



TECNICATURA UNIVERSITARIA EN DISEÑO
INTEGRAL DE VIDEOJUEGOS

FACULTAD DE INGENIERÍA
Universidad Nacional de Jujuy



FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS

TRABAJO PRÁCTICO/ACTIVIDAD

Nº

APELLIDO Y NOMBRE – LU /

VALENTINA ROCIO ANAHI BEJARANO CACERES-



TUV000736

PROFESORES:



MG. ING. ARIEL ALEJANDRO VEGA

ING. CAROLINA CECILIA APAZA

AÑO

| | | |
|---|--|---|
|  | <p>FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS TECNICATURA UNIVERSITARIA EN DISEÑO INTEGRAL DE VIDEOJUEGOS FACULTAD DE INGENIERÍA Universidad Nacional de Jujuy PENSAMIENTO COMPUTACIONAL y PROGRAMACIÓN: Problema y Solución – PC y P – Algoritmos – Principio de la P</p> |  |
|---|--|---|

INDICE

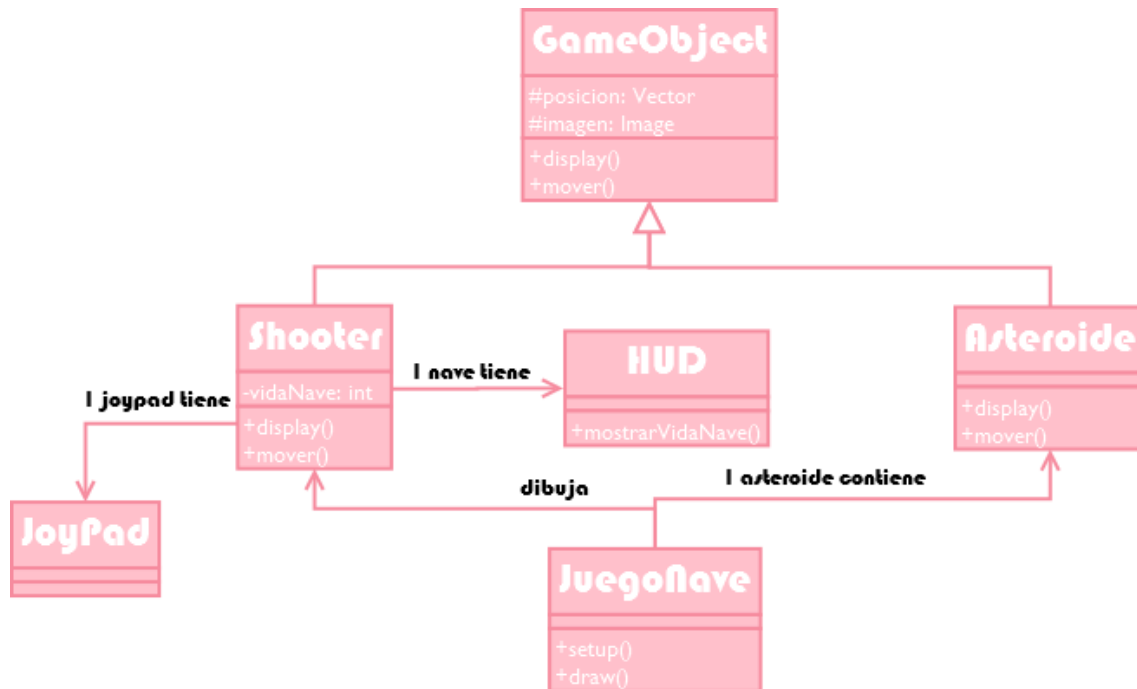
| | | |
|---|---|---|
|  | <p>FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS TECNICATURA UNIVERSITARIA EN DISEÑO INTEGRAL DE VIDEOJUEGOS FACULTAD DE INGENIERÍA Universidad Nacional de Jujuy PENSAMIENTO COMPUTACIONAL y PROGRAMACIÓN: Problema y Solución – PC y P – Algoritmos – Principio de la P</p> |  |
|---|---|---|

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 | |
|--|-----------------------------------|
| CÓDIGO: HU001 | USUARIO: JUGADOR |
| NOMBRE DE HISTORIA DE USUARIO: CONSTRUCCIÓN DE GAMEOBJECTS Y VISUALIZACIÓN DE UN HUD | |
| PRIORIDAD: ALTA | RIESGO DE DESARROLLO: ALTO |
| ESTIMACIÓN: 2 A 3 HORAS | ITERACIÓN ASIGNADA: 1 |
| RESPONSABLE: VALENTINA BEJARANO | |
| DESCRIPCIÓN: COMO JUGADOR, QUIERO PODER VER UN ASTEROIDE QUE ACTÚE COMO OBSTÁCULO CERCA DE MI NAVE. NECESITO PODER MOVERME PARA ESQUIVARLO Y TAMBIÉN VER CLARAMENTE CUÁNTAS VIDAS ME QUEDAN EN LA PANTALLA, PARA ASÍ PLANIFICAR MEJOR MIS MOVIMIENTOS EN EL JUEGO. | |
| CRITERIOS DE ACEPTACIÓN: LA NAVE TIENE QUE PODER SER FÁCILMENTE MANIPULADA POR EL JUGADOR QUE SEA POSIBLE LA VISUALIZACIÓN E INTERACCION DE LOS GAMEOBJECTS PARA LLEVAR RECuento DE LAS VIDAS SE DEBE DE CREAR UN HUD EN LA PANTALLA SE DEBE DE MOSTAR UN CONTADOR PARA TENER EN CUENTA LA CANTIDAD DE VIDAS EN LA PANTALLA SE REPRESENTAR GRÁFICAMENTE AL ASTEROIDE QUE LA NAVE TENGA MOVIMIENTO EN 8 DIRECCIONES DIFERENTES | |
| OBSERVACIONES: EN ESTE MODELO, EL ENFOQUE NO ESTÁ EN LA JUGABILIDAD SINO EN MOSTRAR LAS CLASES UTILIZADAS PARA CREAR CADA OBJETO. POR EJEMPLO, TANTO LA NAVE COMO EL ASTEROIDE HEREDAN ATRIBUTOS Y MÉTODOS DE LA SUPERCLASE GAMEOBJECT. ADEMÁS, SE BUSCA VISUALIZAR LAS VIDAS A TRAVÉS DE UNA CLASE HUD Y CONTROLAR EL MOVIMIENTO DE LA NAVE MEDIANTE UNA CLASE LLAMADA 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.

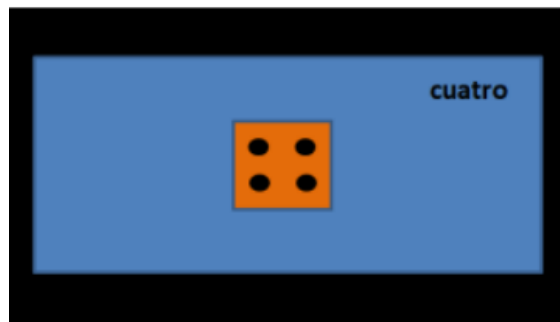
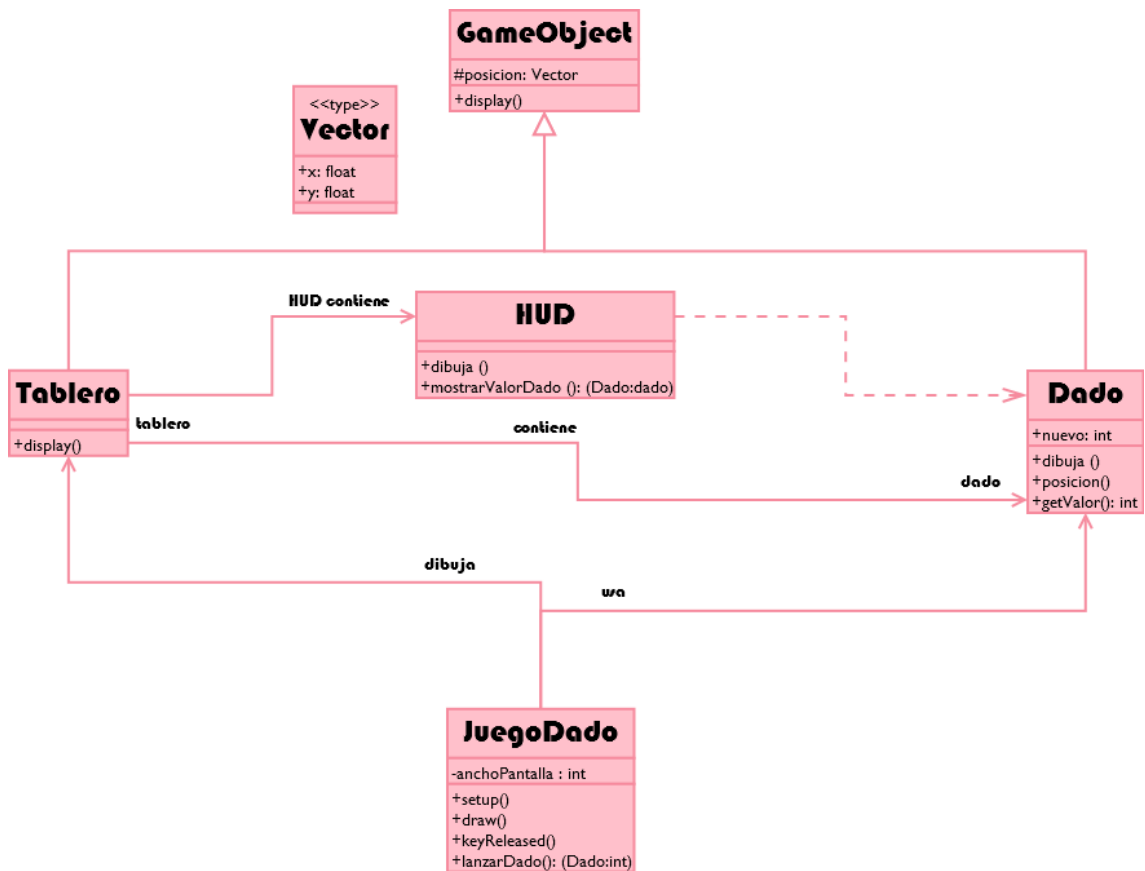


DIAGRAMA DE CLASES

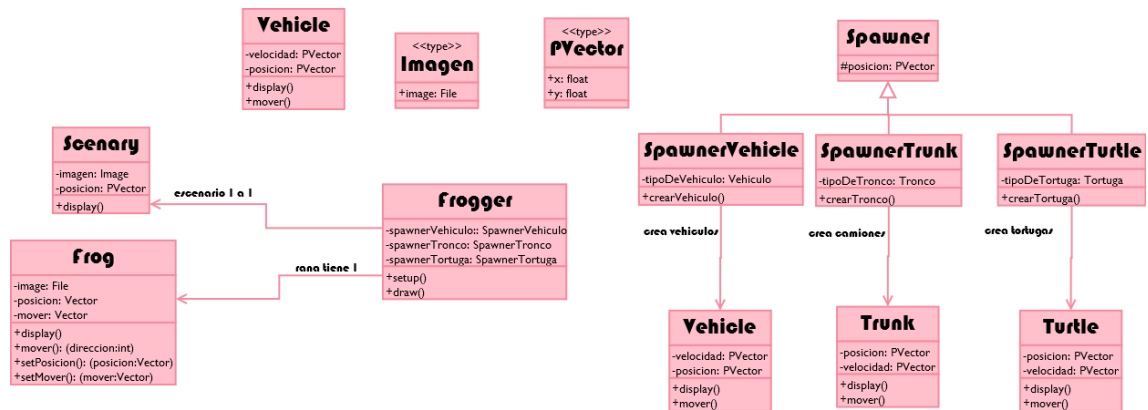


PUNTO 3: REALICE EL MODELADO DE LAS CLASES QUE INTERVIENEN EN EL JUEGO FROGGER A PARTIR DE LA FIG. 1. REALICE LA CONSTRUCCIÓN DE LAS CLASES EN PROCESSING. EL JUEGO DEBE LLEGAR A PODER MOSTRAR EN PANTALLA LA VISUALIZACIÓN DE LOS DIFERENTES OBJETOS MODELADOS. UTILICE HERENCIA Y ENCAPSULAMIENTO PARA LOS VEHÍCULOS. ADEMÁS, LOS VEHÍCULOS DEBEN GUARDARSE EN UNA LISTA DE OBJETOS QUE ES ATRIBUTO DE LA CLASE SPAWNERVEHICULOS.





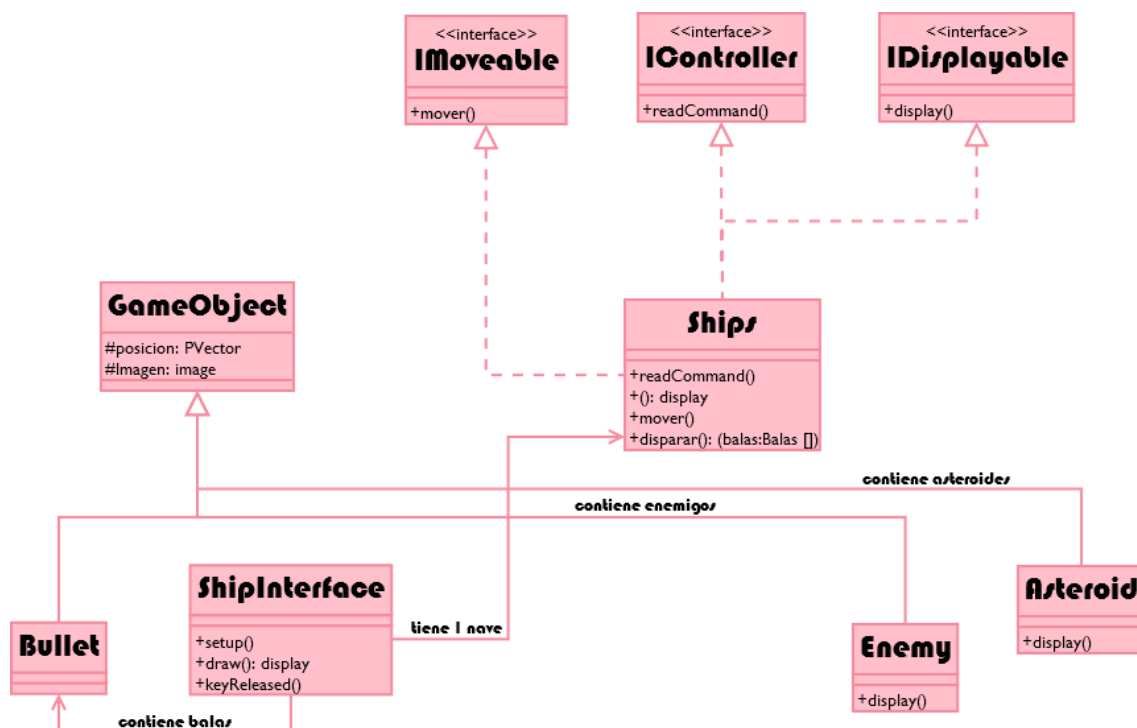
DIAGRAMA DE CLASES





PUNTO 4: CONSIDERE PROGRAMAR UN JUEGO DE NAVES. DEBE USAR IMÁGENES PARA LAS NAVES, LOS ASTEROIDES Y LOS ENEMIGOS. APLIQUE HERENCIA. USE UNA INTERFAZ DENOMINADA IDISPLAYABLE QUE TENGA EL MÉTODO DISPLAY(). DEFINA DOS INTERFAZES MÁS: IMOVEABLE QUE TENGA EL MÉTODO MOVER() Y OTRA ICONTROLLER 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.

DIAGRAMA DE CLASES:



| | | |
|---|---|---|
|  <p>TUDI Videojuegos Fundamentos de Programación Orientada a Objetos</p> | <p>FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS TECNICATURA UNIVERSITARIA EN DISEÑO INTEGRAL DE VIDEOJUEGOS FACULTAD DE INGENIERÍA Universidad Nacional de Jujuy PENSAMIENTO COMPUTACIONAL y PROGRAMACIÓN: Problema y Solución – PC y P – Algoritmos – Principio de la P</p> |  |
|---|---|---|

CONCLUSIÓN

UTILICE LOS VIDEOS DEL AULA XD