

## Práctica Unidad ½

### 1. Galaga

Seguramente conoce el Galaga, un juego clásico de naves. Defina las clases y los miembros que utilizaría si fuese a implementar un clon del mismo.

### 2. Pac-Man

¿Sería posible realizar un diseño de clases para un clon del Pac-Man®? Proponga uno. Implemente los métodos y atributos que crea necesarios

### 3. FPS

Defina una clase que modele al personaje principal de un FPS. Incluya todos los atributos y métodos que crea necesarios para su funcionamiento.

### 4. Plataformas

Realice la misma tarea que en el ejercicio anterior, pero esta vez modele un personaje de un videojuego de plataformas al estilo de Mario Bros® dónde el personaje principal cuenta con sólo un punto de vida, posee una lista de hechizos aprendidos y tiene un atributo que indica si se encuentra cansado.

### 5. Star cough!!

En este ejercicio proponga un diseño de clases para un videojuego de estrategia en el cual existen dos bandos (Docentes y Alumnos). Y en cada bando hay dos unidades. Una de ellas ataca lanzando libros y la otra simplemente cura a los heridos. Las unidades en los dos bandos son muy similares. La diferencia radica en el color de las mismas.

### 6. Heroes

Dada la siguiente clase

```
class Heroe{
private:
    int salud;
    int x,y; //posicion en el mundo
};
```

Realice los cambios que crea necesarios para que la misma cumpla con los siguientes requerimientos:

- El personaje comienza siempre en la posición x=0, y=0
- La salud inicial del personaje es de 500 puntos
- Las demás clases de la aplicación deben poder leer y modificar la salud del heroe.
- El héroe debe contener una lista con los nombres de las princesas que ha rescatado.  
Tenga en cuenta la alocaación de memoria (dinámica o estática y el impacto que esto tiene en los constructores/destructores).
- Implemente todos los métodos necesarios para el funcionamiento de la clase.
- Realice un programa de ejemplo que pruebe la clase( una simple aplicación de consola)

## 7. ¿No funciona?

¿Porqué el siguiente fragmento de código no funciona?

```
class Hechizo{
    int costo;
    void setCosto(int &val){costo=val;}
    int getCosto(){return costo;}
};

int _tmain(int argc, _TCHAR* argv[])
{
    Hechizo h;
    h.setCosto(10);
    std::cout<<h.getCosto();
    return 0;
}
```

## 8. Salidas

Dadas las siguientes definiciones de clase. Indique si es correcto, y de ser así, que mostrará el programa en cada caso:

```
class Unidad{
public:
    virtual void showClass(){cout<< "Unidad Genérica";}
};

class Soldado: public Unidad{
public:
    virtual void showClass(){cout<< "Soldado";}
};

class Granadero: public Unidad{
public:
    virtual void showClass(){cout<< "Granadero";}
};

class Ingeniero: public Unidad{
public:
```

```
virtual void showClass(){cout<< "Ingeniero";}
};
```

a.

```
Soldado *unidad1;
unidad1= new Soldado();
unidad1->showClass();
```

b.

```
Unidad *unidad1;
unidad1= new Soldado();
unidad1->showClass();
```

c.

```
Soldado *unidad1;
unidad1= new Unidad();
unidad1->showClass();
```

d.

```
Soldado *unidad1;
unidad1= new Soldado();
unidad1= new Ingeniero();
unidad1->showClass();
unidad1->showClass();
```

e.

```
Soldado *unidad1;
Ingeniero* unidad2;
unidad1= new Soldado();
unidad2= new Ingeniero();
unidad1->showClass();
unidad2->showClass();
```

f.

```
Soldado *unidad1;
Ingeniero* unidad2;
unidad1= new Soldado();
unidad2= new Ingeniero();
Unidad* unidad3;
unidad3= unidad1;
unidad3->showClass();
```

g.

```
Soldado *unidad1;
Ingeniero* unidad2;
unidad1= new Soldado();
unidad2= new Ingeniero();
```

```
unidad1=unidad2;
```

¿Si no se hubiese utilizado “virtual” el resultado hubiese sido el mismo?

## 9. Consola

Implemente una clase que se encargue de manipular la interacción con el usuario. Es decir, brindar información por consola y recibir los comandos de la misma. La clase debe poseer un método para leer comandos y valores por consola y métodos para solicitar dicha entrada. Considere el siguiente ejemplo de uso:

```
int _tmain(int argc, _TCHAR* argv[])
{
    ConsoleManager manager;
    int jugadores;
    float energía;
    manager.mostrarMensaje("Ingrese la cantidad de jugadores");
    jugadores=manager.leerEntero();
    manager.mostrarMensaje("Ingrese la cantidad de energía");
    energía = manager.leerFlotante();
    manager.mostrarMensaje("Esperamos haya disfrutado perder su tiempo, le
                           agradecemos interactuar con nosotros");

    manager.pausa();
    return 0;
}
```

Dicho programa pide al usuario la cantidad de jugadores y energia de ellos, luego muestra un mensaje de agradecimiento y espera que el usuario presione una tecla para salir. Lo que se solicita es que Ud. implemente la clase ConsoleManager. Es común que existan este tipo de clases en las librerías para videojuegos para simplificar la interacción con el usuario (por supuesto que mucho más completas que nuestro caso académico).