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

*Profesores:*

*Mg. Ing. Ariel Alejandro Vega*

*Ing. Carolina Cecilia Apaza*

*Año*

Trabajo Práctico/Actividad

N°

Apellido y Nombre – LU /

Tarifa, Juan Mateo-

TUV000732

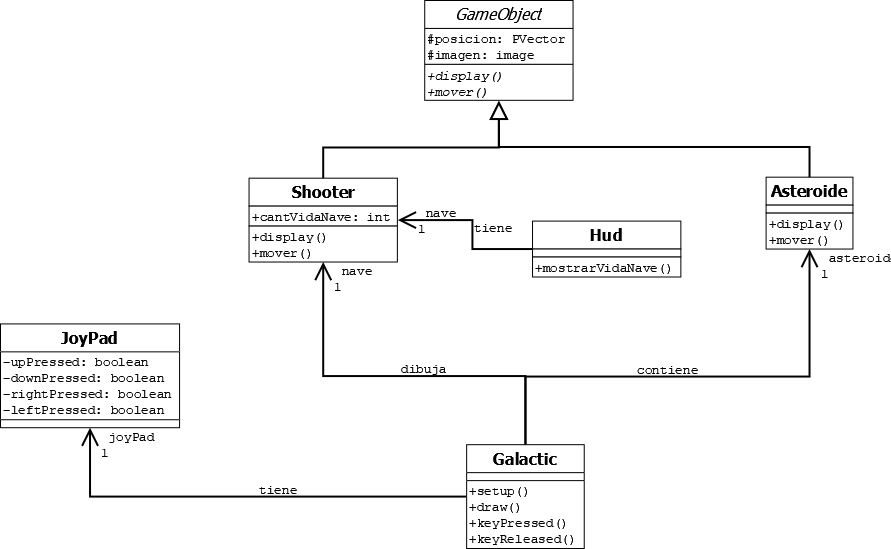
**FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS**

Indice

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.

HISTORIA DE USUARIO

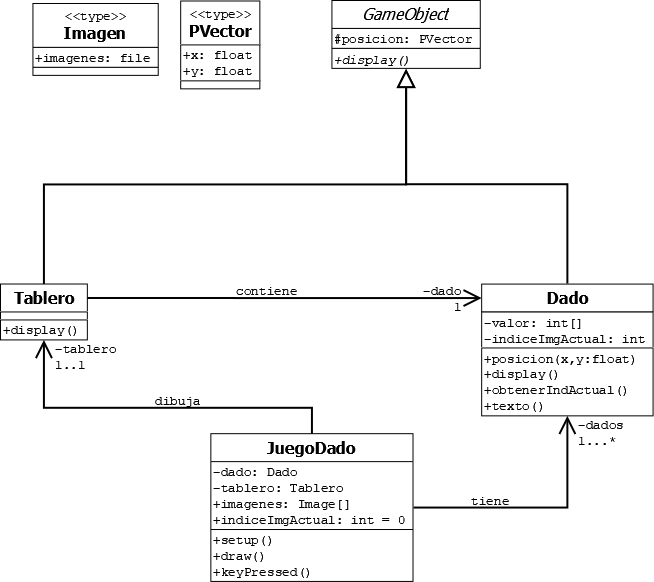
|  |  |
| --- | --- |
| HISTORIA DE USUARIO | |
| |  | | --- | | **Usuario**: Jugador |   **Código**: HU001 | |
| **Nombre de historia de Usuario:** Construcción de GameObjects y visualización de un Hud | |
| **Prioridad:** Alta | **Riesgo de desarrollo**: Alto |
| **Estimación**: 1 a 2 horas | **Iteración asignada**: 1 |
| **Responsable:** Mateo Tarifa | |
| **Descripción**: *Como jugador quiero poder observar un asteroide como obstáculo junto a mi nave que se pueda mover esquivando estos y observar la cantidad de vidas que tengo en pantalla, para poder planear mi estrategia de juego.* | |
| **Criterios de Aceptación:**  Visualización de los GameObjects para poder interactuar con ellos.  El asteroide debe ser un obstáculo representado visualmente en pantalla.  La Nave se puede mover en 8 direcciones.  El jugador debe poder controlar el movimiento de la Nave para esquivar los asteroides y lograr una experiencia de juego gratificante.  Se debe de mostrar en pantalla la cantidad de vidas que posee la nave para que el jugador plantee una estrategia.  Se ha de crear un HUD que represente esa cantidad de vidas. | |
| Observaciones:  En este modelo no se busca la jugabilidad, se busca visualizar las clases que se utilizan para crear cada objeto, una de ellas, el GameObject, de donde la nave y el asteroide recibirán por herencia los atributos y métodos de la Superclase, también se busca mostrar las vidas por una clase HUD y el movimiento de la nave con una clase JoyPad. | |

DIAGRAMA

Punto 2: 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.

DIAGRAMA DE CLASES

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.

DIAGRAMA DE CLASES

Conclusión

Párrafos de las conclusiones

Fuentes bibliográficas

Se deben enunciar las fuentes (apuntes de la materia, páginas web, videos de youtube, libro (nombre, autores, año), etc)