

Taller 1 - Algoritmos

Blizzard le ha contratado para crear un prototipo de un nuevo videojuego. Solicitan que usted desarrolle un juego tipo Space Invaders o Plants vs Zombies, usted escoge el mas le guste ya que ha sido un gran diseñador y desarrollador de videojuegos. Queda a su criterio la mecánica del juego y también le permiten utilizar los conceptos y estilos gráficos que usted desee. Un requerimiento no negociable, es que los enemigos sean infinitos, por tal motivo el mejor jugador será aquel que logre durar más tiempo sin morir. A medida que pase el tiempo el nivel de dificultad deberá aumentar. El juego debe mostrar el tiempo transcurrido y los puntos obtenidos. El juego debe tener al menos 4 pantallas:

- Pantalla inicio.
- Pantalla de instrucciones.
- Pantalla de juego.
- Pantalla resumen.

Contexto

Se debe desarrollar un videojuego, inspirado en Space Invaders o Plants vs. Zombies, que comience con una pantalla de inicio e instrucciones. Luego, en la pantalla de juego, los enemigos son infinitos, el nivel de dificultad debe aumentar a medida que pasa el tiempo y el ganador es quien dure más tiempo sin morir. Al final, se debe mostrar el tiempo y puntaje del jugador, en una pantalla de resumen.

Entidades:

- Principal
- Alien
- Alien Fuerte
- Arma

Requerimientos funcionales

RF1	
Descripción	El programa debe comenzar con una pantalla de inicio con 2 botones, uno de play y otro de instrucciones
Entradas	Archivo en formato png (pantalla inicial con los botones)
Salidas	-
Precondición	Las imágenes deben ya de existir

Postcondición	La imagen debe de aparecer en pantalla
----------------------	--

RF2	
Descripción	El programa debe mostrar una pantalla con las instrucciones, controles del juego y un botón de regreso a la pantalla inicial
Entradas	Archivo en formato png (pantalla instrucciones con el botón)
Salidas	-
Precondición	Se debió de hacer click en el botón instrucciones y las imágenes deben ya de existir
Postcondición	La imagen debe de aparecer en pantalla

RF3	
Descripción	El usuario debe de poder hacer click en los botones de las pantallas para continuar
Entradas	X y Y de la ubicación de los botones en la pantalla
Salidas	cambio de pantallas
Precondición	los botones deben ya de existir
Postcondición	La imagen debe de aparecer en pantalla

RF4	
Descripción	El programa debe mostrar la pantalla de juego
Entradas	Archivo en formato png (pantalla juego)
Salidas	-
Precondición	Se debió de hacer click en el botón play y las imágenes deben ya de existir
Postcondición	La imagen debe de aparecer en pantalla

RF5	
Descripción	El programa debe crear aliens infinitos
Entradas	Archivo en formato png (del alien), posX, posY..
Salidas	-
Precondición	Las imágenes deben ya de existir
Postcondición	Los enemigos deben de aparecer

RF6	
Descripción	El programa debe mostrar el tiempo transcurrido.
Entradas	Framecount
Salidas	-
Precondición	Hacer click en el botón inicio
Postcondición	Mostrar el tiempo transcurrido en la pantalla juego.

RF7	
Descripción	El programa debe crear más enemigos a medida que el tiempo transcurra
Entradas	Archivo en formato png (enemigo), X, Y.
Salidas	-
Precondición	Las imágenes deben ya de existir
Postcondición	Aumento de enemigos

RF9	
Descripción	El programa debe grabar el puntaje del jugador a partir de la vida de los aliens.
Entradas	alien eliminados
Salidas	Suma de score por aliens eliminados
Precondición	Registro de aliens eliminados
Postcondición	Puntaje final en la pantalla resumen

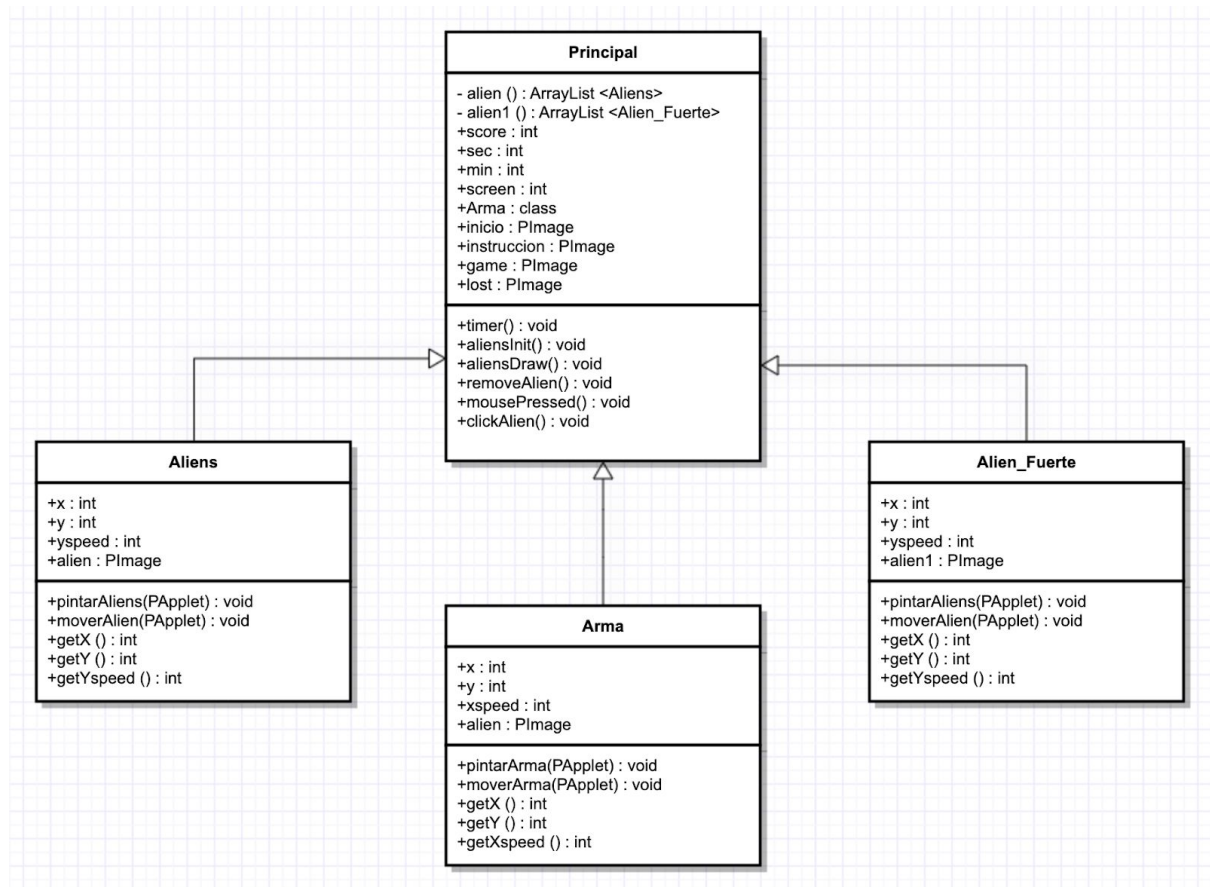
RF10	
Descripción	El programa debe mostrar una pantalla de resumen con un botón de reinicio.
Entradas	Archivo en formato png (pantalla juego con el botón)
Salidas	-
Precondición	Las imágenes deben ya de existir
Postcondición	La imagen debe de aparecer en pantalla

Requerimientos no funcionales

RNF1 El programa debe realizarse con java.

RNF2 El programa debe referenciar la librería de processing

Diagrama UML



BEHANCE

<https://www.behance.net/gallery/115064547/Heart-Attack>