

SPACE INVADERS AAT



ADRIANA ARMENTAL TOMÉ

Índice

Versión Orixinal.....	3
1. Un pouco de historia.....	3
2. Curiosidades.....	3
3. Deseño do xogo.....	4
Versión AAT.....	5
1. Descrición do proxecto.....	5
2. Deseño do xogo.....	5
3. Ferramentas usadas no desenvolvemento da aplicación.....	8
4. Funcionalidades do xogo.....	10
5. Análisis.....	15
6. Documentación por clases.....	23
7. Probas.....	25
8. Propostas de mellora.....	25
9. Conclusións.....	26
Bibliografía.....	26

Versión Orixinal

1. Un pouco de historia

A idea do “Space Invaders” xurdeu da man de Toshihiro Nishikado en 1978, quen programando para a compañía Taito (compañía japonesa dedicada a creación de máquinas arcade) ocorreu-lle a idea dun soldado tras unhas trincheiras que disparara a filas de inimigos aproximándose. A compañía decidiu cambiar os humanos por aliens porque parecía demasiado violento.

O xogador debe vencer 4 filas de aliens que se aproximan a el cada vez máis rápido. A innovación estaba no “High-Score” que foi o leit-motiv (tema central) dos videoxogos durante moito tempo. Creábase un instinto de superación, a batir o record inalcanzable e o orgullo de deixar a nosa puntuación rexistrada na máquina. Outro cambio que rexistrou foi a inclusión de 3 vidas porque ata entón xogábase por tempo, o cal outorgaba ós xogadores a posibilidade de pasar máis tempo xogando.

En conclusión, foi un éxito absoluto, provocou que o goberno nipón cuadruplicara a produción de moedas de yen pola escasez que provocaba o xogo. Cando chegou a América tivo os mesmos resultados. Pasou de bares e salas recreativas a varios negocios.

O longo da súa historia calculase que facturou máis de 500 millóns de dólares deixando varias historias tras de si como a rapaza que roubou 5000\$ ós pais para gastalos na máquina ou o record de Eric Furrer, quen con 12 anos xogou 38 horas e media seguidas acadando 1.114.000 puntos.

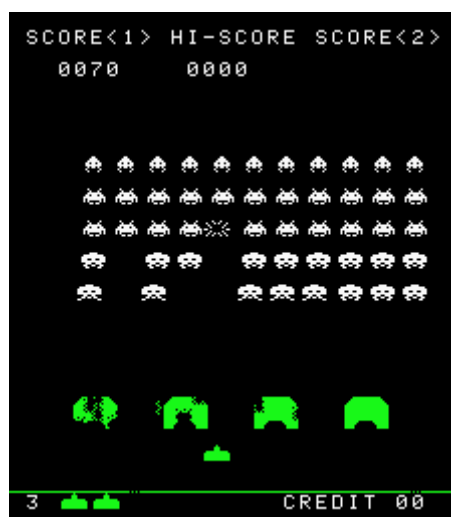


Figura1. Xogo Original

2. Curiosidades

- En Mortal Kombat Trilogy, aparece como un xogo oculto grazas ós kombat kodes.

- Na serie animada Futurama, no capítulo “Antología del interés II” podese ver unha escea na que os aliens de Space Invaders baixan e Fry cos seus amigos subense a unha nave para destruílos como no xogo.
- En Ultimate Mortal Kombat 3 é posible xogar ó Space Invaders.
- En OpenOffice.org co programa calc podese xogar unha variante de Space Invaders cun código secreto.
- En ¡Rompe Ralph! Xunto a máquina de Fix-It Felix Jr. están as máquinas de Pac-Man e Space Invaders.
- No xogo Grand Theft Auto: San Andreas, CJ pode xogar un modelo máis avanzado de Space Invaders.
- Nun episodio de Los Simpson: Bart ten que usar un tirachinas para destruír naves alienígenas, as cales van máis rápido cada vez que eliminas un grupo.
- No vídeo Pixels, aparecen os aliens destruindo os autos de Nueva York.
<https://www.youtube.com/watch?v=P-WnULvCdFA>
- O grupo de Electro Industrial alemán Eisenfunk grabou unha canción en tributo a Space Invaders chamada "Super Space Invaders".
https://www.youtube.com/watch?v=9oAYR6T_-VY
- En Cáceres (España) existe unha ponte sobre a Rivera del Marco, en Fuentefría, cuxo taboleiro consiste nun mosaico que reproduce unha partida de Space Invaders.
- O que produce que as naves baixen con máis velocidade foi producido por un bug pero mantívose porque o programador considerou que "facía o xogo máis interesante".

3. Deseño do xogo

1. Filas de naves alieníxenas avanzan cara abaixo con movementos de dereita a esquerda e viceversa, baixando un pouco cando toca en cada lateral, á vez que lanzan mísiles esporádicamente.
2. Unha nave defende a zona inferior para que non sea traspasada polos aliens, mentras lanza disparos para destruílos. Tendo en conta que non se poderá disparar ata que o disparo sala fora da pantalla ou intercepte un alien.
3. As fortalezas son destruídas polos disparos, tantos dos invasores como da nave.
4. A medida que os aliens diminúen en número aumentan a súa velocidade.
5. Hai catro tipos de aliens con candansúa puntuación (50, 100, 150 ou 300 puntos).
6. Cada 1500 puntos aparece un bonus que aportará unha puntuación aleatoria.
7. O xogador ten 3 vidas.
8. O xogo finaliza cando todos os aliens foron destruídos, cando chegan abaixo ou cando os mísiles interceptaron a nave.

Versión AAT

1. Descripción do proxecto

O proxecto a realizar consiste na realización dunha versión do lexendario videoxogo “Space invaders” mediante a Programación Orientada a Obxectos Java.

Os obxectivos son simples, cumpre que a nave sexa controlada por teclado por un xogador, que os aliens se vaian movendo por toda a pantalla, á vez que lanzan os seus lasers e tamén poidan ser destruídos polo laser da nave e un ovni que pase cada 10 segundos. Así durante 5 fases.

Decidín facer este xogo en Java porque foi un dos xogos que revolucionou na súa época os arcades e emprego este linguaxe porque foi no que máis traballamos en clase e no que máis tiña idea de facer animación. Tamén emprego frames en lugar de applets para que non se poda executar nun buscador e porque para o meu gusto, é mais sinxelo de programar.

2. Deseño do xogo

A diferenza do orixinal esta versión inclúe novas utilidades, á vez que elimina outras.

1. Como na versión orixinal, os aliens baixan cara abaixo cos mesmos movementos, baixando en cada lateral e lanzando os lasers.
2. Sitúase a nave na zona inferior e lanza disparos. Aquí encontrase o primeiro cambio, en lugar de volver a disparar cando se intercepta un alien ou o disparo desapareza da pantalla, a nave so disparará cando se solte a tecla presionada, neste caso espacio.
3. Segundo cambio, nesta versión non hai fortalezas.
4. Terceiro cambio, os aliens aumentan a velocidade pero non a medida que diminúe o seu número senón cando se elimina un bloque de aliens e consecuentemente se cambia de fase e aparece un novo bloque con maior velocidade. 5 fases con 5 fondos distintos.
5. Cuarto cambio. Os aliens sempre son os mesmos durante o xogo e polo tanto teñen a mesma puntuación (10 puntos).
6. Quinto cambio. Referíndo ó anterior, ofrecese a posibilidade no menu inicial, antes de iniciar o xogo, configurar a aparencia deste, podendo elixir entre 3 aliens, 3 naves e 3 fondos.
7. Sexto cambio. O bonus (un ovni) cruzará a pantalla cada 10 segundos e aportará sempre a mesma puntuación (100 puntos).
8. Séptimo cambio. O xogador só ten unha vida, sí é interceptado polos lasers dos aliens ou estes chegan abaixo acabouse a partida.
9. O xogo finaliza cando se consiguen superar as 5 fases destruíndo todos os aliens ou cando se perde, é decir, cando un laser dun alien impacta coa nave ou os aliens conseguiron superar a posición da nave.
10. Inclúese un menu pausa, neste caso pulsando a tecla p, que dará as opcións de reiniciar a

partida (non a fase), reanudar o xogo donde estaba antes de pausalo, salir ó menu principal ou pechar o xogo directamente.

11. No menú principal tamén se inclúe a opción “sonido” na cal se pode habilitar ou deshabilitar este según o gusto do xogador.
12. Tanto si se perde como si se gaña abrírase outro menu indicando a puntuación acadada e cas options de volver a xogar ou pechar o xogo.

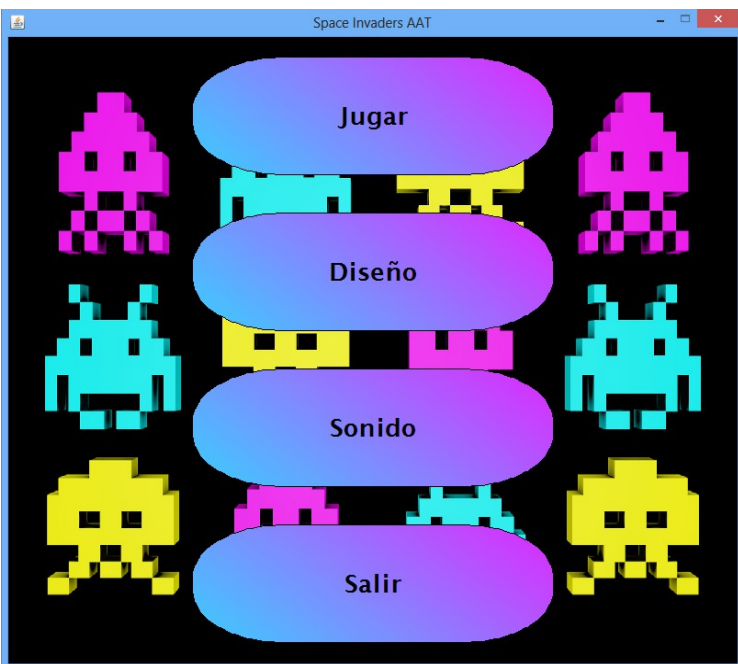


Figura2. Menú Inicial

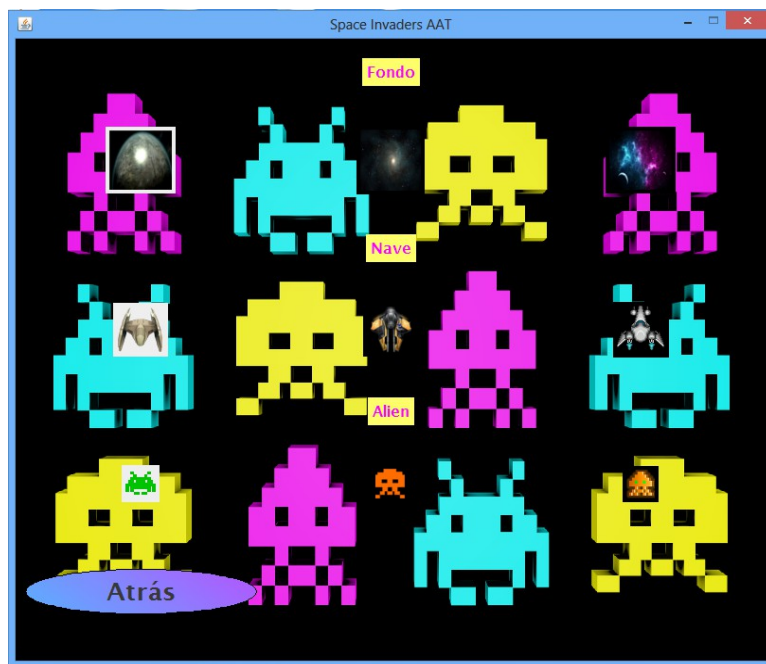


Figura3. Menú Diseño

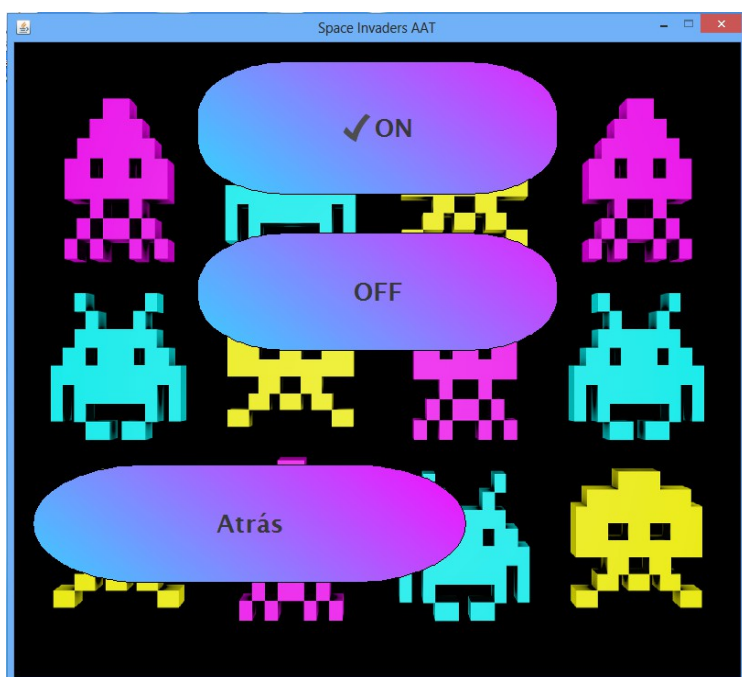


Figura4. Menú Sonido

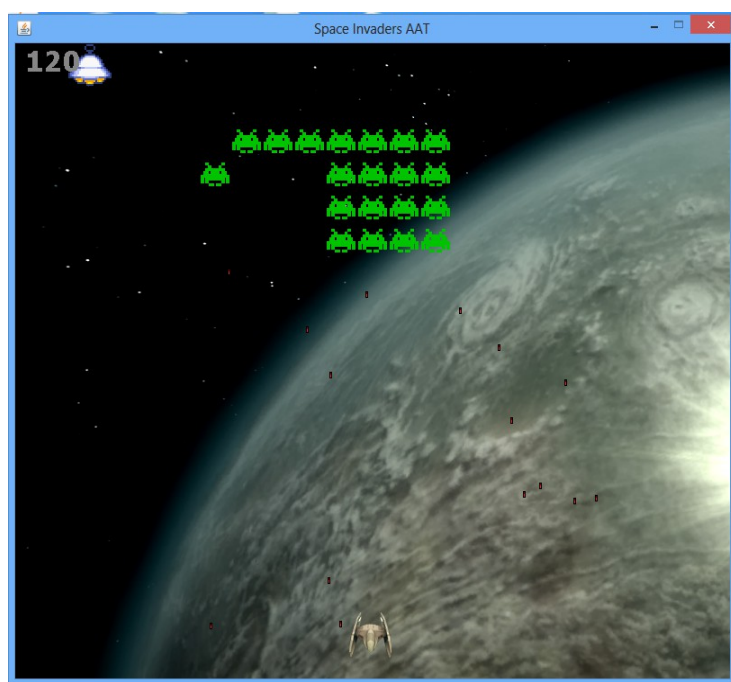


Figura5. Xogo

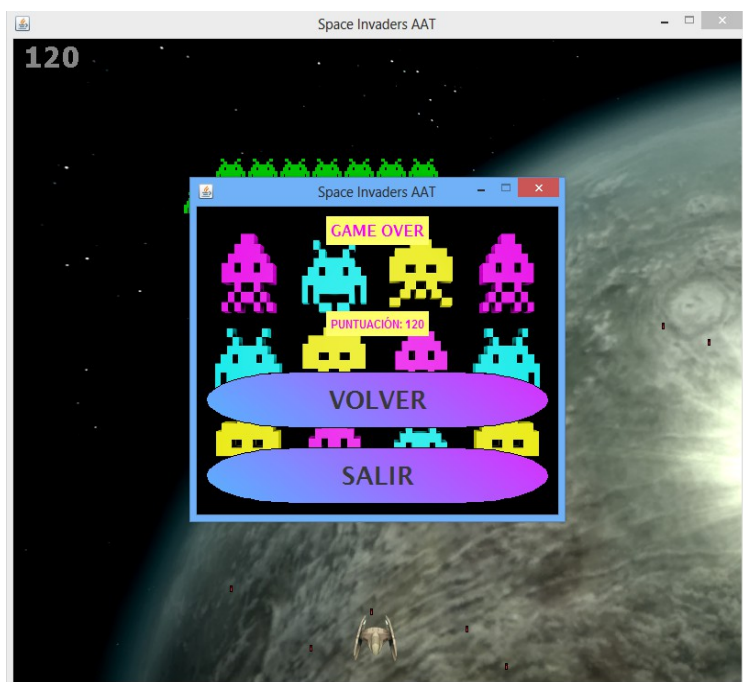


Figura6. Game Over

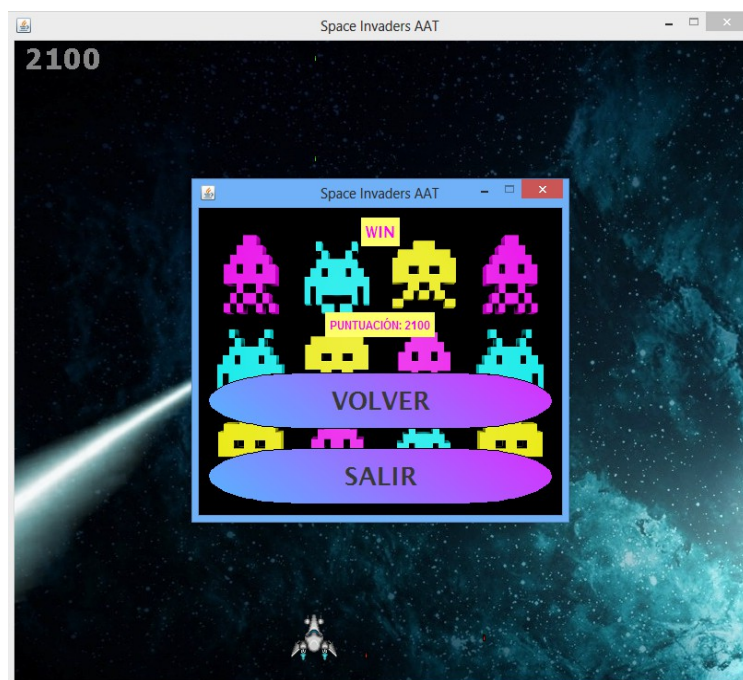


Figura7. Win



Figura8. Menú Pausa

3. Ferramentas usadas no desenvolvemento da aplicación

As ferramentas que se usaron no desenvolvemento do proxecto:

- ECLIPSE IDE JAVA EE DEVELOPERS KEPLER
Ferramenta para desenvolvedores de Java que crean Java EE e aplicacións web, incluíndo un Java IDE.
- JAVA SE
Para a creación, compilación e execución de aplicacións no linguaxe de programación Java, o requisito é a instalación dun entorno de programación J2SE.
- OPENOFFICE WRITER, IMPRESS
Necesarios para a creación da documentación e da presentación.
Encargase do procesamento de palabras, follas de cálculo, presentación, gráficos, bases de datos e máis.
- GLIFFY
Aplicación web para crear os diagramas.
- DROPBOX
Aplicación para o control de versións que empreguei durante o transcurso da realización da aplicación.

Dentro da propia aplicación tamén se empregou:

- JFRAME
Compoñente básica que se require para implementar unha interfaz visual coa librería Swing. Encapsula unha ventana clásica de calquera sistema operativo con entorno gráfico (Windows, Linux, etc). Ten unha serie de propiedades que se poden modificar ó gusto de cada un.
- Compoñentes do JFrame: JBUTTON, JRADIOBUTTON, JLABEL, JPANEL, BUTTONGROUP
Ainda que o JButton non se usa directamente, crease unha clase que hereda del e se modifican as súas propiedades.
Son compoñentes da librería Swing, das cales, o JLabel mostra un texto, o JButton mostra un botón a partir do cal se interactúa mediante eventos e a súa variable, o JRadioButton tense que agrupar nun ButtonGroup de xeito que só se permita a selección dun deles, é dicir, cando se seleccione un se deseleccionen os outros. E o JPanel actúa como contenedor dentro do cal se poden agrupar os anteriores mencionados para darlle unha distribución distinta o JFrame.

- ACTIONLISTENER

É a interfaz de escoita para recibir eventos de acción (ex. pulsar un botón). Rexístrase un obxecto co compoñente addActionListener e cando se produce o evento de acción invocase o método actionPerformed que levara a cabo unha acción indicada.

- KEYLISTENER

Similar ó anterior pero para recibir eventos de teclado (ex. pulsar unha tecla). Rexístrase un obxecto co compoñente addKeyListener e cando se pulsa unha tecla invocase o método KeyEvent que se subdivide en KeyPressed (tecla pulsada), KeyReleased (tecla liberada) e KeyType (tecla escrita).

- SPRITE

Un bitmap (mapa de bits) é a representación dunha imaxe. Un sprite é un bitmap dotado de características propias como anchura, altura, desprazamento, etc, que necesita ser controlado.

Son a representación gráfica dos personaxes que aparecen. Hai un personaxe principal controlado polo xogador, mentras que os inimigos avanza ata el. O obxectivo é repeler o sprite do personaxe principal dos disparos inimigos.

Podese dicir que un sprite é un elemento gráfico determinado con entidade propia e sobre o cal se poden modificar distintos atributos.

- AUDIOCLIP

É unha interfaz abstracta para a reprodución dun clip de sonido. Poden estar múltiples sonidos ó mesmo tempo, e no resultante mézclanse para formar un material composto. Non recoñece formatos mp3.

- ACTOR

Un actor é unha entidade externa ó sistema pero que realiza interaccións con el. O personaxe principal do xogo é o actor e cada unha das súas accións, desprazar ou facer disparar a nave.

- CLASE GRAPHICS E GRAPHICS2D

Graphics é a clase abstracta que proporciona o entorno de traballo para calquera operación gráfica que se realice dentro do AWT.

Para poder pintar, un programa necesita un contexto gráfico válido, representado pola instancia de Graphics. Pero esta clase non se pode instanciar directamente así que

debemoslo pasar como argumento o método paint()).

A clase Graphics soporta tres categorías de operacións gráficas.

Debuxo de formas primitivas, debuxo de texto e presentación de imaxes (todas as imaxes se debuxan non paint).

O api 2D de Java presenta un novo tipo de obxecto Graphics2D que descende da clase Graphics para proporcionar acceso as características avanzadas de renderizado do api 2D de Java.

Para usar as características de api 2D de Java, temos que forzar o obxecto Graphics, pasado ó método de debuxo dun compoñente de Graphics2D.

Solo se emprega para o gradiente dos botóns, no cal se crea unha instancia de Graphics2D e se lle cambia o atributo paint a un recheo de gradiente.

- TIMER

Proporciona facilidade para programar tarefas de execución nun subproceso en segundo plano.

Con esta clase varios subprocesos poden compartir un solo obxecto Timer sin necesidade de sincronización externa.

Fai tarefas de execución dunha soa vez, ou para a execución repetida a intervalos regulares. Todos os construtores empezan unha liña de temporizador.

- THREAD

Un fío é un proceso que se esta executando nun momento determinado no noso sistema operativo e realízase directamente no procesador. Os fíos que aquí se empregan son definidos con un comportamento e inicianse nun momento específico.

En Java, o proceso que sempre se executa é o chamado main que é a partir do cal se inicia practicamente todo o comportamento da nosa aplicación e en ocasións xa lle basta ela sola para funcionar correctamente pero hai outras que necesitan máis dun proceso executándose ó mesmo tempo (multithreading).

4. Funcionalidades do xogo

PARTE JFRAMES

O xogo está realizado en JFrames, ós cales se lle asigna un tamaño de 800 px de ancho e 700 px de alto. Os diferentes JFrames que existen iranse eliminando e facendo visibles según a ventana que se desexe abrir. Foron creados de forma que non se podan pechar na x superior dereita, só se poderá no JFrame do xogo (Figura5), non se lles pode cambiar o tamaño, neste caso o do xogo tampouco e cada vez que se abra unha ventana está colocarse no centro da pantalla.

O menú principal (Figura2) mostra 4 botons dos cales 2 son para iniciar o xogo e para pechalo. Os outros 2 levan a cadansúa ventana na que se elixen diferentes opcións de xogo, son e configuración do deseño do xogo.

No menú de deseño (Figura3) estan as opcións para elixir un fondo inicial, posto que a medida que se vaian avanzando fases vaise ir cambiando, e unha nave e un alien co que xogar as diferentes fases.

A opción elixida de cada grupo será enmarcada. Por defecto estará enmarcada a primeira opción, xa que senón se abrise o menú de deseño ter un alien, unha nave e un fondo cos que xogar.

Se presionamos no botón “Sonido” do menu inicial conducíranos a outra ventana na que se dan as opcións de “ON” e “OFF” (Figura4). Estes botóns se son seleccionados teñen un “check” indicando a opción elixida e poñendo ou quitando o son. Por defecto, e sempre que se inicie o xogo estará con sonido e por supostamente seleccionado o boton “ON”.

Os botons “ATRÁS” que aparecen nas ventanas de opcións, Deseño e Sonido, se son pulsados, pechan a ventana actual e abren o JFrame que levou a elas, é dicir, o menú principal.

Unha vez seleccionados ou non, as diferentes opción que se dan a elixir, vólvese o menú inicial e si se presiona “Jugar” abrírase a ventana de xogo coas imaxes elixidas e no caso de non elixírense ningunha, as que veñen implantadas por defecto.

Xa no xogo, si se desexase parar por calqueira motivo, ou reiniciar a partida si se presiona a tecla “p” abrírase un novo menú que á vez parara o xogo (Figura8). Este ofrece a posibilidade de, ou ben reanudar a partida donde a deixamos ou reiniciala dende a primeira. Tamén aparecen os botóns de “SALIR AL MENU”, ó cal che volve a abrir o primeiro menú como indica e “SALIR” que pecha o xogo directamente que tamén se podería realizar dende a x do xogo.

Durante o xogo tamén se abren dúas novas ventanas pero que se abren solas, sin necesidade de presionar ningunha tecla. As ventanas conforme perdiches ou ganaches, as cales son idénticas, só cambia o título. Unha pon “Game Over” e a outra “Win”. Ambas teñen unha etiqueta que mostra a puntuación acadada e dous botóns, un que leva ó menu principal por si se desexa volver a xogar e outro que pecha a aplicación.

Os botons que ten cada menú están realizados mediante unha clase Boton que hereda de JButton e colle todas as súas propiedades. Modificáronse algunhas como a forma, o borde e o gradiente para darlle un estilo máis vistoso.

Tamén para darlle o fondo ós menús se tivo que crear outra clase na cal se lle asigna unha imaxe que se vai colocar en todos os JFrames que ten a aplicación.

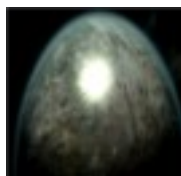
PARTE MULTIMEDIA

A parte multimedia da aplicación está composta polo que ven sendo a animación:

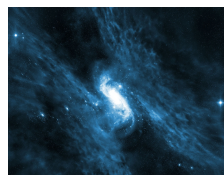
Dende que se abre a aplicación ata que se pecha estará sonando sempre a mesma melodía (fondo.wav) a non ser que se lle quite o son. Para calqueira son que leve a aplicación hai que ter en conta que sonará si ten o sonido activado ou no.

O fondo crease a partir dunha imaxe e colocase na posición (0,0) do JFrame para que ocupe todo o fondo. A medida que se vai pasando de fase, destruindo todos os aliens vai cambiando.

INICIAIS



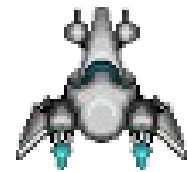
POR FASES



Unha nave feita por unha imaxe e que se situa na zona baixa do JFrame. Esta nave moveráse cara á esquerda ou cara á dereita coas teclas de dirección esquerda e dereita cunha velocidade indicada.

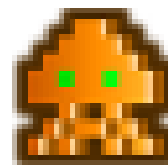
Coa barra espaciadora a nave soltará un laser e indicarse tamén cun son (lnave.wav). Moveráse en traxectoria ascendente. Este laser tamén é unha imaxe. Cando este láser choca cun ovni ou cun alien, ambos desaparecen. Tamen leva un son (colision.wav).

Ó eliminar todo un grupo de aliens cambiase de fase e cando todos os aliens foron eliminados gañase a partida. Se o laser non impacta con ningún alien nin ovni segue o seu recorrido ata que percorre todo o JFrame e entón desaparece.



Un grupo de aliens formado sempre pola mesma imaxe que empezan na zona superior esquerda do JFrame e se irán desplazando cara á dereita e unha vez chegado o límite baixan un pouco e volven cara á esquerda e alí volven a baixar e así sucesivamente ata que, si non son destruídos chegan abaixo e se perde a partida.

Teñen unha velocidade tamén que irá aumentando consecuentemente se cambie de fase. Estes tamén lanzan un laser que leva un son pero distinto o laser da nave (lalien.wav), pero eles lanzano de maneira aleatoria e en traxectoria descendente. Cando este laser impacta coa nave sona unha explosión (explosion.wav) e tamén se perde a partida e no caso de non impactar coa nave recorre o JFrame ata que desaparece o traspasar a zona inferior.



Un ovni pero do cal non se pode elixir a imaxe (sempre é a mesma, para todo), que cruzará a pantalla de esquerda a dereita cada 10 segundos, sexa eliminado ou non. Leva unha velocidade contante que non varía anque se cambie de fase e o contrario que os demais non dispara lasers, solo trascorre pola ventana a modo de bonus. Xa que aporta maior puntuación, non é necesario destruílo para gañar o xogo.



Na esquina superior dereita está debuxado o marcador que irá sumando as puntuacións consecuentemente se eliminen aliens ou ovnis.

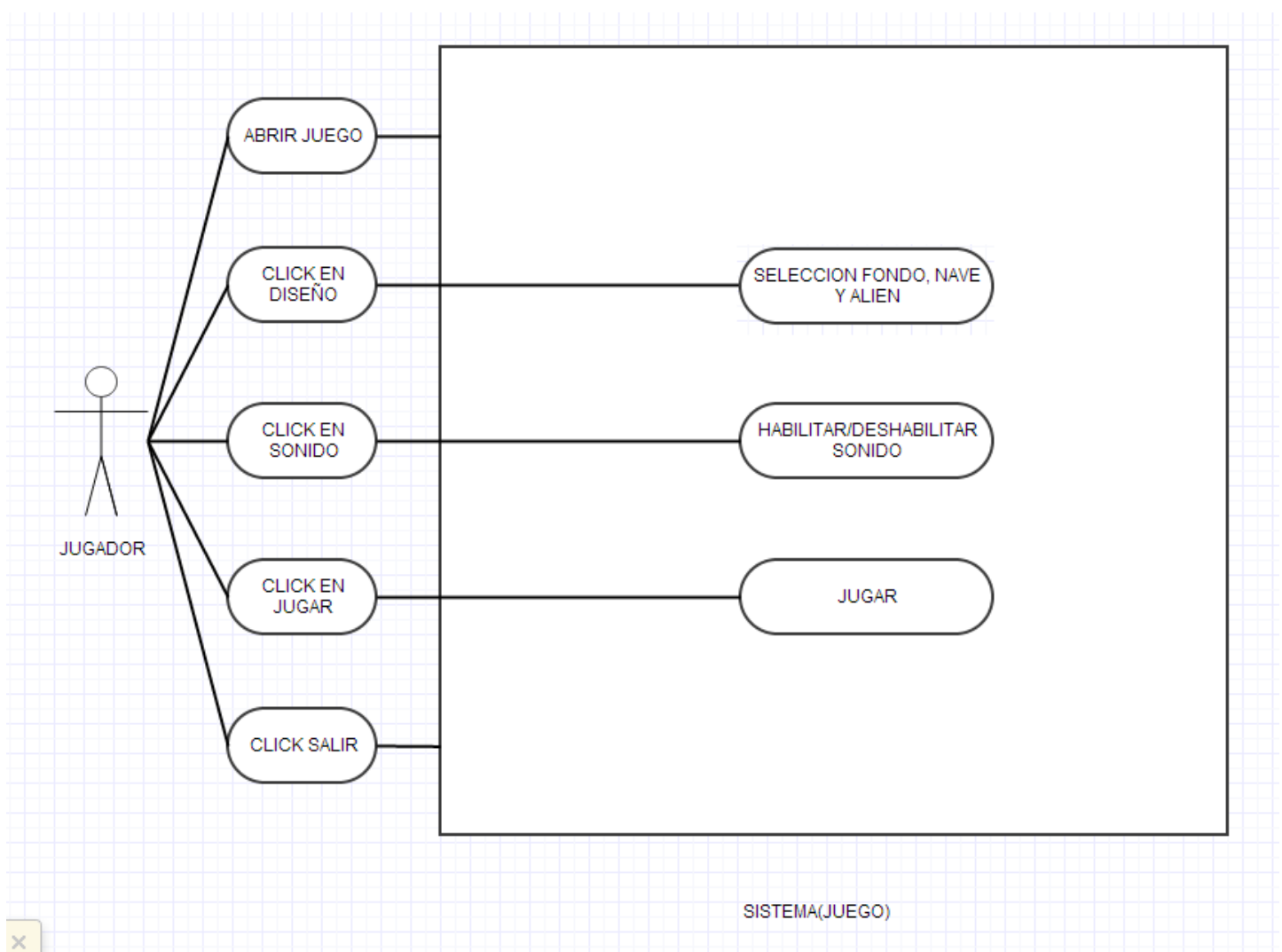
40

1900

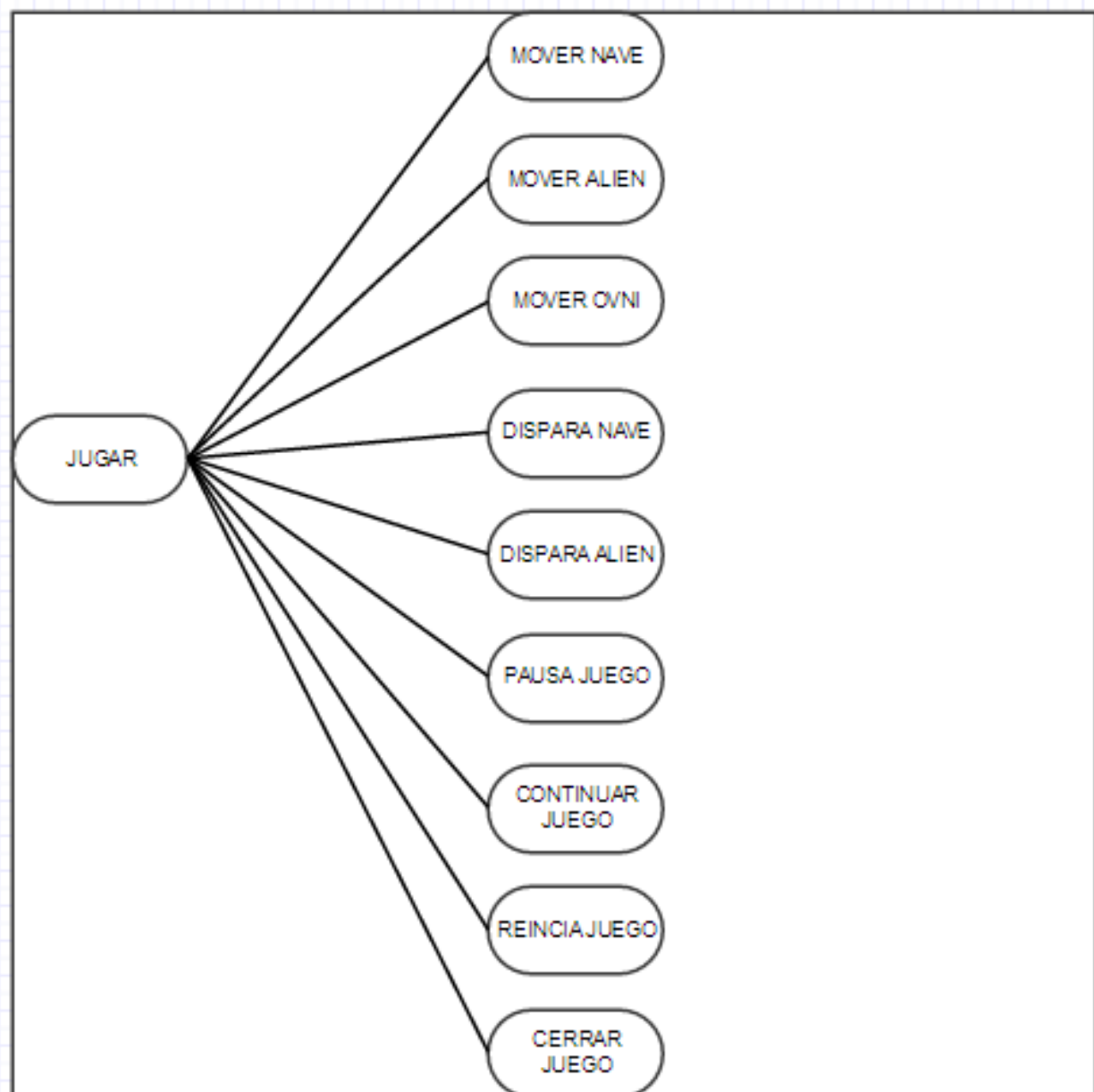
5. Análisis

- Diagrama de Casos de Uso

O diagrama de casos de uso representa a forma en como un Cliente (Actor) opera co sistema en desenvolvemento, ademáis da forma, tipo e orden en como os elementos interactúan. Nesta caso mostráanse as distintas formas nas que se pode atopar o xogo.



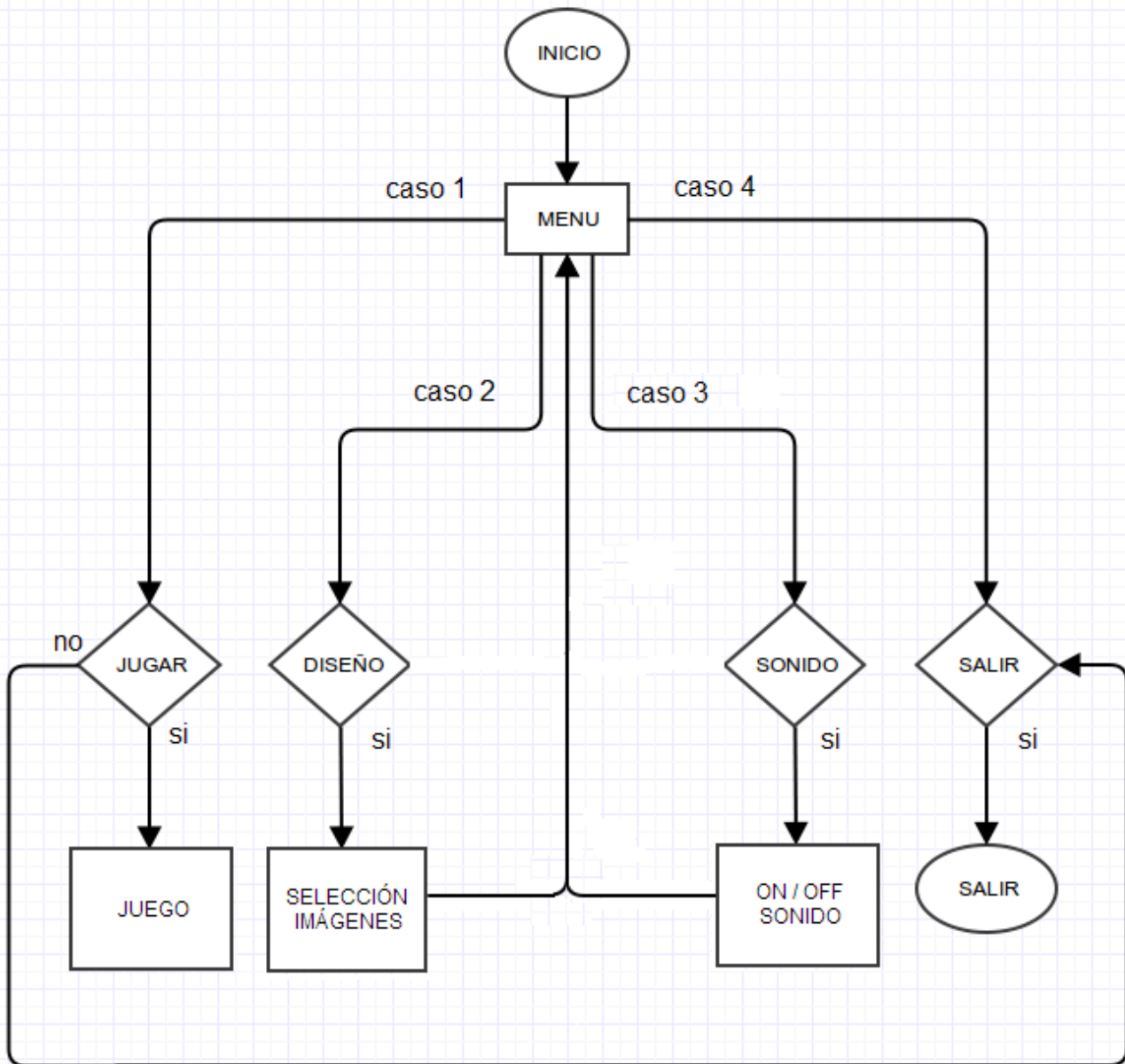
E neste caso mostráanse o que conleva no sistema a operación “jugar”.



SISTEMA(JUEGO)

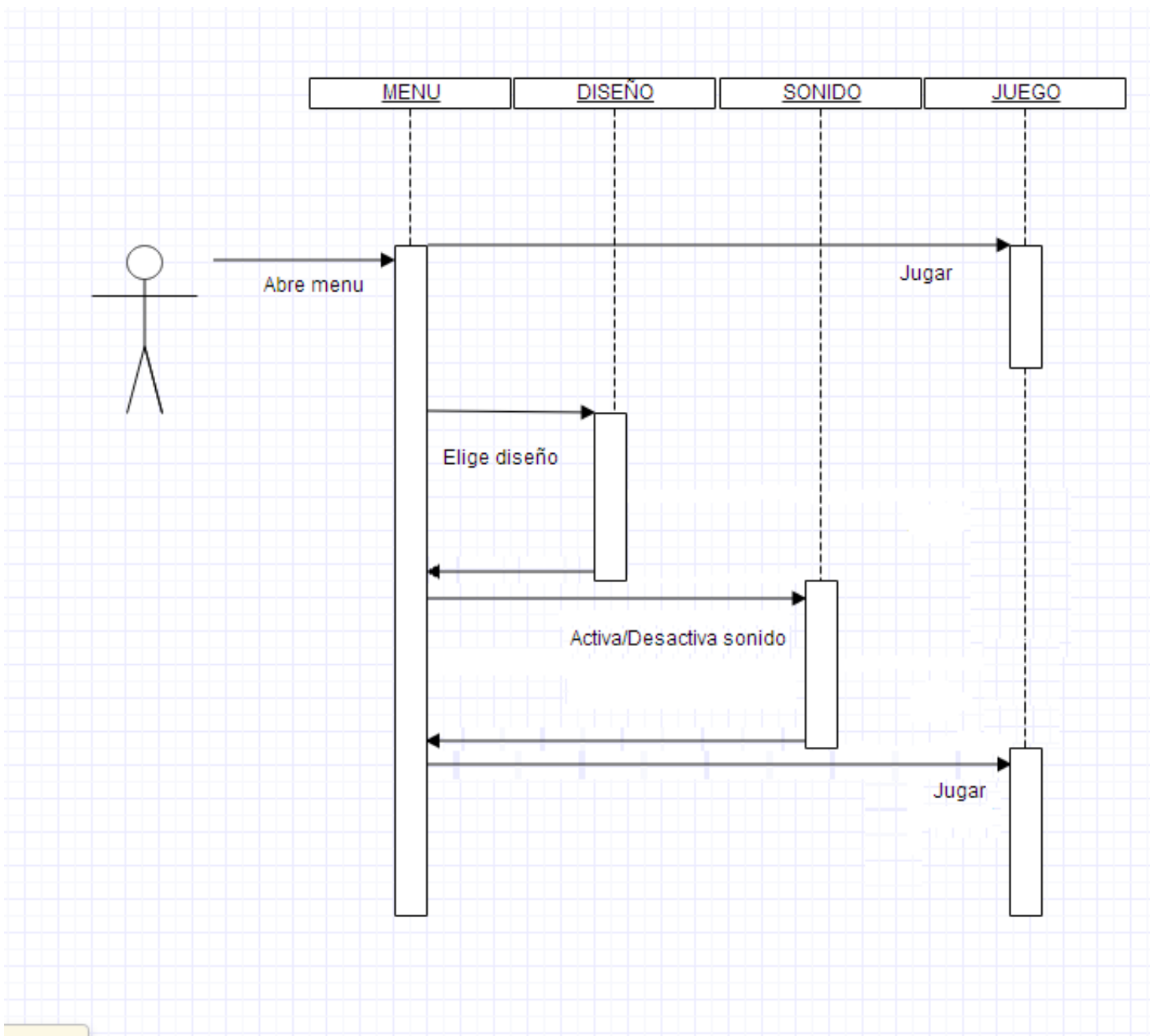
- Diagrama de Fluxo

O diagrama de fluxo é unha representación gráfica dun proceso. Ofrece unha descrición visual das actividades implicadas nun proceso mostrando a relación secuencial entre elas, facilitando a rápida comprensión de cada actividade e a súa relación coas demais.



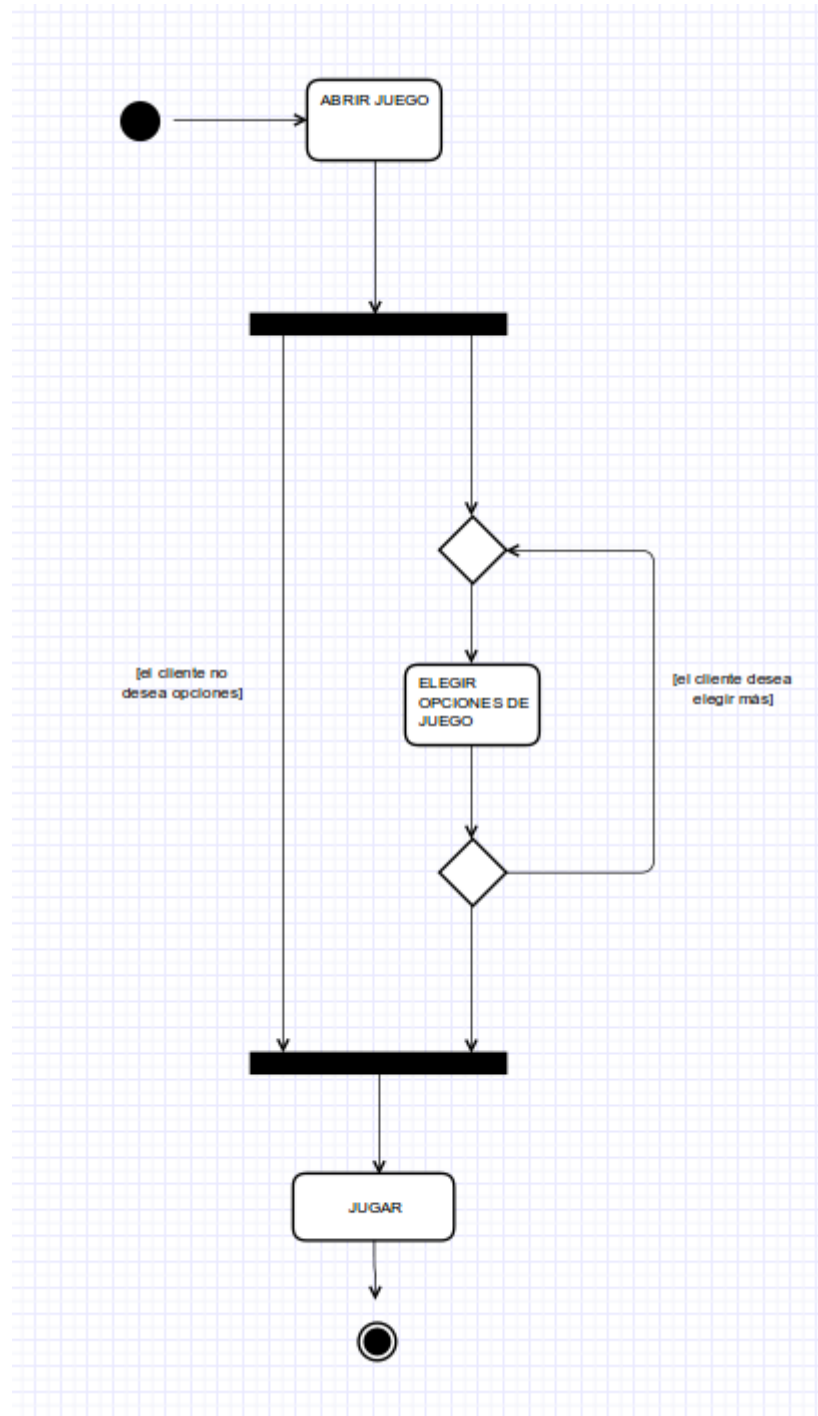
- Diagrama de Secuencia

Mostra unha interacción que representa a secuencia de mensaxes entre as instancias das clases, compoñentes, subsistemas ou actores. O tempo flúe cara abaixo no diagrama e mostra o fluxo de control dun participante a outro.



- Diagrama de Actividades

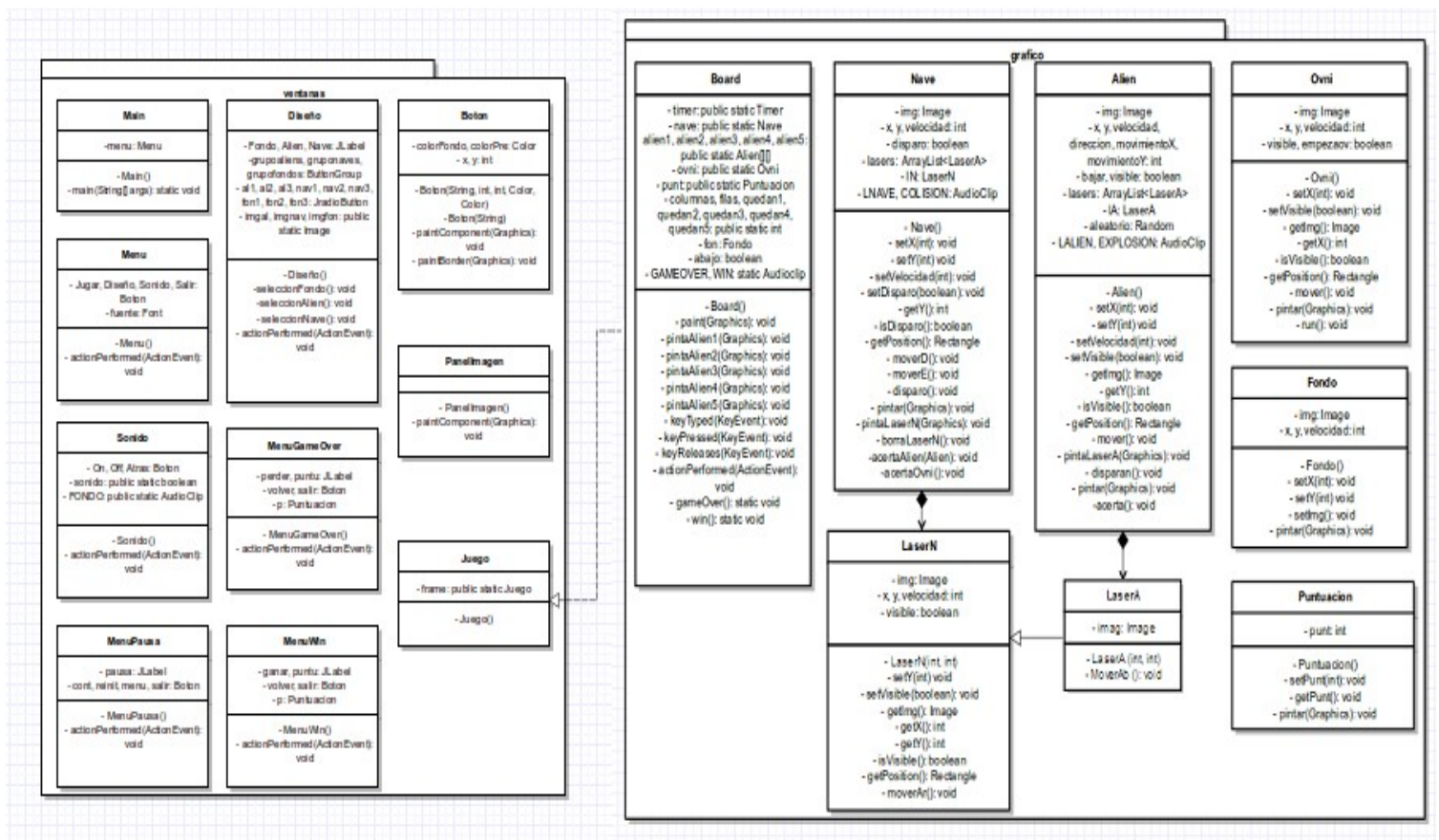
Mostrase un proceso de negocio ou un proceso de software como un fluxo de traballo a través dunha serie de accións que poden ser levadas a cabo por persoas, compoñentes de software ou equipos.

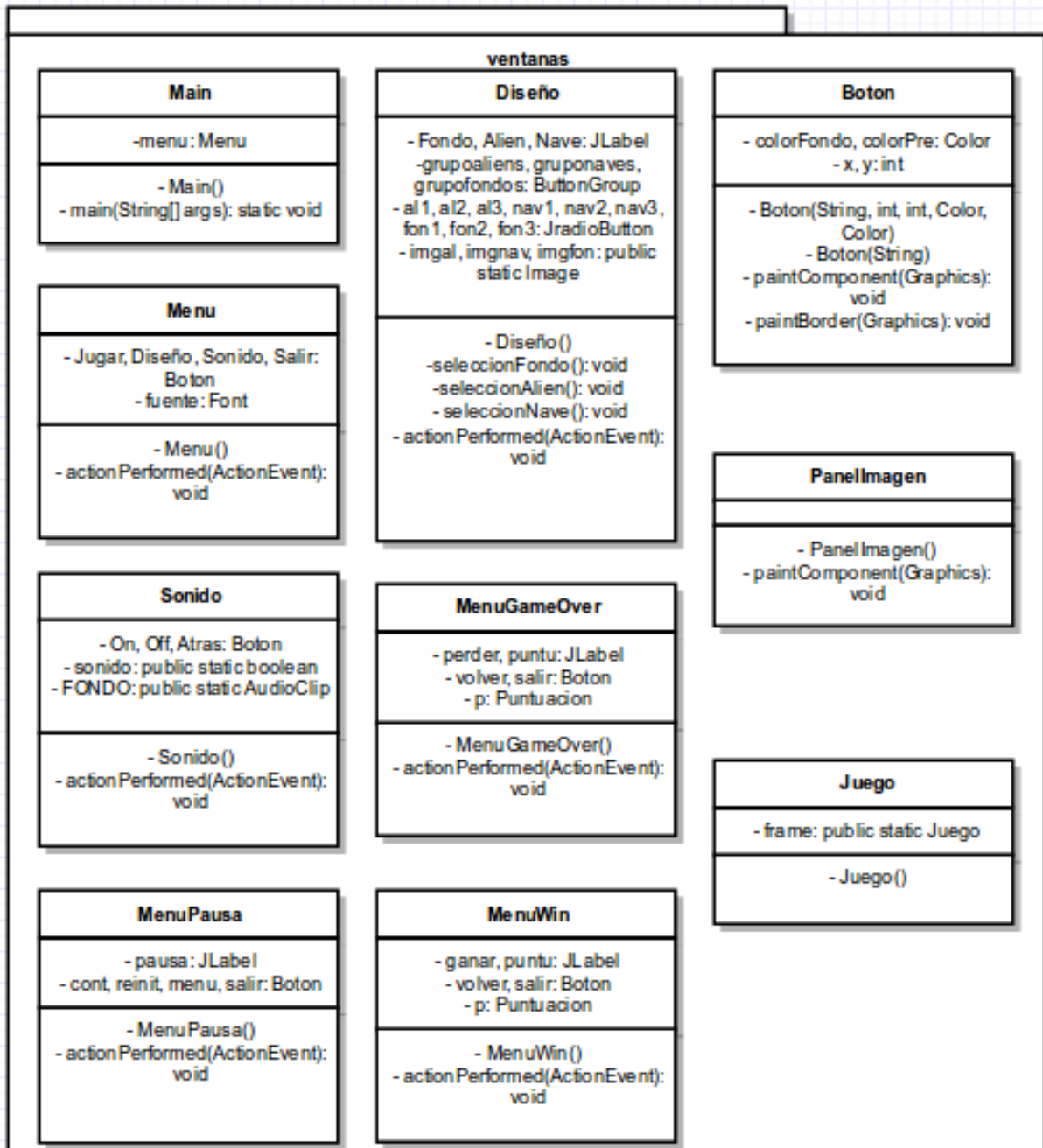


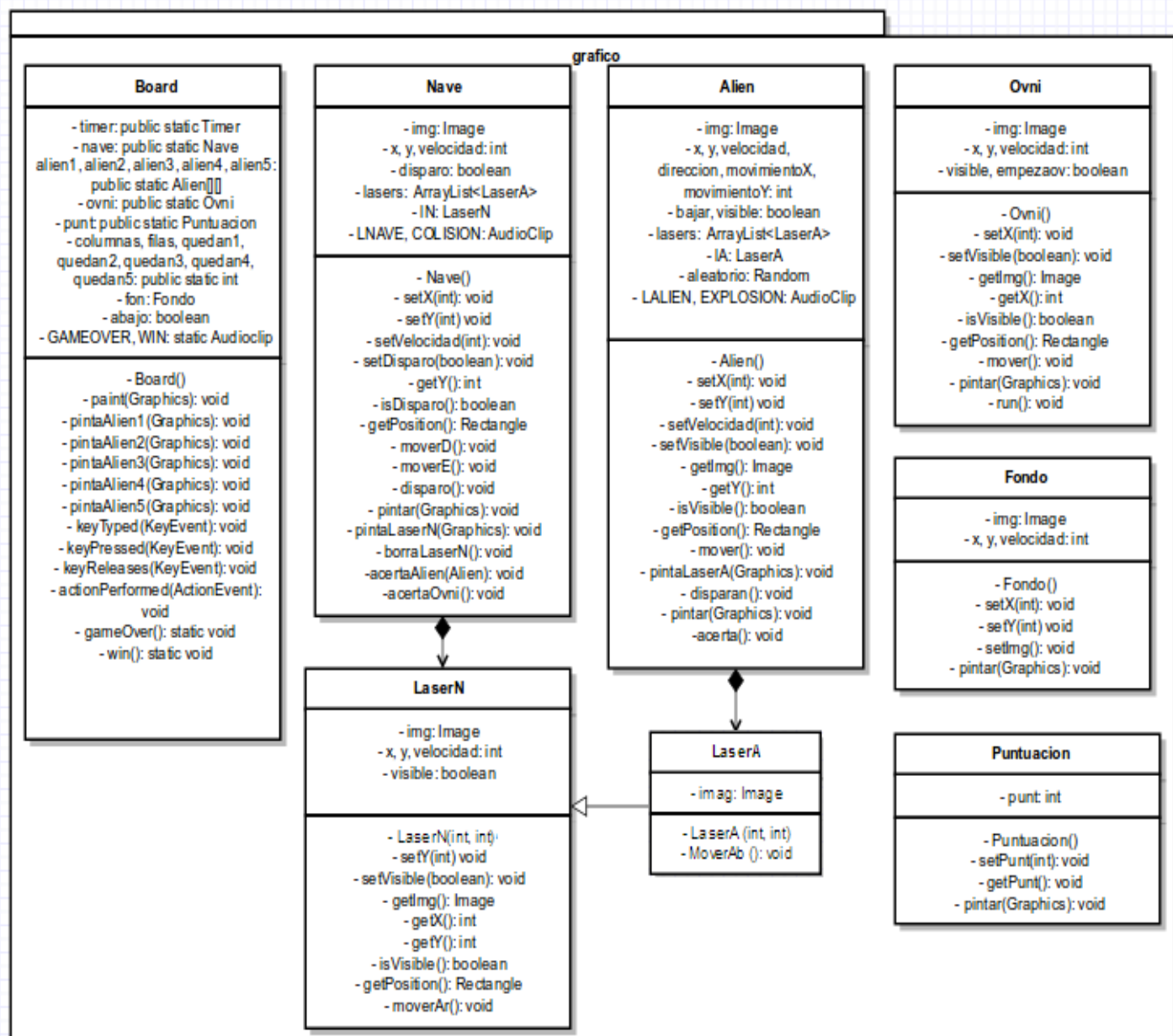
- Diagrama de Clases

Sirve para visualizar as relacións entre as clases que involucran o sistema, as cales poden ser asociativas, de herencia, de uso e de contemento.

A relación que une ambos paquetes realízase na inclusión da clase Board o JFrame Juego. Podense ver ambos paquetes con mais detalle.







6. Documentación por clases

Definíronse dous paquetes, ventanas que vai conter todos os JFrames para darlle soporte a aplicación e graficos que vai levar toda a lóxica e entornos multimedia do xogo.

VENTANAS

A todos os JFrames se lles cambiou as mesmas propiedades.

En todos se crea unha serie de etiquetas ou botóns que se van distribuír mediante un GridBagLayout. A estes modifícanse algunhas propiedades como a cor de fondo, cor de letra ou poñer a escoita, así como o seu consecuente actionPerformed para levar a cabo as accións de cambiar de ventana, eliminado a actual e facendo visible a seguinte.

Para definir os JFrames sempre se modifican as mesmas propiedades. Colocarlle un panel dunha imaxe de fondo, darlle tamaño, colocalo no medio da pantalla e que non se poda pechar na x da esquina superior.

Á clase Diseño súmaselle a maiores a formación de grupos de botóns que van recoller os JRadiobuttons, ós que tamén se lles vai asignar unha imaxe e os paneles nos que van ir as etiquetas por darlle color de fondo.

Na clase Sonido tamén aparece a opción de quitar ou poñer sonido polo cal xa aparece o AudioClip para paralo ou reanudalo.

A clase Xogo só ten como diferencia das demais que inclúe o panel no que se vai pintar o xogo para que apareza nun entorno visual.

Xa cando se esta executando o xogo e polo tanto nos atopamos no JFrame da clase Xogo, podese abrir o menu pausa que ven sendo como unha mezcla dos anteriores. Ten a súa distribución de GridBagLayout e os compoñentes a escoita e tamén os paneles para darlle fondo as etiquetas.

Cando se perde ou se gaña abrense dous menus iguais, só cambia o texto da etiqueta principal que indica si se gañou ou perdeu e o son que os acompaña. Nestas clases empregáse o mesmo que xa expliquei anteriormente.

O panellmagen vai coller un tamaño dado para adecualo ós JFrames, neste caso deille un tamaño maior para que quedara máis expandido e co método paintComponent debuxase a imaxe.

Tamén a clase Boton, ó heredar de JButton, emprega ese mesmo método, no cal se crea un Graphics2D para poder darlle un gradiente de cores o botón, e o método paintBorder para pintar o borde.

GRAFICO

Podese dicir que en todas as clases que pertencen a este paquete se colle unha imaxe e posición na que se atopa si hai unha colision, menos no Board e na Puntuacion.

Á clase LaserN indícaselle un movemento ascendente. Só movendo a coordenada y. A clase LaserA que hereda de LaserN, cambiaselle a imaxe e o movemento descendente.

Na clase Alien defínese a maiores o movemento que ten que levar, posicionando as coordenadas x e y. Tamén se crean os métodos de debuxo no cal se vai añadir un arraylist de lasers que vai soltar cun random acompañados dun audioclip. Creáse un método para saber si colisionou o laserA coa Nave, en cuxo caso tamén se denotaría co sonido da explosión ou polo máis evidente, que se perdeu. Finalmente pintar todo.

A clase Nave é similar a anterior, pero os movementos da nave só se realizan horizontalmente na parte inferior e están controlados polo xogador. Tamén se crea o arraylist de lasers con cadanseu son pero que van estar controlados polo xogador. Idénticamente tamén existe o metodo para averiguar si o laser colisionou co alien, que o eliminaría ou se é o último gañaría o xogo. Como tamén existe un ovni tamen se crea un método para saber si o laser chocou con el. E por suposto pintar todo.

A clase Ovni creáse como un fío, xa que sempre vai aparecer cada 10 segundos cun movemento horizontal na parte superior. E un método pintar para incluír no Board.

A clase Fondo non ten maior complicación, xa que solo colle a imaxe e se pinta de xeito que ocupe todo o espazo.

Na clase Puntuación só se crea un enteiro que vaia levando conta dos puntos que vai facendo o xogador cada vez que destrúe un alien e o método pintar no que se di que se debuxe un texto, donde e outra serie de propiedades.

A clase Board, o panel no que se vai xuntar todo, creáse un obxecto de cada clase anterior para poder debuxalas todas no mesmo contexto e tamén se lle da funcionalidade o teclado que vai controlar a nave.

E finalmente a clase principal Main, na cal se abre o menú porque é o primeiro que aparece ó abrir o xogo e tamén se inclúe a melodía que vai acompañar a todo o xogo.

7. Probas

Proba de sistema

Comprobar que o sistema funcione correctamente. A proba de todo o sistema en conxunto.

Como é un videoxogo centrase nas teclas que controlan a nave ou abren o menu pausa, tales como as de dirección, “Espacio” e “P”. Como en toda aplicación sempre hai a posibilidade de abarcar máis, con novas clases ou novos métodos e unha vez acabados todos volveríase realizar esta proba.

Proba unitaria

Comprobar os módulos individuais do sistema.

Durante o desenvolvemento das clases da aplicación, fixéronse probas individuais para detectar máis facilmente calquera problema que puidese xurdir. Foron máis que necesarias para controlar a animación das imaxes contidas, sobre todo a hora de realizar o fío do ovni. Finalmente o seu resultado foi o correcto.

Proba de integración

Comprobar que as partes do sistema dun certo nivel funcionan correctamente todas xuntas.

Todas as clases, tanto as de JFrame como as do entorno gráfico interactúan entre elas correctamente, incluídas as pulsacións de tecla por parte do xogador.

Proba de aceptación

O cliente ou alguén que represente ó cliente valida que a aplicación realmente cumpre o seu propósito.

Esta proba non ten moito sentido xa que non está destinada a ningún cliente pero foi probada por varias persoas, as cales non lle atoparon erro ningún.

8. Propostas de mellora

Como videoxogo que é, sempre se lle poden facer cambios que afecten ou a dificultade ou o aspecto gráfico. Un dos que se podería ter en conta.

- ◆ Facer fortalezas.
- ◆ Facer cohetes que destrúan parte do grupo de aliens.
- ◆ Facer que aparezan imaxes que si colisionan ca nave aumente en vidas.
- ◆ Facerlle propiedades a cada nave.
 - A primeira nave que aguante 2 disparos dos aliens.
 - A segunda nave que dispare máis rápido.
 - A terceira nave teña maior velocidade.

- ◆ Incluir unha tecla para que abra unha axuda e un manual de como se xoga.

9. Conclusións

Esta aplicación é un xogo que na súa época foi unha innovación total pero co tempo sempre se pode mellorar, é o igual que min, moita xente fixo as súas propias versións. Ten unha xogabilidade sinxela e directa e pode chegar a ser aditivo.

Bibliografía

http://space-invaders.publijuegos.com/historia_de_space_invaders.html

http://es.wikipedia.org/wiki/Space_Invaders

<http://docs.oracle.com/javase/7/docs/api/>