**Trabajo Práctico Nº2**

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. GameObject 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. Utilce un JoyPad para generar los movimientos.

|  |  |  |
| --- | --- | --- |
| **HISTORIA DE USUARIO** | | |
| **Código:** FT0516 | **Usuario:** Jugador | |
| **Nombre de Historia de Usuario:** Construcción del escenario y ubicación de GameObject | | |
| **Prioridad:** Alta | | **Riesgo de desarrollo:** Alto |
| **Estimación:** días | | **Iteración asignada:** 1 |
| **Responsable:** Milagros Flores | | |
| **Descripción:** Como jugador, quiero observar el escenario, ubicación y movimientos del shooter y el asteroide. Sumando la cantidad de vida del shooter. | | |
| **Criterios de aceptación:**  El jugador se mueve en las 8 direcciones  El asteroide se mueve horizontalmente    El HUD debe visualizar la cantidad de vida del jugador | | |
| **Observaciones:** Este modelo no considera el disparo del shooter. | | |

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.

|  |  |  |
| --- | --- | --- |
| **HISTORIA DE USUARIO** | | |
| **Código:** FT0517 | **Usuario:** Jugador | |
| **Nombre de Historia de Usuario:** Construcción de un tablero y generación aleatoria del número de dados. | | |
| **Prioridad:** Alta | | **Riesgo de desarrollo:** Alto |
| **Estimación:** días | | **Iteración asignada:** 1 |
| **Responsable:** Milagros Flores | | |
| **Descripción:** Como jugador, quiero observar el tablero y el dado con su ubicación y que al presionar un botón me genere una aleatoriedad de las caras del dado y con otro botón poder detener el mismo para obtener el número que me ha tocado al azar. | | |
| **Criterios de aceptación:**  El jugador debe visualizar el tablero con el dado  El dado debe mostrar sus caras aleatoriamente    El HUD debe visualizar el número que corresponde a la cara del dado cuando este está detenido | | |
| **Observaciones:** | | |

Estructura de aplicación

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.

|  |  |  |
| --- | --- | --- |
| **HISTORIA DE USUARIO** | | |
| **Código:** FT0518 | **Usuario:** Jugador | |
| **Nombre de Historia de Usuario:** Construcción de un escenario con diferentes objetos inspirado en el juego Frogger. | | |
| **Prioridad:** Alta | | **Riesgo de desarrollo:** Alto |
| **Estimación:** días | | **Iteración asignada:** 1 |
| **Responsable:** Milagros Flores | | |
| **Descripción:** Como jugador, quiero observar los diferentes objetos del juego Frogger con sus respectivas ubicaciones y posiciones. | | |
| **Criterios de aceptación:**  El jugador debe visualizar los diferentes objetos | | |
| **Observaciones:** | | |

Estructura de aplicación

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.

|  |  |  |
| --- | --- | --- |
| **HISTORIA DE USUARIO** | | |
| **Código:** FT0519 | **Usuario:** Jugador | |
| **Nombre de Historia de Usuario:** | | |
| **Prioridad:** Alta | | **Riesgo de desarrollo:** Alto |
| **Estimación:** días | | **Iteración asignada:** 1 |
| **Responsable:** Milagros Flores | | |
| **Descripción:** | | |
| **Criterios de aceptación:** | | |
| **Observaciones:** | | |

Estructura de aplicación