lose]" << endl;
cout << Jonny_Deep.name << ": [lose]" << endl;
}
}
```

Observaciones

Se crean variables, propiedades de los jugadores donde tales propiedades se ponen a prueba, se siguen las indicaciones:

Utilizar por lo menos 4 propiedades para realizar el enfrentamiento.

La asignación de propiedades se hará mediante un random de 0 a 10.

Los contendientes tendrán que compararse contra los demás para saber quien es el mayor y menor de las propiedades.

Para darle un cambio a las cosas, existirá un enumerador con powerup el cual mediante un random asignará un extra a los valores.

Y al final se imprime quien fue el vencedor

Conclusiones

Las propiedades son de 0 a 10

El power up es de 70 para darle un gran cambio al azar.

Se ejecutan los valores de las propiedades y se proyecta a el vencedor.