|  |  |  |
| --- | --- | --- |
| Texto  Descripción generada automáticamente con confianza media | TECNICATURA UNIVERSITARIA EN DISEÑO INTEGRAL DE VIDEOJUEGOS  FACULTAD DE INGENIERÍA  Universidad Nacional de Jujuy |  |

Trabajo Práctico

N°2

Tactaca, Cecilia Yazmin – LU TUV000764

*Profesores:*

*Mg. Ing. Ariel Alejandro Vega*

*Año 2024*

**FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS**

Índice

[Punto 1: 3](#_Toc166532640)

[Desarrollo del ejercicio 1: 3](#_Toc166532641)

[Fase de análisis: 3](#_Toc166532642)

[Fase de diseño: 3](#_Toc166532643)

[Punto 2: 3](#_Toc166532644)

[Fase de análisis: 4](#_Toc166532645)

[Fase de desarrollo: 4](#_Toc166532646)

[Codificación: 4](#_Toc166532647)

[Punto 3: 4](#_Toc166532648)

[Fase de análisis: 5](#_Toc166532649)

[Fase de desarrollo: 5](#_Toc166532650)

[Codificación: 5](#_Toc166532651)

[Punto 4: 5](#_Toc166532652)

[Fase de análisis: 5](#_Toc166532653)

[Fase de desarrollo: 5](#_Toc166532654)

[Codificación: 5](#_Toc166532655)

[Fuentes bibliográficas 6](#_Toc166532656)

Punto 1: Desarrolle una historia de usuario, en la cual defina la visualización y movimiento de una clase GameObject, de la que heredan Shooter y Asteroide. GameObjects es abstracta, y posee atributos protegidos: posición, imagen; además del método abstracto display() y mover(). Además, debe poseer un HUD que visualice la cantidad de vidas del Shooter. Utilice un JoyPad para generar los movimientos.

## Desarrollo del ejercicio 1:

### Fase de análisis:

**TARJETA DE HISTORIA DE USUARIO**

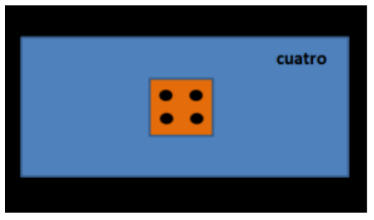
|  |  |
| --- | --- |
| HISTORIA DE USUARIO | |
| **Código:** HU001 | **Usuario:** Jugador |
| **Nombre de historia de usuario:** Construcción de escenario y ubicación de GameObject. | |
| **Prioridad:** Alta | **Riesgo de desarrollo:** Alta |
| **Estimación:** 1 hora | **Iteración asignada:** 1 |
| **Responsable:** Tactaca, Cecilia Y. | |
| **Descripción:** Como jugador, quiero poder ver en pantalla la ubicación y el movimiento del Shooter y el Asteroide, además poder visualizar un texto que me diga cuantas vidas me quedan; esto me ayudara a la hora de desarrollar una estrategia de juego. | |
| **Criterios de aceptación:**   * El jugador puede mover la nave Shooter, que tiene movimientos de izquierda a derecha. * La nave se encuentra inicialmente en la parte inferior de la pantalla, y en el centro. * El control de los movimientos de Shooter se realizan con el JoyPad. * Los Asteroides tienen un movimiento de arriba hacia abajo, desaparecen al salir de pantalla. * Los Asteroides aparecen en la parte superior de la pantalla en ubicaciones aleatorias. * El HUD permite visualizar la cantidad de vidas del Shooter. Tanto la cantidad inicial y como las va perdiendo. * El Shooter pierde vidas si es impactado por un Asteroide. | |
| **Observaciones:**  En este modelo no se ha considerado el fondo en el juego. | |

### Fase de diseño:

**ESTRUCTURA DE APLICACIÓN**

Punto 2: Desarrolle un videojuego que cumpla con las siguientes especificaciones:

Realice un diagrama de clases



Como se observa se trata de un dado. El cual al presionar un botón debe generar un número aleatorio entre 1 y 6 y dibujarlo. Además, debe mostrar el número en la parte superior derecha. Repetir esto cuantas veces lo desee y al finalizar (con otro botón) debe dibujar por consola y agrupado en filas de 4 columnas los dados obtenidos.

Al momento de programar utilice constructores sobrecargados. Considere que el dado se muestra en un tablero, este tablero contiene al dado, y al texto.

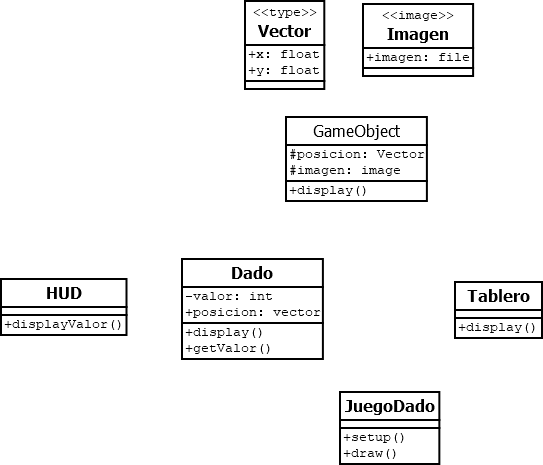
Además, almacene cada dado obtenido en un arreglo. Considere aplicar la herencia respecto de que existe una clase abstracta padre GameObject, de la que hereda la posición y el método abstracto display(). Luego recrear otra versión donde use imágenes en lugar de dibujar con las primitivas.

### Fase de análisis:

**TARJETA DE HISTORIA DE USUARIO**

|  |  |
| --- | --- |
| HISTORIA DE USUARIO | |
| **Código:** HU002 | **Usuario:** Jugador |
| **Nombre de historia de usuario:** Construcción de escenario y ubicación de GameObject. | |
| **Prioridad:** Alta | **Riesgo de desarrollo:** Alta |
| **Estimación:** 1 hora | **Iteración asignada:** 1 |
| **Responsable:** Tactaca, Cecilia Y. | |
| **Descripción:** Como jugador, quiero poder ver en pantalla la ubicación y el valor del dado en cada tiro, y al finalizar los valores que se obtuvieron en la partida. | |
| **Criterios de aceptación:**   * El jugador tira el Dado al presionar una tecla, la barra espaciadora; el número generado (valor del dado) es aleatoria y es un entero entre 1 y 6. * El jugador termina el juego al presionar otra tecla, la tecla enter. * Se visualiza en pantalla los últimos Dados tirados. * El HUD se puede ver en la parte derecha superior de la pantalla con el valor del Dado escrito en letras. | |
| **Observaciones:** | |

### Fase de desarrollo:



### Codificación:

Punto 3: Realice el modelado de las clases que intervienen en el juego frogger a partir de la Fig. 1. Realice la construcción de las clases en processing. El juego debe llegar a poder mostrar en pantalla la visualización de los diferentes objetos modelados. Utilice herencia y encapsulamiento para los vehículos. Además, los vehículos deben guardarse en una lista de objetos que es atributo de la clase SpawnerVehiculos.



Figura 1. Modelo juego Fogger

### Fase de análisis:

**TARJETA DE HISTORIA DE USUARIO**

|  |  |
| --- | --- |
| HISTORIA DE USUARIO | |
| **Código:** HU003 | **Usuario:** Jugador |
| **Nombre de historia de usuario:** Construcción de escenario y ubicación de GameObject. | |
| **Prioridad:** Alta | **Riesgo de desarrollo:** Alta |
| **Estimación:** 1 hora | **Iteración asignada:** 1 |
| **Responsable:** Tactaca, Cecilia Y. | |
| **Descripción:** Como jugador, quiero | |
| **Criterios de aceptación:**   * El jugador | |
| **Observaciones:**  El modelo no considera el fondo del juego ni el comportamiento de los GameObjects cuando salen de pantalla. | |

### Fase de desarrollo:

### Codificación:

Punto 4: Considere programar un juego de naves. Debe usar imágenes para las naves, los asteroides y los enemigos. Aplique herencia. Use una interface denominada IDisplayable que tenga el método display(). Defina dos interfaces más: IMoveable que tenga el método mover() y Otra IControler que tenga el método readCommand();

Usando el sentido común haga que las clases Nave, Asteroid y Enemy implementen las interfaces correspondientes. Finalmente use la dependencia para que la nave dispare balas que serán almacenadas en una lista de balas. Las balas se deben destruir cuando salen de pantalla.

### Fase de análisis:

**TARJETA DE HISTORIA DE USUARIO**

|  |  |
| --- | --- |
| HISTORIA DE USUARIO | |
| **Código:** HU004 | **Usuario:** Jugador |
| **Nombre de historia de usuario:** Construcción de escenario y ubicación de GameObject. | |
| **Prioridad:** Alta | **Riesgo de desarrollo:** Alta |
| **Estimación:** 1 hora | **Iteración asignada:** 1 |
| **Responsable:** Tactaca, Cecilia Y. | |
|  | |
| **Criterios de aceptación:**   * El jugador | |
| **Observaciones:** | |

### Fase de desarrollo:

### Codificación:

# Fuentes bibliográficas

* Mg. Ing. Vega, A. DISEÑO DE ALGORITMOS: Expresiones aritméticas. Facultad de Ingeniería. Universidad Nacional de Jujuy.
* <https://www.youtube.com/watch?v=gQgX-YbRJAw>
* <https://www.youtube.com/watch?v=qzg8uK4xmzI>