



FRANCISCO DE LOS RÍOS
INSTITUTO DE EDUCACIÓN SECUNDARIA

50
Aniversario

MEDIEVAL GAME



Asignatura: Entornos de Desarrollo

Curso: 1º CSFP Desarrollo de aplicaciones multiplataforma

Estudiantes: Juan Miguel, Alejandro, Antonio, Pedro, Raul

Profesor: Miguel Morales Carmona

Fecha: 17 de febrero de 2022

Resumen

Diseñar en editor Dia las clases que cada uno tenga asignadas. Cada uno debe subir al GitHub un único archivo .dia con todas sus clases en él, con el formato nombreDiagramas.dia ; Ejemplo: JuanMiguelDiagramas.dia

Así mismo las subidas se realizaran a través de git mediante claves ssh y no de forma manual. Para ello debeis crear una clave ssh en cada ordenador que useis y añadirlo a vuestra cuenta de github.

Enlace a Github: <https://github.com/juanmirivas8/MedievalGame>

Instrucciones:

1. git init
2. git remote add *origin* git@github.com:juanmirivas8/MedievalGame.git

Índice

| | |
|---|----------|
| 1. Antonio | 2 |
| 1.1. Clase Ciudad | 2 |
| 1.1.1. Atributos | 2 |
| 1.1.2. Metodos | 2 |
| 2. Alejandro | 3 |
| 2.1. Clase Poblacion | 3 |
| 2.1.1. Atributos | 3 |
| 2.1.2. Metodos | 3 |
| 3. Raul | 3 |
| 3.1. Clase Deuda | 3 |
| 3.1.1. Atributos | 3 |
| 3.1.2. Metodos | 3 |
| 3.2. Clase Edificio (Abstracto) | 4 |
| 3.2.1. Atributos | 4 |
| 3.2.2. Metodos | 4 |
| 3.3. Clase Controlador | 4 |
| 3.3.1. Atributos | 4 |
| 3.3.2. Metodos | 4 |
| 4. Pedro | 4 |
| 4.1. Clase Privilegio (Abstracta) | 4 |
| 4.1.1. Atributos | 4 |
| 4.1.2. Metodos | 4 |
| 4.2. Clase Evento (Abstracta) | 5 |
| 4.2.1. Atributos | 5 |
| 4.2.2. Metodos | 5 |
| 4.3. Clase Vista | 5 |
| 4.3.1. Metodos | 5 |
| 5. Juan Miguel | 5 |
| 5.1. Clase | 5 |
| 5.1.1. Atributos | 5 |
| 5.1.2. Metodos | 5 |

1. Antonio

1.1. Clase Ciudad

Esta clase representa a una ciudad y contiene todos los atributos y métodos necesarios para modelarla

1.1.1. Atributos

- `private Double ingresos`
- `private Double oro`
- `private ArrayList<Deuda> listaDeudas`
- `private Double inflacion`
- `private Integer estabilidad`
- `private Double agitacion`
- `private Integer soldadesca`
- `private PuebloHumano humanos`
- `private Poblacion humanos`
- `private Poblacion nohumanos`
- `private Double mantenimientoEjercito`
- `private Double mantenimientoAdministracion`
- `private ArrayList<Privilegio> listaPolíticas`
- `private ArrayList<Edificio> listaEdificios`
- `private ArrayList<Evento> eventosActivos`
- `private Double modificadorImpuestos`
- `private Double modificadorSoldadesca`
- `private Double corrupcion`
- `private Double propiedadPublica`

1.1.2. Metodos

- `public getGastos()`
- `public getRecaudacion()`
- `public getIngresos()`
- Resto de getters y setters publicos

2. Alejandro

2.1. Clase Poblacion

Clase que representa un estamento y contiene todos los atributos y métodos para manejarlos.

2.1.1. Atributos

- `protected String nombre`
- `protected Integer Cantidad`
- `protected Double influencia`
- `protected Double propiedadPrivada`
- `protected Double modificadorImpuestos`
- `protected Double modificadorSoldadesca`
- `protected Integer maximaSoldadesca`
- `protected Double crecimiento`
- `private ArrayList<Privilegio> listaPrivilegios`

2.1.2. Metodos

- Getters y setters publicos

3. Raul

Ésta clase simula un préstamo que se pide. Tiene un interés, el cual determina cuanto se paga al final y cada turno.

3.1. Clase Deuda

3.1.1. Atributos

- `private Double importe`
- `private Double interes`

3.1.2. Metodos

- `public Double getImporte()`
- `public void setImporte()`
- `public Double getInteres()`
- `public Double setInteres()`

Clase abstracta la cual será implementada por diferentes tipos de edificios, los cuales tendrán distintos efectos y costes.

3.2. Clase Edificio (Abstracto)

3.2.1. Atributos

- `private Double precio`
- `private Double precioVenta`
- `private String Descripcion`

3.2.2. Metodos

- `public String aplicarEfecto(Ciudad c)`
- `public String revertirEfecto(Ciudad c)`
- Getters y setters de los otros atributos

3.3. Clase Controlador

Controlador del programa que coordina la vista con el modelo, el cual es una clase de tipo partida.

3.3.1. Atributos

- `private Vista v`
- `private Partida p`

3.3.2. Metodos

- `public void run()`
- metodos privados

4. Pedro

Clase Privilegio, el cual ofrece una serie de beneficios a aquella entidad que lo posea, ya sea el gobernante o alguna población.

4.1. Clase Privilegio (Abstracta)

4.1.1. Atributos

- `private String Descripcion`

4.1.2. Metodos

- `public String aplicarEfecto(Ciudad c)`
- `public String revertirEfecto(c)`
- Resto de getters y setters

4.2. Clase Evento (Abstracta)

Clase abstracta que luego será implementado por diferentes eventos, tanto positivos como negativos. Los eventos pueden ser lanzados aleatoriamente o también si se cumplen una serie de condiciones.

4.2.1. Atributos

- `private Integer turnosRestantes`
- `private String Descripcion`

4.2.2. Metodos

- `public String aplicarEvento(Ciudad c)`
- `public String revertirEvento(c)`
- Getters y setters

4.3. Clase Vista

La clase vista contendrá multitud de métodos de entrada y salida para la correcta interacción del programa con el usuario.

4.3.1. Metodos

- Metodos de lectura y escritura

5. Juan Miguel

5.1. Clase

5.1.1. Atributos

- `private Ciudad c`
- `private ArrayList<Evento> listaEventos`

5.1.2. Metodos

- **Multitud eventos funcionamiento:** pagarDeuda, lanzarEventoAleatorio, darPrivilegio, etc