



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

**FACULTAD DE ESTUDIOS SUPERIORES
ARAGÓN**

**“DISEÑO E IMPLEMENTACIÓN EN SOFTWARE
DE UN JUEGO DE ESTRATEGIA
COLABORATIVA”**

T E S I S

**QUE PARA OBTENER EL TÍTULO DE :
INGENIERO EN COMPUTACIÓN
P R E S E N T A :
ALBERTO MURILLO OLASCOAGA**

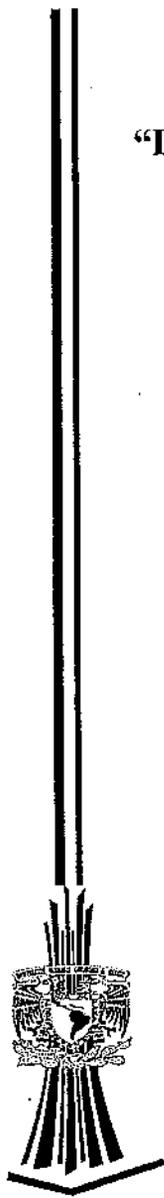
ASESOR:

M. EN C. MARCELO PÉREZ MEDEL

MÉXICO

2005

m. 345510





Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

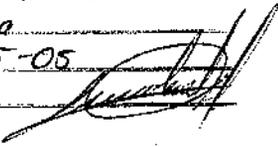
El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Autorizo a la Dirección General de Bibliotecas de la UNAM a digitalizar en formato electrónico e impreso el contenido de mi archivo personal.

NOMBRE: Alberto Morillo

Olascoaga

FECHA: 20-5-05

FIRMA: 

AGRADECIMIENTOS

A MI MADRE

Sería imposible expresar en unos cuantos renglones, todo el agradecimiento y sentimiento que siento por ti, por todo el amor, apoyo, ayuda, soporte, guía, y tantas y tantas cosas más que incondicional y completamente me has dado.

En mi vida ha habido momentos muy difíciles, pero contigo a mi lado todas esas cosas las he superado y ahora se han convertido en experiencias; experiencias que me han dado la fortaleza que ahora tengo.

Todavía falta mucho camino por andar, muchas cosas por hacer, afortunadamente sé que seguiré contando contigo por el resto de mi vida, **GRACIAS MAMÁ!**

A MI FAMILIA

Quiero agradecerle a todos ustedes el gran apoyo que siempre me han dado, su cariño y comprensión. Espero que los años sigan pasando y que la unión familiar que tenemos se mantenga tan cordial, respetuosa y amorosa por el resto de nuestras vidas.

Quiero que sepan que siempre contarán con mi apoyo incondicional, **GRACIAS!**

**DISEÑO E IMPLEMENTACIÓN EN SOFTWARE DE UN JUEGO DE
ESTRATEGIA COLABORATIVA**

ÍNDICE

INTRODUCCIÓN

CAPÍTULO 1	Diseño conceptual del juego (Batel – Nah)	3
1.1	Idea Conceptual de Batel – Nah	3
1.2	Análisis comparativo entre el Ajedrez y Batel - Nah	4
1.3	Movimientos	18
1.4	Eliminación	30
1.5	Versatilidad	38
CAPÍTULO 2	Creación de las piezas virtuales	43
2.1	Creación de Tablero	43
2.2	Creación de Piezas Batel- Nah	45
2.3	Creación de Logotipo	62
2.4	Creación de Cursores e Iconos	63
2.5	Creación de Imágenes Gif	66
CAPÍTULO 3	Diseño de algoritmos del juego	67
3.1	Diseño de los algoritmos de identificación del tablero	68
3.2	Definición de algoritmos de los movimientos de la piezas	75
3.3	Lógica de movimientos	83
CAPÍTULO 4	Implementación del Juego para PC	90
4.1	Tablas y valores iniciales	90
4.2	Los movimientos y la eliminación	102
4.3	Turnos y Aliados	106
4.4	Ejemplo de programación en Director	109
CAPÍTULO 5	Creación de páginas Web (Proyección Futura)	112
5.1	Diseño de Páginas	113
5.2	Creación de Animaciones	125
5.3	Creación de Protectores de pantalla	136
CONCLUSIONES		140

Introducción

Desde hace siglos el hombre ha buscado la manera de entretenerse y divertirse de varias formas. Una de ellas es por medio de juegos de mesa, en los cuales, ya sea por medio de cartas, fichas, dados, piezas, etc, se intenta derrotar al oponente en base a las instrucciones y reglas de dicho juego.

Dentro de este "mundo" de juegos existen muchos y muy variados, los hay para pasar el tiempo, para diversión, educativos, y colaborativos, siendo estos últimos los de mayor beneficio para los participantes.

En esos tiempos, cuando existían reinos y reyes, épocas de grandes conflictos bélicos, se desarrolló un juego basado en la realidad de esa época, las guerras, este juego es el Ajedrez.

Aquí dos reinos peleaban entre sí, realizaban avances de su ejército, y finalmente lograban eliminar al Rey oponente, con lo cual ganaban el juego. El ajedrez como tal, es un juego de estrategia en donde un grupo de piezas se mueven de forma estratégica, para lograr vencer al contrario; pero éste es un juego individual que no permite el trabajo en equipo con otros jugadores, por lo que ¿porqué no crear un juego en el que se trabaje en equipo?, ¿porqué no crear un juego con mayor versatilidad, haciéndolo más interesante?, ¿porqué no crear una variante de un juego que ha existido durante siglos?

Es así como nace Batel - Nah, un juego colaborativo, un juego que desarrollará el trabajo en equipo, que con la tecnología actual y el desarrollo de Internet se hace posible que los participantes en éste se encuentren en diferentes lugares, promoviendo así la comunicación, el trabajo en equipo y abriendo las oportunidades de relación fuera del propio entorno.

El problema es crear un portal en Internet capaz de llevar a cabo una partida con la intervención de 2, 3 o 4 jugadores sin importar en que país estén, un portal que sea capaz de permitir a los jugadores comunicarse y compartir sus ideas, un portal que una a personas de diferentes lugares del mundo con un objetivo común.

Con la realización de este proyecto se busca crear un software con el que se pueda jugar interactivamente por Internet, para lo cual se utilizarán diversas utilerías para crear las piezas y gráficas por computadora, las interfaces de comunicación, programas para validación de reglas, tutoriales, etc.

CAPÍTULO 1

Diseño conceptual del juego (Batel – Nah)

1.1 Idea Conceptual.

En el ajedrez existen algunas limitaciones tanto en el tablero como en las piezas, esto hace que la "guerra" que se está llevando a cabo tienda a ser, al paso de muchas partidas jugadas, repetitiva, ya que los jugadores van creando sus estrategias de ataque a su estilo, y por lo general, siempre son las mismas, al no haber interacción con un compañero, no existe alguien que aconseje, ayude, o cree nuevas variantes de ataque y defensa, motivo por el cual diseñé un juego, con los mismos principios pero con una mayor versatilidad, tanto del tablero como de las piezas, además de la interacción con otros jugadores.

El Ajedrez es un juego sólo para 2 personas, por lo que no hay interacción con nadie más dentro del mismo bando, es un juego individual.

En toda guerra existen los aliados de cada bando, por lo que ahora se agregan estos aliados a Batel- Nah para ofrecer varias modalidades de juego, con las cuales pueden jugar hasta 4 personas a la vez, por lo que la interacción y el trabajo en equipo serán de gran importancia para lograr la victoria.

Las modalidades son:

- Reino contra Reino para 2 personas
- Reino y Aliados contra Reino y Aliados para 2 personas
- Reino y Aliados contra Reino y Aliados para 3 personas
- Reino y Aliados contra Reino y Aliados para 4 personas

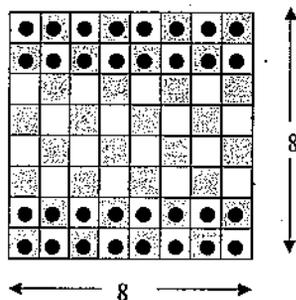
En el ajedrez cada cual tiene un turno, primero mueve un jugador y después mueve el otro de esta manera los ataques son uno a uno, por lo que la defensa debe prever un sólo ataque a la vez, en cambio, en Batel – Nah por cada ataque que uno realice, se recibirán dos ataques, así que la labor del aliado será de suma importancia, lo que generará un gran trabajo en equipo, creando estrategias de ataque y defensa de manera colaborativa.

1.2 Análisis comparativo entre el Ajedrez y Batel - Nah

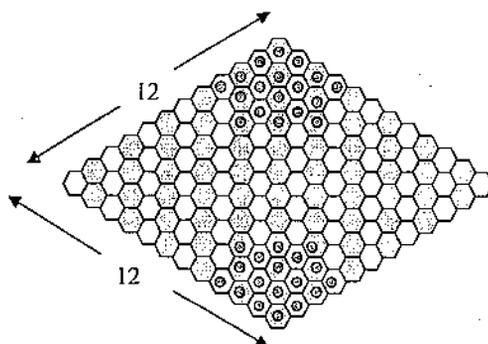
El Tablero.

El tablero del ajedrez está formado por espacios cuadrados siendo de 8 cuadros por 8 cuadros, esto da un total de 64 espacios, en total, si tomamos en cuenta que 32 ya están ocupados por las piezas, esto deja únicamente 32 espacios posibles para mover a las piezas, lo que resulta muy limitante.

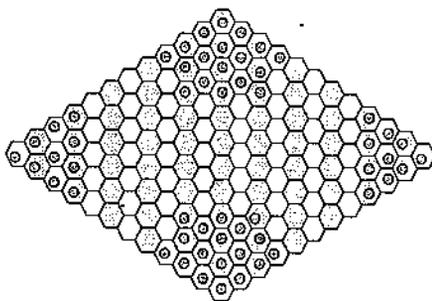
El tablero de Batel - Nah está formado por espacios hexagonales, estos espacios se unen y forman un área de 12 por 12 hexágonos, proporcionando un total de 144 espacios, de los cuales, en la modalidad de Reino contra Reino se ocupan 36 para las piezas, quedando libres 108 espacios para mover, y en la modalidad de 4 jugadores el total de piezas es de 56 dejando libres 56 espacios, otorgando de esta manera una mayor libertad de movimiento, así como el incremento de las opciones estratégicas de ataque.



Ajedrez – 32 espacios libres



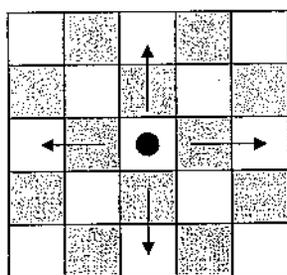
Batel – Nah (dos jugadores) – 108 espacios libres



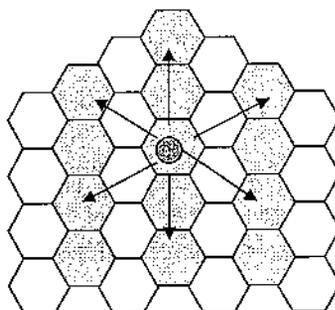
Batel – Nah (cuatro jugadores) – 56 espacios libres

El ajedrez, al estar formado por cuadrados, sólo tiene 4 posibles ángulos de ataque para la mayoría de las piezas, por lo que éstos son en cierta forma limitados. En Batel - Nah al ser

hexagonales los espacios, se incrementa el rango de movilidad ya que ahora hay 6 direcciones de movimiento, por ejemplo, existen los soldados, que son de igual forma que en el ajedrez, la pieza con menor valor, también son lentos, pero hay dos grandes diferencias, una, que se pueden mover en las 6 direcciones del hexágono, por lo que ahora podrán estar frente a frente con otra pieza y retroceder o esquivarla si es necesario, y la segunda, que el soldado puede eliminar piezas tanto hacia adelante como hacia atrás, lo que da un gran valor de ataque a esta pieza.



Ajedrez



Batel - Nah

El Ejército.

El ejército del Ajedrez está formado por 16 piezas, de las cuales 8 son peones, por lo que el 50% del ejército es lento, después siguen las torres (2), de gran valor, los caballos (2), los alfiles (2), el Rey y finalmente la Reina, que es la más versátil de todas las piezas.

En Batel - Nah existen 18 piezas : 8 son soldados lo que da un 44% de piezas débiles, hay templos (2), brujos (2), guerreros (2), rey (1), reina (1), y ahora el ejército tiene 2 piezas más, el Bufón y el Príncipe, por lo que aunado al hecho de que todas las piezas tienen mayor ataque y movilidad, el juego será más real , completo e interesante.

Rey (ajedrez) vs. Rey (Batel – Nah)

Ajedrez.

Movimiento.

El rey es una pieza que se mueve únicamente un espacio, lo cual da 4 posibles opciones de movimiento.

Ataque.

Su ataque está basado en su movimiento teniendo 4 posibles opciones de atacar.

Batel – Nah.

Movimiento

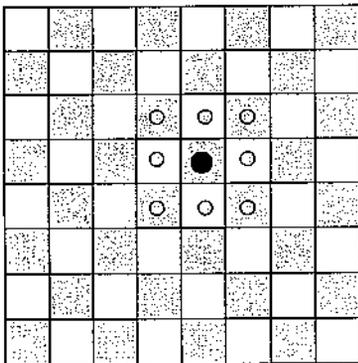
El rey se podrá mover un sólo espacio, aunque ahora tiene 6 opciones de movimiento

Ataque.

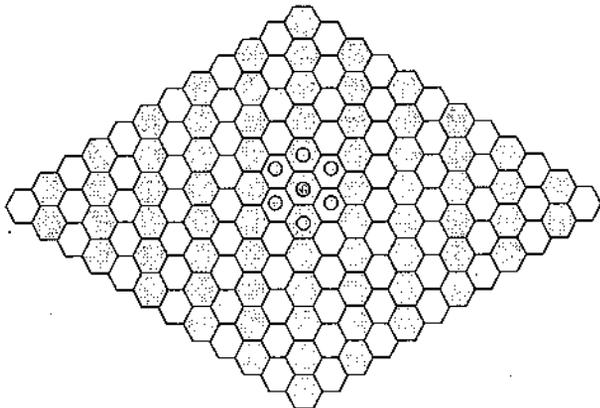
Tiene 6 posibles opciones de ataque basándose en su movimiento

Movimiento (puntos amarillos)

Ajedrez

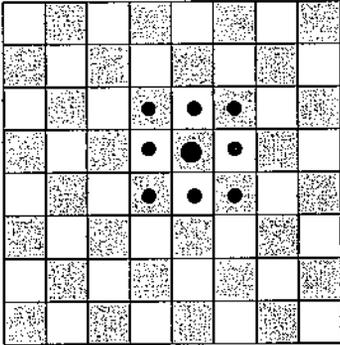


Batel – Nah

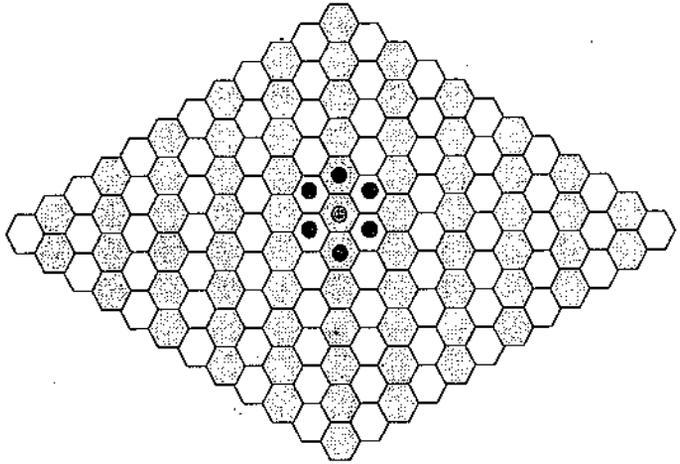


Ataque (puntos rojos)

Ajedrez



Batel - Nah



Reina (ajedrez) vs. Reina (Batel - Nah)

Ajedrez.

Movimiento.

La reina en el ajedrez es la pieza más importante en lo que a fuerza de ataque se refiere, ya que es la que tiene mayor movilidad, se puede desplazar horizontal, diagonal y verticalmente la cantidad de espacios que quiera. Dentro del objetivo principal del ajedrez, no está eliminar a la reina, por lo que cuando ésta es eliminada, sólo se pierde una pieza, claro, la más peligrosa, pero finalmente una pieza.

Ataque

La reina puede atacar en cualquiera de los espacios en los que puede caer según su movimiento.

Batel - Nah

Movimiento.

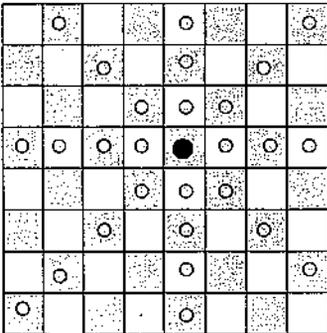
La importancia de la reina en Batel - Nah no es por su nivel de ataque, ya que éste está limitado, su importancia radica en el hecho de que el objetivo del juego es eliminar a las 2 piezas reales, por lo que al lograrlo se habrá conseguido el 50% del objetivo del juego.

Ataque.

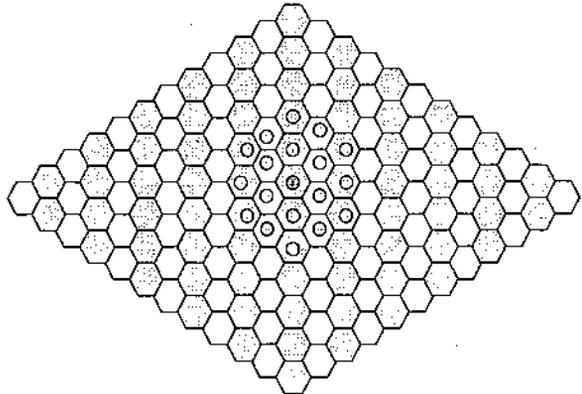
Como aquí la reina no es una pieza de pelea, su ataque es limitado, y está considerado en base a su movimiento, pudiendo eliminar oponentes sólo dentro de este rango; a corta distancia es muy peligrosa

Movimiento (puntos amarillos)

Ajedrez

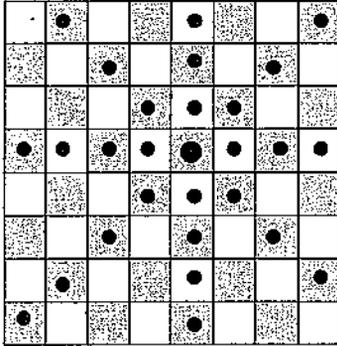


Batel - Nah

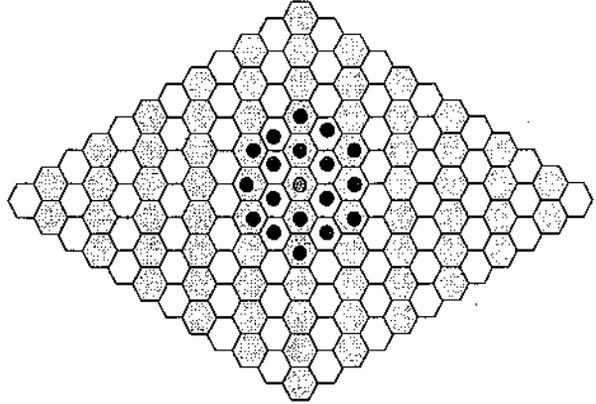


Ataque (puntos rojos)

Ajedrez



Batel - Nah



Torre (ajedrez) vs. Guerrero (Batel – Nah)

Ajedrez (Torre)

Movimiento.

La torre en el ajedrez es una de las piezas más importantes y poderosas, ya que tienen un gran rango de movimiento y ataque, su movimiento es horizontal y vertical y pueden llegar de un lado a otro del tablero en una tirada.

Ataque.

El ataque se realiza basándose en su movimiento.

Batel – Nah (guerrero)

Movimiento.

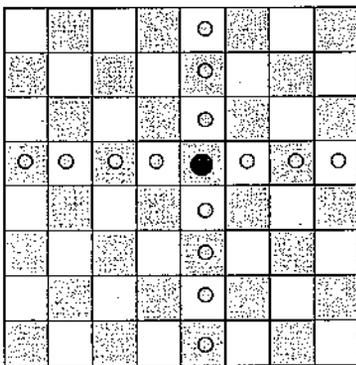
Al igual que la torre en el ajedrez el guerrero es una de las piezas más importantes y poderosas, ya que tienen un gran rango de movimiento y ataque, su movimiento es diagonal, debido a la forma del tablero y pueden llegar de un lado a otro del tablero en una tirada.

Ataque.

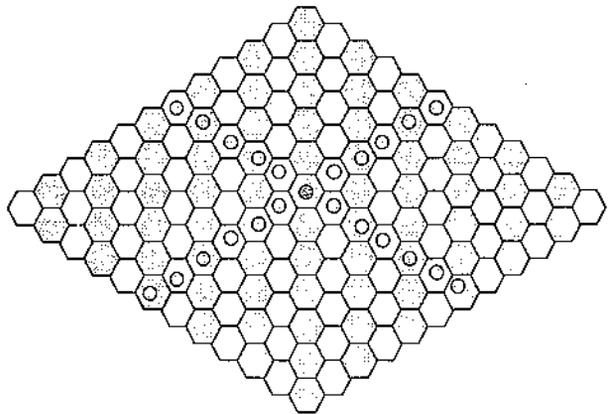
El ataque se realiza basándose en su movimiento

Movimiento (puntos amarillos)

Ajedrez

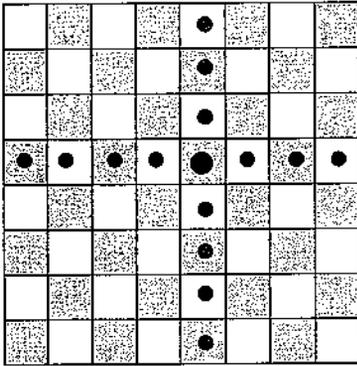


Batel - Nah

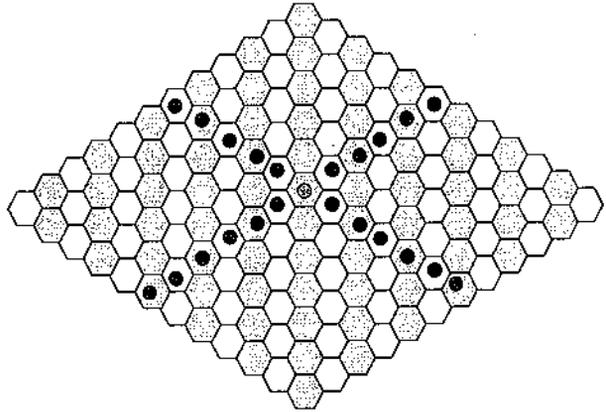


Ataque (puntos rojos)

Ajedrez



Batel - Nah



Caballo (ajedrez) vs. Brujo (Batel – Nah)

Ajedrez (caballo)

Movimiento.

El caballo es una pieza que tiene un desplazamiento lento, pero su rango de ataque lo hace peligroso, su movimiento es de tal forma que puede brincar otras piezas.

Ataque

El ataque está dado en función de su movimiento, por lo que al eliminar a un oponente deberá caer exactamente en el espacio correspondiente.

Al igual que en su movimiento, en el ataque sólo tiene 8 posibles puntos de eliminación del oponente.

Batel – Nah (brujo)

Movimiento.

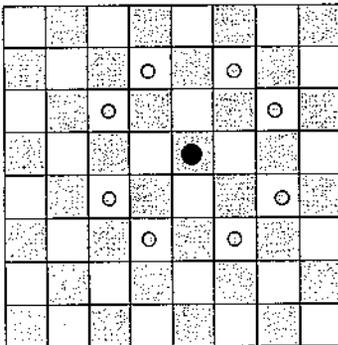
El movimiento es básicamente el mismo que el del caballo del ajedrez, pero ahora, al ser hexagonales los espacios del tablero, se incrementan los posibles puntos de eliminación a 12, lo que lo hace más peligroso. El brujo también podrá brincar a otras piezas.

Ataque

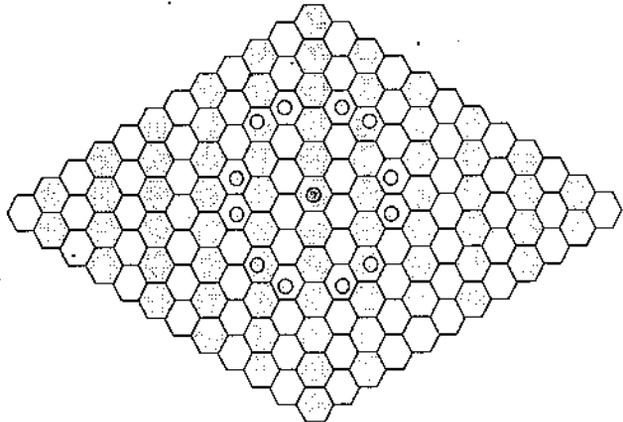
El ataque se da en función de su movimiento, por lo que al eliminar a un oponente deberá caer exactamente en el espacio correspondiente.

Movimiento (puntos amarillos)

Ajedrez

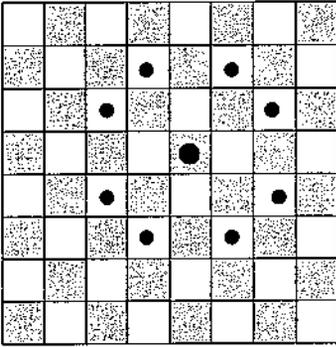


Batel - Nah

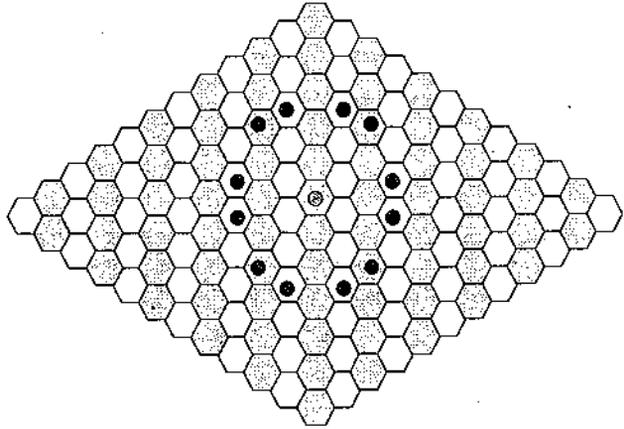


Ataque (puntos rojos)

Ajedrez



Batel - Nah



Alfil (ajedrez) vs. Templo (Batel – Nah)

Ajedrez (alfil)

Movimiento.

El movimiento del alfil es en forma diagonal en dirección a cada una de las esquinas del cuadrado sobre el que está colocado, existe un alfil que se desplaza por los cuadros negros y otro que se desplaza por los cuadros blancos. Una vez que uno de ellos es eliminado, el color sobre el que estaba, queda descubierto, ya que el otro alfil no puede cambiar de color, por lo que un solo alfil pierde el 50% de los espacios posibles del tablero.

Ataque

El ataque se realiza en función de su movimiento, por lo que al eliminar a un oponente deberá caer exactamente en el espacio correspondiente.

Batel – Nah (templo)

Movimiento

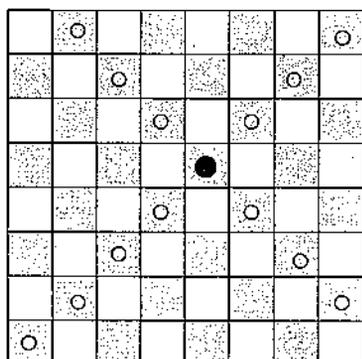
El movimiento del templo es en forma horizontal y vertical con respecto al tablero, lo que provoca que al igual que el alfil, tiene que seguir su trayectoria por un solo color, pero a diferencia de este último, el templo puede cambiarse de color, con lo que podrá reemplazar a un templo eliminado, moviéndose y atacando sobre la misma trayectoria que el otro templo tenía.

Ataque

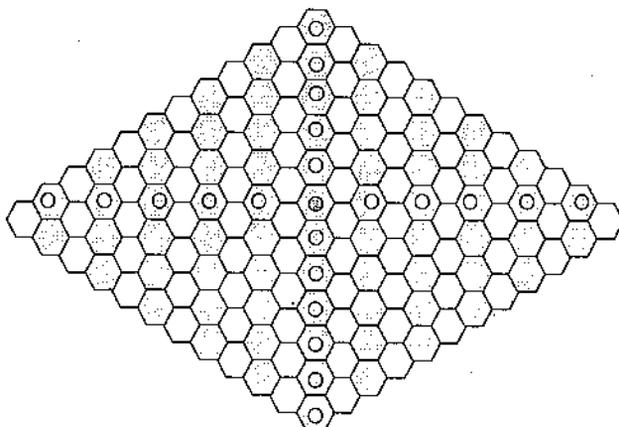
El ataque está dado en función de su movimiento, por lo que al eliminar a un oponente deberá caer exactamente en el espacio correspondiente.

Movimiento (puntos amarillos)

Ajedrez

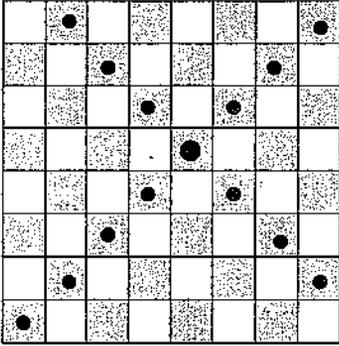


Batel - Nah

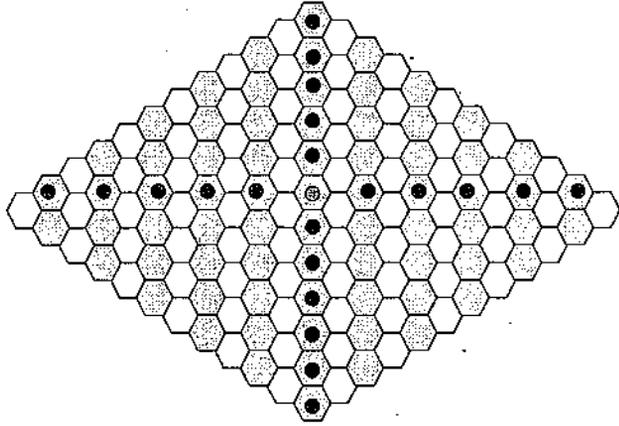


Ataque (puntos rojos)

Ajedrez



Batel - Nah



Peón (ajedrez) vs. Soldado (Batel – Nah)

Ajedrez (peón)

Movimiento

El peón en el ajedrez es la pieza mas débil, sólo se mueve hacia delante, uno o dos espacios en su línea inicial y después un único espacio, si dos peones se encuentran de frente, no se podrán mover hasta que otras piezas intervengan y permitan ser eliminadas por alguno o eliminando a uno de ellos, lo que resulta muy limitante

Ataque

Para eliminar a su oponente, el peón debe realizar su movimiento en forma diagonal un solo espacio hacia delante, no puede atacar hacia atrás, por lo que es una pieza de muy poco riesgo.

Batel – Nah (soldado)

Movimiento

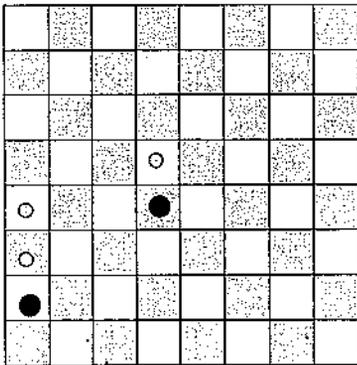
A diferencia del peón, el soldado es una pieza mucho más versátil y agresiva, se puede mover un espacio en las seis direcciones del hexágono, pudiendo retroceder y esquivar a los oponentes, e ir y venir por todo el tablero.

Ataque

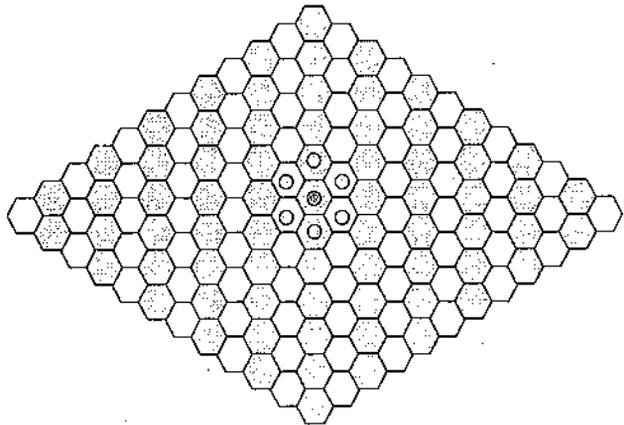
Al ser más completo su movimiento también debe de aumentar su grado de ataque, por lo que esta pieza puede atacar hacia delante y hacia atrás, con lo que ahora el enemigo tendrá que cuidarse de un ataque mayor.

Movimiento (puntos amarillos)

Ajedrez

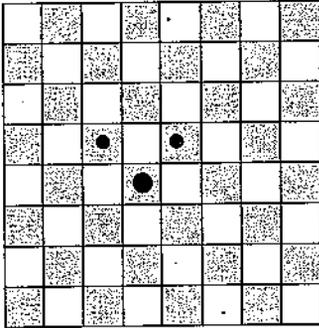


Batel - Nah

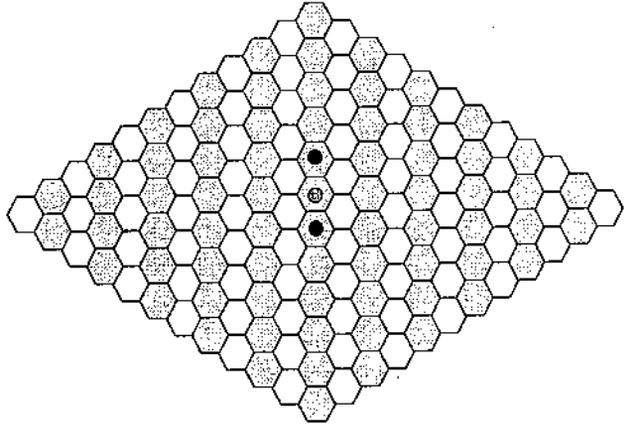


Ataque (puntos rojos)

Ajedrez



Batel - Nah



¡ Dos piezas Nuevas !

Batel – Nah cuenta con dos piezas nuevas para hacer al ejército más fuerte y completo, estas piezas son: El Príncipe y El Bufón

El príncipe es la pieza más fuerte de Batel – Nah, ya que se mueve en todas direcciones la cantidad de espacios que quiera, y elimina en base a este movimiento.

El bufón se mueve de igual forma que la reina, pero a diferencia de ésta, va eliminado oponentes de una forma especial, mientras que todas las demás piezas ocupan la posición del oponente eliminado, el bufón se coloca en el espacio siguiente en dirección de la pieza, y su movimiento de ataque es en forma diagonal.

Más adelante se mostrarán las reglas de movimientos y eliminación del Príncipe y el Bufón.

1.3 Movimientos

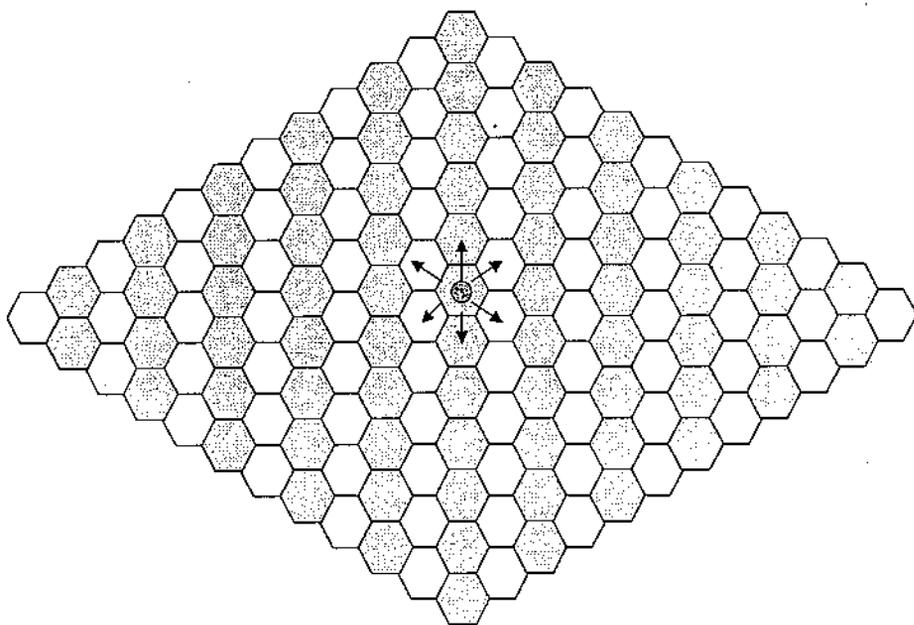
Cuando empecé a desarrollar este juego, una de las situaciones que más me interesaba, era hacerlo más emocionante y completo, por lo que al momento de diseñar las piezas fue muy importante no romper totalmente con el esquema ya establecido por el ajedrez, y a su vez, incrementar la fuerza del ejército.

Debido a la forma de los espacios del tablero, esta mejora en los movimientos se vio directamente mejorada, pero, piezas como el peón y el alfil, definitivamente tenían que cambiar.

En esta sección se mostrará e ilustrará más a fondo el movimiento de cada una de las piezas.

El Rey.

El rey es una pieza que por tradición debe de seguir con sus mismas facultades, es una pieza lenta de un ataque muy corto. Como el objetivo del juego es eliminar a las dos piezas reales (rey y reina), el jugador podrá decidir a cual de los dos dejará morir, en caso necesario, y si tomamos en cuenta que la reina tiene mayor movilidad y ataque, es lógico pensar que primero caerá el rey, sin que esto implique la pérdida de la partida. El rey se mueve un sólo espacio en cualquier dirección.

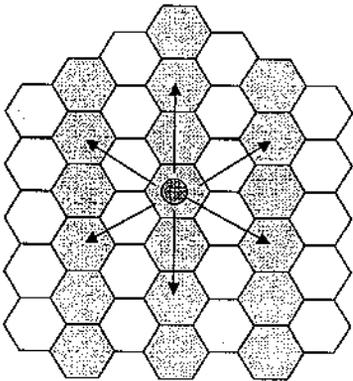


La Reina

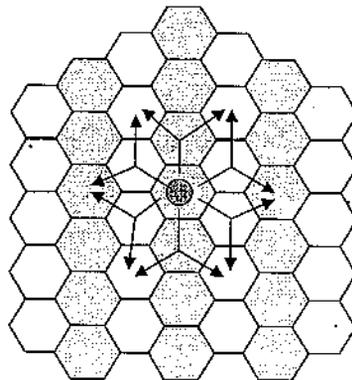
La reina es ahora el 50% del objetivo del juego, por lo que no podía conservar mucha movilidad, aunque su movimiento a corta distancia es muy completo, pudiendo abarcar un total de 18 posibles lugares de colocación.

Se mueve un total de 2 espacios, estos pueden ser en línea recta o no, en las siguientes gráficas se mostrarán las diferentes variantes de movimiento.

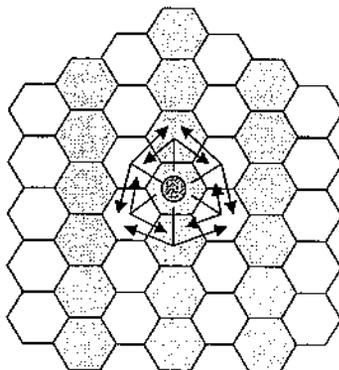
Dos espacios en línea recta



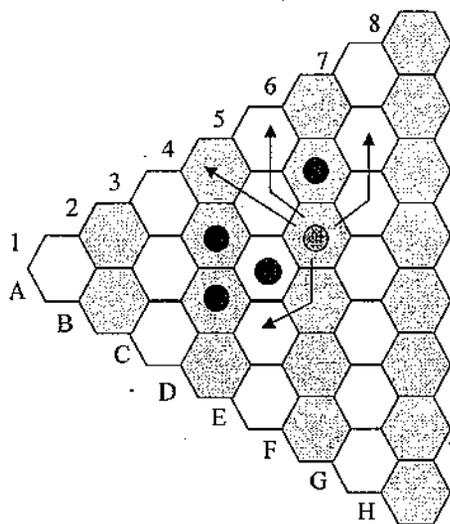
Un espacio hacia una dirección y otro en diagonal hacia delante, con respecto a la dirección del primer movimiento



Un espacio hacia una dirección y otro en diagonal hacia atrás, con respecto a la dirección del primer movimiento



Una regla muy importante que la reina siempre debe de cumplir es pasar por 2 espacios en su movimiento, en la figura anterior con el movimiento hacia atrás pareciera que sólo se movió un espacio desde su posición original, pero no es así, primero tuvo que cumplir con el avance hacia una dirección y después el retroceso, esto se ilustra en la siguiente figura.



Aquí la reina puede pasar entre las piezas que están ubicadas en las posiciones B5 y B3, puede esquivarlas como a las piezas B5 y C2, pero no puede colocarse en la posición B4, ya que ahí no completa su segundo movimiento.

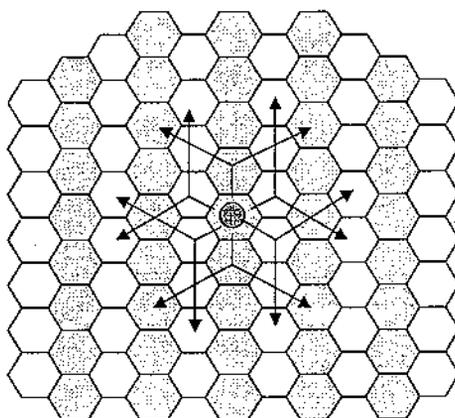
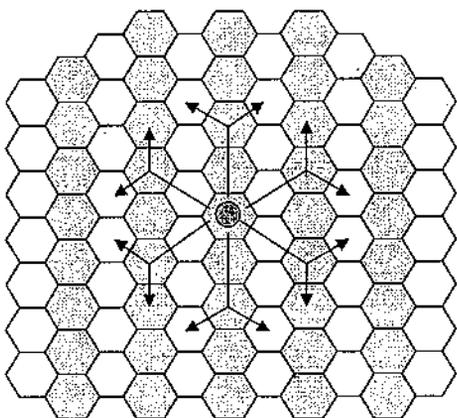
El Brujo

Esta pieza se mueve de una manera lenta, pero puede abarcar varios espacios a su alrededor, lo que lo hace muy peligroso

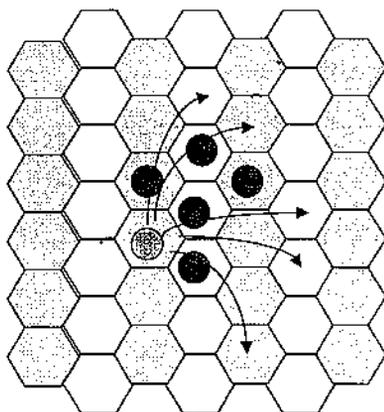
Su movimiento se basa en las siguientes reglas:

Dos espacios en una dirección y un espacio en dirección diagonal hacia adelante con respecto al primer movimiento.

Un espacio en una dirección y dos espacios en dirección diagonal hacia adelante con respecto al primer movimiento.



El brujo es la única pieza que en su movimiento puede saltar a otras piezas, tanto aliadas como enemigas. (El bufón puede saltar otras piezas pero sólo en su ataque).

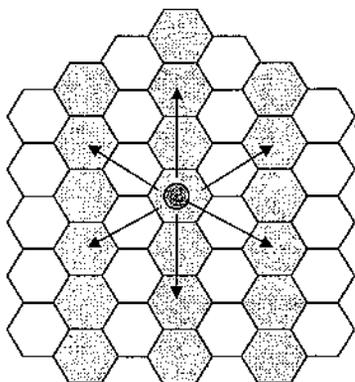


El Bufón

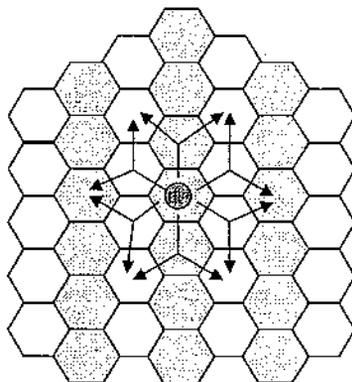
Esta es una nueva pieza que complementará el ataque del ejército, su movimiento es igual que el de la reina, pero su ataque es diferente.

Se mueve un total de 2 espacios, estos pueden ser en línea recta o no, en las siguientes gráficas se mostrarán las diferentes variantes de movimiento.

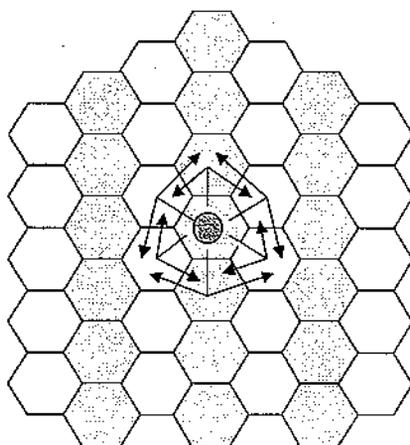
Dos espacios en línea recta



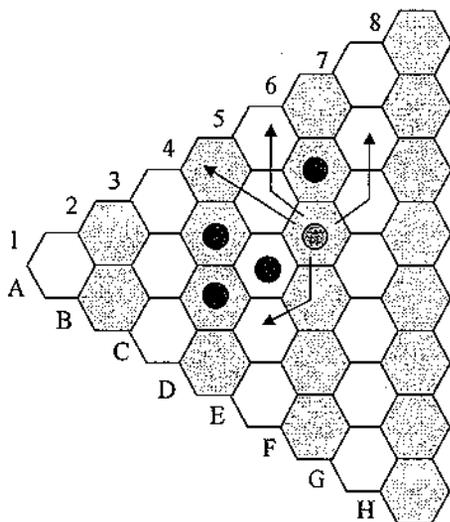
Un espacio hacia una dirección y otro en diagonal hacia adelante, con respecto a la dirección del primer movimiento



Un espacio hacia una dirección y otro en diagonal hacia atrás, con respecto a la dirección del primer movimiento



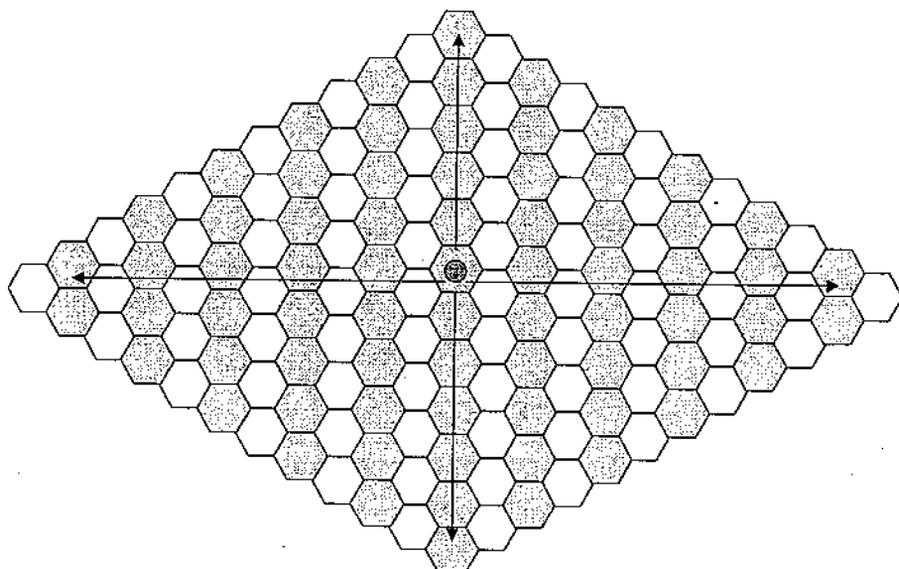
Una regla muy importante que siempre se debe de cumplir es pasar por 2 espacios en su movimiento, en la figura anterior con el movimiento hacia atrás pareciera que sólo se movió un espacio desde su posición original, pero no es así, primero tuvo que cumplir con el avance hacia una dirección y después el retroceso, esto se ilustra en la siguiente figura.



Aquí el bufón puede pasar entre las piezas que están ubicadas en las posiciones B5 y B3, puede esquivarlas como a las piezas B5 y C2, pero no puede colocarse en la posición B4, ya que ahí no completa su segundo movimiento.

El Templo

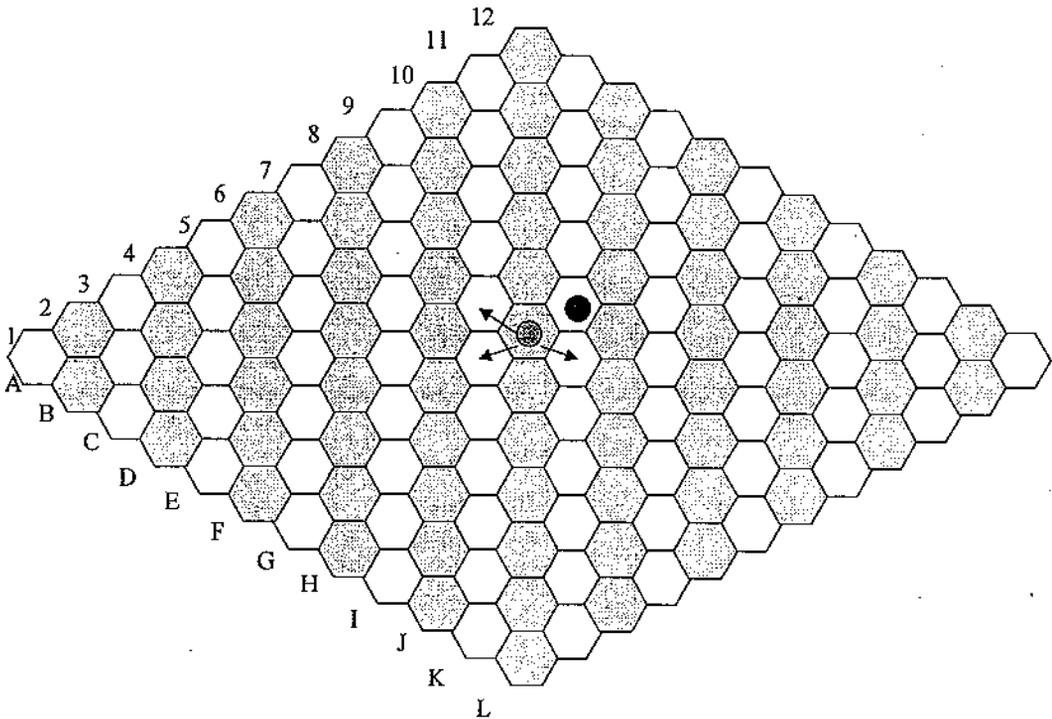
Esta pieza es muy importante, ya que tiene una gran movilidad, su movimiento es horizontal y vertical, lo que hace que únicamente pueda ir en un sólo color, al igual que el alfil en el ajedrez, pero ahora hay una gran diferencia, ya que el Templo puede cambiarse de color si así lo desea, este cambio se considera como una tirada, por lo que después de realizarlo seguirá el turno del siguiente jugador.



Como se muestra en la figura, puede abarcar una gran cantidad de espacios pero su trayectoria va sobre un mismo color, en este caso el espacio oscuro.

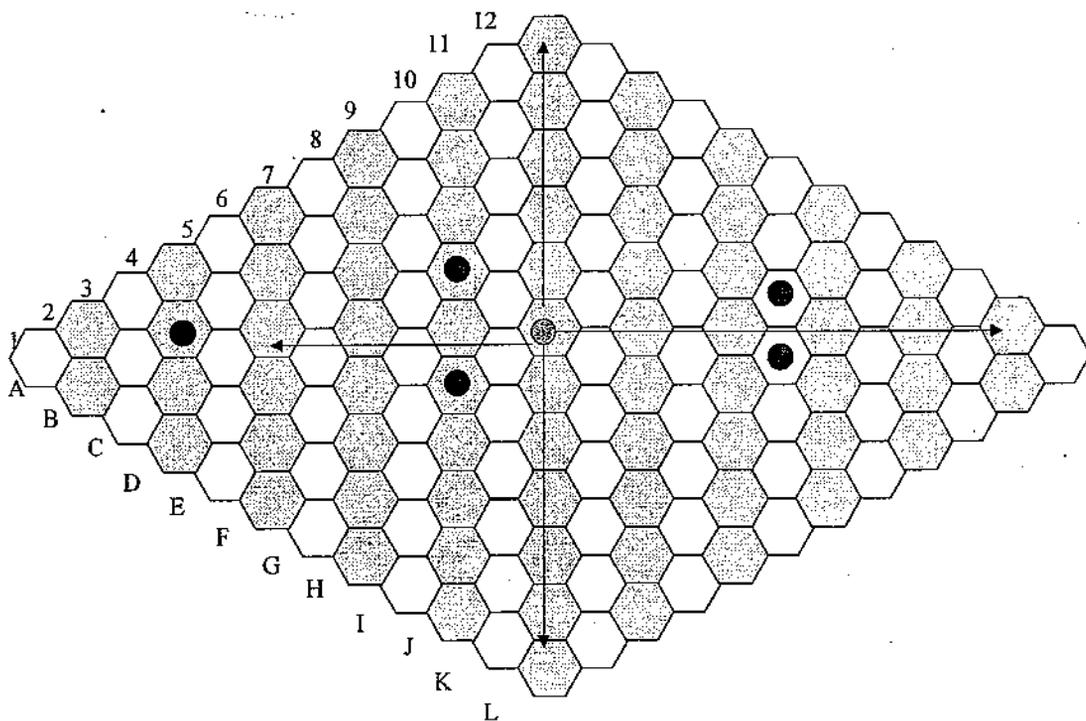
Cambio de color.

Cuando un jugador quiera puede cambiar de color de trayectoria a su Templo, no importando el color en que este colocado el otro. Esto se realiza de la siguiente forma: el Templo se moverá un sólo espacio hacia cualquiera de los espacios del color diferente al de su posición actual, siempre y cuando éste este libre, el Templo no puede eliminar oponentes en su movimiento de cambio de color.



En este caso el Templo está en la posición F7 que es un espacio oscuro, se podrá cambiar de color a los espacios F6, E7, G7, y no podrá cambiarse al espacio F8, ya que no puede eliminar piezas con este movimiento en especial, una vez que se ha realizado el cambio seguirá el turno del siguiente jugador. Realizado el cambio, el Templo deberá de respetar su nuevo color de trayectoria y sólo se moverá por el nuevo color, podrá realizar los cambios que quiera durante el juego.

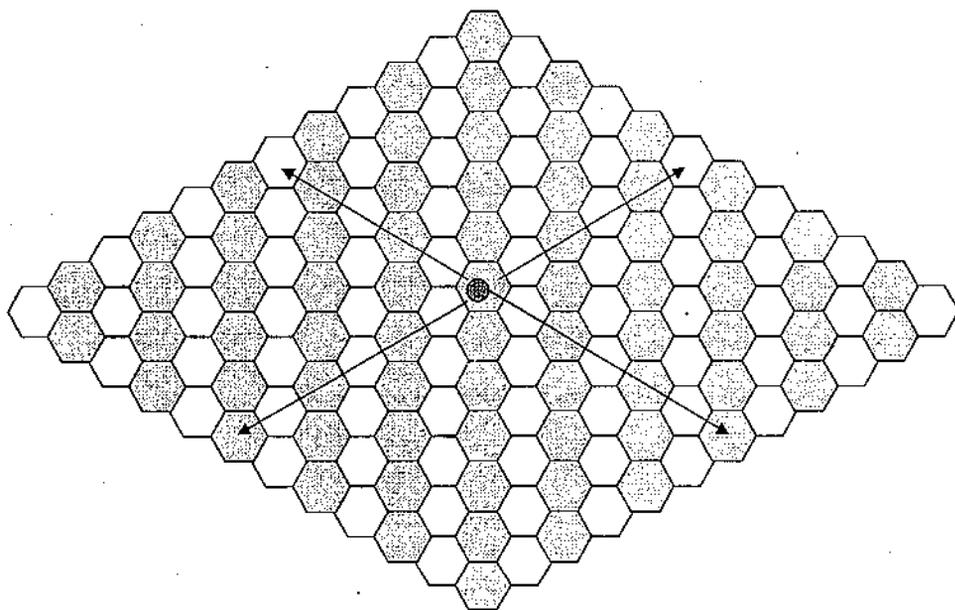
Debido a la forma del tablero, puede presentarse una confusión en el movimiento del Templo entre otras piezas, por lo que se ilustrará en la siguiente figura.



En la figura, el Templo puede pasar por los espacios E6, D5, y C4 hacia la izquierda, pasando entre otras dos piezas, hacia arriba por E8, D9, C10, B11, A12, hacia abajo por G6, H5, I4, J3, K2, L1, hacia la derecha por G8, H9, I10, J11, K12, es importante observar que aunque las piezas que están colocadas en las posiciones H10, e I9, están en espacios contiguos, éstas no obstaculizan el desplazamiento del Templo, ya que no están en el color de su trayectoria, como en el caso de la pieza que está en B3.

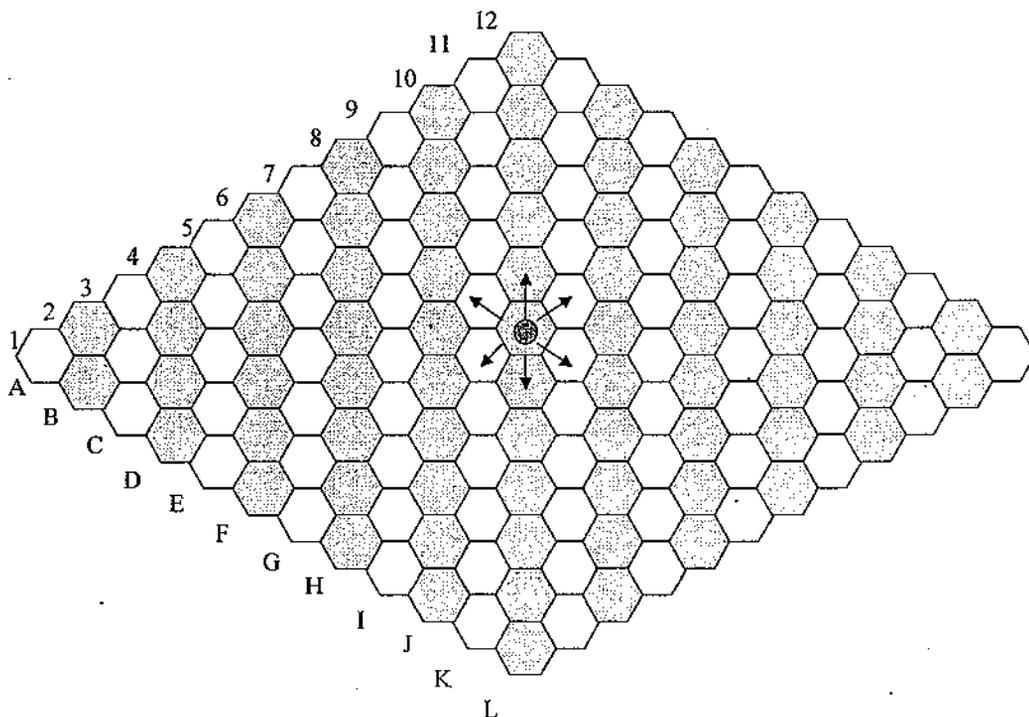
El Guerrero.

Es una de las piezas más importantes, ya que al igual que el Templo tiene gran movilidad, pero como el movimiento es diagonal, podrá abarcar todos los espacios del tablero sin importar el color.



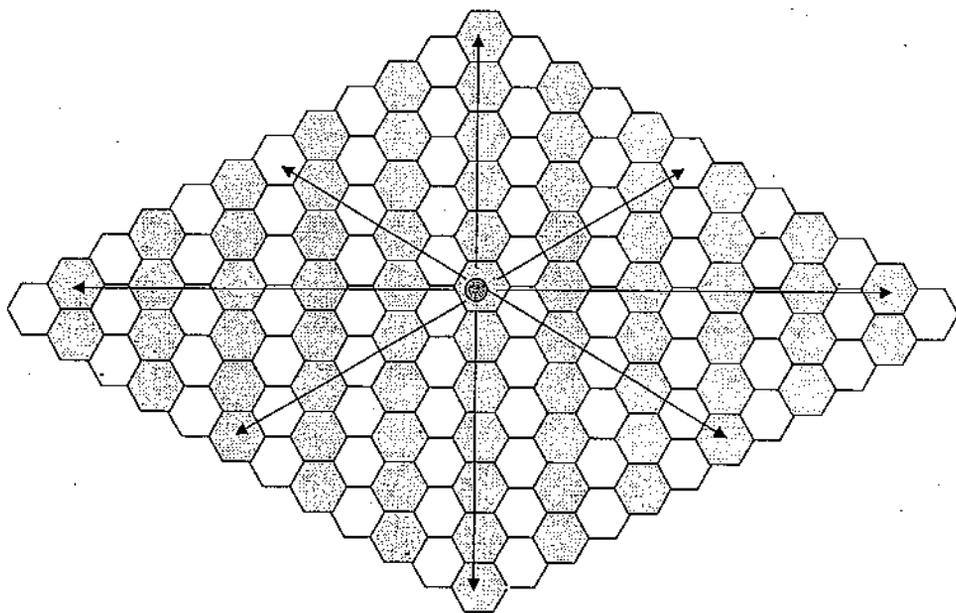
El Soldado.

El soldado es una pieza lenta pero peligrosa de cerca se mueve un sólo espacio en cualquier dirección, por lo que los posibles espacios donde podrá colocarse estando en el espacio F7 son: E8, E7, F6, G6, G7, F8, por lo que al poder retroceder se vuelve más versátil.



El Príncipe.

Esta es la pieza más importante, ya que es la que tiene la mayor movilidad, se podrá mover en dirección horizontal, vertical y diagonal la cantidad de espacios que quiera.

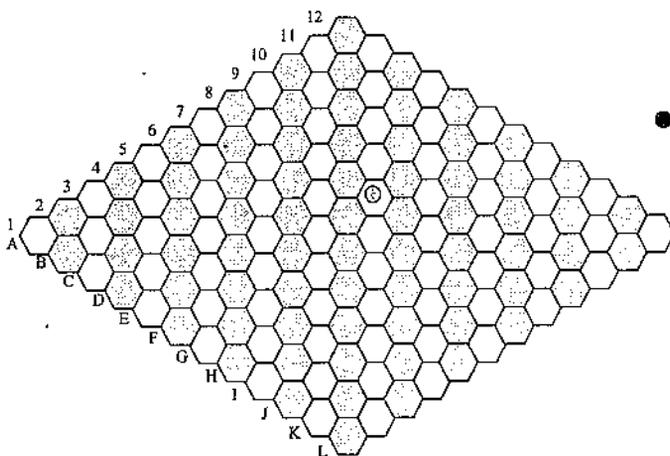
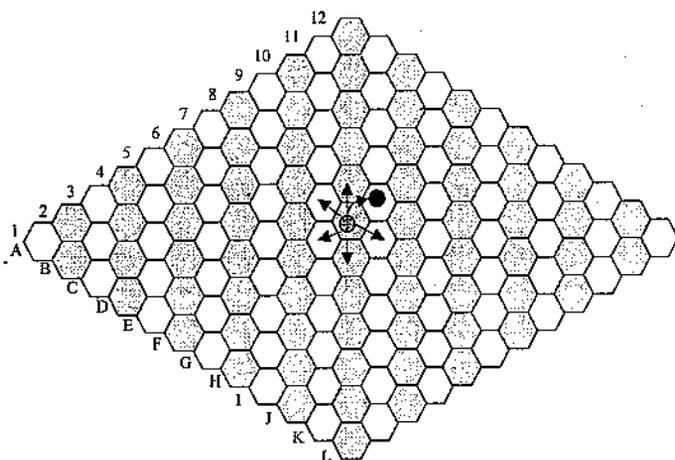


1.3 Eliminación

Al igual que los movimientos, los ataques son más completos, debido a la forma de los espacios del tablero, hay más direcciones de ataque, dándole al juego un mejor nivel de competitividad haciendo que los jugadores deban cuidarse más.

El Rey.

El rey eliminará oponentes según su movimiento, por lo que tiene 6 posibles opciones de ataque, al eliminar a un oponente ocupará el espacio de la pieza. En la figura el Rey va a eliminar a la pieza que está en la posición F8

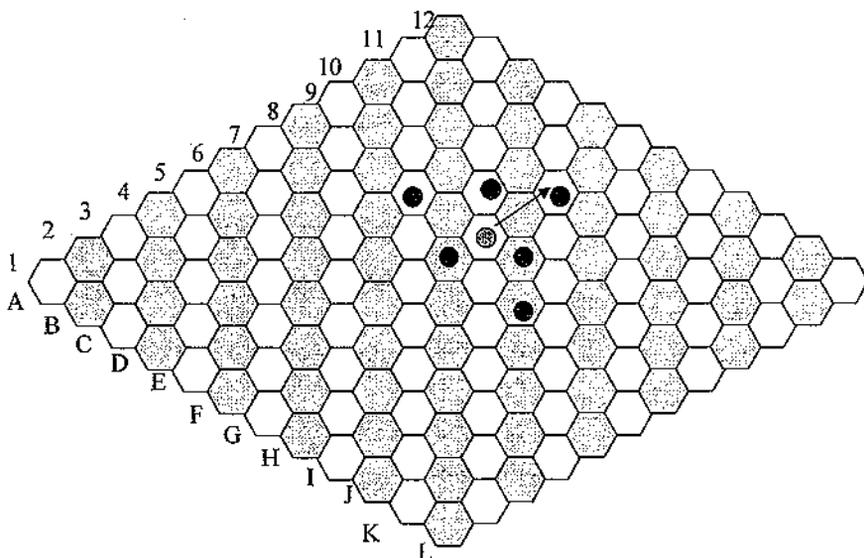


Una vez que el Rey elimina al oponente ocupa su lugar, en este caso se deberá colocar en la posición F8 y quitar del tablero la pieza eliminada.

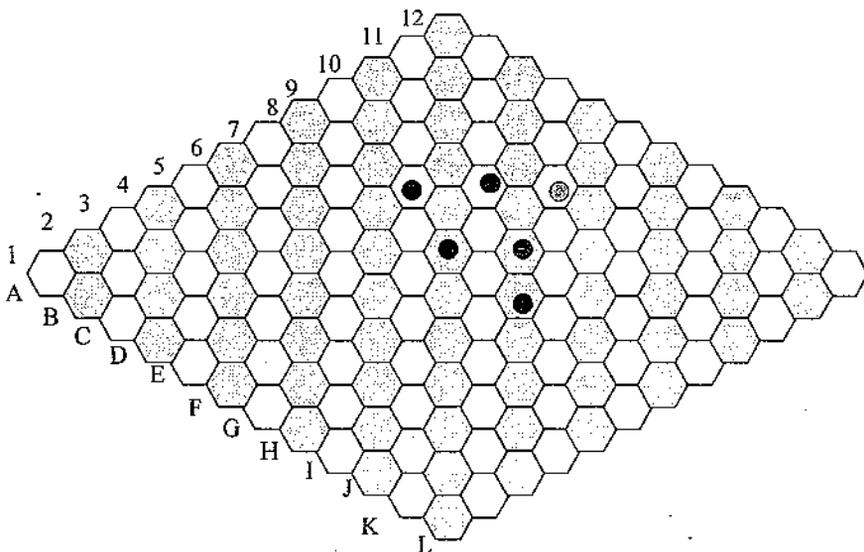
La Reina

Su ataque es basándose en su movimiento, por lo que de igual forma deberá cumplir con la regla de recorrer dos espacios en su ataque, sea cual sea su forma o dirección.

En la siguiente gráfica la reina podría eliminar a cualquiera de las piezas rojas, en este caso se eliminará la pieza que está en F10.



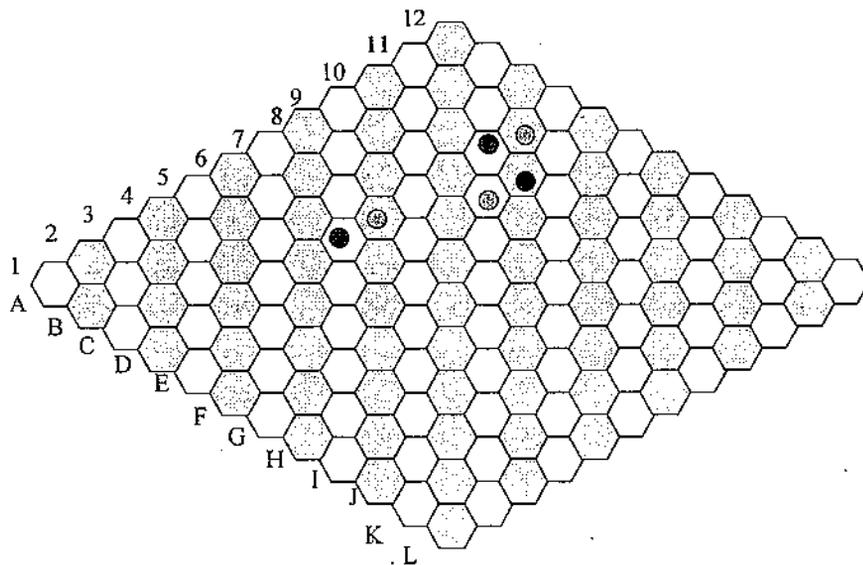
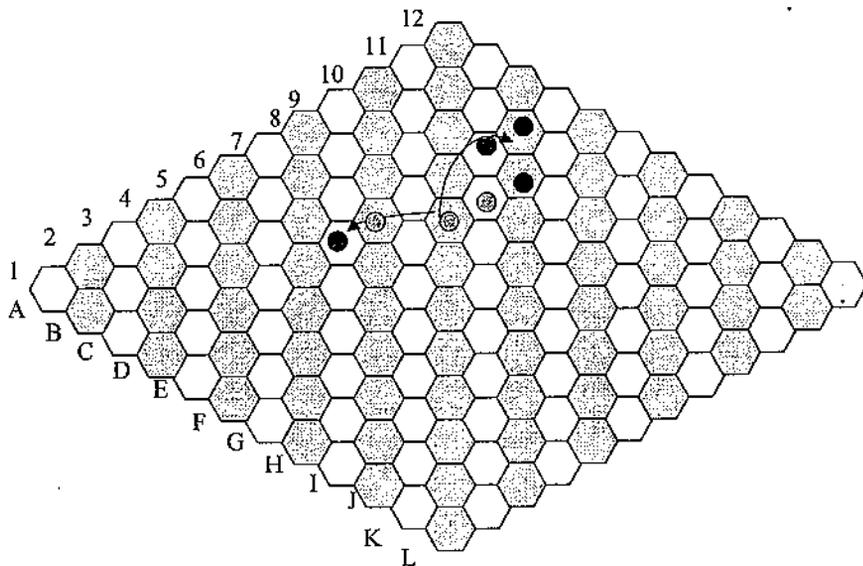
Aquí la reina puede pasar entre las piezas que están ubicadas en las posiciones E9 y G83, y eliminar a la pieza que está en la posición F10



La reina ocupará el espacio de la pieza eliminada, y ésta última se retirará del tablero.

El Brujo

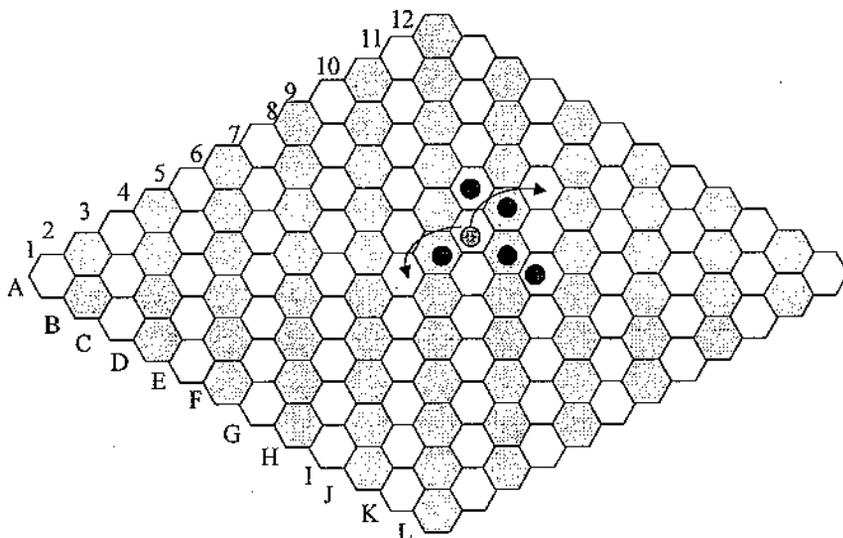
Esta pieza se mueve de una manera lenta, pero puede abarcar varios espacios a su alrededor, lo que lo hace muy peligroso. El brujo en su movimiento puede saltar a otras piezas, tanto aliadas como enemigas, por lo que de igual forma lo hará en su ataque. En la siguiente figura el brujo que está en la posición E8, puede eliminar a las piezas que están en las posiciones D6 y D11, en la primera no saltará ninguna pieza, en cambio en la segunda tendrá que saltar a dos piezas aliadas.



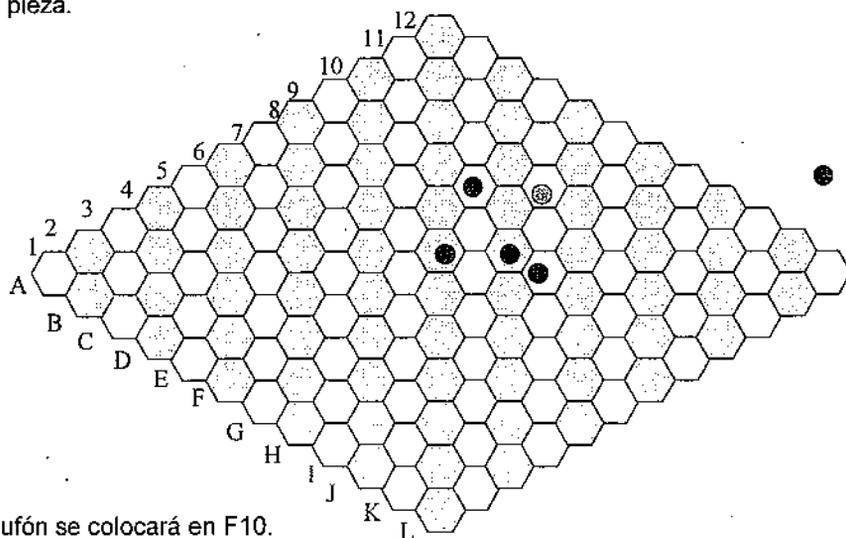
Se colocará el Brujo en el lugar de la pieza eliminada.

El Bufón

Esta es una nueva pieza que complementará el ataque del ejército, su movimiento es igual que el de la reina, pero su ataque es diferente. La manera en que eliminará al oponente es colocándose junto a la pieza a eliminar en forma diagonal; el bufón es la única pieza que al atacar no se coloca en el espacio de la pieza atacada, sino en el espacio siguiente en la dirección del movimiento del ataque, y sólo podrá hacerlo si el espacio donde caerá se encuentra libre.



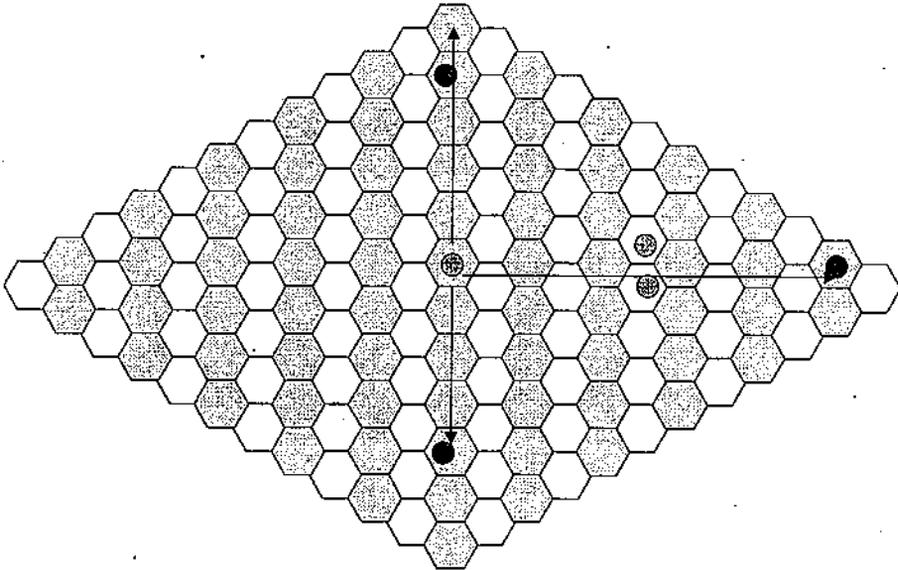
En la grafica el bufón podrá atacar a las piezas que se encuentran en la posición F9 y F7, y aunque está en posición diagonal contigua a la pieza que está colocada en G8, no podrá eliminarla ya que al atacar debería de colocarse en el espacio H8 el cual está ocupado por otra pieza.



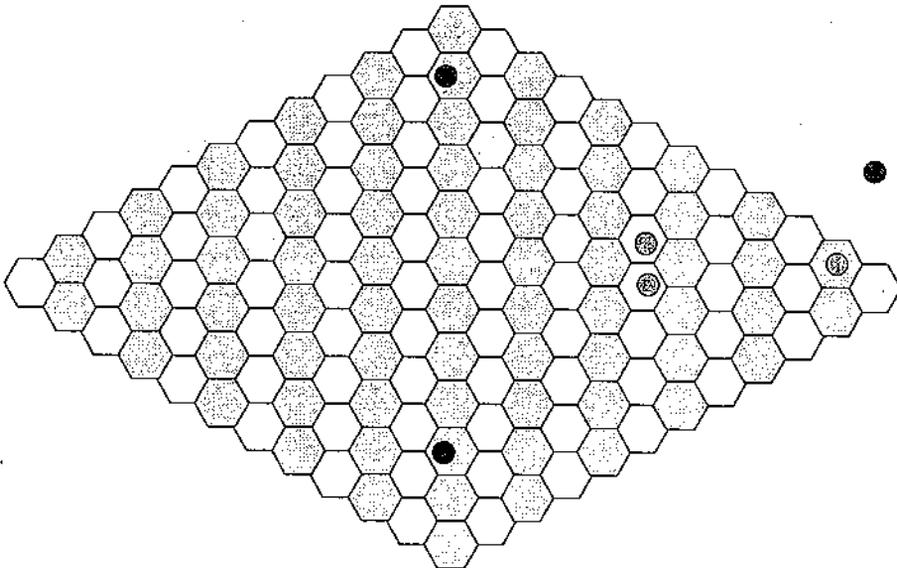
El bufón se colocará en F10.

El Templo

Eliminará a las piezas oponentes según su movimiento y se ubicará en el lugar de la pieza eliminada. Cuando realice el cambio de color de trayectoria, NO podrá eliminar oponentes .

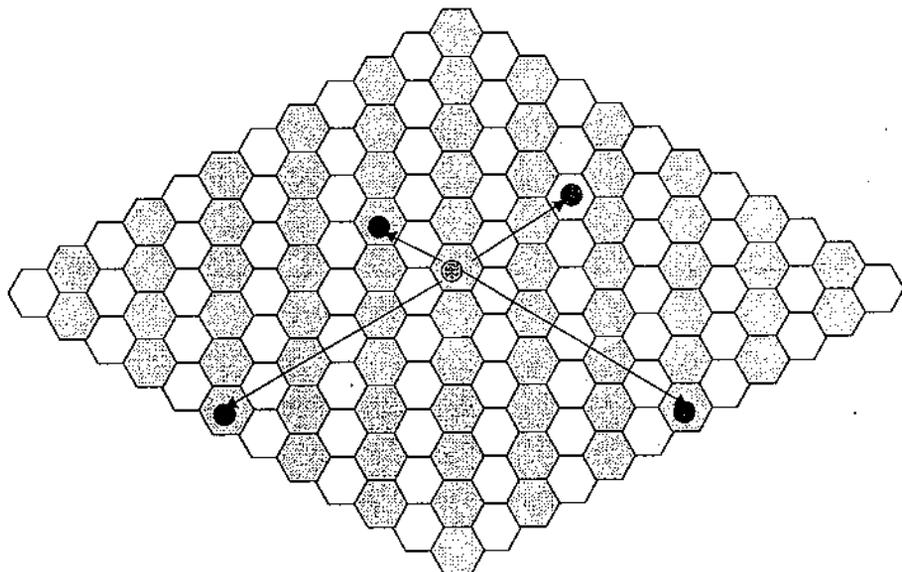


Como se muestra en la figura, puede abarcar una gran cantidad de espacios pero su trayectoria va sobre un mismo color, en este caso el espacio oscuro. Se colocará en el lugar de la pieza eliminada.

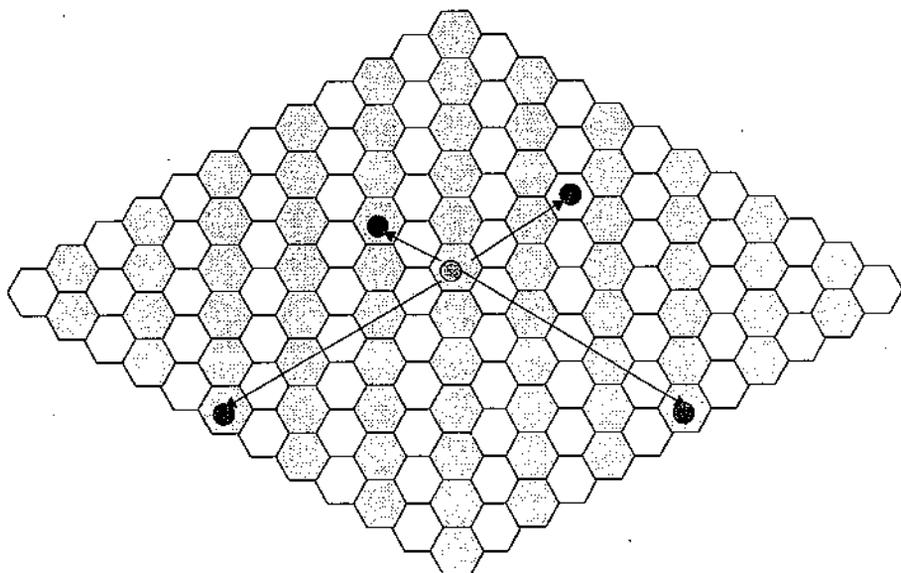


El Guerrero.

Es una de las piezas más importantes, puede atacar en dirección diagonal, por lo que llega a muchos espacios tanto blancos como negros.

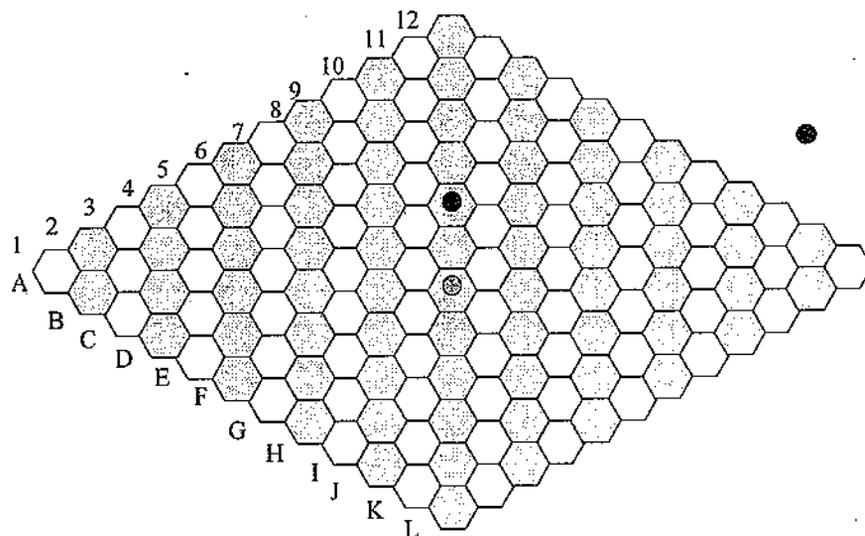
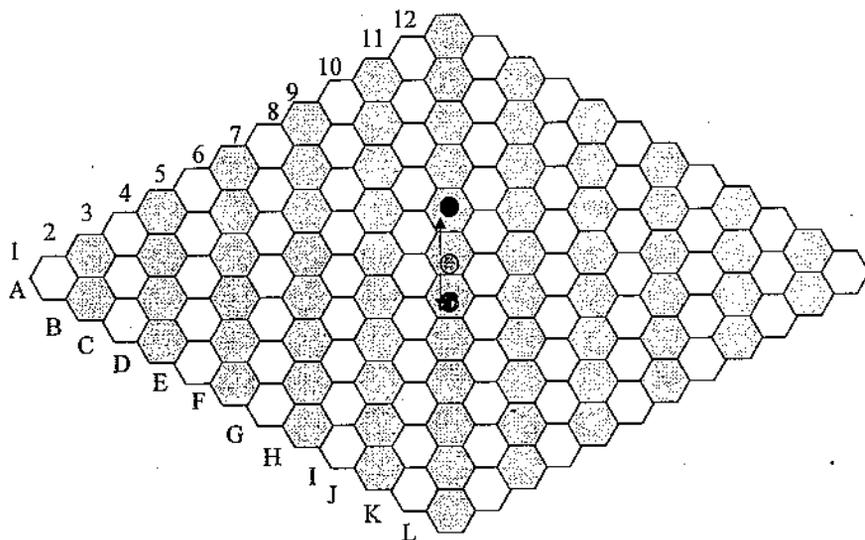


El guerrero se colocará en el espacio de la pieza eliminada.



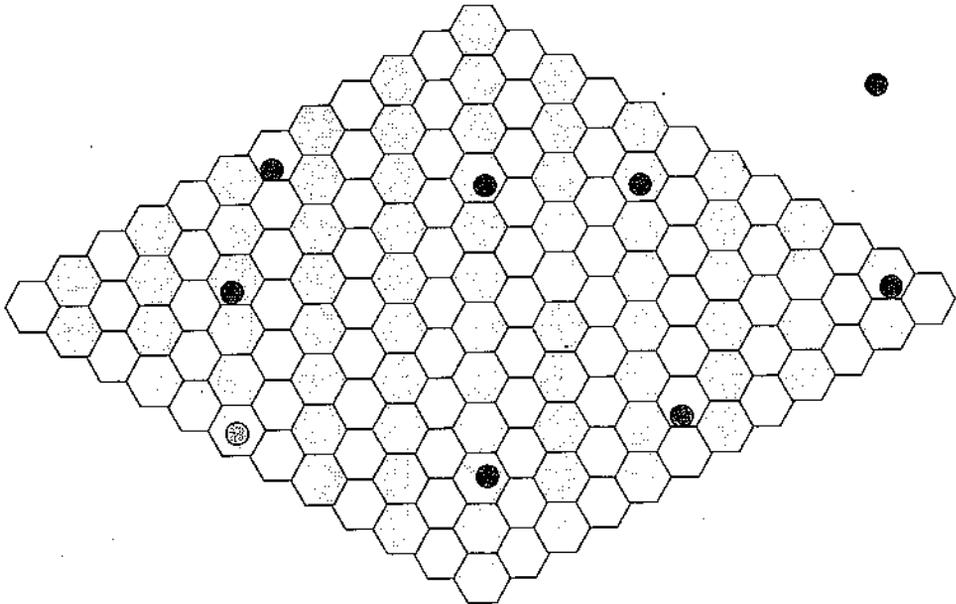
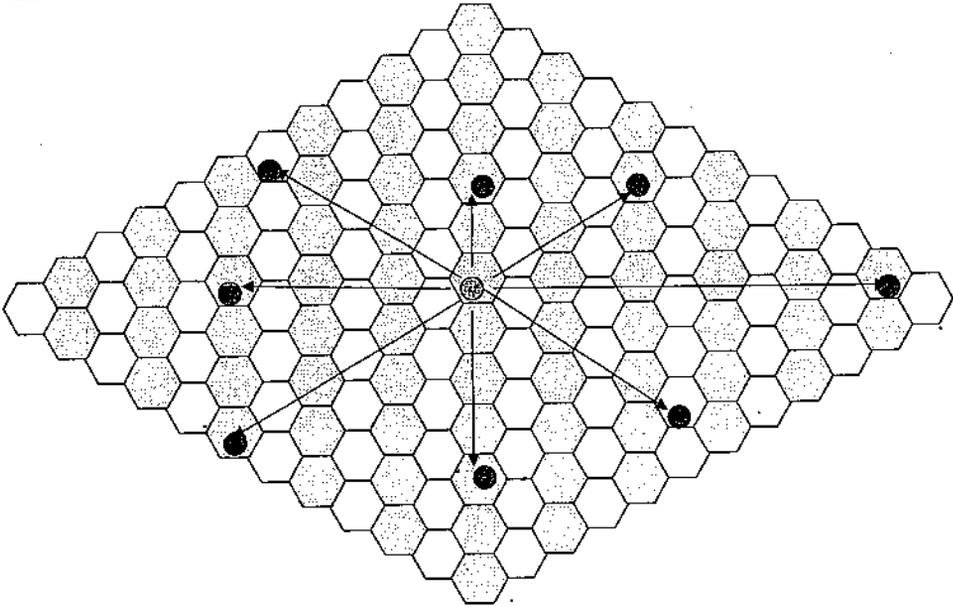
El Soldado.

El soldado es una pieza lenta pero con su nuevo rango de movimiento y ataque es mucho más peligrosa. Puede atacar hacia delante y hacia atrás, puede retroceder y moverse de lado.



El Príncipe.

Esta pieza tiene un gran rango de ataque, por lo que es la más peligrosa de todas, ataca hacia 6 direcciones, la cantidad de espacios que quiera, al atacar ocupará la posición de la pieza eliminada.

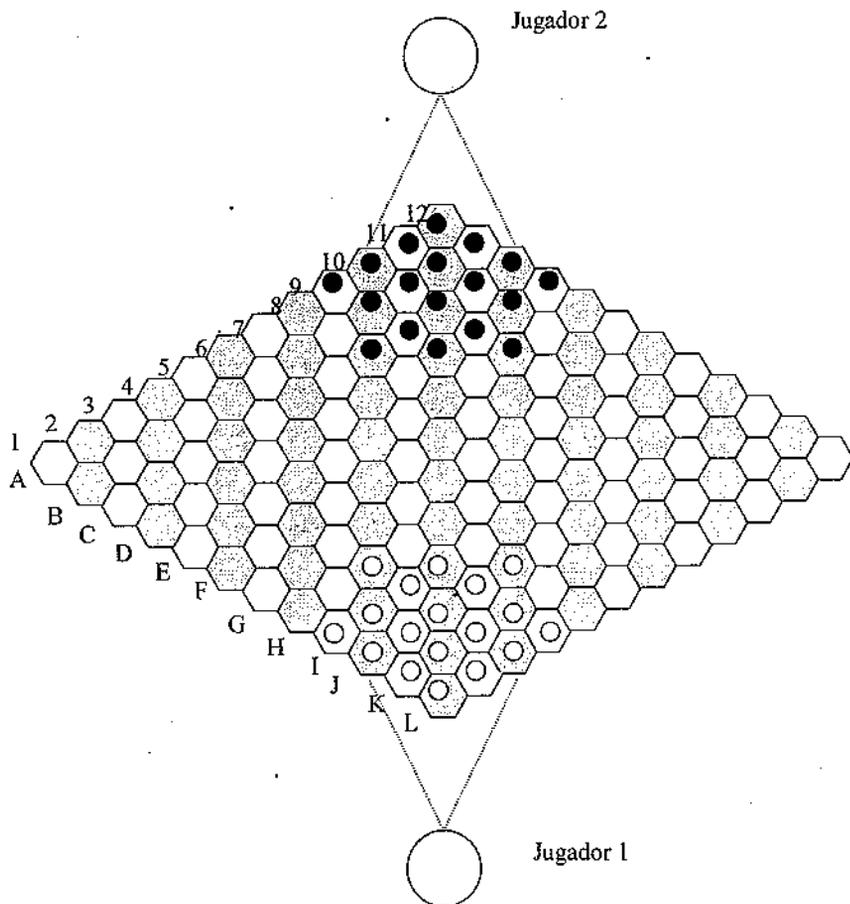


1.5 Versatilidad

A diferencia del ajedrez, Batel - Nah es un juego que se puede jugar entre más de dos personas, además de que se pueden formar equipos y aconsejarse entre ellos para planear las diversas estrategias de ataque y de defensa, existen 4 diferentes modalidades las cuales se mostrarán a continuación.

Juego para 2 personas (Reino vs. Reino)

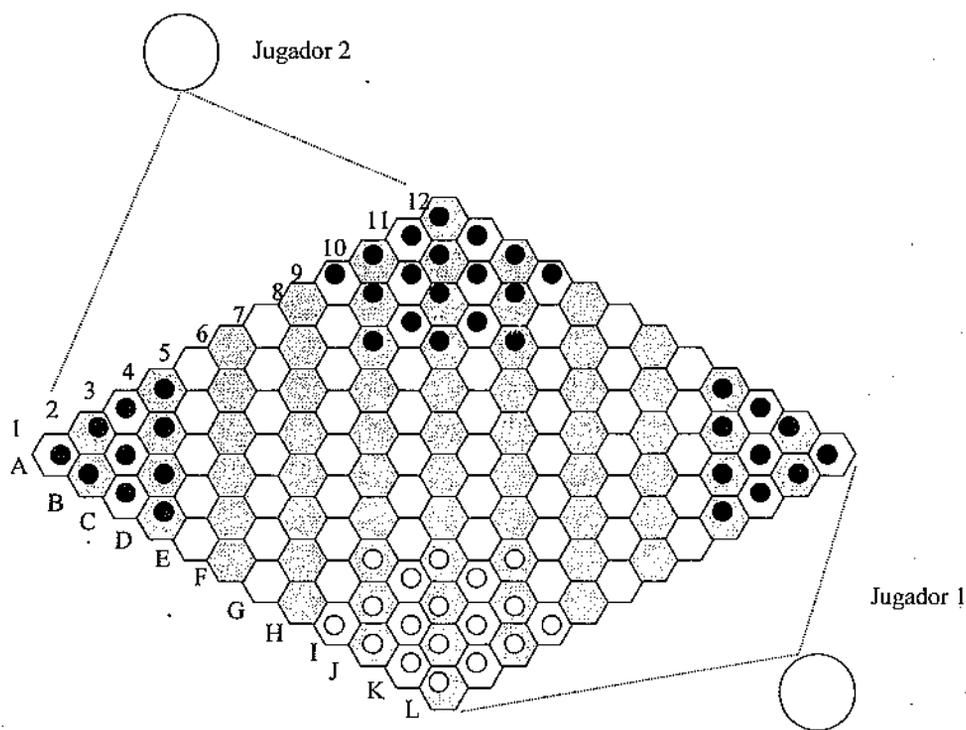
Esta modalidad es como la del ajedrez, en donde un jugador con su reino pelea contra otro jugador y su reino, esta modalidad es la más sencilla y todo el juego es individual. El primer turno es de las Blancas y el segundo turno para las Negras.



Posición inicial de (Reino y aliados vs. Reino y aliados)

Juego para 2 personas (Reino y aliados vs. Reino y aliados)

Esta modalidad es más avanzada ya que requiere de mayor concentración y habilidad por parte de los jugadores, ahora cada jugador tendrá un ejército que es el reino y otro que son los aliados, por lo que tendrá que coordinar los ataques y la defensa usando ambos ejércitos, un turno por ejército, el turno 1 es para las blancas, el turno 2 para las negras, el turno 3 para las azules y por último, el turno 4 para las rojas, lo que quiere decir que los jugadores atacarán de forma alternada, por lo que, por ejemplo una vez que hayan tirado las piezas blancas, tendrán que esperar 2 movimientos enemigos y un movimiento aliado para volver a tirar, esta situación hace que si no se planean las estrategias en conjunto tanto de defensa como de ataque, existirá una gran desventaja contra una buena planeación y trabajo en equipo.



Posición inicial de (Reino y aliados vs. Reino y aliados)

Juego para 3 personas (Reino y aliados vs. Reino y aliados)

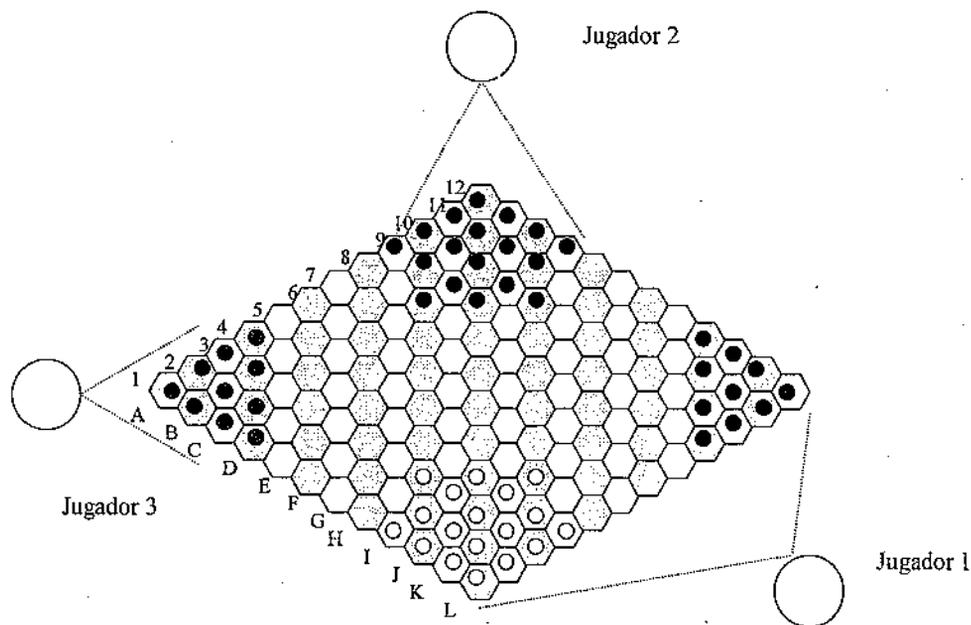
En todas las guerras hay aliados de cada ejército principal, estos aliados se podrán incorporar al juego en cualquier momento, ya sea al inicio o a media partida, siempre y cuando sea aceptado por los dos jugadores principales, en esta modalidad, al ser solo 3 jugadores, implica que un jugador jugará de pareja con otro, y el tercer jugador jugará solo.

Los aliados (pequeños pueblos), no tienen la misma cantidad de elementos en su ejército, tampoco tienen piezas reales, ya que al estar apoyando a un reino, sólo puede existir un Rey, debido a esto, el papel de los aliados es como apoyo para el reino fuerte, por lo que deberán de coordinarse bien, planear la estrategia de ataque y defensa.

Es muy importante que el reino cuide a su aliado, ya que cada uno tiene un turno para tirar y si el aliado es minimizado o eliminado, el reino contrario tendrá ventaja sobre los turnos de tiro.

Una vez que han sido aceptados los aliados, tirará primero el aliado de las blancas, o sea los azules, claro, después de que las negras hayan realizado su tiro, después de que tiren las blancas podrán tirar los aliados de color rojo, por lo que los turnos quedaran: blancas, negras, azules y rojas.

El trabajo en equipo es vital para ganar el juego, pero no está permitido hablar en el transcurso del mismo, por lo que la manera de comunicarse será por medio escrito, usando las coordenadas del tablero para ubicar posiciones.

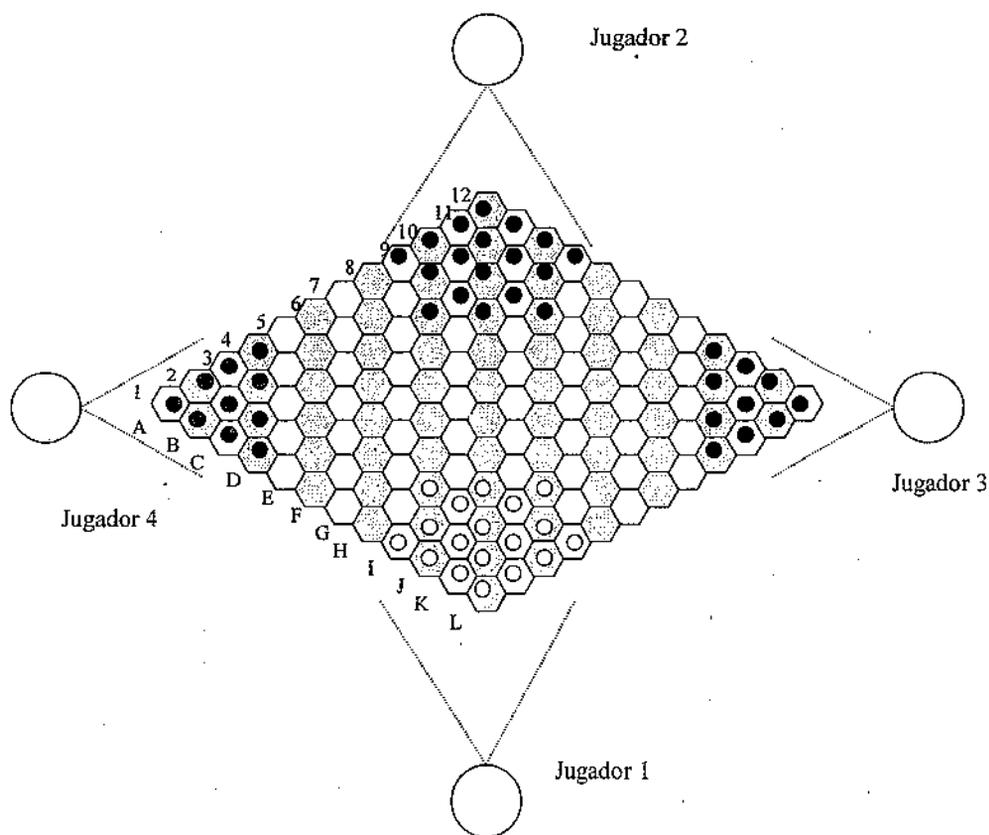


Juego para 4 personas (Reino y aliados vs. Reino y aliados)

Esta modalidad es la más completa y divertida, ya que ahora habrá 4 jugadores, y la competencia se volverá más interesante y en igualdad de circunstancias, los dos reinos tendrán consejeros, por lo que se fomentará la comunicación, la planeación y el trabajo en equipo.

De igual forma que en las modalidades anteriores los turnos de tiro serán: blancas, negras, azules y rojas.

El trabajo en equipo es vital para ganar el juego, pero no está permitido hablar en el transcurso del juego, por lo que la manera de comunicarse será por medio escrito, usando las coordenadas del tablero para ubicar posiciones.



Reglas especiales.

Se deberán de tomar en cuenta las siguientes reglas:

1. No se permite intercambiar turnos en ningún caso.
2. No se permite hablar en el transcurso del juego
3. No se permite intercambiar piezas.
4. Si un jugador abandona la partida, su compañero podrá seguir jugando con las piezas del primero, respetando los turnos.
5. Durante el juego, cuando se amenace a la primera pieza real (rey o reina), se dará un único aviso al oponente para que mueva esa pieza y la salve de ser eliminada, por lo que en las siguientes ocasiones en que se amenace a una pieza real no se dirá nada. Para dar este aviso se deberá de decir la palabra "Kolba", que en maya significa "corre o huye"
6. Una vez que un jugador ha eliminado las dos piezas reales deberá de decir la palabra "Hok-Sah" que en maya significa "vete o desaloja", con lo cual se habrá ganado el juego

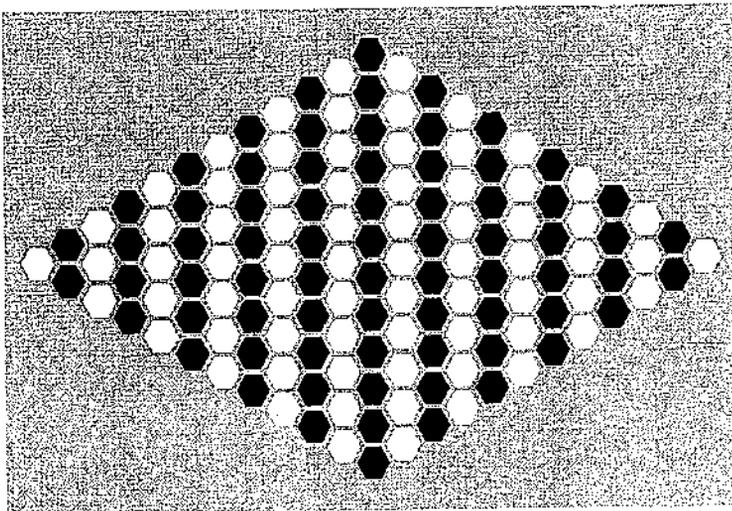
Para crear las piezas y el tablero del juego utilicé diversos programas de diseño por computadora, era de vital importancia que al diseñar individualmente cada pieza, fuera de forma tal que éstas resultasen atractivas para los jugadores, pero el mejor diseño fue el tablero, ya que es la estructura base del juego, permitió habilitar a más jugadores así como las opciones de trayectoria de los movimientos de las piezas, dentro de un escenario compacto y bien distribuido.

2.1 Diseño del Tablero.

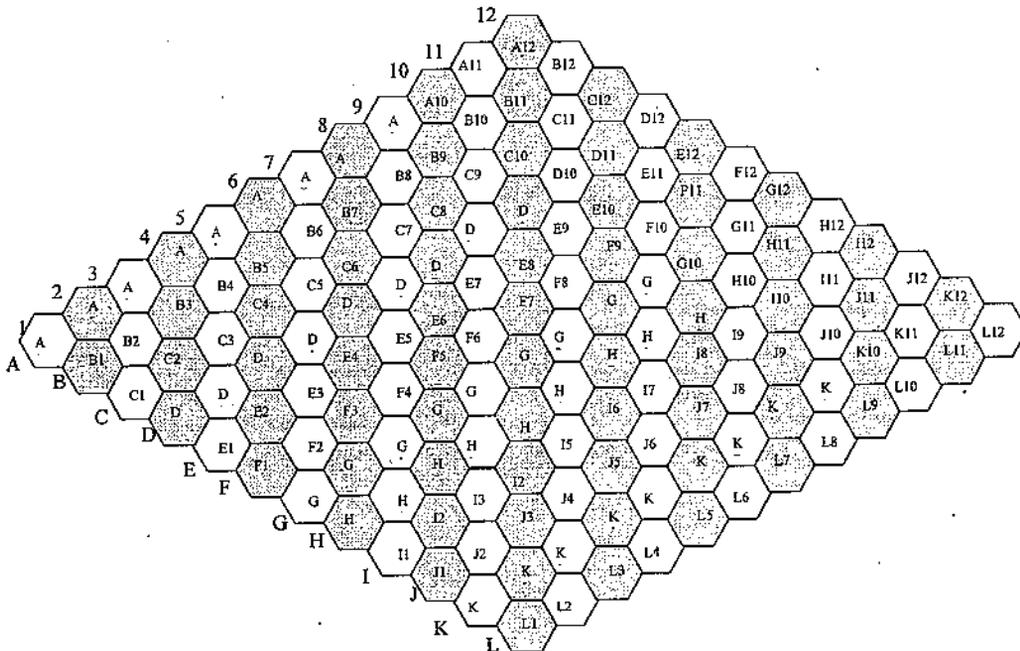
Primero diseñé el tablero, para lo cual utilicé una agrupación de hexágonos de manera tal que permitiera crear un área de batalla que sirviera para las diferentes modalidades del juego.

Primero tuve que definir el área, que consta de 12 hexágonos por 12 hexágonos, dando un total de 144 hexágonos, suficiente espacio para que jueguen 4 personas y no para que únicamente jueguen 2.

La forma hexagonal de los espacios habilita 6 diferentes direcciones de movimiento, con lo que las piezas adquieren automáticamente mayor rango de ataque, haciendo más completa su funcionalidad.



Después le asigné las coordenadas para la planeación estratégica entre jugadores, los rangos van de 1-12 y A-L, con lo que cada espacio tiene su identificador.



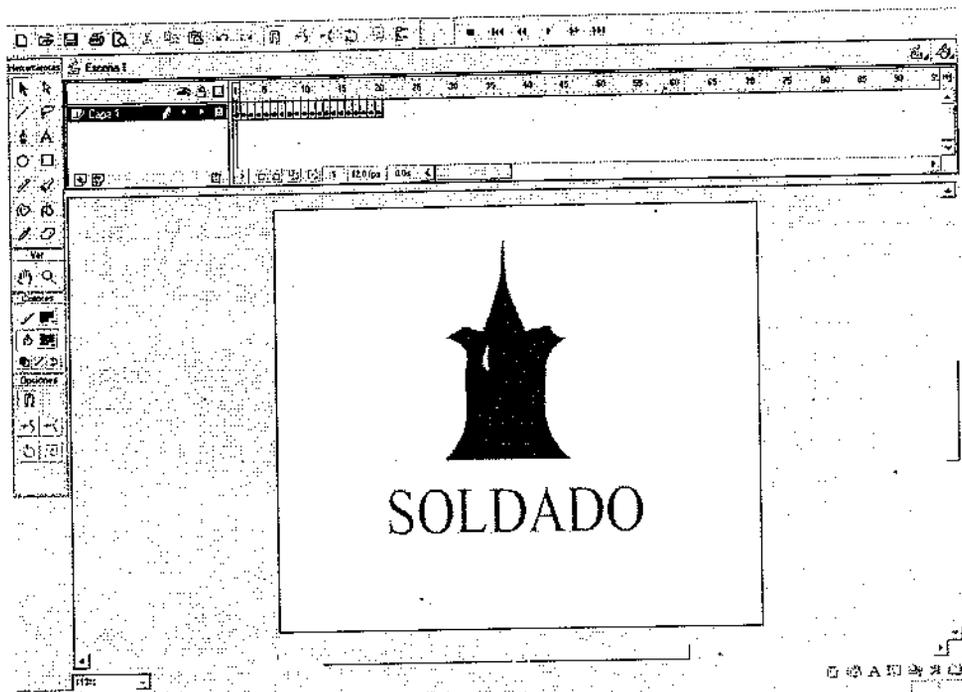
2.2 Creación de piezas Batef - Nah.

Para crear cada una de las piezas realicé un análisis en función del tablero, para que cada una tuviera su movimiento específico y que su ataque individual fuera más amplio, haciendo así, que al atacar en conjunto se convirtiera en un ejército poderoso. Utilicé diversos software para su creación, todas las piezas se crearon de la misma forma, los colores, texturas y luminosidad se puede cambiar permitiendo crear los diferentes ejércitos, a continuación mostraré brevemente como fue este proceso.

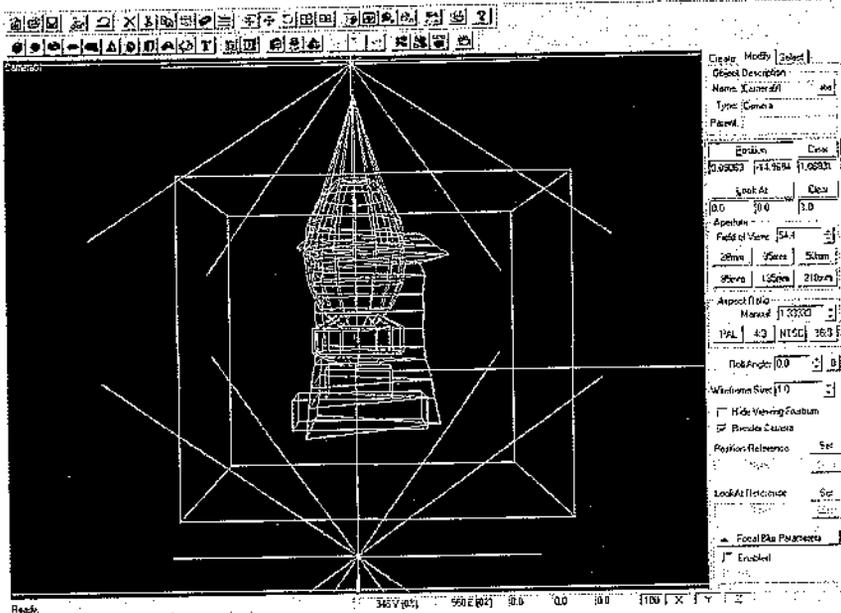
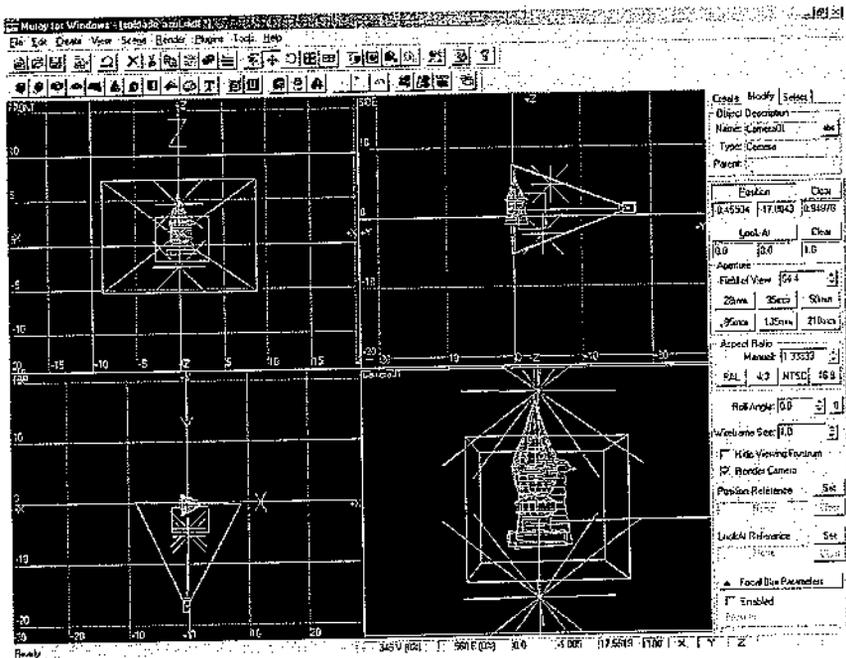
El soldado

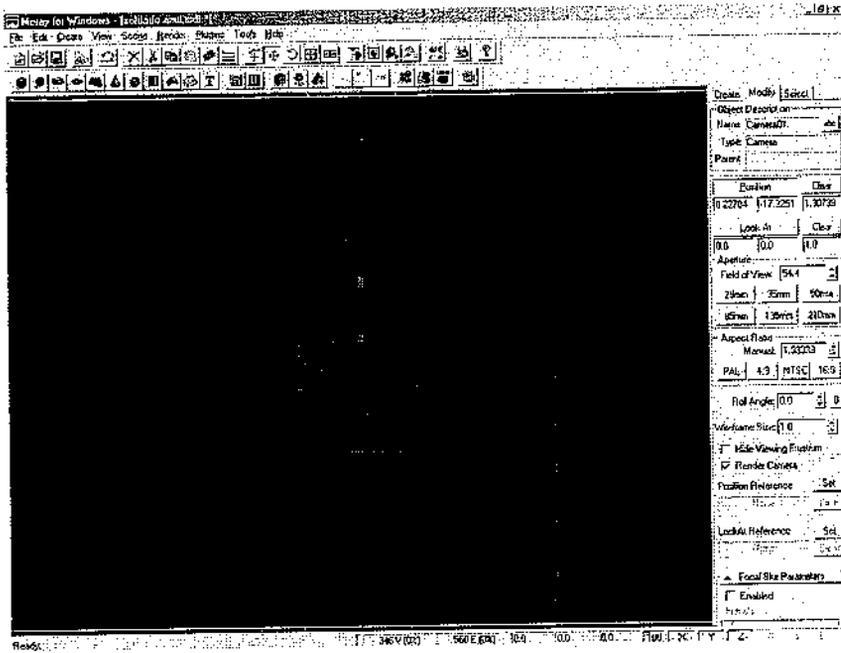
La figura del soldado representa al elemento lento, con poco ataque, pero mayor en número, son piezas sacrificables en la batalla.

Diseñé su figura usando Flash.



La pieza como tal, ahora ya estaba diseñada, pero tenía que darle dimensiones para que tuviera mucha más presencia gráfica, por lo que busque otro software para desarrollar las piezas tridimensionales, este software fue Moray. Este es un software muy completo y amigable por lo que con un poco de práctica se puede diseñar casi cualquier cosa.





Así, de ésta manera quedó terminada la creación de esta pieza, tanto en su forma plana (dos dimensiones), como en su forma tridimensional.



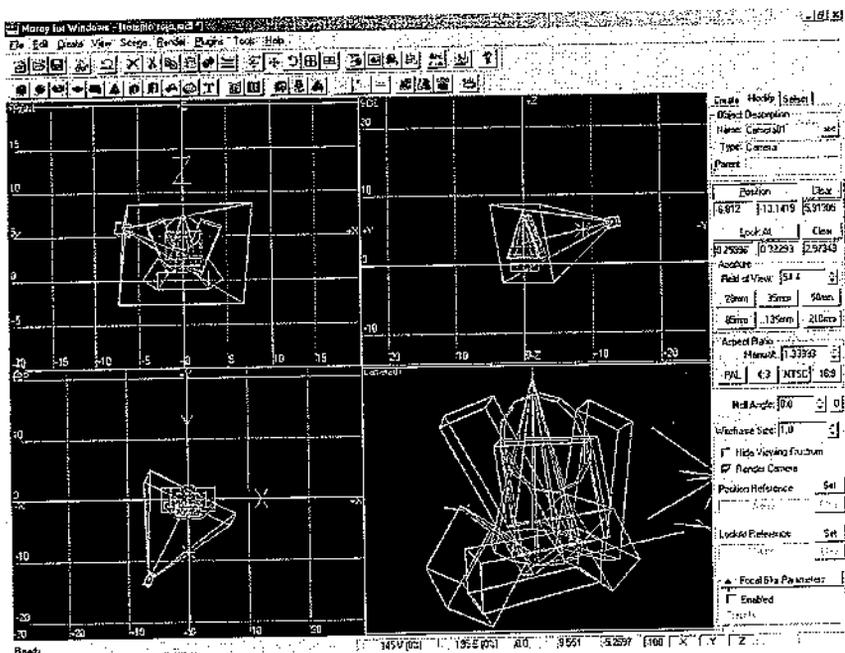
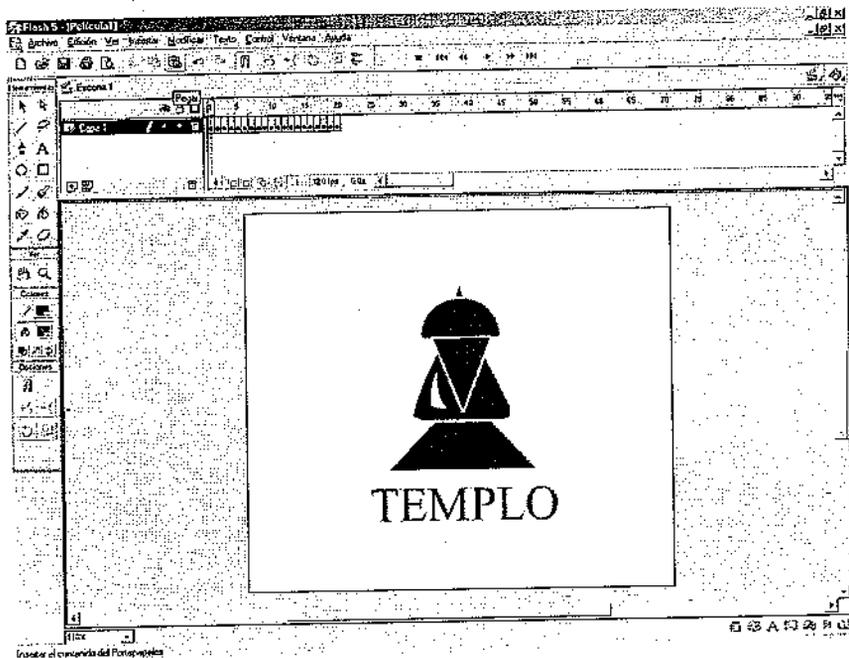
SOLDADO

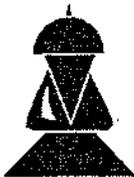
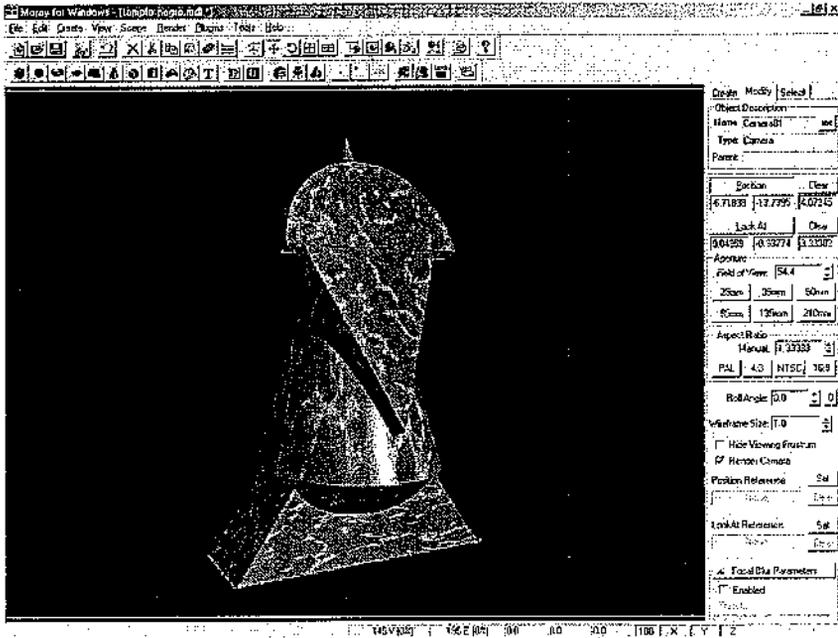
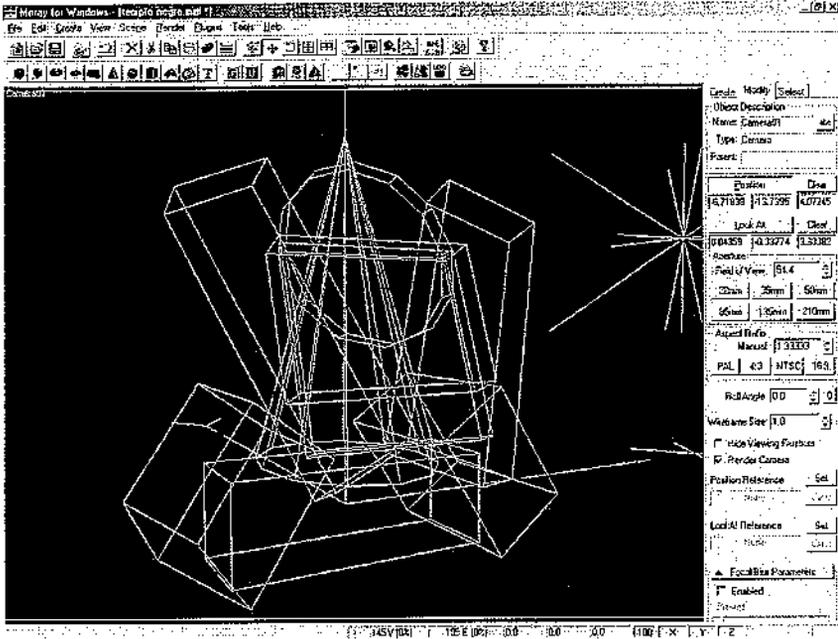


SOLDADO

El templo

Representa a elementos que son rápidos, pero sólo tienen definido cierto rango de ataque.





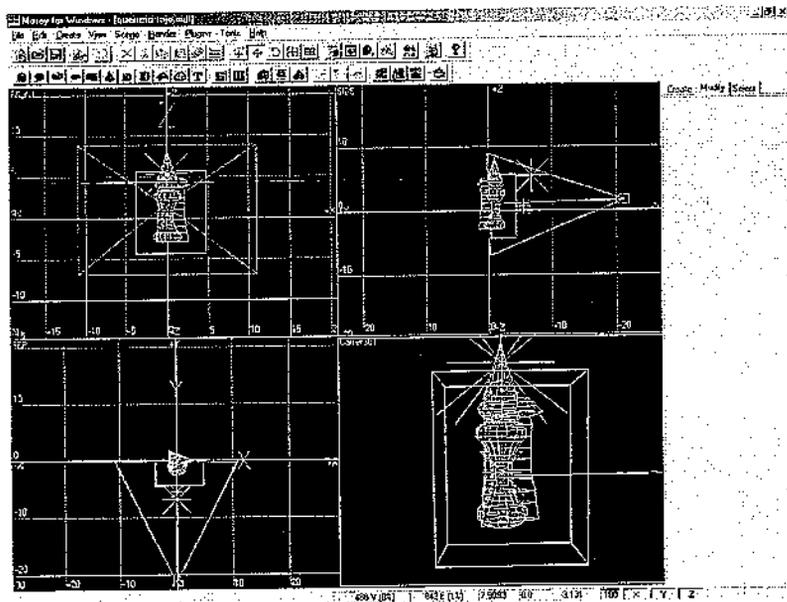
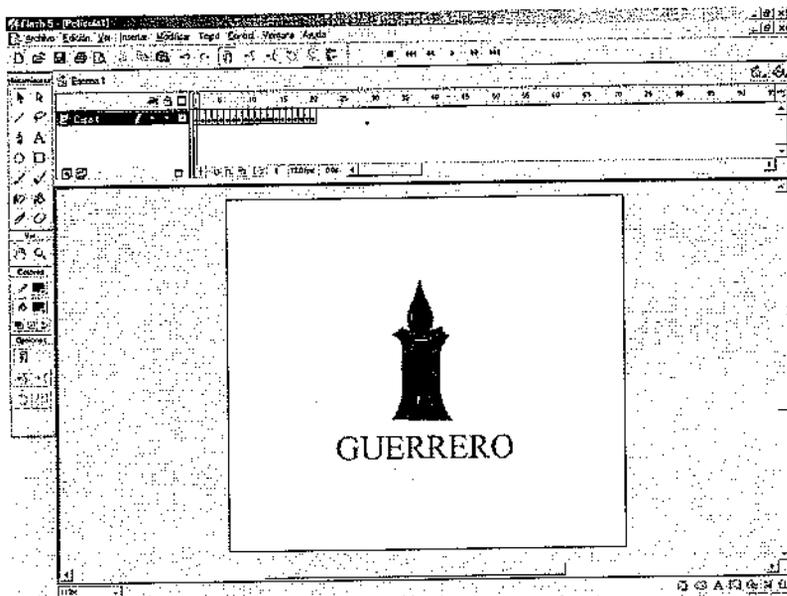
TEMPLO

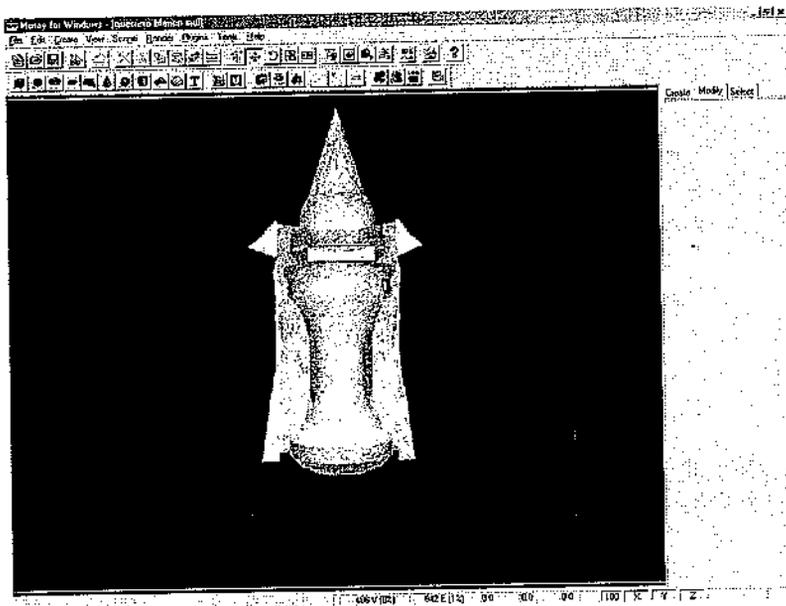
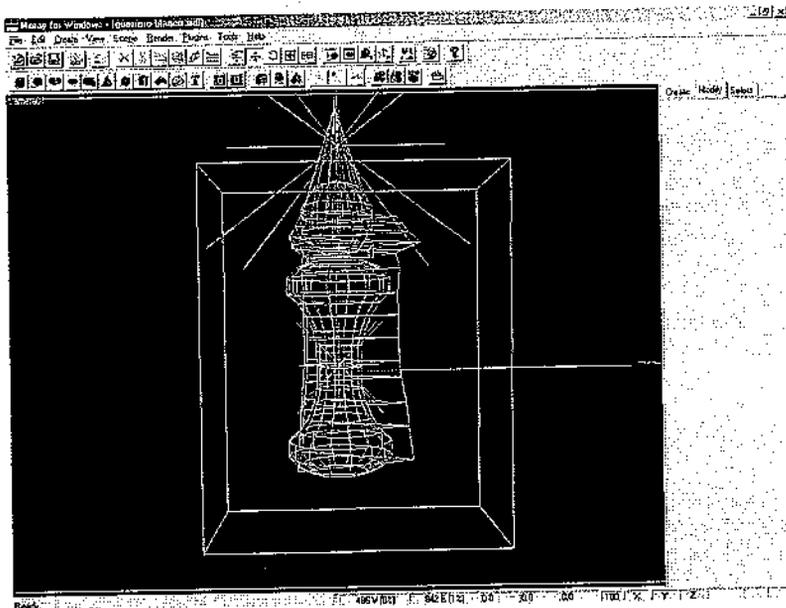


TEMPLO

El guerrero

Representa a los elementos especializados, más rápidos y más letales.





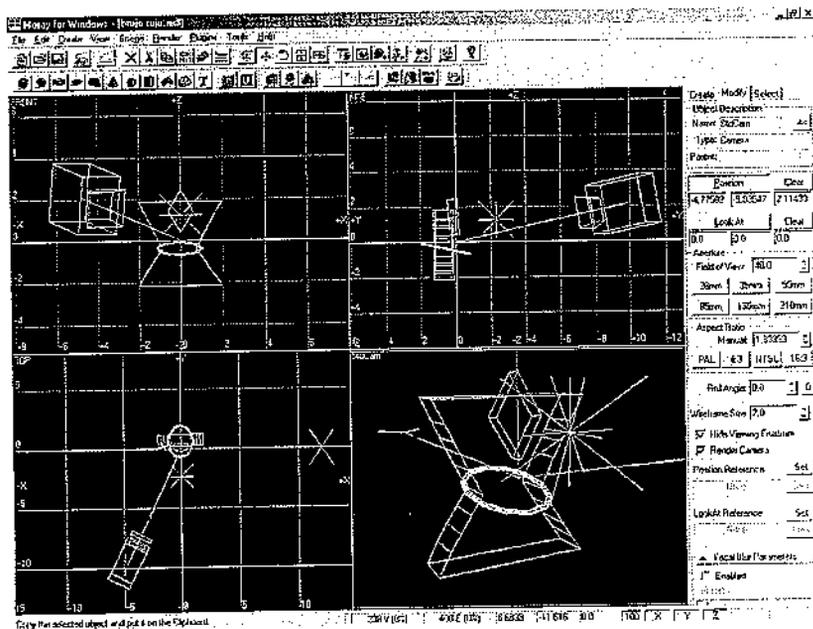
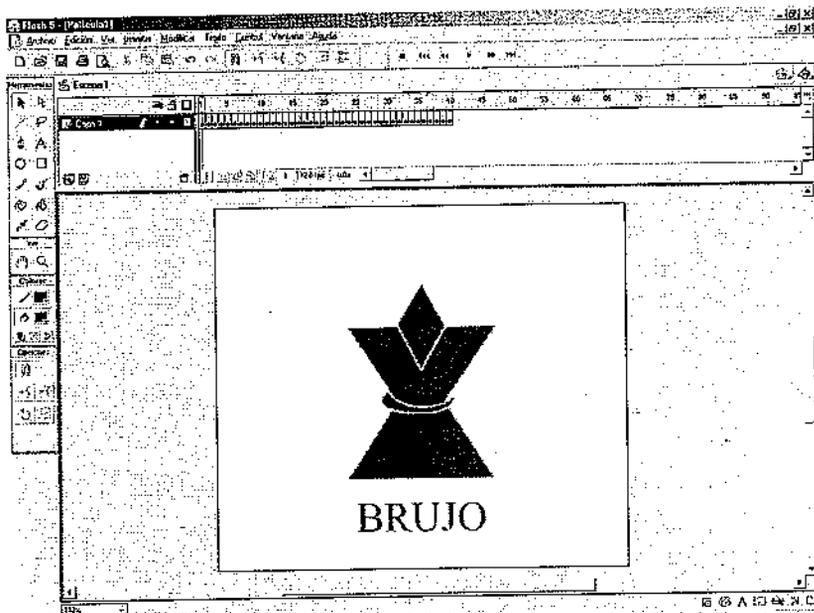
GUERRERO

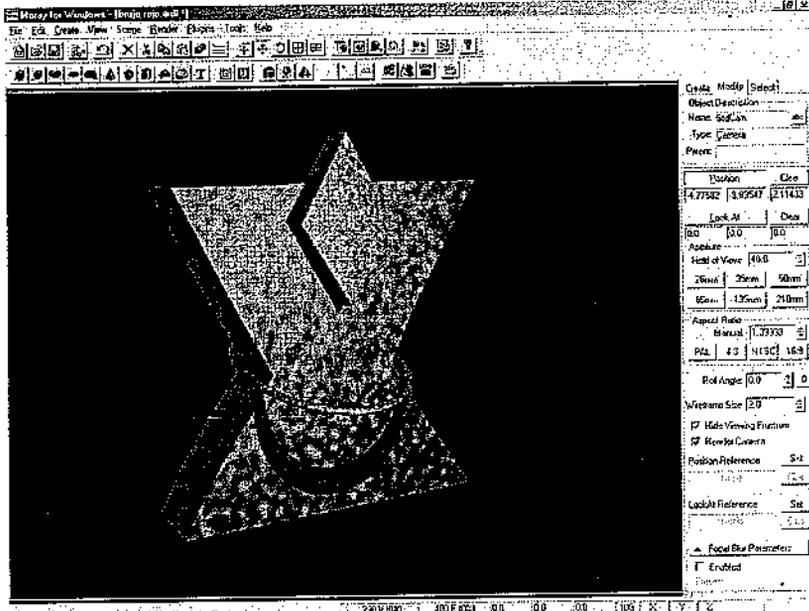
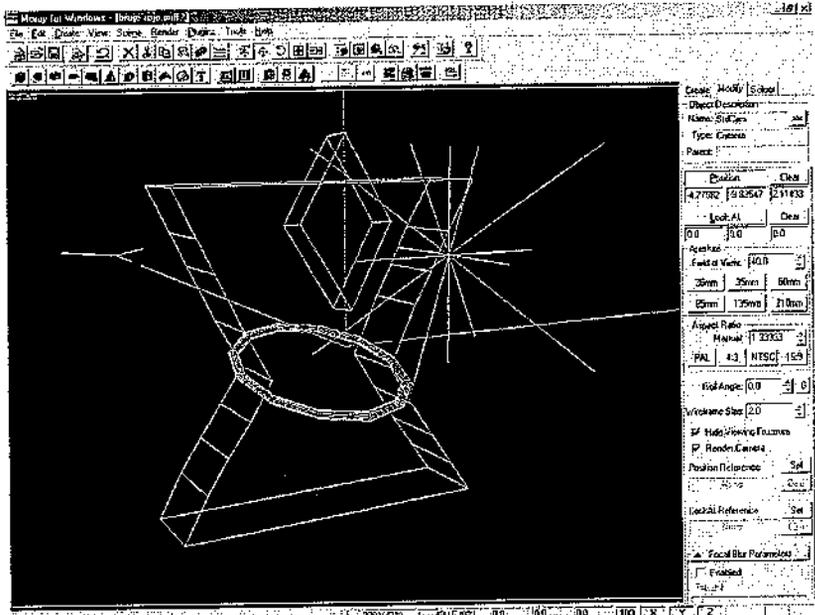


GUERRERO

El brujo

Representa a los elementos entrenados con tácticas especiales, que los hace fuera de lo común.





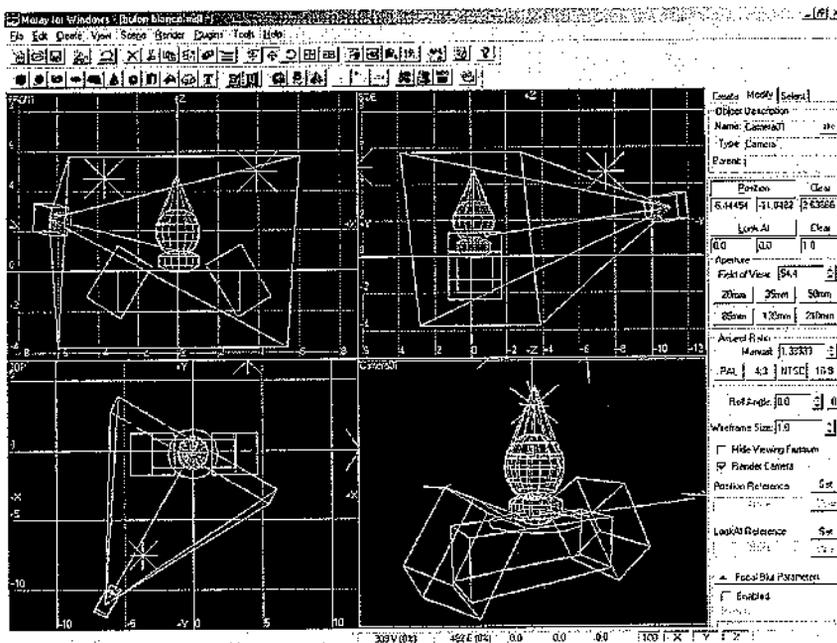
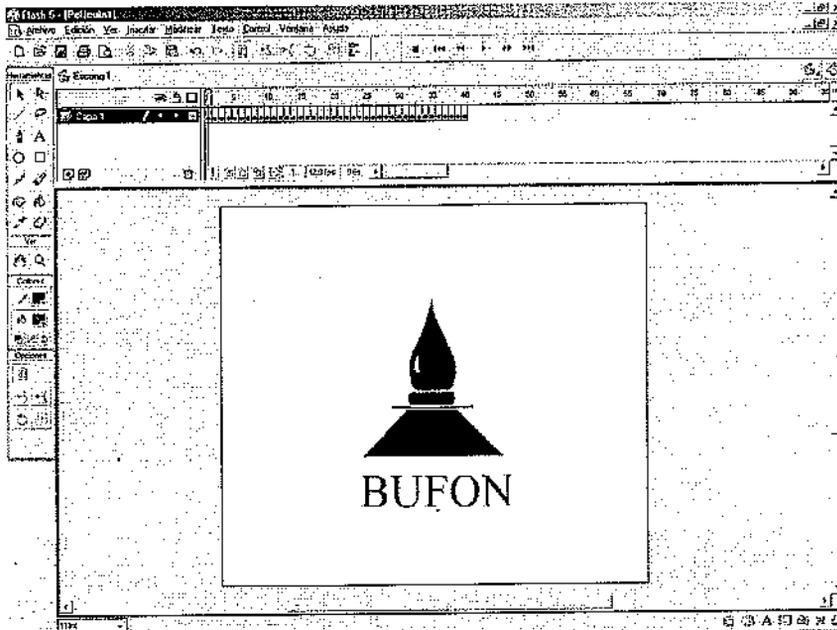
BRUJO

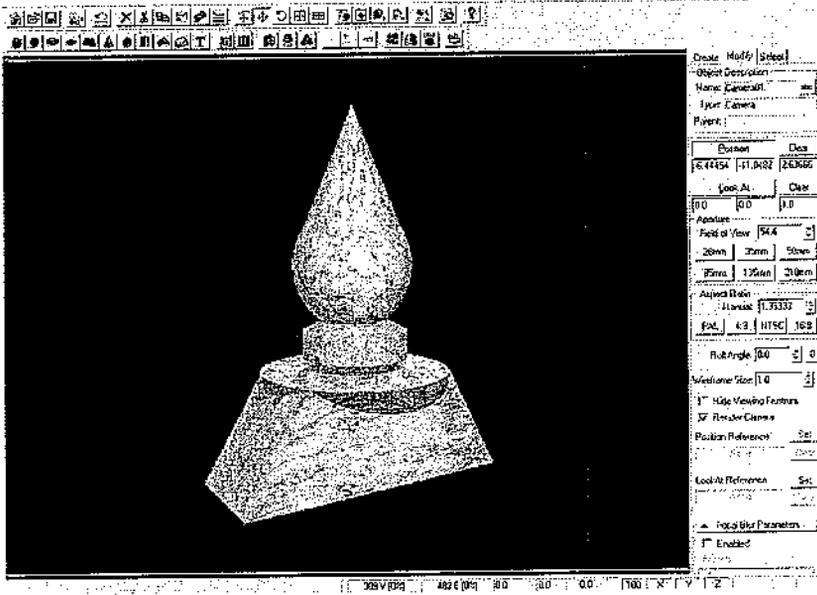
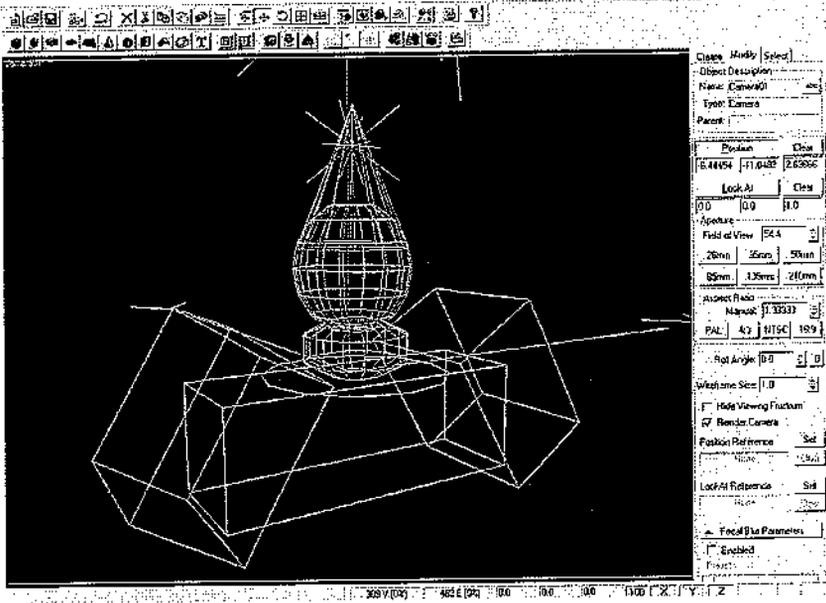


BRUJO

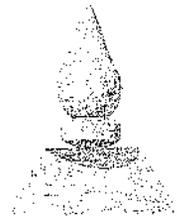
El bufón

Representa a los elementos que en apariencia no son peligrosos, pero una vez que atacan, son efectivos y elusivos.





BUFON

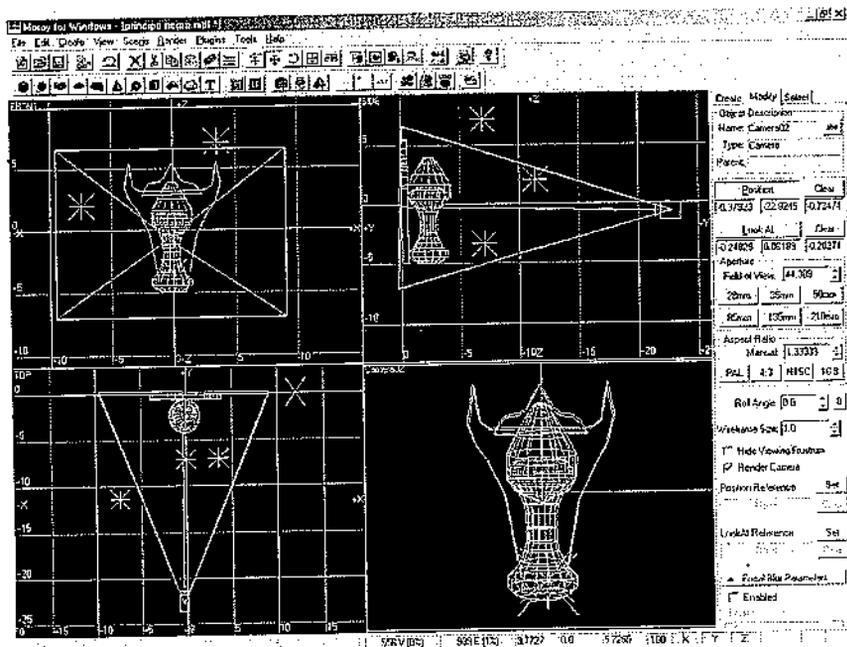
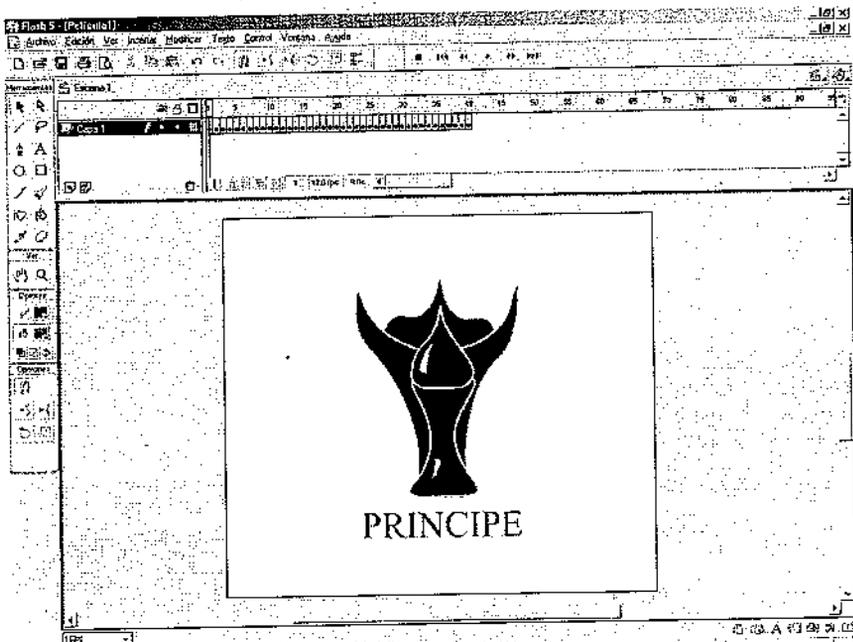


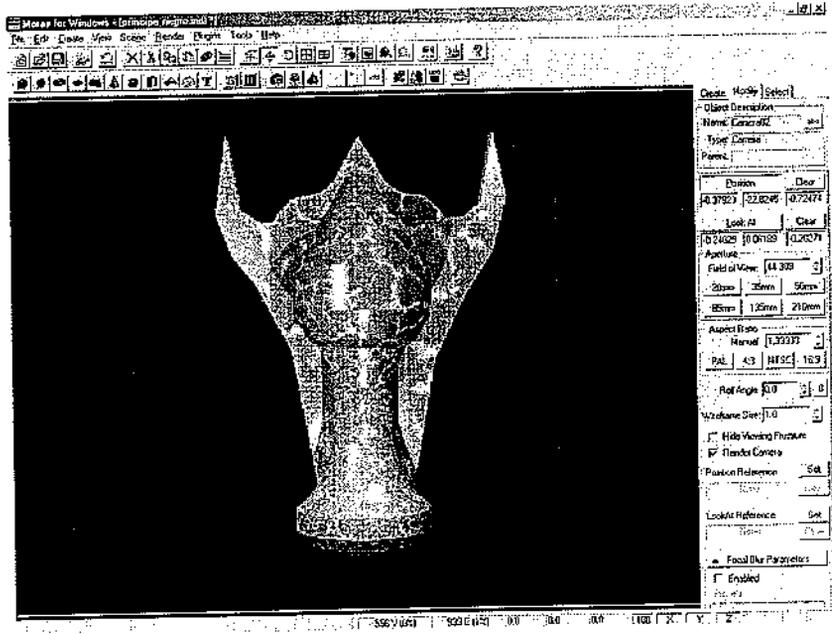
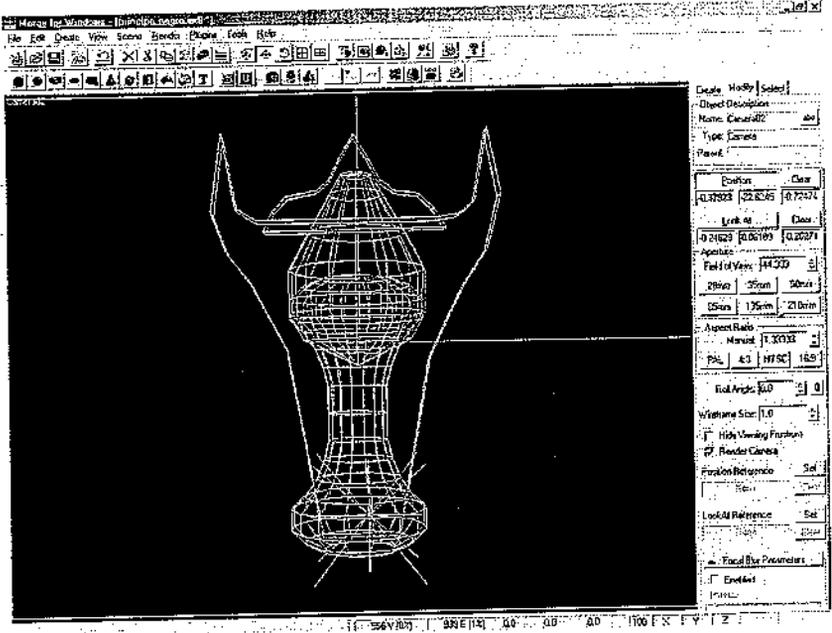
BUFON

El príncipe

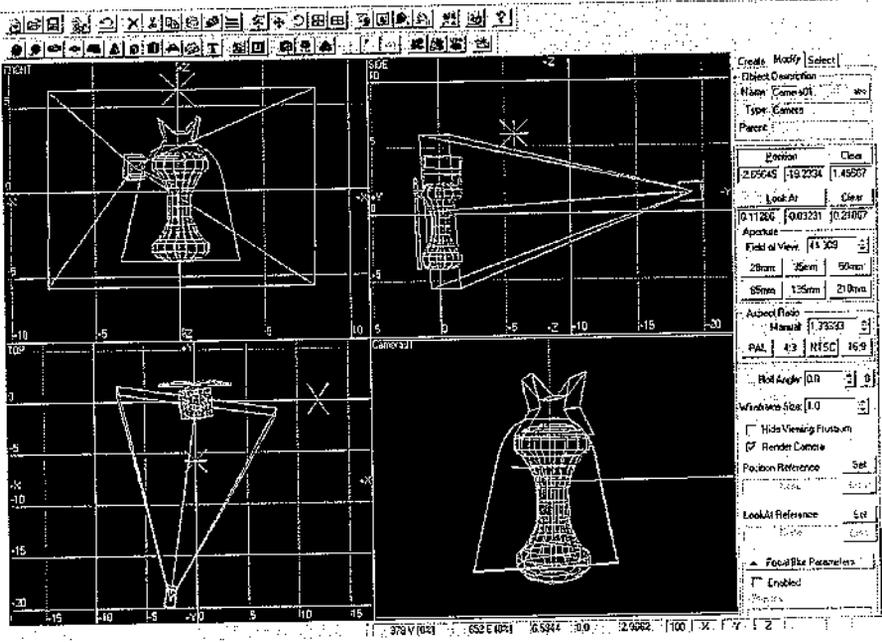
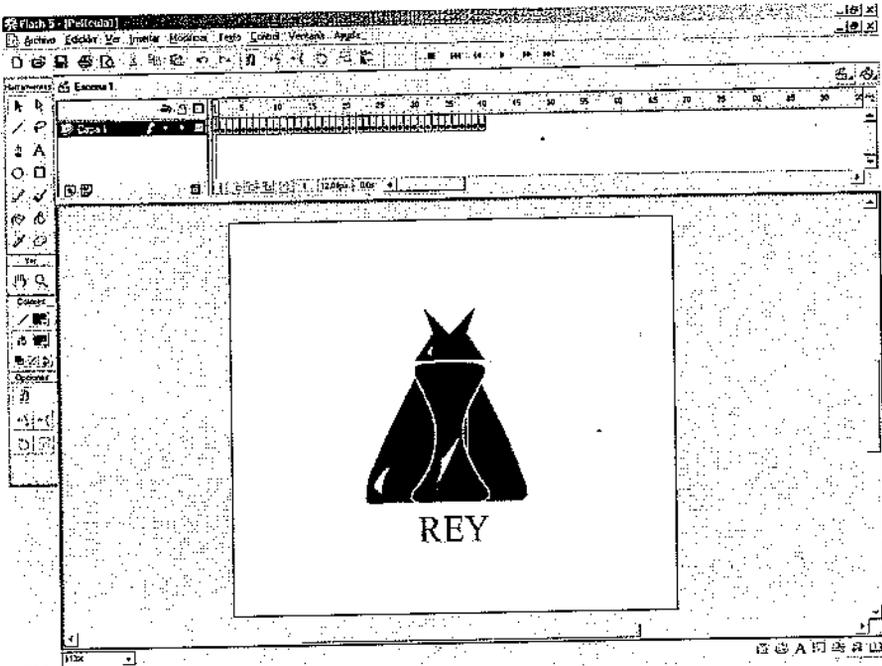
Representa al guerrero poderoso e impetuoso, que gracias a su estatus y habilidades es muy peligroso.

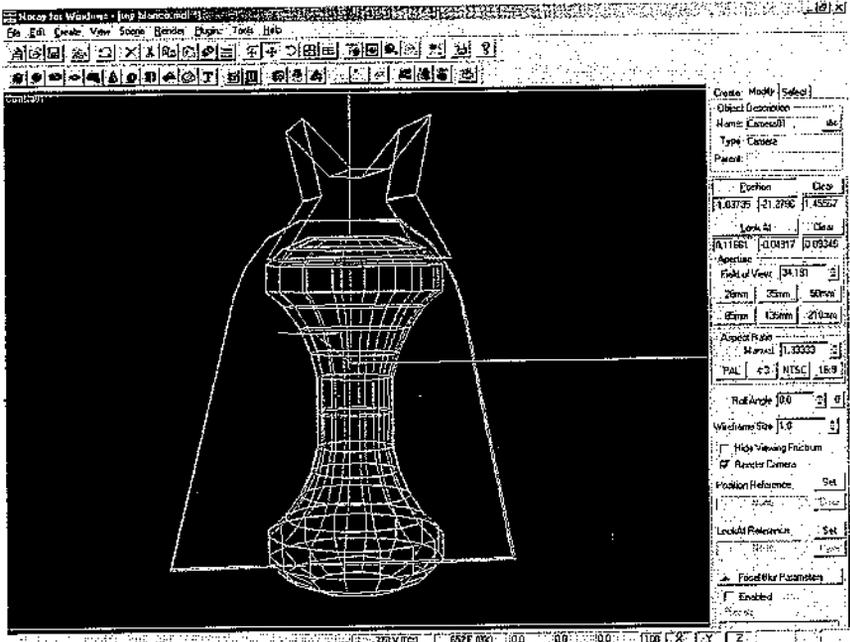
9





El Rey



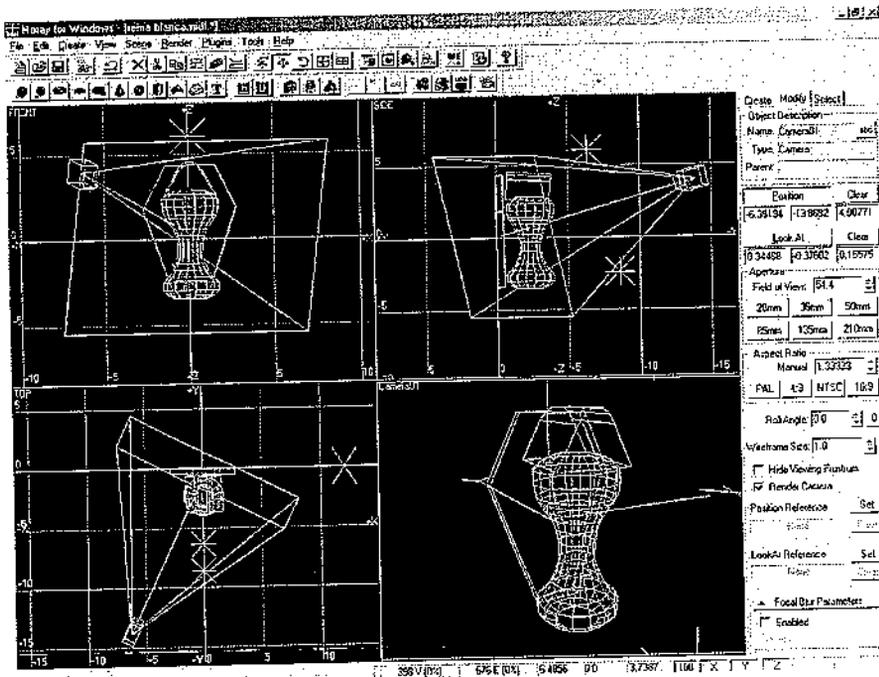
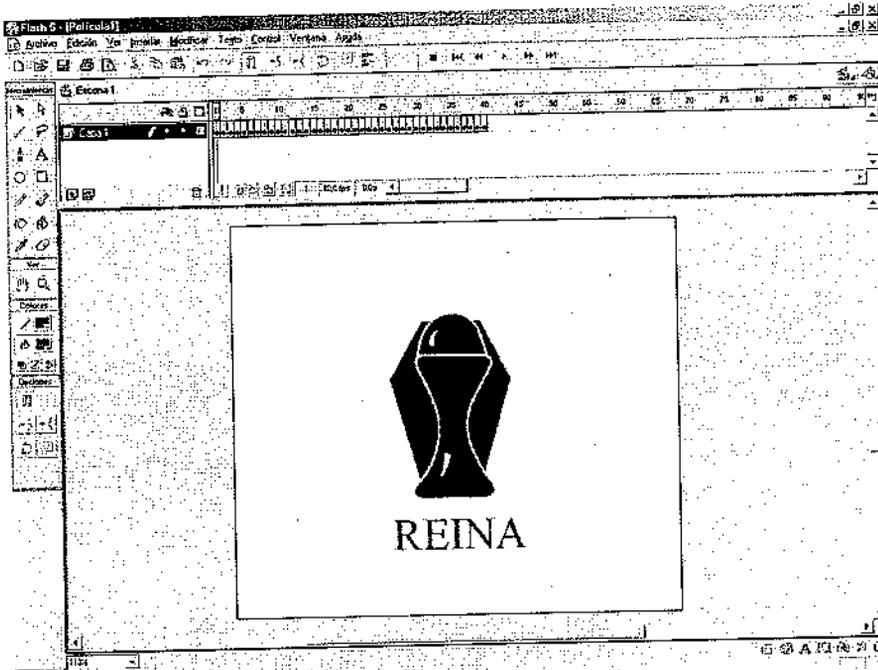


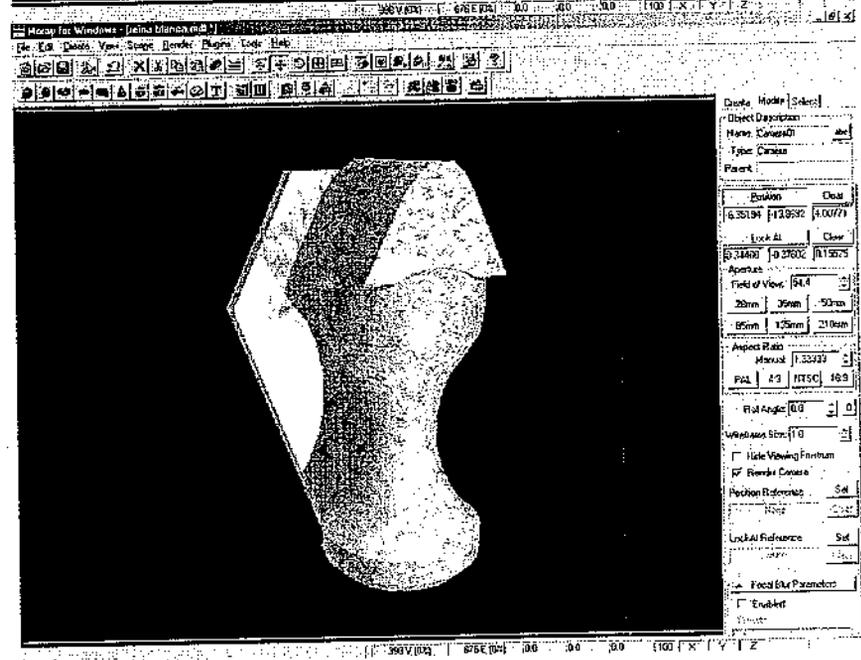
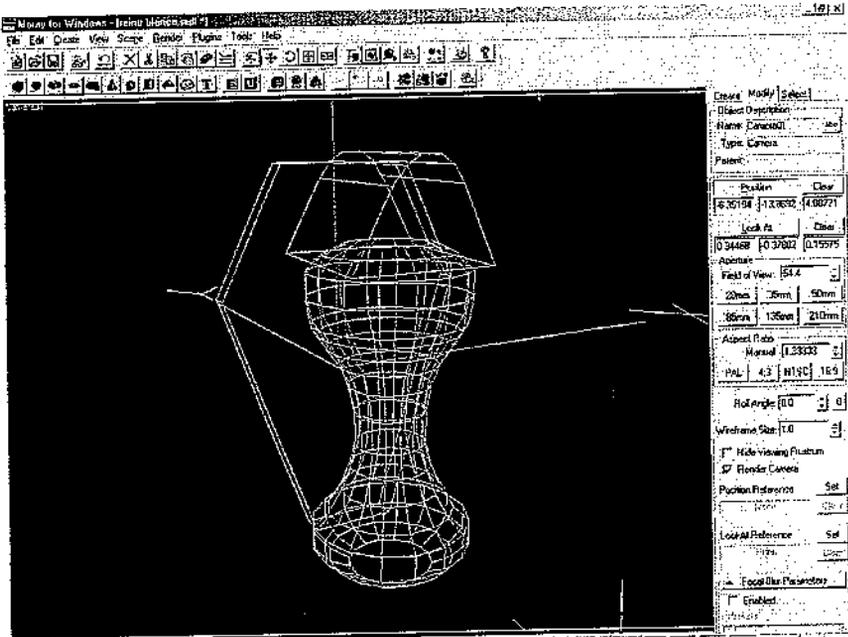
REY



REY

La Reina





REINA



REINA

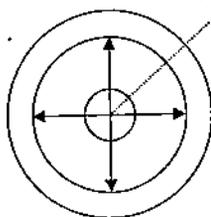
2.3 Creación de logotipo.

El logotipo será la figura representativa del juego, utilicé al príncipe porque es una pieza nueva, además de que es la más poderosa por lo que los jugadores la valorarán mucho.

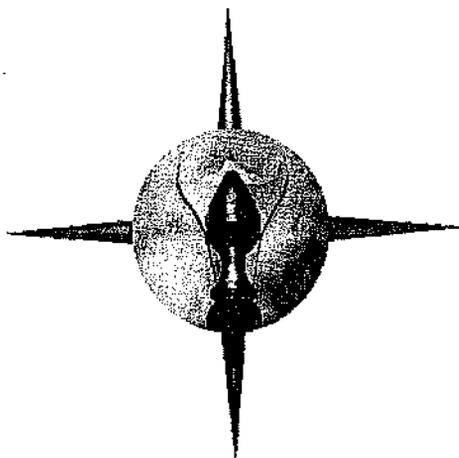


PRINCIPE

Elegí al círculo, porque su estructura representa solidez y a su vez mantiene una igualdad desde su origen, que es el centro hacia cualquier punto de su figura. Para mí representa estabilidad y equilibrio.



Por último, escogí la estrella que representa los 4 puntos cardinales, al igual que el tablero que sus 4 esquinas están orientadas hacia estos 4 ejes.

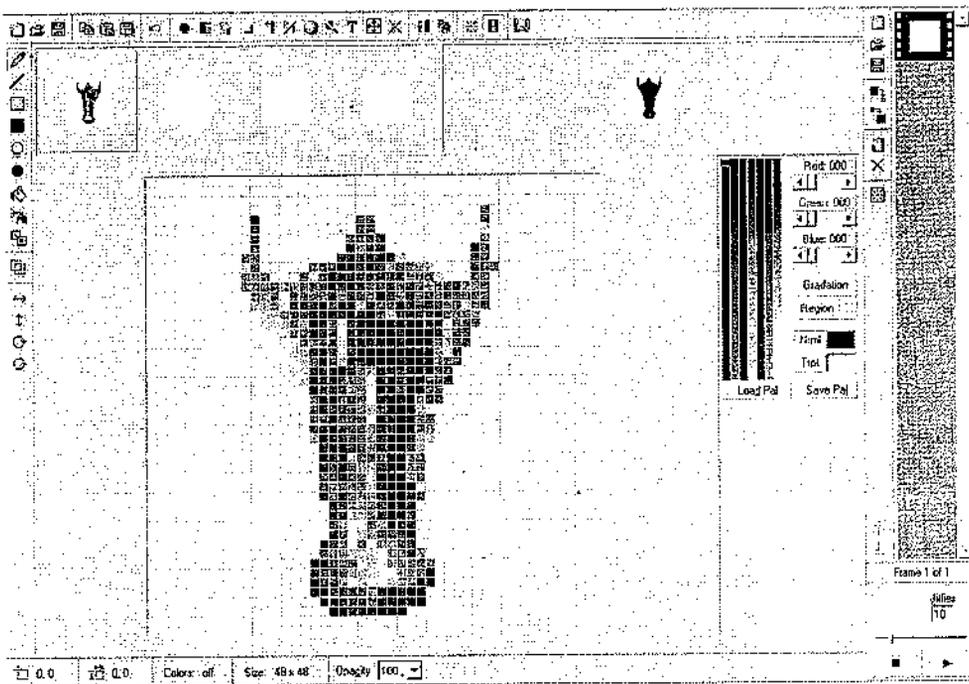


2.4 Creación de Cursores e iconos.

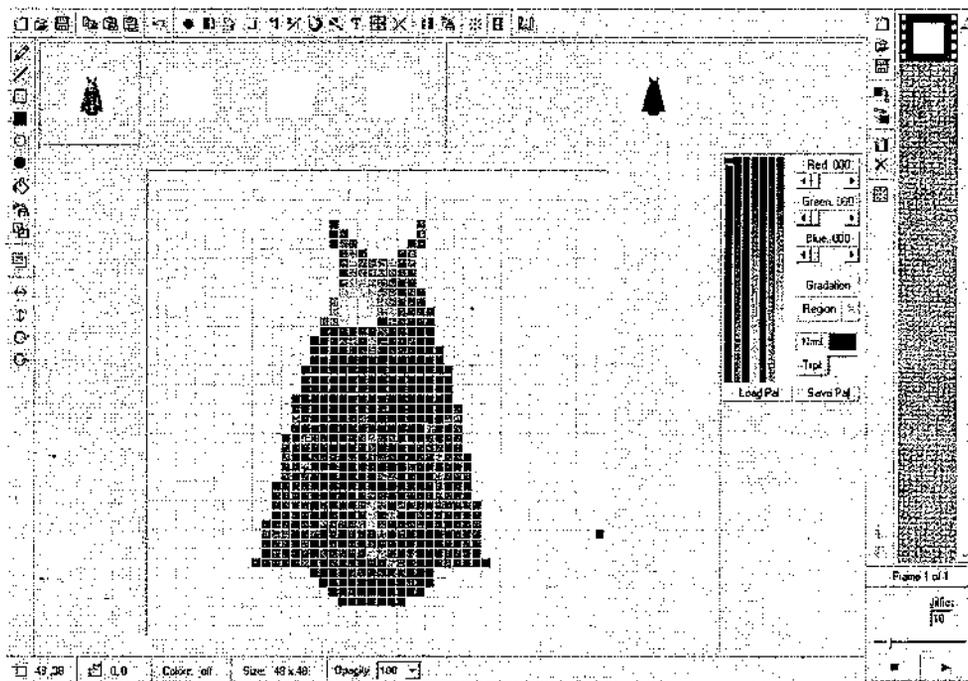
Todo software necesita interfases gráficas para su funcionalidad, en el caso de programas para aplicaciones con sistema operativo Windows, como las computadoras personales (PC) es necesario crear tanto iconos, que son los elementos gráficos que representan y nos direccionan hacia la ejecución de un programa, como los cursores o punteros, que son los elementos que representan el posicionamiento del mouse.

Para la creación de estos elementos utilicé tres softwares: el primero fue el Pov-Ray para crear los diferentes cuadros de movimiento de cada icono, después utilicé Paraben's icons builder.

Primero se diseña la imagen que se desea convertir en icono, después se importa al editor de Paraben's icons builder. Hoy existen varias opciones para editar o modificar el dibujo, una vez que queda completamente definido como lo queremos se guarda con la extensión .ico, este archivo ahora será utilizado por Windows como icono.



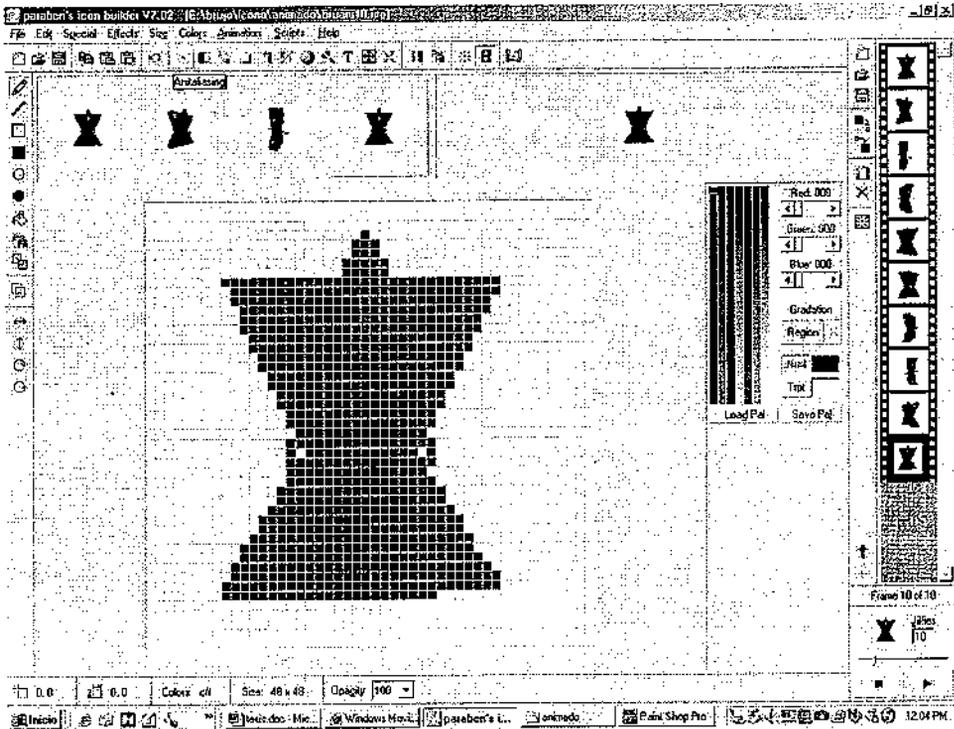
Para crear el cursor se realiza la misma operación, pero ahora se deberá guardar con la extensión .cur, con esto Windows reconocerá este archivo como un cursor o puntero de mouse.



Para crear el cursor animado es necesario crear la secuencia de imágenes, esta secuencia se hace con Pov-Ray, programando una rotación sobre el eje z para que se vayan generando las imágenes y al momento de hacer la animación parezca que la pieza está rotando.

Del lado derecho de la figura anterior se pueden ver las imágenes que crearán la secuencia de movimiento, utilicé 10 cuadros para crear el cursor animado de cada figura del juego.

Una vez creados se podrán utilizar como cualquier otra aplicación de Windows, para configurar los cursores se va al panel de control en el icono del mouse, ahí uno puede asignar el puntero que se desee.

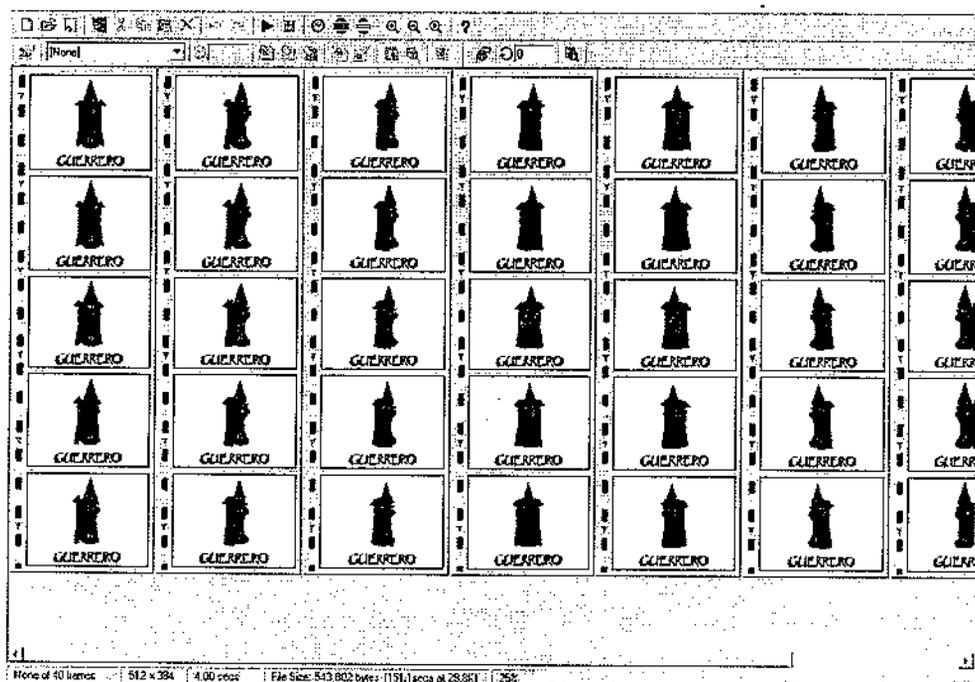


2.5 Creación de Imágenes Gif.

El juego Batel.-Nah se va a implementar en varias formas, una de ellas es dentro de una página Web. El diseño debe de ser visualmente muy atractivo, por lo que es necesario crear imágenes y animaciones que proporcionen dicho atractivo.

Para esto creé animaciones con formato gif, que son archivos que pueden contener una secuencia de imágenes, las cuales se despliegan automáticamente dentro de una página Web, por lo que así se pueden crear diversas animaciones.

Utilice Pov-Ray para crear la secuencia de imágenes de la animación, una vez creadas se importan al programa Gif Movie Gear, el cual junta estas imágenes y genera una animación, que se guarda en un archivo con extensión .gif. Además, este programa es capaz de crear archivos en formato .avi, por lo que es de gran utilidad para desarrollar secuencias de imágenes o películas.



CAPÍTULO 3

Diseño de algoritmos del juego (Batel – Nah)

El sistema que creé debe de ser capaz de evaluar varias situaciones y procedimientos para hacer que el juego sea jugado conforme a las reglas establecidas, así como a los espacios válidos del tablero, a continuación numero las diversas situaciones que se deberán de evaluar e implementar.

1. Cada espacio debe de estar identificado como unidad, si está libre u ocupado, por quien está ocupado
2. Al inicio del juego el sistema debe colocar las piezas en su lugar y partiendo de esta base, sabrá que piezas se pueden mover y hacia dónde.
3. El sistema debe de saber todas las reglas individuales de movimiento de cada pieza.
4. El sistema debe de validar los intentos de movimiento que cada jugador haga, tanto en trayectoria como en ubicación final
5. El sistema debe de identificar los turnos de juego y cuando un jugador ha completado su movimiento.
6. El sistema deberá de saber que tipo de juego se realizará, Reino vs. Reino o Reino y Aliados vs. Reino y Aliados.
7. El sistema deberá de contar con un reloj que limite el tiempo de tiro de cada jugador en turno.

3.1 Diseño de los algoritmos de identificación del tablero

Como ya se ha mencionado el tablero está formado por un área de 12 por 12 hexágonos, lo que da un total de 144 espacios hexagonales, esta área tiene coordenadas con las cuales se identificará cada uno de los espacios, estas coordenadas van de 1 hasta 12 y de A a L, por lo que se define la siguiente tabla:

A 1	B 1	C 1	D 1	E 1	F 1	G 1	H 1	I 1	J 1	K 1	L 1
A 2	B 2	C 2	D 2	E 2	F 2	G 2	H 2	I 2	J 2	K 2	L 2
A 3	B 3	C 3	D 3	E 3	F 3	G 3	H 3	I 3	J 3	K 3	L 3
A 4	B 4	C 4	D 4	E 4	F 4	G 4	H 4	I 4	J 4	K 4	L 4
A 5	B 5	C 5	D 5	E 5	F 5	G 5	H 5	I 5	J 5	K 5	L 5
A 6	B 6	C 6	D 6	E 6	F 6	G 6	H 6	I 6	J 6	K 6	L 6
A 7	B 7	C 7	D 7	E 7	F 7	G 7	H 7	I 7	J 7	K 7	L 7
A 8	B 8	C 8	D 8	E 8	F 8	G 8	H 8	I 8	J 8	K 8	L 8
A 9	B 9	C 9	D 9	E 9	F 9	G 9	H 9	I 9	J 9	K 9	L 9
A 10	B 10	C 10	D 10	E 10	F 10	G 10	H 10	I 10	J 10	K 10	L 10
A 11	B 11	C 11	D 11	E 11	F 11	G 11	H 11	I 11	J 11	K 11	L 11
A 12	B 12	C 12	D 12	E 12	F 12	G 12	H 12	I 12	J 12	K 12	L 12

Para que el sistema pueda saber a qué espacio se está haciendo referencia, estas coordenadas que acabamos de obtener, aunque están unidas para definir el área, las dividiremos para su manipulación, por lo que la posición de la coordenada que está dada por las letras tendrá un valor y de igual forma la de los números.

Cada espacio será definido por una posición, y esta estará dada por la fórmula sig.

$$\text{posición} = \text{coordenada 1} + \text{coordenada 2}$$

$$\text{posición} = \text{letra} + \text{número}$$

$$\text{posición} = A + 12 = A12$$

Cada coordenada tendrá un valor, resultando la siguiente tabla:

A = 1	1 = 1
B = 2	2 = 2
C = 3	3 = 3
D = 4	4 = 4
E = 5	5 = 5
F = 6	6 = 6
G = 7	7 = 7
H = 8	8 = 8
I = 9	9 = 9
J = 10	10 = 10
K = 11	11 = 11
L = 12	12 = 12

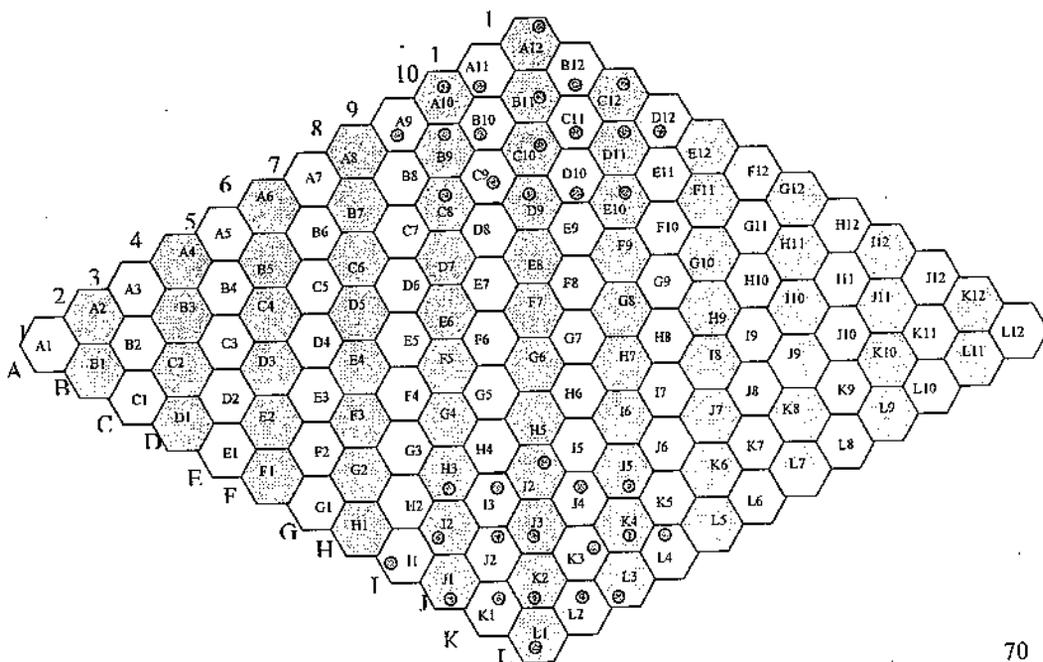
Es necesario definir algún método para identificar de alguna manera cuando un espacio se encuentra ocupado o vacío, esto se hará asignado un valor a cada coordenada, para los espacios ocupados será 1 y para los espacios libres 0

A5(1) = posición A5 ocupada

F8(0) = posición F8 vacío

Al inicio del juego hay algunos espacios que ya están ocupados, a continuación se muestra la grafica y la tabla correspondiente.

Los puntos muestran los espacios ocupados, para el juego de Reino vs. Reino

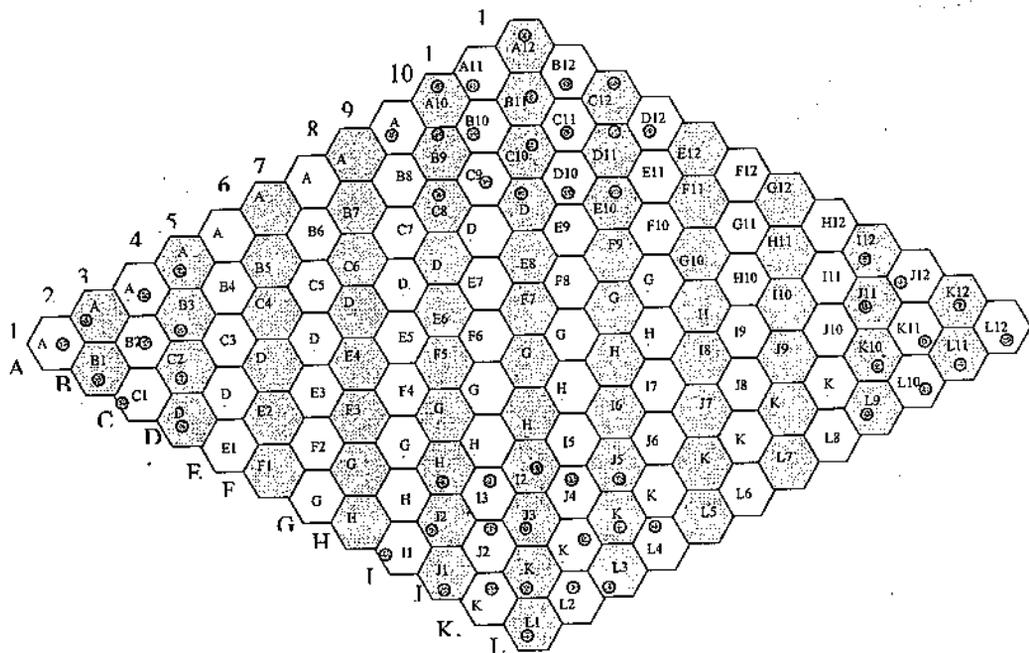


La tabla quedaría de la siguiente forma

A 1 =0	B 1 =0	C 1 =0	D 1 =0	E 1 =0	F 1 =0
A 2 =0	B 2 =0	C 2 =0	D 2 =0	E 2 =0	F 2 =0
A 3 =0	B 3 =0	C 3 =0	D 3 =0	E 3 =0	F 3 =0
A 4 =0	B 4 =0	C 4 =0	D 4 =0	E 4 =0	F 4 =0
A 5 =0	B 5 =0	C 5 =0	D 5 =0	E 5 =0	F 5 =0
A 6 =0	B 6 =0	C 6 =0	D 6 =0	E 6 =0	F 6 =0
A 7 =0	B 7 =0	C 7 =0	D 7 =0	E 7 =0	F 7 =0
A 8 =0	B 8 =0	C 8 =1	D 8 =0	E 8 =0	F 8 =0
A 9 =1	B 9 =1	C 9 =1	D 9 =1	E 9 =0	F 9 =0
A 10 =1	B 10 =1	C 10 =1	D 10 =1	E 10 =1	F 10 =0
A 11 =1	B 11 =1	C 11 =1	D 11 =1	E 11 =0	F 11 =0
A 12 =1	B 12 =1	C 12 =1	D 12 =1	E 12 =0	F 12 =0

G 1 =0	H 1 =0	I 1 =1	J 1 =1	K 1 =1	L 1 =1
G 2 =0	H 2 =0	I 2 =1	J 2 =1	K 2 =1	L 2 =1
G 3 =0	H 3 =0	I 3 =1	J 3 =1	K 3 =1	L 3 =1
G 4 =0	H 4 =0	I 4 =1	J 4 =1	K 4 =1	L 4 =1
G 5 =0	H 5 =0	I 5 =0	J 5 =1	K 5 =0	L 5 =0
G 6 =0	H 6 =0	I 6 =0	J 6 =0	K 6 =0	L 6 =0
G 7 =0	H 7 =0	I 7 =0	J 7 =0	K 7 =0	L 7 =0
G 8 =0	H 8 =0	I 8 =0	J 8 =0	K 8 =0	L 8 =0
G 9 =0	H 9 =0	I 9 =0	J 9 =0	K 9 =0	L 9 =0
G 10 =0	H 10 =0	I 10 =0	J 10 =0	K 10 =0	L 10 =0
G 11 =0	H 11 =0	I 11 =0	J 11 =0	K 11 =0	L 11 =0
G 12 =0	H 12 =0	I 12 =0	J 12 =0	K 12 =0	L 12 =0

Ahora lo muestro, para el juego de Reino y aliados vs. Reino y aliados

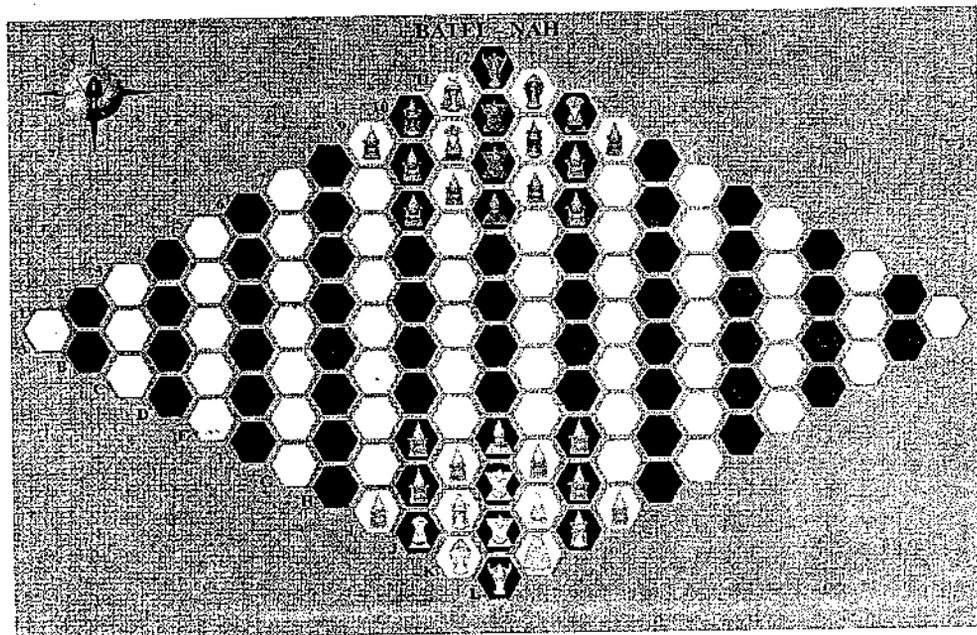


A 1 =1	B 1 =1	C 1 =1	D 1 =1	E 1 =0	F 1 =0
A 2 =1	B 2 =1	C 2 =1	D 2 =0	E 2 =0	F 2 =0
A 3 =1	B 3 =1	C 3 =0	D 3 =0	E 3 =0	F 3 =0
A 4 =1	B 4 =0	C 4 =0	D 4 =0	E 4 =0	F 4 =0
A 5 =0	B 5 =0	C 5 =0	D 5 =0	E 5 =0	F 5 =0
A 6 =0	B 6 =0	C 6 =0	D 6 =0	E 6 =0	F 6 =0
A 7 =0	B 7 =0	C 7 =0	D 7 =0	E 7 =0	F 7 =0
A 8 =0	B 8 =0	C 8 =1	D 8 =0	E 8 =0	F 8 =0
A 9 =1	B 9 =1	C 9 =1	D 9 =1	E 9 =0	F 9 =0
A 10 =1	B 10 =1	C 10 =1	D 10 =1	E 10 =1	F 10 =0
A 11 =1	B 11 =1	C 11 =1	D 11 =1	E 11 =0	F 11 =0
A 12 =1	B 12 =1	C 12 =1	D 12 =1	E 12 =0	F 12 =0

G 1 =0	H 1 =0	I 1 =1	J 1 =1	K 1 =1	L 1 =1
G 2 =0	H 2 =0	I 2 =1	J 2 =1	K 2 =1	L 2 =1
G 3 =0	H 3 =1	I 3 =1	J 3 =1	K 3 =1	L 3 =1
G 4 =0	H 4 =0	I 4 =1	J 4 =1	K 4 =1	L 4 =1
G 5 =0	H 5 =0	I 5 =0	J 5 =1	K 5 =0	L 5 =0
G 6 =0	H 6 =0	I 6 =0	J 6 =0	K 6 =0	L 6 =0
G 7 =0	H 7 =0	I 7 =0	J 7 =0	K 7 =0	L 7 =0
G 8 =0	H 8 =0	I 8 =0	J 8 =0	K 8 =0	L 8 =0
G 9 =0	H 9 =0	I 9 =0	J 9 =0	K 9 =0	L 9 =1
G 10 =0	H 10 =0	I 10 =0	J 10 =0	K 10 =1	L 10 =1
G 11 =0	H 11 =0	I 11 =0	J 11 =1	K 11 =1	L 11 =1
G 12 =0	H 12 =0	I 12 =1	J 12 =1	K 12 =1	L 12 =1

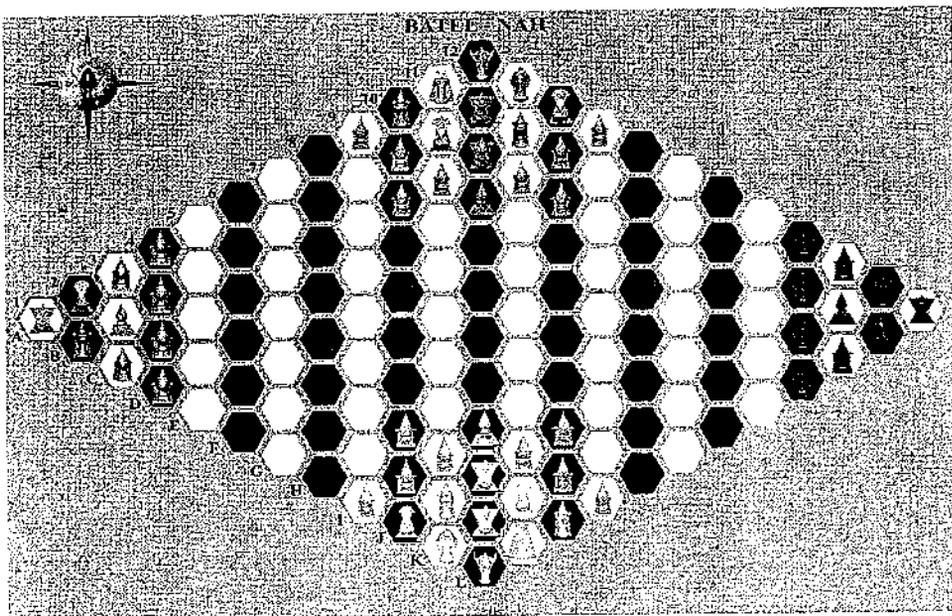
Al inicio del juego las piezas ya están colocadas en un espacio, a esta situación inicial la llamaré "Matriz Inicial de Carga" MIC.

MIC para dos jugadores.



Blancas				Negras			
Pieza	posinic	edoesp	edopza	Pieza	posinic	edoesp	edopza
pb	L1	1	1	pn	A12	1	1
rnb	K1	1	1	rn	B12	1	1
ryb	L2	1	1	ryn	A11	1	1
g1b	J2	1	1	g1n	A10	1	1
g2b	L3	1	1	g2n	C11	1	1
t1b	J1	1	1	t1n	C12	1	1
t2b	K3	1	1	t2n	B10	1	1
b1b	K2	1	1	b1n	B11	1	1
b2b	J3	1	1	b2n	C10	1	1
bfb	I4	1	1	bfn	D9	1	1
s1b	I1	1	1	s1n	A9	1	1
s2b	I2	1	1	s2n	B9	1	1
s3b	H3	1	1	s3n	C8	1	1
s4b	I3	1	1	s4n	C9	1	1
s5b	J4	1	1	s5n	D10	1	1
s6b	J5	1	1	s6n	E10	1	1
s7b	K4	1	1	s7n	D11	1	1
s8b	L4	1	1	s8n	D12	1	1

MIC para cuatro jugadores.



Pieza	Blancas			Negras			Azules			Rojas					
	posinic	edoesp	edopza	Pieza	posinic	edoesp	edopza	Pieza	posinic	edoesp	edopza	Pieza	posinic	edoesp	edopza
pb	L1	1	1	pn	A12	1	1	ga	K12	1	1	gr	B1	1	1
mb	K1	1	1	mn	B12	1	1	ta	L11	1	1	tr	A2	1	1
rb	L2	1	1	rn	A11	1	1	ba	L12	1	1	br	A1	1	1
g1b	J2	1	1	g1n	A10	1	1	bfa	K11	1	1	bfr	B2	1	1
g2b	L3	1	1	g2n	C11	1	1	s1a	I12	1	1	s1r	A4	1	1
t1b	J1	1	1	t1n	C12	1	1	s2a	J11	1	1	s2r	B3	1	1
t2b	K3	1	1	t2n	B10	1	1	s3a	K10	1	1	s3r	C2	1	1
b1b	K2	1	1	b1n	B11	1	1	s4a	L9	1	1	s4r	D1	1	1
b2b	J3	1	1	b2n	C10	1	1	s5a	J12	1	1	s5r	C1	1	1
bfb	I4	1	1	bfn	D9	1	1	s6a	L10	1	1	s6r	A3	1	1
s1b	I1	1	1	s1n	A9	1	1								
s2b	I2	1	1	s2n	B9	1	1								
s3b	H3	1	1	s3n	C8	1	1								
s4b	I3	1	1	s4n	C9	1	1								
s5b	J4	1	1	s5n	D10	1	1								
s6b	J5	1	1	s6n	E10	1	1								
s7b	K4	1	1	s7n	D11	1	1								
s8b	L4	1	1	s8n	D12	1	1								

El resto de los espacios tiene un valor de cero al inicio de la partida, la columna de edopza, indica si la pieza ha sido eliminada o no.

Una vez iniciada la partida, la MIC, deberá ir actualizándose con cada tiro, esta matriz ahora recibirá el nombre de Matriz de Juego MDJ

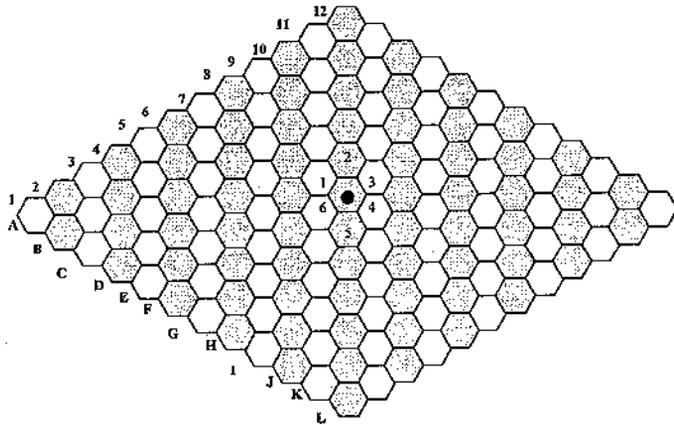
3.2 Definición de algoritmos de los movimientos de la piezas

Cada pieza tiene su movimiento particular, por lo que es necesario crear el algoritmo de movimiento para cada una, al igual que el tablero, su posición estará dada por las coordenadas, partiendo de la posición en que se encuentre ubicada, además las piezas indicarán los espacios que están siendo ocupados y por quien.

El sistema validará que los movimientos que el usuario haga se realicen en espacios válidos, cuando esto no sea así la pieza se regresará automáticamente a su posición de origen.

De igual forma se asignará un valor binario para definir cuando una pieza está "viva", o ha sido eliminada, y por lo tanto deberá ser quitada del tablero de juego, dichos valores serán 1 para viva y 0 para eliminada

El rey
Esta pieza tiene 6 posibles opciones de movimiento

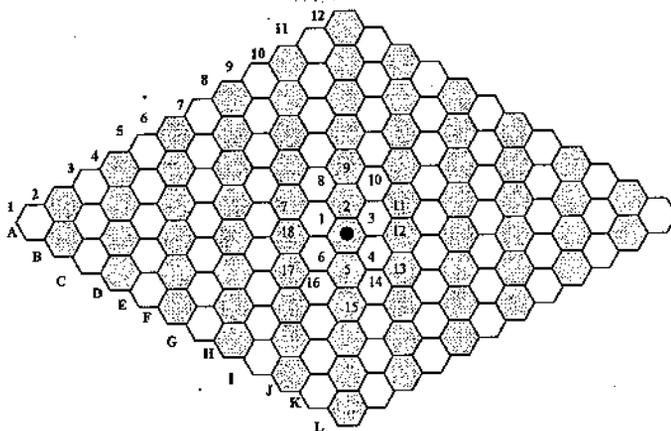


Movimiento del rey (ry)

- 1(coord1-1) + (coord2)
- 2(coord1-1) + (coord2+1)
- 3(coord1) + (coord2+1)
- 4(coord1+1) + (coord2)
- 5(coord1+1) + (coord2-1)
- 6(coord1) + (coord2-1)

La Reina.

Tiene 18 posibles opciones de movimiento

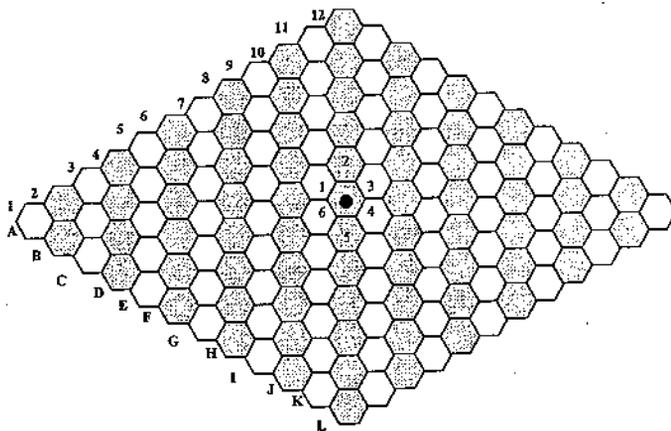


Movimiento de la reina (m)

- 1(coord1-1) + (coord2)
- 2(coord1-1) + (coord2+1)
- 3(coord1) + (coord2+1)
- 4(coord1+1) + (coord2)
- 5(coord1+1) + (coord2-1)
- 6(coord1) + (coord2-1)
- 7(coord1-2) + (coord2)
- 8(coord1-2) + (coord2+1)
- 9(coord1-2)+ (coord2+2)
- 10(coord1-1) + (coord2+2)
- 11(coord1) + (coord2+2)
- 12(coord1+1) + (coord2+1)
- 13(coord1+2) + (coord2)
- 14(coord1+2) + (coord2-1)
- 15(coord1+2) + (coord2-2)
- 16(coord1+1) + (coord2-2)
- 17(coord1) + (coord2-2)
- 18(coord1-1) + (coord2-1)

El soldado.

Tiene 6 posibles opciones de movimiento



Movimiento del soldado (s)

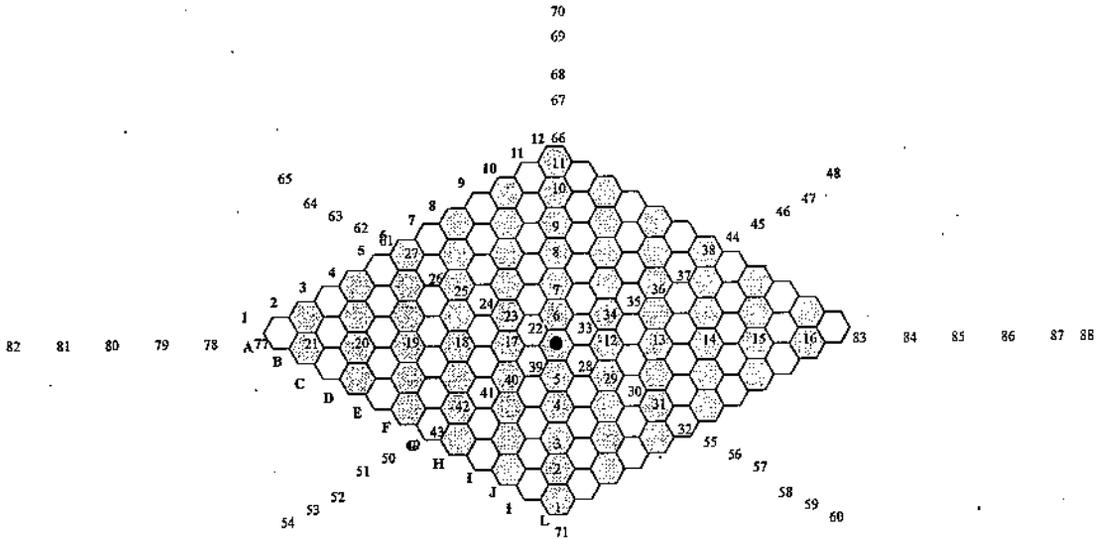
- 1(coord1-1) + (coord2)
- 2(coord1-1) + (coord2+1)
- 3(coord1) + (coord2+1)
- 4(coord1+1) + (coord2)
- 5(coord1+1) + (coord2-1)
- 6(coord1) + (coord2-1) .

Para eliminar una pieza sólo en:

- 2(coord1-1) + (coord2+1)
- 5(coord1+1) + (coord2-1)

El Príncipe

Tiene 88 posibles opciones de movimiento



Movimiento del príncipe (p)

- 1(coord1+5) + (coord2-5)
- 2(coord1+4) + (coord2-4)
- 3(coord1+3) + (coord2-3)
- 4(coord1+2) + (coord2-2)
- 5(coord1+1) + (coord2-1)
- 6(coord1-1) + (coord2+1)
- 7(coord1-2) + (coord2+2)
- 8(coord1-3) + (coord2+3)
- 9(coord1-4) + (coord2+4)
- 10(coord1-5) + (coord2+5)
- 11(coord1-6) + (coord2+6)
- 12(coord1+1) + (coord2+1)
- 13(coord1+2) + (coord2+2)
- 14(coord1+3) + (coord2+3)
- 15(coord1+4) + (coord2+4)
- 16(coord1+5) + (coord2+5)
- 17(coord1-1) + (coord2-1)
- 18(coord1-2) + (coord2-2)
- 19(coord1-3) + (coord2-3)
- 20(coord1-4) + (coord2-4)
- 21(coord1-5) + (coord2-5)
- 22(coord1-1) + (coord2)
- 23(coord1-2) + (coord2)
- 24(coord1-3) + (coord2)
- 25(coord1-4) + (coord2)
- 26(coord1-5) + (coord2)
- 27(coord1-6) + (coord2)
- 28(coord1+1) + (coord2)
- 29(coord1+2) + (coord2)
- 30(coord1+3) + (coord2)

- 31(coord1+4) + (coord2)
- 32(coord1+5) + (coord2)
- 33(coord1) + (coord2+1)
- 34(coord1) + (coord2+2)
- 35(coord1) + (coord2+3)
- 36(coord1) + (coord2+4)
- 37(coord1) + (coord2+5)
- 38(coord1) + (coord2+6)
- 39(coord1) + (coord2-1)
- 40(coord1) + (coord2-2)
- 41(coord1) + (coord2-3)
- 42(coord1) + (coord2-4)
- 43(coord1) + (coord2-5)

si p está en línea l

- 44(coord1) + (coord2+7)
- 45(coord1) + (coord2+8)
- 46(coord1) + (coord2+9)
- 47(coord1) + (coord2+10)
- 48(coord1) + (coord2+11)

si p está en línea l2

- 49(coord1) + (coord2-6)
- 50(coord1) + (coord2-7)
- 51(coord1) + (coord2-8)
- 52(coord1) + (coord2-9)
- 53(coord1) + (coord2-10)
- 54(coord1) + (coord2-11)

72
73
74
75
76

si p está en línea A

- 55(coord1+6) + (coord2)
- 56(coord1+7) + (coord2)
- 57(coord1+8) + (coord2)
- 58(coord1+9) + (coord2)
- 59(coord1+10) + (coord2)
- 60(coord1+11) + (coord2)

si p está en línea L

- 61(coord1-7) + (coord2)
- 62(coord1-8) + (coord2)
- 63(coord1-9) + (coord2)
- 64(coord1-10) + (coord2)
- 65(coord1-11) + (coord2)

si p está en L1

- 66(coord1-7) + (coord2+7)
- 67(coord1-8) + (coord2+8)
- 68(coord1-9) + (coord2+9)
- 69(coord1-10) + (coord2+10)
- 70(coord1-11) + (coord2+11)

si p está en A12

- 71(coord1+6) + (coord2-6)
- 72(coord1+7) + (coord2-7)
- 73(coord1+8) + (coord2-8)
- 74(coord1+9) + (coord2-9)
- 75(coord1+10) + (coord2-10)
- 76(coord1+11) + (coord2-11)

si p está en L12

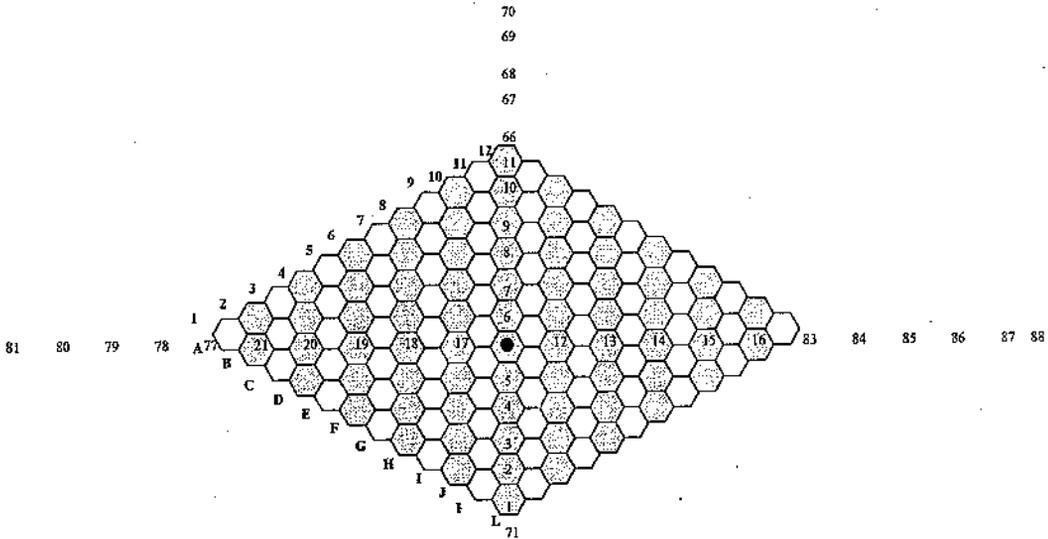
- 77(coord1-6) + (coord2-6)
- 78(coord1-7) + (coord2-7)
- 79(coord1-8) + (coord2-8)
- 80(coord1-9) + (coord2-9)
- 81(coord1-10) + (coord2-10)
- 82(coord1-11) + (coord2-11)

si p está en A1

- 83(coord1+6) + (coord2+6)
- 84(coord1+7) + (coord2+7)
- 85(coord1+8) + (coord2+8)
- 86(coord1+9) + (coord2+9)
- 87(coord1+10) + (coord2+10)
- 88(coord1+11) + (coord2+11)

El templo.

Tiene 44 posibles opciones de movimiento



Movimiento del templo (t)

- 1(coord1+5) + (coord2-5)
- 2(coord1+4) + (coord2-4)
- 3(coord1+3) + (coord2-3)
- 4(coord1+2) + (coord2-2)
- 5(coord1+1) + (coord2-1)
- 6(coord1-1) + (coord2+1)
- 7(coord1-2) + (coord2+2)
- 8(coord1-3) + (coord2+3)
- 9(coord1-4) + (coord2+4)
- 10(coord1-5) + (coord2+5)
- 11(coord1-6) + (coord2+6)
- 12(coord1+1) + (coord2+1)
- 13(coord1+2) + (coord2+2)
- 14(coord1+3) + (coord2+3)
- 15(coord1+4) + (coord2+4)
- 16(coord1+5) + (coord2+5)
- 17(coord1-1) + (coord2-1)
- 18(coord1-2) + (coord2-2)
- 19(coord1-3) + (coord2-3)
- 20(coord1-4) + (coord2-4)
- 21(coord1-5) + (coord2-5)

si t esta en L1

- 22(coord1-7) + (coord2+7)
- 23(coord1-8) + (coord2+8)
- 24(coord1-9) + (coord2+9)
- 25(coord1-10) + (coord2+10)
- 26(coord1-11) + (coord2+11)

si t esta en A12

- 27(coord1+6) + (coord2-6)
- 28(coord1+7) + (coord2-7)
- 29(coord1+8) + (coord2-8)
- 30(coord1+9) + (coord2-9)
- 31(coord1+10) + (coord2-10)
- 32(coord1+11) + (coord2-11)

si t esta en L12

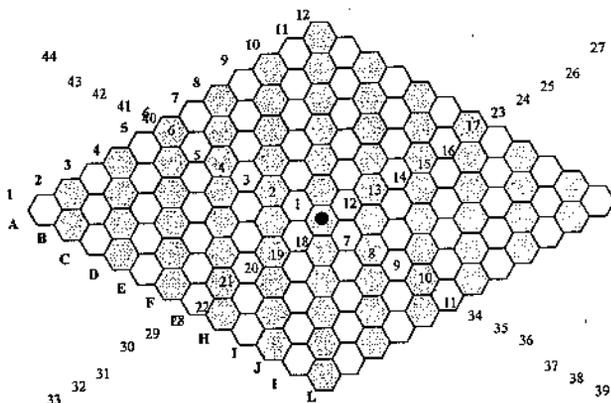
- 33(coord1-6) + (coord2-6)
- 34(coord1-7) + (coord2-7)
- 35(coord1-8) + (coord2-8)
- 36(coord1-9) + (coord2-9)
- 37(coord1-10) + (coord2-10)
- 38(coord1-11) + (coord2-11)

si t esta en A1

- 39(coord1+6) + (coord2+6)
- 40(coord1+7) + (coord2+7)
- 41(coord1+8) + (coord2+8)
- 42(coord1+9) + (coord2+9)
- 43(coord1+10) + (coord2+10)
- 44(coord1+11) + (coord2+11)

El guerrero

Tiene 44 posibles opciones de movimiento



Movimiento del guerrero (g)

Si g esta en línea 1

- 1(coord1-1) + (coord2)
- 2(coord1-2) + (coord2)
- 3(coord1-3) + (coord2)
- 4(coord1-4) + (coord2)
- 5(coord1-5) + (coord2)
- 6(coord1-6) + (coord2)
- 7(coord1+1) + (coord2)
- 8(coord1+2) + (coord2)
- 9(coord1+3) + (coord2)

- 23(coord1) + (coord2+7)
- 24(coord1) + (coord2+8)
- 25(coord1) + (coord2+9)
- 26(coord1) + (coord2+10)
- 27(coord1) + (coord2+11)

si g esta en línea 12

- 28(coord1) + (coord2-6)
- 29(coord1) + (coord2-7)
- 30(coord1) + (coord2-8)
- 31(coord1) + (coord2-9)
- 32(coord1) + (coord2-10)
- 33(coord1) + (coord2-11)

si g esta en línea A

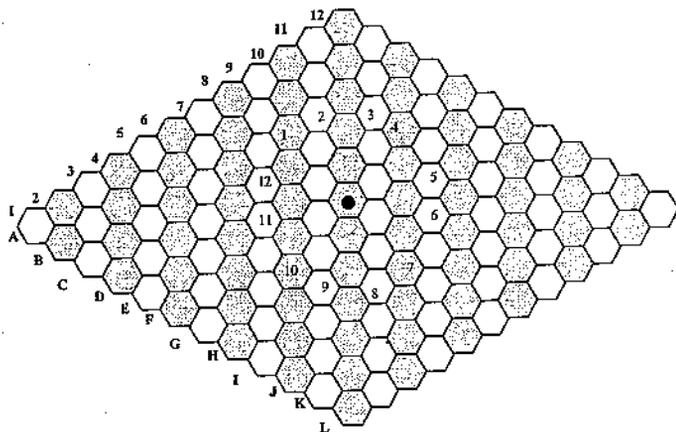
- 34(coord1+6) + (coord2)
- 35(coord1+7) + (coord2)
- 36(coord1+8) + (coord2)
- 37(coord1+9) + (coord2)
- 38(coord1+10) + (coord2)
- 39(coord1+11) + (coord2)

si g esta en línea L

- 40(coord1-7) + (coord2)
- 41(coord1-8) + (coord2)
- 42(coord1-9) + (coord2)
- 43(coord1-10) + (coord2)
- 44(coord1-11) + (coord2)

El brujo

Tiene 12 posibles opciones de movimiento

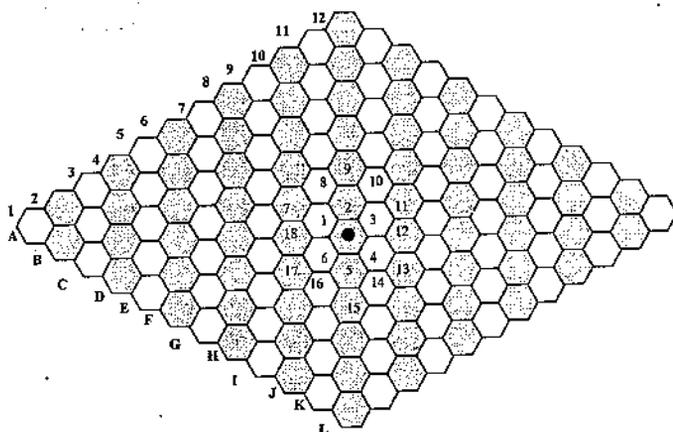


Movimiento del brujo (b)

- 1(coord1-3) + (coord2+1)
- 2(coord1-3) + (coord2+2)
- 3(coord1-2) + (coord2+3)
- 4(coord1-1) + (coord2+3)
- 5(coord1+1) + (coord2+2)
- 6(coord1+2) + (coord2+1)
- 7(coord1+3) + (coord2-1)
- 8(coord1+3) + (coord2-2)
- 9(coord1+2) + (coord2-3)
- 10(coord1+1) + (coord2-3)
- 11(coord1-1) + (coord2-2)
- 12(coord1-2) + (coord2-1)

El bufón

Tiene 18 posibles opciones de movimiento



Movimiento del bufón (bf)

- 1(coord1-1) + (coord2)
- 2(coord1-1) + (coord2+1)
- 3(coord1) + (coord2+1)
- 4(coord1+1) + (coord2)
- 5(coord1+1) + (coord2-1)
- 6(coord1) + (coord2-1)
- 7(coord1-2) + (coord2)
- 8(coord1-2) + (coord2+1)
- 9(coord1-2)+(coord2+2)
- 10(coord1-1) + (coord2+2)
- 11(coord1) + (coord2+2)
- 12(coord1+1) + (coord2+1)
- 13(coord1+2) + (coord2)
- 14(coord1+2) + (coord2-1)
- 15(coord1+2) + (coord2-2)
- 16(coord1+1) + (coord2-2)
- 17(coord1) + (coord2-2)
- 18(coord1-1) + (coord2-1)

Para eliminar una pieza

Sólo si la pieza a eliminar está en el espacio siguiente en forma diagonal con respecto al tablero, el bufón se colocará

en el espacio siguiente a la pieza eliminada en la misma dirección diagonal.

Ej,

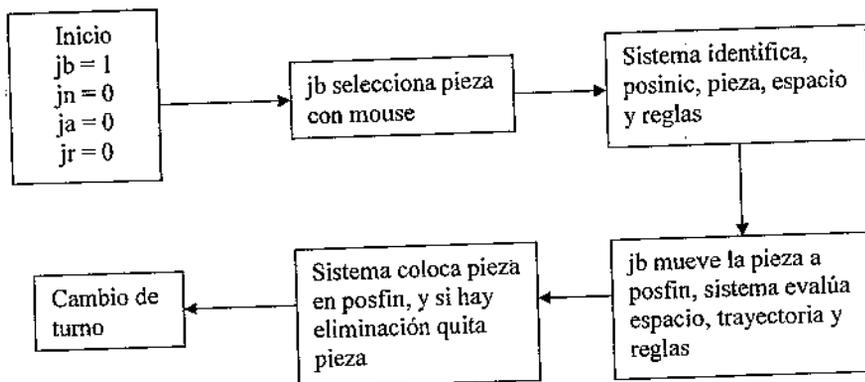
Posición inicial bufón	Posición pieza a eliminar	Posición final bufón	Movimiento
G6	F6	E6	7(coord1-2)+(coord2)
G6	G7	G8	11(coord1)+(coord2+2)
G6	H6	I6	13(coord1+2)+(coord2)
G6	G5	G4	17(coord1)+(coord2-2)

3.3 Lógica de Movimientos

Al inicio del juego el sistema deberá saber en qué espacio va cada pieza, qué espacios están libres, conocer los movimientos válidos, e identificar de quién es turno de tiro, además en cada movimiento deberá validar si las trayectorias de desplazamiento están libres u ocupadas. Para lograr esto el elemento base serán las piezas, dentro de la programación de cada una de ellas se incluirán todas estas validaciones, por lo que explicaré esto con un diagrama de flujo y tablas.

jb = jugador blanco posinic = posición inicial
jn = jugador negro posfin = posición final
ja = jugador azul
jr = jugador rojo

El que tenga el turno de tiro tendrá el valor de 1

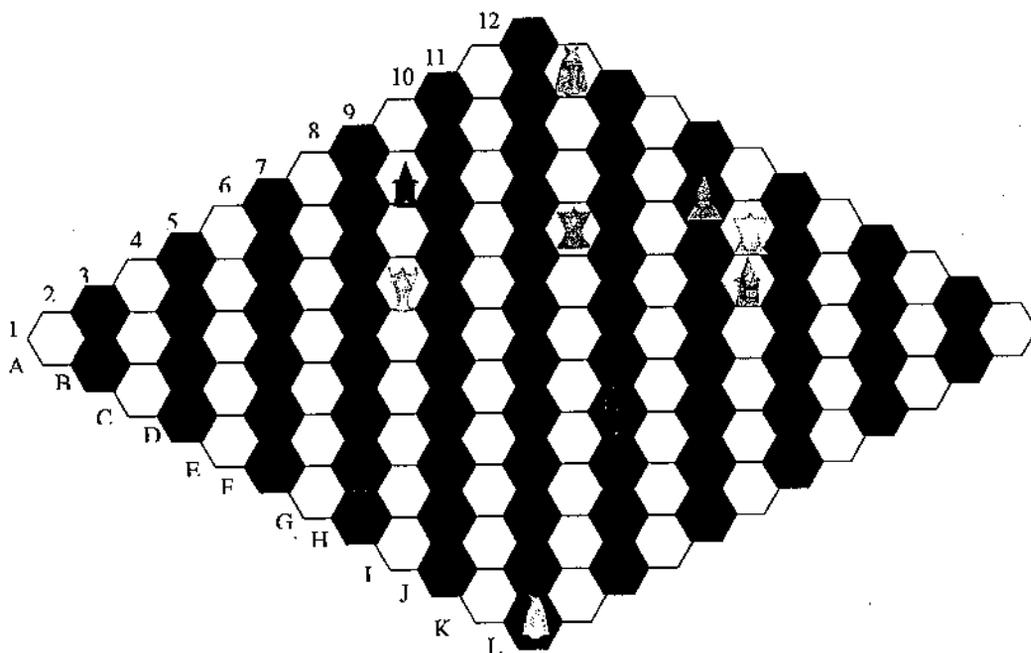


Cuando el sistema evalúa la regla de movimiento de una pieza y encuentra un espacio ocupado, o igual a 1, todo el rango siguiente de esa trayectoria en específico se invalidará como opción de movimiento.

Sólo aplicará excepción en el caso del brujo y del bufón, con estas piezas se utilizarán reglas especiales.

En el caso de los extremos del tablero cuando el sistema realiza el análisis de movimiento en función de las reglas de cada pieza, también realizará una evaluación para definir en dónde se acaba el tablero, por lo que usando las coordenadas comparará el valor de la posición, por lo que un valor mayor de "L" y un valor mayor de "12", serán inválidos y retornará la pieza a su posición original.

Ejemplo. Supongamos que tenemos el siguiente tablero:



Para este ejemplo se generaría la siguiente tabla de datos, en donde tenemos el identificador de cada pieza, su posición inicial, las opciones de espacios a los que se podría mover según las reglas individuales de cada una de las piezas, las opciones válidas reales para mover la pieza, y los topes para cada trayectoria.

Pieza	posinic	opesp	opval	topes
SA	B8=1	A8=0,A9=0,B7=0,B9=0,C7=0,C8=0	A8,A9,B7,B9,C7,C8	
RyN	B12=1	A12=0,B11=0,C11=0,C12=0	A12,B11,C11,C12	
PB	D6=1	A3=0,A6=0,A9=0,A10=0, B4=0,B6=0,B8=1,C5=0,C7=0, d1=0,D2=0,D3=0,D4=0,D5=0,D7=0,D8=0,D9=0, D10=0,D11=0,D12=0,E5=0, E6=0,F4=0,F6=0, F8=0, G3=0,G6=0,G9=0,H2=0,H6=0,H10=1,I1=0,I6=1, I11=0,J6=0,J12=0,K6=0,L6=0	A3,A6,B4,B6,C5,C7,D1,D2,D3,D4,D5,D7,D8,D9,D10,D11,D12, E5,E6,F4,F6,F8,G3,G6, G9,H2,H6,I1	SA(B8), SN(H10), GA(I6)
BN	E9=1	B10=0,B11=0,C8=0,C12=0,D7=0,D12=0, F6=0,G6=0,H7=0,H8=0,G10=0,F11=1	B10,B11,C8,C12,D7,D12,F6,G6,H7,H8,G10	
BFN	F11=1	D11=0,D12=0,E10=0,E11=0,E12=0, F9=0,F10=0,F12=0,G9=0,G10=0,G11=1, G12=0,H9=0,H10=0,H1=0	D11,D12,E10,E11,E12,F9,F10, F12,G9,G10,G12,H9,H11	
BRB	G11=1	D12=0,E10=0,F9=0,H8=0,I8=0,I9=0,J10=0,I12=0	D12,E10,F9,H8,I8,J9,J10,I12	
SN	H10=1	G10=0,G11=1,H9=0,H11=0,I9=0,I10=0	G10,G11,H9,H11,I9,I10	BRB(G11)
GA	I6=1	A6=0,B6=0,C6=0,D6=1,E6=0,F6=0,G6=0, H6=0,I1=0,I2=0,I3=0,I4=0,I5=0,I7=0,I8=0,I9=0, I10=0,I11=0,I12=0,J6=0,K6=0,L8=0	D6,E6,F6,G6,H6,I1,I2,I3,I4,I5,I7,I8,I9,I10,I11,I12,J6,K6,L6	PB(D6)
RyB	L1	K1=0,K2=0,L2=0	K1,K2,L2	

Posinic = posición inicial

Opesp = opción de espacio

Opval = opción de espacio válida

Tope = pieza que ocupa el espacio en la trayectoria de movimiento de otra pieza, impidiendo el paso continuo sobre dicha trayectoria.

El sistema también deberá identificar qué piezas son aliadas y cuáles no, por lo que se dará un identificador a cada pieza por su color, esto estará dado por la letra final del identificador individual de las piezas.

	Blancas		Negras		Azules		Rojas	
	id pieza	id color						
Rey	Ry	b	Ry	n	Ry	a	Ry	r
Reina	Rn	b	Rn	n	Rn	a	Rn	r
Príncipe	P	b	P	n	P	a	P	r
Guerrero	G	b	G	n	G	a	G	r
Templo	T	b	T	n	T	a	T	r
Brujo	B	b	B	n	B	a	B	r
Bufón	Bf	b	Bf	n	Bf	a	Bf	r
Soldado	S	b	S	n	S	a	S	r

De esta manera el sistema tiene toda la información de la ubicación de las piezas, así como la manera de calcular los movimientos, por lo que al momento en que un usuario realice su tirada, sólo habrá que aplicar la regla para la pieza seleccionada, y validar en el momento en que el usuario suelte la pieza en su posición final.

Usando el tablero anterior supongamos que el usuario es el de las piezas blancas, y mueve al príncipe para eliminar al soldado negro.

Primero el sistema identificará que el usuario quiere mover al príncipe cuando éste de un clic sobre la pieza, en ese momento el sistema obtendrá la posición inicial, validará los espacios válidos según las reglas, y validará cuales son opciones válidas para colocar las piezas, de igual forma identificará las posibles piezas ubicadas en la trayectoria, verificando si son aliadas o enemigas



En ese momento se genera la Matriz de Juego (MDJ) y obtiene todos los datos necesarios para la validación de movimientos y eliminación.

tope					
pieza	posinic	opesp	opvai	enem	aliado
PB	D6=1	A3=0,A6=0,A9=0,A10=0, B4=0,B6=0,B8=1,C5=0, C7=0, d1=0,D2=0,D3=0, D4=0,D5=0,D7=0,D8=0, D9=0, D10=0,D11=0, D12=0, E5=0, E6=0,F4=0, F6=0, F8=0, G3=0,G6=0,G9=0, H2=0,H6=0,H10=1, I1=0,I6=1,I11=0,J6=0, J12=0,K6=0,L6=0	A3,A6,B4,B6, C5,C7,D1,D2, D3,D4,D5,D7, D8, D9,D10,D11, D12,E5,E6,F4, F6,F8,G3,G6, G9,H2,H6, I1	SN(H10)	SA(B8),GA(I6)

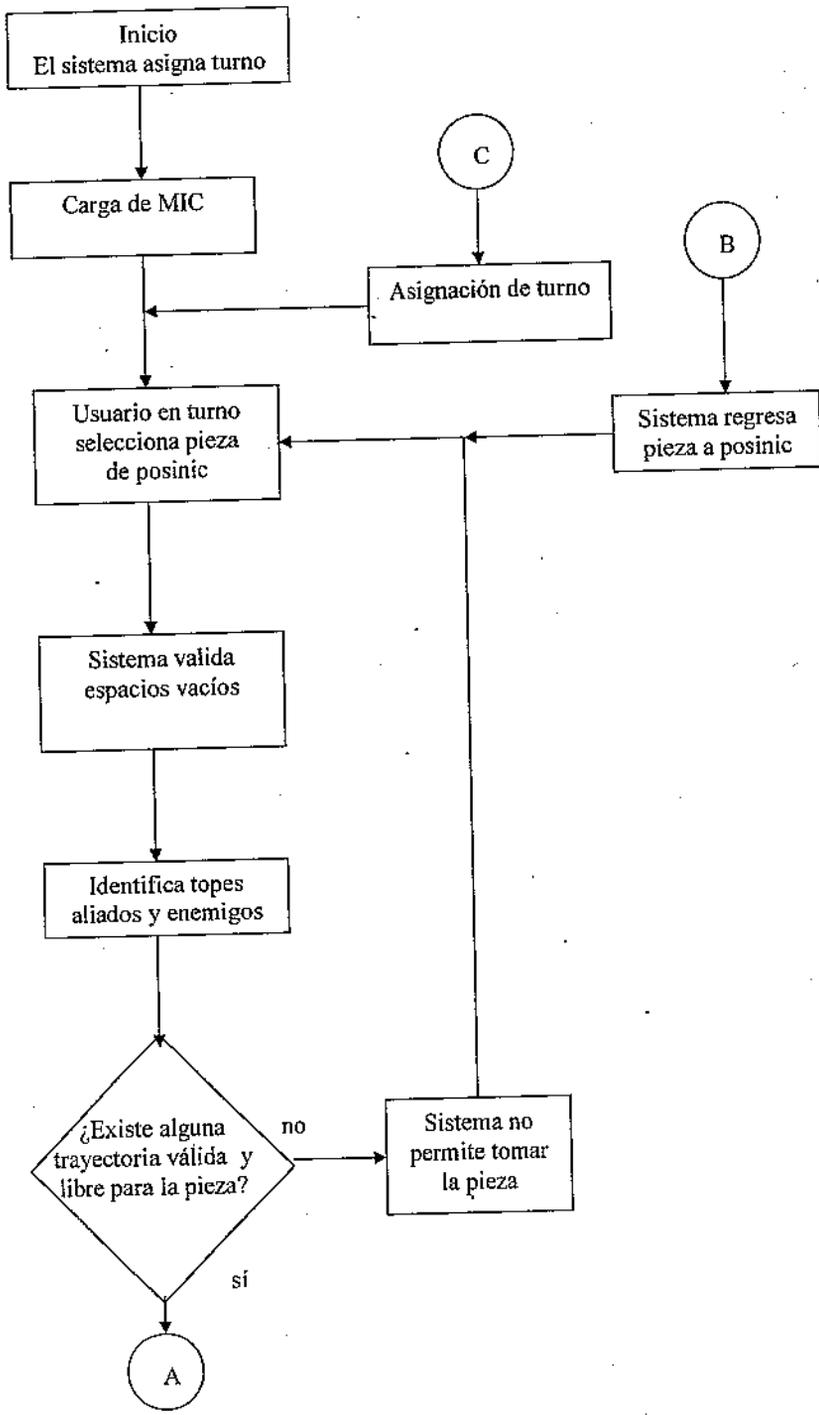
Como las piezas blancas son aliadas de las azules el sistema separará las piezas a eliminar, quedando sólo el soldado negro. Al final del movimiento el príncipe quedará en la posición H10 y el soldado será eliminado del tablero

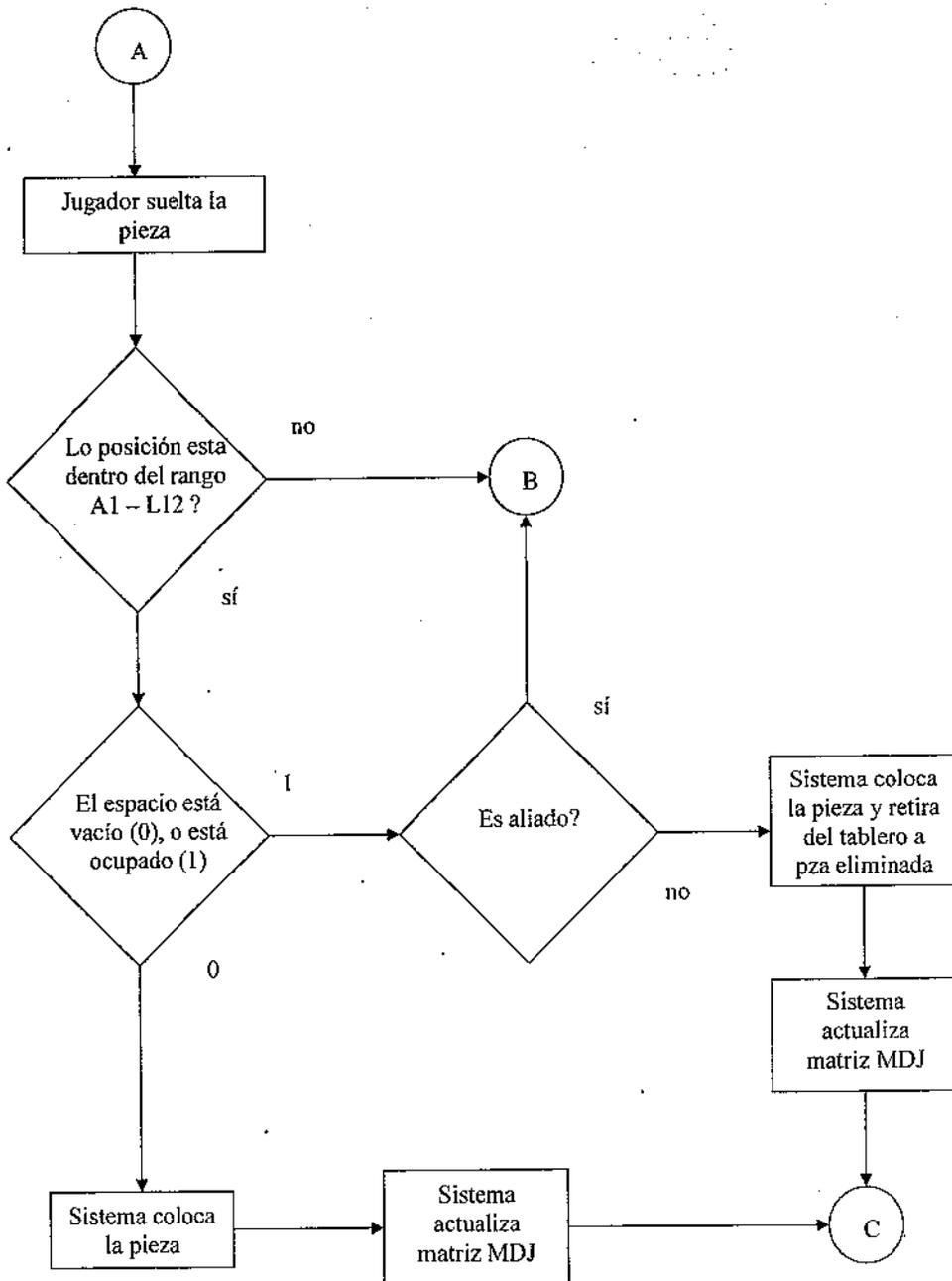
La pieza se "pega al puntero" mientras se mantenga presionado el botón, movimiento de "drag and drop", que hace que la pieza se mueva junto con el puntero en cualquier dirección. Al momento de soltarla el sistema comparará la posición final donde fue colocada, las validaciones a realizar son las siguientes:

1. ¿La coordenada 1 es menor que 1 y mayor a 12?
2. ¿La coordenada 2 es menor que 1 y mayor a 12?
3. ¿La trayectoria de movimiento es válida y estaba libre?
4. ¿ El espacio donde se colocó está ocupado?
5. Si está ocupado, ¿lo está por un enemigo?

Una vez que "libera la pieza" soltando el clic del mouse en una posición válida el sistema actualizará su nueva posición y asignará el turno al siguiente jugador.

A continuación detallaré este procedimiento en un diagrama de flujo.





CAPÍTULO 4 *Implementación de la Interfase gráfica stand-alone*

Para poder crear el juego como tal, poder verlo y jugarlo en la PC, se realizó todo lo que se ha visto previamente, y ahora llegó el momento de decírselo a la máquina esto se hace por medio de un lenguaje de programación, dicho lenguaje fue Lingo, lenguaje de Director, Software de Macromedia.

Cuando un usuario ve el juego, sólo puede ver el tablero y las piezas, pero no puede ver todo lo que hay detrás de él, cada pieza, cada movimiento, cada validación, cada eliminación, cada movimiento incorrecto, etc, etc implica cientos de líneas de código para hacerlo posible.

A continuación mostraré como se realizó este juego.

4.1 Tablas y valores iniciales

Primero se creó una lista (tabla), con los contenidos de las posiciones iniciales, con esta lista se controla el curso del juego ya que sobre ella se irán actualizando los movimientos.

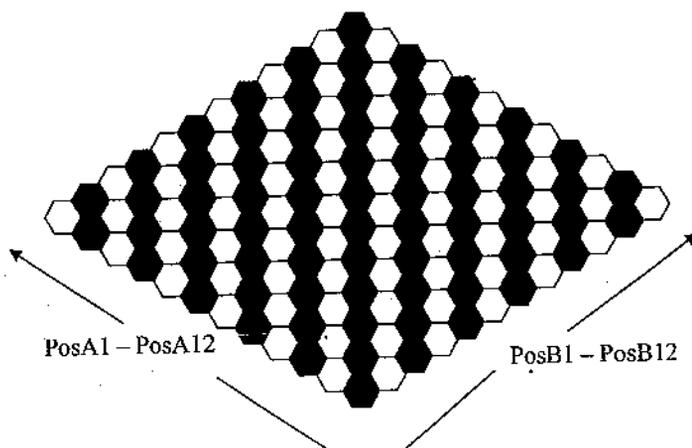
Esta lista inicialmente se carga con los valores de la versión de Reino vs. Reino, pero cuando se acciona el botón para cambiar de modo a Reino y Aliados vs. Reino y Aliados, debe ser capaz de actualizarse con todo el contenido de la lista, dicha actualización también será gráfica, por lo que deberán de aparecer todas las piezas.

Tabla Inicial de Lista

A1	0 - 50	E1	0	I1	11
A2	0 - 51	E2	0	I2	12
A3	0 - 54	E3	0	I3	13
A4	0 - 55	E4	0	I4	10
A5	0	E5	0	I5	0
A6	0	E6	0	I6	0
A7	0	E7	0	I7	0
A8	0	E8	0	I8	0
A9	47	E9	0	I9	0
A10	37	E10	44	I10	0
A11	32	E11	0	I11	0
A12	31	E12	0	I12	0 - 28
B1	0 - 52	F1	0	J1	9

B2	0 - 53	F2	0	J2	6
B3	0 - 56	F3	0	J3	5
B4	0	F4	0	J4	15
B5	0	F5	0	J5	18
B6	0	F6	0	J6	0
B7	0	F7	0	J7	0
B8	0	F8	0	J8	0
B9	46	F9	0	J9	0
B10	38	F10	0	J10	0
B11	34	F11	0	J11	0 - 27
B12	33	F12	0	J12	0 - 29
C1	0 - 59	G1	0	K1	3
C2	0 - 57	G2	0	K2	4
C3	0	G3	0	K3	8
C4	0	G4	0	K4	16
C5	0	G5	0	K5	0
C6	0	G6	0	K6	0
C7	0	G7	0	K7	0
C8	48	G8	0	K8	0
C9	45	G9	0	K9	0
C10	35	G10	0	K10	0 - 26
C11	36	G11	0	K11	0 - 23
C12	39	G12	0	K12	0 - 22
D1	0 - 58	H1	0	L1	1
D2	0	H2	0	L2	2
D3	0	H3	14	L3	7
D4	0	H4	0	L4	17
D5	0	H5	0	L5	0
D6	0	H6	0	L6	0
D7	0	H7	0	L7	0
D8	0	H8	0	L8	0
D9	40	H9	0	L9	0 - 25
D10	43	H10	0	L10	0 - 24
D11	42	H11	0	L11	0 - 21
D12	41	H12	0	L12	0 - 20

Después fue necesario crear un script que controlara las dimensiones del tablero por medio de las coordenadas, por lo que las trayectorias diagonales marcaban dichas dimensiones, quedando:



En donde cada variable de PosA y PosB, tomará un valor de coordenada del 1 al 12, por lo que cualquier valor menor que 1 y mayor que 12 indicará lugar fuera del tablero

En la siguiente tabla se muestran los valores numéricos de cada espacio.

Posiciones de coordenadas de los espacios:

Hex	posSild (posA)	posSdli (posB)	Hex	posSild (posA)	posSdli (posB)	Hex	posSild (posA)	posSdli (posB)
A1	12	1	E1	8	1	I1	4	1
A2	12	2	E2	8	2	I2	4	2
A3	12	3	E3	8	3	I3	4	3
A4	12	4	E4	8	4	I4	4	4
A5	12	5	E5	8	5	I5	4	5
A6	12	6	E6	8	6	I6	4	6
A7	12	7	E7	8	7	I7	4	7
A8	12	8	E8	8	8	I8	4	8
A9	12	9	E9	8	9	I9	4	9
A10	12	10	E10	8	10	I10	4	10
A11	12	11	E11	8	11	I11	4	11
A12	12	12	E12	8	12	I12	4	12
B1	11	1	F1	7	1	J1	3	1
B2	11	2	F2	7	2	J2	3	2
B3	11	3	F3	7	3	J3	3	3
B4	11	4	F4	7	4	J4	3	4
B5	11	5	F5	7	5	J5	3	5
B6	11	6	F6	7	6	J6	3	6

B7	11	7	F7	7	7	J7	3	7
B8	11	8	F8	7	8	J8	3	8
B9	11	9	F9	7	9	J9	3	9
B10	11	10	F10	7	10	J10	3	10
B11	11	11	F11	7	11	J11	3	11
B12	11	12	F12	7	12	J12	3	12
C1	10	1	G1	6	1	K1	2	1
C2	10	2	G2	6	2	K2	2	2
C3	10	3	G3	6	3	K3	2	3
C4	10	4	G4	6	4	K4	2	4
C5	10	5	G5	6	5	K5	2	5
C6	10	6	G6	6	6	K6	2	6
C7	10	7	G7	6	7	K7	2	7
C8	10	8	G8	6	8	K8	2	8
C9	10	9	G9	6	9	K9	2	9
C10	10	10	G10	6	10	K10	2	10
C11	10	11	G11	6	11	K11	2	11
C12	10	12	G12	6	12	K12	2	12
D1	9	1	H1	5	1	L1	1	1
D2	9	2	H2	5	2	L2	1	2
D3	9	3	H3	5	3	L3	1	3
D4	9	4	H4	5	4	L4	1	4
D5	9	5	H5	5	5	L5	1	5
D6	9	6	H6	5	6	L6	1	6
D7	9	7	H7	5	7	L7	1	7
D8	9	8	H8	5	8	L8	1	8
D9	9	9	H9	5	9	L9	1	9
D10	9	10	H10	5	10	L10	1	10
D11	9	11	H11	5	11	L11	1	11
D12	9	12	H12	5	12	L12	1	12

Ahora, ya que se puede delimitar el tablero en base a una relación de coordenadas es necesario crear el área de juego, dicha área está formada por dos partes: una, la gráfica que es lo que el usuario ve (un mapa de bits), y la otra, la lógica, que es la que emplea el sistema para determinar cada espacio del tablero. De igual forma tuve que crear una tabla con los valores de las posiciones individuales de cada área, para que el sistema pudiera evaluar cada movimiento.

La siguiente tabla muestra las áreas lógicas del juego con sus coordenadas en píxeles.

Tabla de posiciones Hexágonos Blancos

37 * 31 píxeles

Del centro al lado izquierdo del tablero

Del centro al lado derecho del tablero

posición tablero	coordenadas rectángulo						posición tablero	coordenadas rectángulo					
	xsi	ysi	xid	yid	locH	locV		xsi	ysi	xid	yid	locH	locV
A11	324	72	361	103	343	88	B12	386	72	423	103	405	88
B10	324	106	361	137	343	122	C11	386	106	423	137	405	122
C9	324	140	361	171	343	156	D10	386	140	423	171	405	156
D8	324	174	361	205	343	190	E9	386	174	423	205	405	190
E7	324	208	361	239	343	224	F8	386	208	423	239	405	224
F6	324	242	361	273	343	258	G7	386	242	423	273	405	258
G5	324	276	361	307	343	292	H6	386	276	423	307	405	292
H4	324	310	361	341	343	326	I5	386	310	423	341	405	326
I3	324	344	361	375	343	360	J4	386	344	423	375	405	360
J2	324	378	361	409	343	394	K3	386	378	423	409	405	394
K1	324	412	361	443	343	428	L2	386	412	423	443	405	428
A9	262	106	299	137	281	122	D12	448	106	485	137	467	122
B8	262	140	299	171	281	156	E11	448	140	485	171	467	156
C7	262	174	299	205	281	190	F10	448	174	485	205	467	190
D6	262	208	299	239	281	224	G9	448	208	485	239	467	224
E5	262	242	299	273	281	258	H8	448	242	485	273	467	258
F4	262	276	299	307	281	292	I7	448	276	485	307	467	292
G3	262	310	299	341	281	326	J6	448	310	485	341	467	326
H2	262	344	299	375	281	360	K5	448	344	485	375	467	360
I1	262	378	299	409	281	394	L4	448	378	485	409	467	394
A7	200	140	237	171	219	156	F12	510	140	547	171	529	156
B6	200	174	237	205	219	190	G11	510	174	547	205	529	190
C5	200	208	237	239	219	224	H10	510	208	547	239	529	224
D4	200	242	237	273	219	258	I9	510	242	547	273	529	258
E3	200	276	237	307	219	292	J8	510	276	547	307	529	292
F2	200	310	237	341	219	326	K7	510	310	547	341	529	326
G1	200	344	237	375	219	360	L6	510	344	547	375	529	360
A5	138	174	175	205	157	190	H12	572	174	609	205	591	190
B4	138	208	175	239	157	224	I11	572	208	609	239	591	224
C3	138	242	175	273	157	258	J10	572	242	609	273	591	258
D2	138	276	175	307	157	292	K9	572	276	609	307	591	292
E1	138	310	175	341	157	326	L8	572	310	609	341	591	326
A3	76	208	113	239	95	224	J12	634	208	671	239	653	224

B2	76	242	113	273	95	258	K11	634	242	671	273	653	258
C1	76	276	113	307	95	292	L10	634	276	671	307	653	292
A1	14	242	51	273	33	258	L12	696	242	733	273	715	258

Tabla de posiciones Hexágonos Negros

Del centro al lado izquierdo del tablero

Del centro al lado derecho del tablero

posición tablero	coordenadas rectángulo						posición tablero	coordenadas rectángulo					
	xsí	ysi	xid	yid	loch	locV		xsí	ysi	xid	yid	loch	locV
A12	355	56	392	87	374	72	C12	417	90	454	121	436	106
B11	355	90	392	121	374	106	D11	417	124	454	155	436	140
C10	355	124	392	155	374	140	E10	417	158	454	189	436	174
D9	355	158	392	189	374	174	F9	417	192	454	223	436	208
E8	355	192	392	223	374	208	G8	417	226	454	257	436	242
F7	355	226	392	257	374	242	H7	417	260	454	291	436	276
G6	355	260	392	291	374	276	I6	417	294	454	325	436	310
H5	355	294	392	325	374	310	J5	417	328	454	359	436	344
I4	355	328	392	359	374	344	K4	417	362	454	393	436	378
J3	355	362	392	393	374	378	L3	417	396	454	427	436	412
K2	355	396	392	427	374	412							
L1	355	430	392	461	374	446	E12	479	124	516	155	498	140
							F11	479	158	516	189	498	174
A10	293	90	330	121	312	106	G10	479	192	516	223	498	208
B9	293	124	330	155	312	140	H9	479	226	516	257	498	242
C8	293	158	330	189	312	174	I8	479	260	516	291	498	276
D7	293	192	330	223	312	208	J7	479	294	516	325	498	310
E6	293	226	330	257	312	242	K6	479	328	516	359	498	344
F5	293	260	330	291	312	276	L5	479	362	516	393	498	378
G4	293	294	330	325	312	310							
H3	293	328	330	359	312	344	G12	541	158	578	189	560	174
I2	293	362	330	393	312	378	H11	541	192	578	223	560	208
J1	293	396	330	427	312	412	I10	541	226	578	257	560	242
							J9	541	260	578	291	560	276
A8	231	124	268	155	250	140	K8	541	294	578	325	560	310
B7	231	158	268	189	250	174	L7	541	328	578	359	560	344
C6	231	192	268	223	250	208							
D5	231	226	268	257	250	242	I12	603	192	640	223	622	208
E4	231	260	268	291	250	276	J11	603	226	640	257	622	242
F3	231	294	268	325	250	310	K10	603	260	640	291	622	276
G2	231	328	268	359	250	344	L9	603	294	640	325	622	310
H1	231	362	268	393	250	378							
							K12	665	226	702	257	684	242

A6	169	158	206	189	188	174	L11	685	260	702	291	684	276
B5	169	192	206	223	188	208							
C4	169	226	206	257	188	242							
D3	169	260	206	291	188	276							
E2	169	294	206	325	188	310							
F1	169	328	206	359	188	344							
A4	107	192	144	223	126	208							
B3	107	226	144	257	126	242							
C2	107	260	144	291	126	276							
D1	107	294	144	325	126	310							
A2	45	226	82	257	64	242							
B1	45	260	82	291	64	276							

Con estos valores se puede determinar cuando una pieza interseca la coordenada de un espacio, de esta forma el sistema puede saber que la pieza está en ese espacio.

Para poder obtener el valor del espacio en base a la coordenada correspondiente utilicé la siguiente tabla, en donde cada "nombre" de coordenada (Ej. A1) se relaciona con su valor numérico en la lista.

Posiciones de los hexágonos en la lista

Posición	Núm	Posición	Núm	Posición	Núm
A1	1	E1	49	I1	97
A2	2	E2	50	I2	98
A3	3	E3	51	I3	99
A4	4	E4	52	I4	100
A5	5	E5	53	I5	101
A6	6	E6	54	I6	102
A7	7	E7	55	I7	103
A8	8	E8	56	I8	104
A9	9	E9	57	I9	105
A10	10	E10	58	I10	106
A11	11	E11	59	I11	107
A12	12	E12	60	I12	108
B1	13	F1	61	J1	109
B2	14	F2	62	J2	110
B3	15	F3	63	J3	111
B4	16	F4	64	J4	112
B5	17	F5	65	J5	113
B6	18	F6	66	J6	114
B7	19	F7	67	J7	115
B8	20	F8	68	J8	116
B9	21	F9	69	J9	117

B10	22	F10	70	J10	118
B11	23	F11	71	J11	119
B12	24	F12	72	J12	120
C1	25	G1	73	K1	121
C2	26	G2	74	K2	122
C3	27	G3	75	K3	123
C4	28	G4	76	K4	124
C5	29	G5	77	K5	125
C6	30	G6	78	K6	126
C7	31	G7	79	K7	127
C8	32	G8	80	K8	128
C9	33	G9	81	K9	129
C10	34	G10	82	K10	130
C11	35	G11	83	K11	131
C12	36	G12	84	K12	132
D1	37	H1	85	L1	133
D2	38	H2	86	L2	134
D3	39	H3	87	L3	135
D4	40	H4	88	L4	136
D5	41	H5	89	L5	137
D6	42	H6	90	L6	138
D7	43	H7	91	L7	139
D8	44	H8	92	L8	140
D9	45	H9	93	L9	141
D10	46	H10	94	L10	142
D11	47	H11	95	L11	143
D12	48	H12	96	L12	144

Para que el sistema pueda identificar a cada pieza, a cada una de éstas se le debe asignar un valor, dicho valor se muestra en la tabla siguiente:

Valores de las Piezas (res)

– Piezas Blancas rango de 1 a 18

Pb = 1	-- Príncipe Blanco
Rb = 2	-- Rey Blanco
Ryb = 3	-- Reina Blanca
Brb1 = 4	-- Brujo Blanco 1
Brb2 = 5	-- Brujo Blanco 2
Gb1 = 6	-- Guerrero Blanco 1
Gb2 = 7	-- Guerrero Blanco 2
Tb1 = 8	-- Templo Blanco 1
Tb2 = 9	-- Templo Blanco 2
Bfb = 10	-- Bufón Blanco
Sb1 = 11	-- Soldado Blanco 1

– Piezas negras rango de 31 a 48

Pn = 31	-- Príncipe Negro
Rn = 32	-- Rey Negro
Ryn = 33	-- Reina Negra
Brn1 = 34	-- Brujo Negro 1
Brn2 = 35	-- Brujo Negro 2
Gn1 = 36	-- Guerrero Negro 1
Gn2 = 37	-- Guerrero Negro 2
Tn1 = 38	-- Templo Negro 1
Tn2 = 39	-- Templo Negro 2
Bfn = 40	-- Bufón Negro
Sn1 = 41	-- Soldado Negro 1

Sb2 = 12 -- Soldado Blanco 2
 Sb3 = 13 -- Soldado Blanco 3
 Sb4 = 14 -- Soldado Blanco 4
 Sb5 = 15 -- Soldado Blanco 5
 Sb6 = 16 -- Soldado Blanco 6
 Sb7 = 17 -- Soldado Blanco 7
 Sb8 = 18 -- Soldado Blanco 8

Sn2 = 42 -- Soldado Negro 2
 Sn3 = 43 -- Soldado Negro 3
 Sn4 = 44 -- Soldado Negro 4
 Sn5 = 45 -- Soldado Negro 5
 Sn6 = 46 -- Soldado Negro 6
 Sn7 = 47 -- Soldado Negro 7
 Sn8 = 48 -- Soldado Negro 8

-- Piezas Azules rango de 20 a 29

--Piezas Rojas rango de 50 a 59

Bra = 20 -- Brujo Azul
 Ta = 21 -- Templo Azul
 Ga = 22 -- Guerrero Azul
 Bfa = 23 -- Bufón Azul
 Sa1 = 24 -- Soldado Azul 1
 Sa2 = 25 -- Soldado Azul 2
 Sa3 = 26 -- Soldado Azul 3
 Sa4 = 27 -- Soldado Azul 4
 Sa5 = 28 -- Soldado Azul 5
 Sa6 = 29 -- Soldado Azul 6

Brr = 50 -- Brujo Rojo
 Tr = 51 -- Templo Rojo
 Gr = 52 -- Guerrero Rojo
 Bfr = 53 -- Bufón Rojo
 Sr1 = 54 -- Soldado Rojo 1
 Sr2 = 55 -- Soldado Rojo 2
 Sr3 = 56 -- Soldado Rojo 3
 Sr4 = 57 -- Soldado Rojo 4
 Sr5 = 58 -- Soldado Rojo 5
 Sr6 = 59 -- Soldado Rojo 6

Cada pieza tiene un nombre de variable, su ubicación inicial, su tamaño (en pixeles), su ubicación en el área de juego y su valor para ser usada por el sistema, esto se muestra a continuación:

Dimensiones y Ubicación del Tablero y piezas posición inicial

	nomenc.	hex	ancho	alto	locH	locV	sprite
Tablero			752	518	378	260	1
Príncipe Blanco	Pb	L1	18	26	374	446	2
Rey Blanco	Rb	L2	18	27	405	428	3
Reina Blanca	Ryb	K1	16	29	343	428	4
Brujo Blanco 1	Brb1	K2	21	24	374	412	5
Brujo Blanco 2	Brb2	J3	21	24	374	378	6
Guerrero Blanco 1	Gb1	J2	14	31	343	394	7
Guerrero Blanco 2	Gb2	L3	14	31	436	412	8
Templo Blanco 1	Tb1	K3	19	28	405	394	9
Templo Blanco 2	Tb2	J1	19	28	312	411	10
Bufón Blanco	Bfb	I4	21	25	374	344	11
Soldado Blanco 1	Sb1	I1	17	27	281	394	12
Soldado Blanco 2	Sb2	I2	17	27	312	378	13
Soldado Blanco 3	Sb3	I3	17	27	343	360	14
Soldado Blanco 4	Sb4	H3	17	27	312	344	15
Soldado Blanco 5	Sb5	J4	17	27	405	360	16
Soldado Blanco 6	Sb6	K4	17	27	436	378	17
Soldado Blanco 7	Sb7	L4	17	27	467	394	18
Soldado Blanco 8	Sb8	J5	17	27	436	344	19

Brujo Azul	Bra1	L12	21	24	715	258	20
Templo Azul	Ta	L11	19	28	684	276	21
Guerrero Azul	Ga2	K12	14	31	684	242	22
Bufón Azul	Bfa	K11	21	25	653	258	23
Soldado Azul 1	Sa1	L10	17	27	653	292	24
Soldado Azul 2	Sa2	L9	17	27	622	310	25
Soldado Azul 3	Sa3	K10	17	27	622	276	26
Soldado Azul 4	Sa4	J11	17	27	622	242	27
Soldado Azul 5	Sa5	I12	17	27	622	208	28
Soldado Azul 6	Sa6	J12	17	27	653	224	29
Príncipe Negro	Pn	A12	18	26	374	72	30
Rey Negro	Rn	A11	18	27	343	88	31
Reina Negra	Ryn	B12	16	29	405	88	32
Brujo Negro 1	Brn1	B11	21	24	374	106	33
Brujo Negro 2	Brn2	C10	21	24	374	140	34
Guerrero Negro 1	Gn1	C11	14	31	405	122	35
Guerrero Negro 2	Gn2	A10	14	31	312	106	36
Templo Negro 1	Tn1	B10	19	28	343	122	37
Templo Negro 2	Tn2	C12	19	28	436	106	38
Bufón Negro	Bfn	D9	21	25	374	174	39
Soldado Negro 1	Sn1	D12	17	27	467	122	40
Soldado Negro 2	Sn2	D11	17	27	436	140	41
Soldado Negro 3	Sn3	D10	17	27	405	156	42
Soldado Negro 4	Sn4	E10	17	27	436	174	43
Soldado Negro 5	Sn5	C9	17	27	343	156	44
Soldado Negro 6	Sn6	B9	17	27	312	140	45
Soldado Negro 7	Sn7	A9	17	27	281	122	46
Soldado Negro 8	Sn8	C8	17	27	312	174	47
Brujo Rojo	Brr	A1	21	24	33	258	48
Templo Rojo	Tr	A2	19	28	64	242	49
Guerrero Rojo	Gr	B1	14	31	64	276	50
Bufón Rojo	Bfr	B2	21	25	95	258	51
Soldado Rojo 1	Sr1	A3	17	27	95	224	52
Soldado Rojo 2	Sr2	A4	17	27	126	208	53
Soldado Rojo 3	Sr3	B3	17	27	126	242	54
Soldado Rojo 4	Sr4	C2	17	27	126	276	55
Soldado Rojo 5	Sr5	D1	17	27	126	310	56
Soldado Rojo 6	Sr6	C1	17	27	95	292	57

Para realizar la acción de cuándo se elimina una pieza, había 2 formas de hacerlo: una era que cuando se eliminará alguna pieza ésta cambiará de posición afuera del tablero, y cambiar todos sus atributos, y la otra que cuando se eliminará esta pieza, se hiciera invisible y al instante otra que ya estaba afuera invisible se hiciera visible, siendo esta última opción la que utilicé, la siguiente tabla muestra los valores de las piezas eliminadas:

Nombre	Sprite
Príncipe Blanco	86
Rey Blanco	87
Reina Blanca	88
Brujo Blanco 1	89
Brujo Blanco 2	90
Guerrero Blanco 1	91
Guerrero Blanco 2	92
Templo Blanco 1	93
Templo Blanco 2	94
Bufón Blanco	95
Soldado Blanco 1	96
Soldado Blanco 2	97
Soldado Blanco 3	98
Soldado Blanco 4	99
Soldado Blanco 5	100
Soldado Blanco 6	101
Soldado Blanco 7	102
Soldado Blanco 8	103
Brujo Azul	104
Templo Azul	105
Guerrero Azul	106
Bufón Azul	107
Soldado Azul 1	108
Soldado Azul 2	109
Soldado Azul 3	110
Soldado Azul 4	111
Soldado Azul 5	112
Soldado Azul 6	113
Príncipe Negro	68
Rey Negro	69
Reina Negra	70
Brujo Negro 1	71
Brujo Negro 2	72
Guerrero Negro 1	73
Guerrero Negro 2	74
Templo Negro 1	75
Templo Negro 2	76
Bufón Negro	77
Soldado Negro 1	78
Soldado Negro 2	79
Soldado Negro 3	80
Soldado Negro 4	81
Soldado Negro 5	82
Soldado Negro 6	83
Soldado Negro 7	84
Soldado Negro 8	85
Brujo Rojo	58
Templo Rojo	59
Guerrero Rojo	60

Bufón Rojo	61
Soldado Rojo 1	62
Soldado Rojo 2	63
Soldado Rojo 3	64
Soldado Rojo 4	65
Soldado Rojo 5	66
Soldado Rojo 6	67

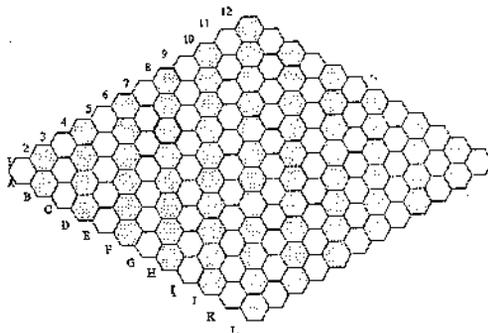
4.2 Los movimientos y la eliminación

Ahora explicaré cómo el sistema realiza los movimientos, para hacer esto, primero tenemos que contestar las siguientes preguntas:

1. ¿Cómo sé que el puntero está sobre la pieza?
2. ¿Cómo sabe el sistema que la quiero tomar?
3. ¿Cuando la tomo cómo sabe el sistema que la solté?
4. ¿Cuando la suelto cómo sé si se hizo dentro del tablero?
5. ¿Cómo el sistema calcula todas las posibles opciones válidas para cada pieza?
6. ¿Cómo sé si la solté en una posición válida?
7. ¿Cómo sé que en la trayectoria de movimiento está el paso libre?
8. ¿Cómo sé hasta dónde está el límite de la trayectoria por una pieza intermedia?
9. ¿Cómo sé qué pieza hay en el espacio que la solté?
10. ¿Cómo sabe el sistema qué acción tomar dependiendo el la pieza sobre la que caí?
11. ¿Cómo el sistema sabe cuando eliminé a un enemigo, y saca la pieza del juego?
12. ¿Cómo se controlan los turnos de tiro?
13. ¿Cómo sabe el sistema qué hacer si suelto la pieza sobre un aliado o una pieza propia?
14. ¿Cómo el sistema puede calcular los espacios válidos después de los movimientos de las piezas?

Primero ubiquemos el área sobre la que vamos a trabajar, o sea el tablero. Supongamos que vamos a ubicarnos en el espacio con el contorno en rojo, el C6, con lo cual en base a las tablas anteriores sabemos lo siguiente:

Espacio → C6
coordenadas (pixeles) → (231,192,268,223)
Número de espacio en lista → 30
Coordenadas A y B → (10,6)
Valor de gráfico → 0

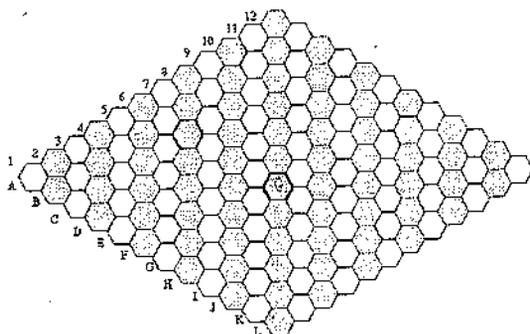


Ahora agreguemos una pieza, el príncipe blanco ubicado en el hex. con contorno azul

Ahora el sistema debe de tomar en cuenta:

Espacio → C6
Coordenadas (pixeles) → (231,192,268,223)
Número de espacio en lista → 30
Coordenadas A y B → (10,6)
Valor de gráfico → 0

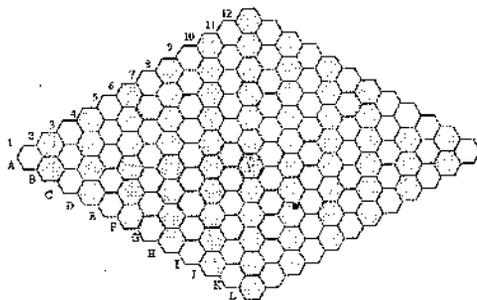
Espacio → G6
Coordenadas → (355,260,392,291)
Número de espacio en lista → 78
Coordenadas A y B → (6,6)
Valor de Pieza → 1
Coordenadas de la pieza (374,276)
Estado de la pieza (movible, visible)
Estado de la pieza eliminada (no-movible, no-visible)



Hasta el momento sólo hemos definido dos espacios y una pieza, pero ahora es cuando empieza lo complicado, ya que cada pieza tiene sus reglas de movimiento por lo que algunas se moverán en una dirección, otras en otra, unas eliminarán piezas de una manera, otras de otra, etc, etc.

El príncipe se mueve en dirección horizontal, vertical, y diagonal, tantos espacios como quiera, siempre y cuando en la trayectoria de movimiento no haya ninguna pieza "estorbando" dicha trayectoria, por lo que en el ejemplo los espacios válidos para el príncipe en esa posición serían:

A6, B6, C6, D6, E6, F6, H6, I6, J6, K6, L6
G1, G2, G3, G4, G5, G7, G8, G9, G10, G11, G12
A12, B11, C10, D9, E8, F7, H5, I4, J3, K2, L1
B1, C2, D3, E4, F5, H7, I8, J9, K10, L11



Dando un total de 44 espacios posibles para el movimiento, el sistema debe ser capaz de saber dónde sí se puede colocar y dónde no, dónde elimina o dónde regresa la pieza a su posición original por un mal tiro.

Si tomamos en cuenta las posibles opciones de cada pieza esto sería como se indica:

- 2 príncipes = 88
- 2 reyes = 12
- 2 reinas = 36
- 6 guerreros = 132
- 6 templos = 156
- 6 brujos = 72
- 4 bufones = 72
- 28 soldados = 168

Dando como total 736 opciones de espacio que el sistema debe tener bien definidas para que al momento de jugar no se produzcan errores, el código fue hecho de manera explícita, por lo que es de un gran tamaño.

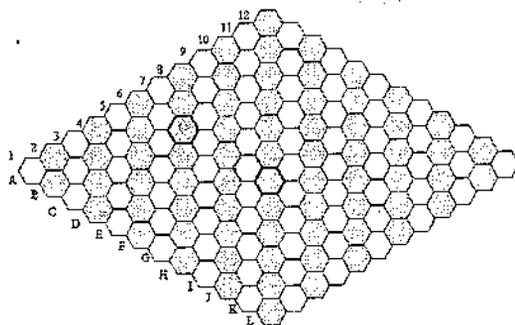
Volviendo al ejemplo, vamos a suponer que la pieza se quiere mover al espacio rojo, el sistema deberá de evaluar al momento de soltar la pieza:

1. Si el espacio está dentro del rango de movimiento válido para esa pieza
2. Si la trayectoria de desplazamiento está libre (D6, E6, F6)
3. Si no lo está, deberá regresar la pieza a su posición inicial
4. Si el espacio es válido, evaluar si está libre u ocupado
5. Si está libre, colocar la pieza e indicar que el espacio anteriormente ocupado, ahora está libre.
6. Si está ocupado por enemigos, eliminarlos e indicar que el espacio anteriormente ocupado ahora está libre.
7. Si está ocupado por aliado o compañero regresar a la posición original.

Una vez que las validaciones se han realizado, el sistema realiza la acción correspondiente, que en este caso será mover la pieza a un lugar vacío, actualizará la lista para que ahora el espacio C6 esté ocupado por la pieza y el espacio G6 ahora quede libre.

La tabla quedaría:

Espacio → C6
 Coordenadas (píxeles) → (231,192,268,223)
 Número de espacio en lista → 30
 Coordenadas A y B → (10,6)
 Valor de gráfico → 2
 Valor de Pieza → 1
 Coordenadas de la pieza (250,208)
 Estado de la pieza (movible, visible)
 Estado de la pieza eliminada (no-movible, no-visible)

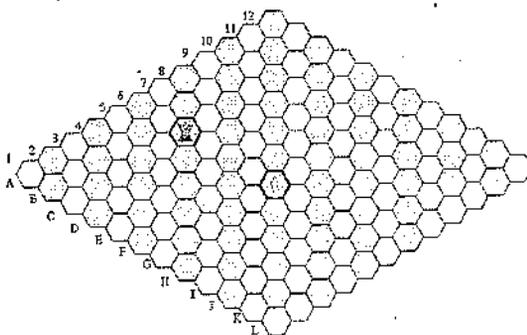


Espacio → G6
 Coordenadas → (355,260,392,291)
 Número de espacio en lista → 78
 Coordenadas A y B → (6,6)
 Valor de gráfico → 0

Para la eliminación sería del mismo modo; pero ahora hay que agregar las variables de la otra pieza.

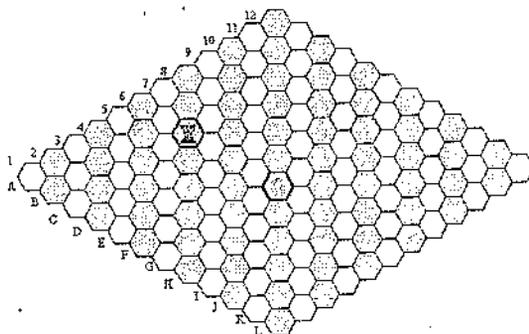
Espacio → C6
 Coordenadas (píxeles) → (231,192,268,223)
 Número de espacio en lista → 30
 Coordenadas A y B → (10,6)
 Valor de gráfico → 50
 Valor de Pieza → 48
 Coordenadas de la pieza (250,208)
 Estado de la pieza (movible, visible)
 Estado de la pieza eliminada (no-movible, no-visible)

Espacio → G6
 Coordenadas → (355,260,392,291)
 Número de espacio en lista → 78
 Coordenadas A y B → (6,6)
 Valor de gráfico → 2
 Valor de Pieza → 1
 Coordenadas de la pieza (374,276)
 Estado de la pieza (movible, visible)
 Estado de la pieza eliminada (no-movible, no-visible)



Una vez que seleccione al príncipe, el sistema está en espera de donde lo ubique, para realizar todas la evaluaciones, y realizar la acción, como en este caso todo indica que sí puedo eliminar al brujo rojo, la tabla se presentaría de la siguiente manera:

Espacio → C6
Coordenadas (píxeles) → (231,192,268,223)
Número de espacio en lista → 30
Coordenadas A y B → (10,6)
Valor de gráfico → 2
Valor de Pieza → 1
Coordenadas de la pieza (250,208)
Estado de la pieza (movible, visible)
Estado de la pieza eliminada (no-movible, no-visible)



Espacio → G6
Coordenadas → (355,260,392,291)
Número de espacio en lista → 78
Coordenadas A y B → (6,6)
Valor de gráfico → 0
Valor de Pieza → 0

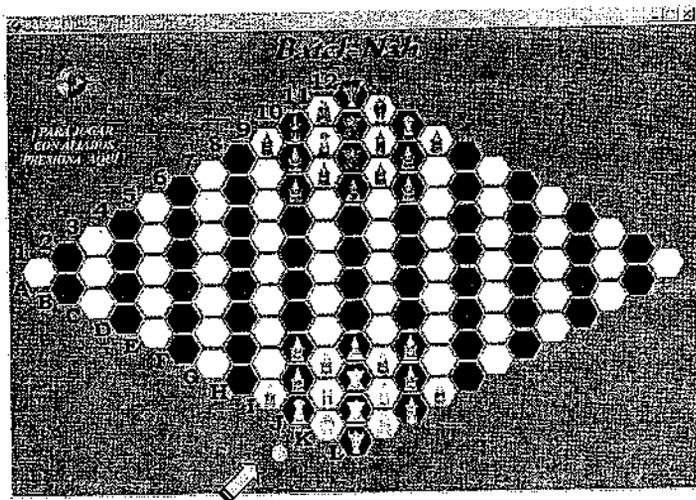
El sistema cambiará las propiedades de la pieza que fue eliminada y la que esté bien, aparecerá en su lugar.

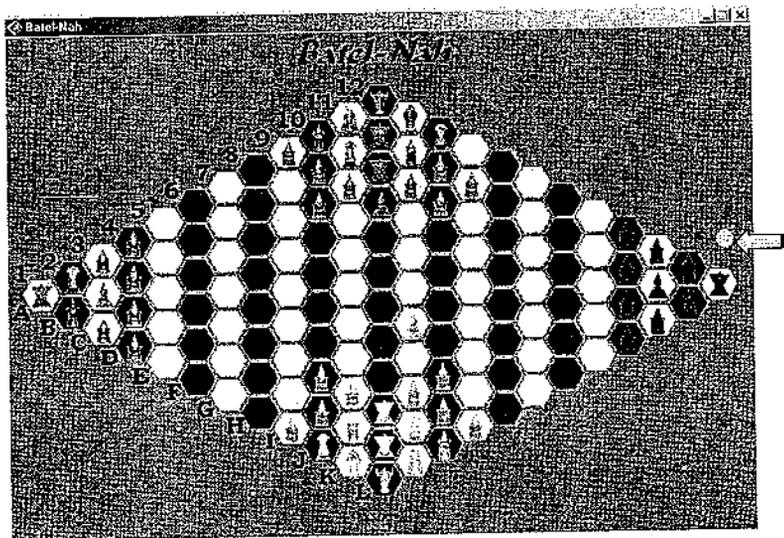
Estado de la pieza (no-movible, no-visible)
Estado de la pieza eliminada (no-movible, visible).

4.3 Turnos y Aliados

Como en todo juego debe de existir un turno de tiro para cada jugador, aquí el turno está indicado por un círculo de color amarillo, dicho círculo ira cambiando al siguiente jugador en turno. Cada jugador tiene un turno, ya sea en el juego de Reino vs. Reino, o con los aliados, por lo que el sistema debe de ser capaz de identificar cuándo y de quién es el turno, para lo cual se tuvo que crear un código especial para controlarlos.

Cuando son 2 jugadores, el turno deberá de ir de un Reino al otro con cada tiro, pero cuando se agregan los aliados, se deberán incluir sus turnos, por lo que la secuencia será: Blancas, Negras, Azules, Rojas. Una vez que se han unido los aliados, éstos no se pueden quitar.





Al inicio del juego sólo aparecen los dos reinos principales, en este momento se puede empezar a jugar sólo con estos dos oponentes, pero, tanto al inicio del juego como ya avanzado éste, es posible que entren a jugar los aliados, para esto se creó un botón, que al darle clic, aparecerán los dos aliados de cada reino



Pero ¿qué pasa si el jugador que está en turno no tira?, ¿qué pasa si durante el juego se elimina totalmente a alguno de los aliados?

Para solucionar estas situaciones se tuvo que crear un código para poner un reloj, el cual dará 5 min. de tiempo a cada jugador para tirar, en caso de que no lo haga se cambiará el turno al jugador siguiente, perdiendo así su turno, esto es importante ya que en ninguna guerra se puede perder el tiempo.

Ahora, tomando en cuenta una situación en donde nos eliminen a todos nuestros aliados, obviamente la desventaja será muy grande, ya que aparte de ser menos elementos en la batalla, ahora recibiremos dos ataques mientras que sólo podremos hacer uno, ya que al no haber elementos para tirar, el turno se pasaría al jugador siguiente, por lo que es de suma importancia que cuidemos a nuestros aliados, ya que son parte indispensable en el desarrollo del juego. Lo que servirá para fomentar el trabajo en equipo.

4.4 Ejemplo de programación en Director

Director es un programa creador de presentaciones gráficas, animaciones, programas interactivos, etc. Su lenguaje de programación es Lingo, el cual es muy parecido a otros lenguajes de uso común como Lenguaje C.

Una de las virtudes de Director es que maneja ambientes gráficos de una manera muy amigable, por lo que el programador coloca un objeto (foto, gráfico, texto, etc.) y el programa automáticamente le asigna diversas características y propiedades, las cuales resultan de gran utilidad.

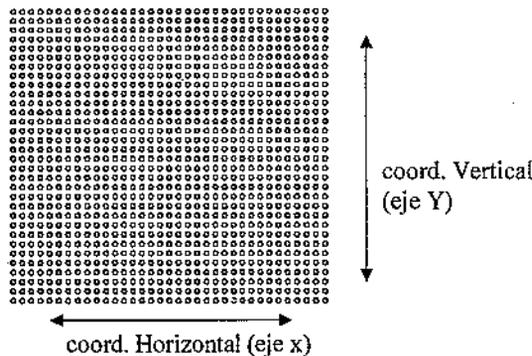
Una de estas propiedades es de que por medio de algunos comandos, el sistema puede saber cuándo un objeto toca a otro, cuándo el cursor pasa por encima de éste, etc, etc.

Por esto creí que resultaría fácil crear un objeto correspondiente a cada espacio (hexágono), y un objeto para cada pieza, y lo hubiera sido de no ser porque Director sólo acepta un número limitado de objetos, aprox. 150, por lo que tomando en cuenta que sumando el número de espacios, más las piezas en juego, más las piezas eliminadas, más otros gráficos necesitaría más de 200 objetos, para lo cual Director es insuficiente.

Lo anterior se convirtió en un gran problema, ya que tuve que buscar alternativas para ubicar tanto a las piezas como a los espacios, afortunadamente se pueden crear áreas virtuales invisibles las cuales se comportan como objetos, sin llegar a serlo.

A continuación mostrare cómo se crean estas áreas, y cómo se manejaron para que el juego funcionara.

Primero imaginemos que tenemos un espacio vacío que está delimitado por una cantidad de puntos tanto horizontal como verticalmente



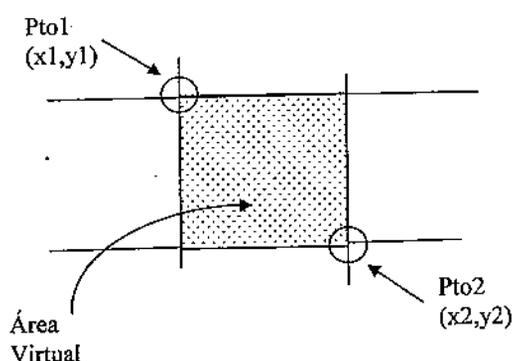
En base a esto, cada punto tiene un valor en X y Y, de esta forma, se puede ubicar perfectamente un punto en este espacio.

Para poder crear las áreas de los espacios se utilizaron los siguientes comandos:

`point(cordx,cordy)` éste devuelve un punto específico en el espacio

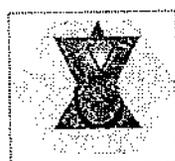
después para crear el área necesitamos 2 puntos, el superior izquierdo y el inferior derecho,

`set area=rect(Pto1, Pto2)` así es como se genera un área cuadrada, ésta área es invisible.



De esta forma generé los 144 espacios del tablero, ubicando dentro de cada área virtual un hexágono.

Una vez teniendo todas éstas áreas ubicadas en base a su posición real en píxeles, continué con las piezas, que, como son objetos, el programa automáticamente indica su posición en píxeles dentro del espacio, por lo que puede modificársele para colocarla en cualquier lugar.



Para Director esto se vería gráficamente como la figura anterior, para el usuario sólo se vería el hexágono y la pieza, pero para Lingo es completamente diferente, a continuación muestro un poco del código para ilustrarlo.

Primero se coloca una imagen con el tablero dibujado como fondo, para ubicar las posiciones de todos los hexágonos.

Después se crea cada área virtual con el siguiente código.

```
ptoincHBrn1 = 374 - Indico la coordenada inicial de la posición de la pieza en eje horizontal  
ptoincVBrn1 = 106 - Indico la coordenada inicial de la posición de la pieza en eje vertical
```

```
ptoiniH1Brn1 = 355- Indico la coordenada inicial de la posición de punto1 en eje horizontal  
ptoiniV1Brn1 = 90 - Indico la coordenada inicial de la posición de punto1 en eje vertical  
ptoiniH2Brn1 = 392- Indico la coordenada inicial de la posición de punto2 en eje horizontal  
ptoiniV2Brn1 = 121 - Indico la coordenada inicial de la posición de punto2 en eje vertical
```

```
puntoinH = point (ptoiniH1Brn1, ptoiniV1Brn1) -- Genero el punto superior izquierdo  
puntoinV = point (ptoiniH2Brn1, ptoiniV2Brn1) -- Genero el punto inferior derecho
```

```
set areainicBrn1 = rect(puntoinH,puntoinV) -- Genero la área
```

Después en base a incrementos y decrementos creo las diferentes áreas necesarias según sea el caso:

```
desp1H = 31  
desp1V = 85
```

```
xsi1 = ptoiniH1Brn1 - desp1H  
ysi1 = ptoiniV1Brn1 - desp1V  
xsi2 = ptoiniH2Brn1 - desp2H  
ysi2 = ptoiniV2Brn1 - desp2V
```

```
puntosi1 = point(xsi1,ysi1)  
puntosi2 = point(xsi2,ysi2)
```

```
set área1Brn1 = rect (puntosi1,puntosi2)
```

Finalmente con el comando **put the clickOn**, le indicamos al sistema, que considere a la pieza sobre la que se le dio el clic cómo la pieza activa, así quedaría de la siguiente manera:

```
Put the clickOn into piezaBrn1 -- El sistema tomó a la pieza Brujo negro como la activa
```

Ahora que ya están generadas las áreas y las piezas, se pueden usar otros comandos para interactuar entre ellas, validando su intersección o su ubicación en el tablero, por supuesto que para realizar las validaciones necesarias, y las actualizaciones gráficas es necesario mucho más código.

CAPÍTULO 5

Creación de páginas Web (Proyección Futura)

El desarrollo de este capítulo es con fines a futuro, ya que pienso implementarlo en Internet, a continuación hago una propuesta de cómo podría quedar este portal.

Ahora que ya tengo creado el juego para la PC, es necesario incluirlo en una página html o pagina web, así es como comúnmente se les conoce a los portales de Internet, que a grosso modo son unos "visores" donde se colocan las fotos, textos, aplicaciones, ligas, etc. etc., que realmente va a utilizar el usuario.

Para implementar la aplicación completa del juego, y así lograr que el usuario tenga acceso a toda la información, es necesario crear varias páginas, para presentaciones, tutoriales; gráficos, el juego en sí, etc.

Para esto utilicé varios programas así como los gráficos ya creados con anterioridad en capítulos anteriores.

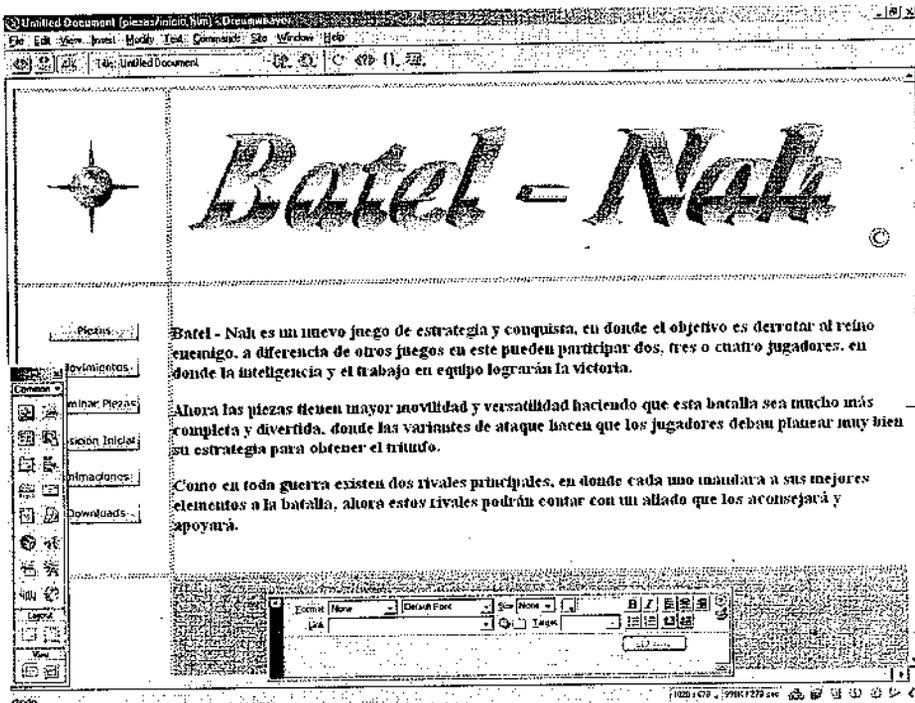
El programa que empleé fue DreamWeaver, que es un programa muy amigable, y permite al usuario diseñar páginas Web con mucha facilidad.

La aplicación final básicamente está formada por dos aspectos, uno, el gráfico, que permite la visualización del juego, la interacción, etc y el otro es el lógico, que corresponde a lo que va a hacer el sistema para realizar todas las operaciones del juego.

5.1 Diseño de la Página Inicial.

En esta página coloque una pequeña introducción, botones de direccionamiento, el logotipo, el nombre, y un gráfico del juego.

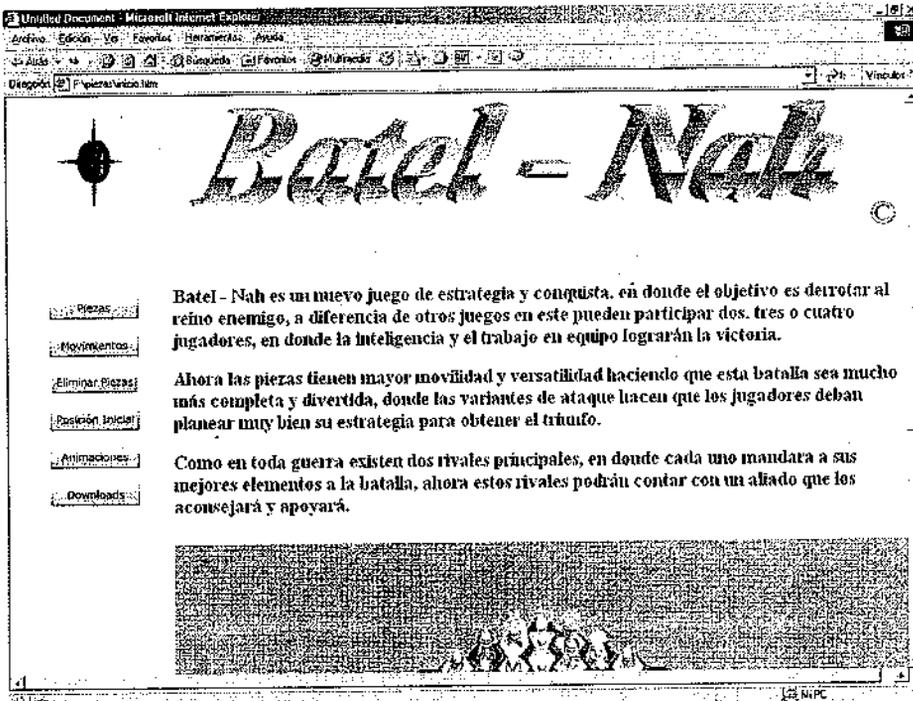
El logotipo es una animación gif, ese tipo de archivos tiene la facultad de ejecutarse automáticamente en una página html, por lo que al acceder el usuario a ésta vera la animación.



Esta es la ventana de Dreamweaver, con la ayuda de los cuadros de herramientas es relativamente sencillo crear cualquier tipo de página Web, es casi como sólo "cortar y pegar" lo que queremos colocar, para insertar links en el cuadro de herramientas y nos despliega las opciones para establecer el path de cada vínculo.

Describiré cada una de las páginas del portal de Internet

Esta es la página inicial de la interfase, a través de ella se accede a las diversas opciones del sistema.



Los botones sirven para direccionar, o invocar un archivo o un procedimiento, en esta primera página coloque 8 botones los cuales son:

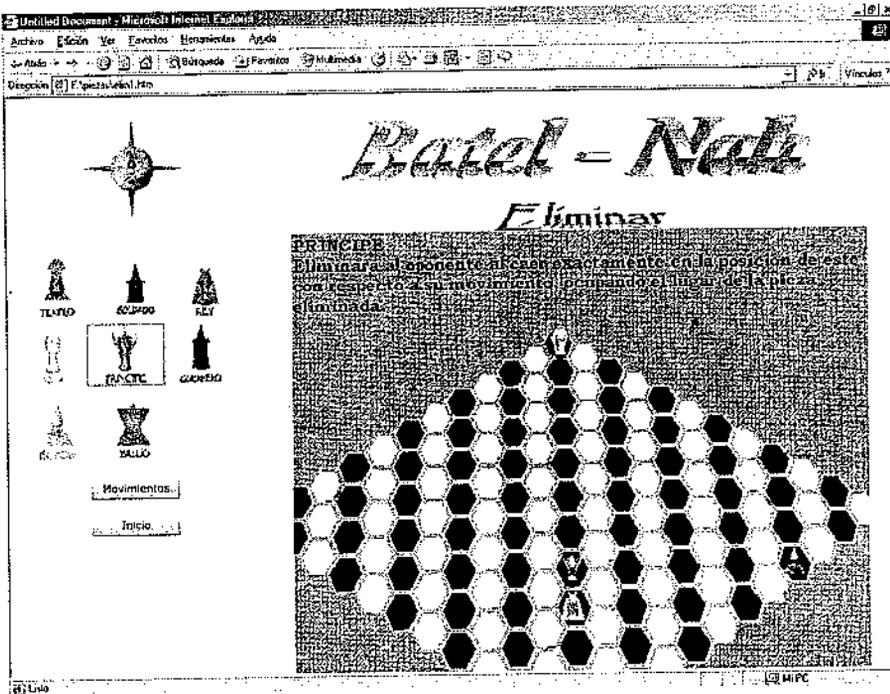
Botón Piezas. Con este botón llamo a la página en donde se muestran los nombres de las figuras, y al pasar el mouse por encima de cada una de ellas se despliega el gráfico de la figura tridimensional, esto con el fin de que los nuevos jugadores se vayan familiarizando con el aspecto de cada figura



Como se puede apreciar dentro de esta página hay más botones que nos direccionan hacia otra página, esta situación es muy común en el diseño de interfaces para Internet, ya que es imposible mostrar toda la información que se desea en un solo portal.

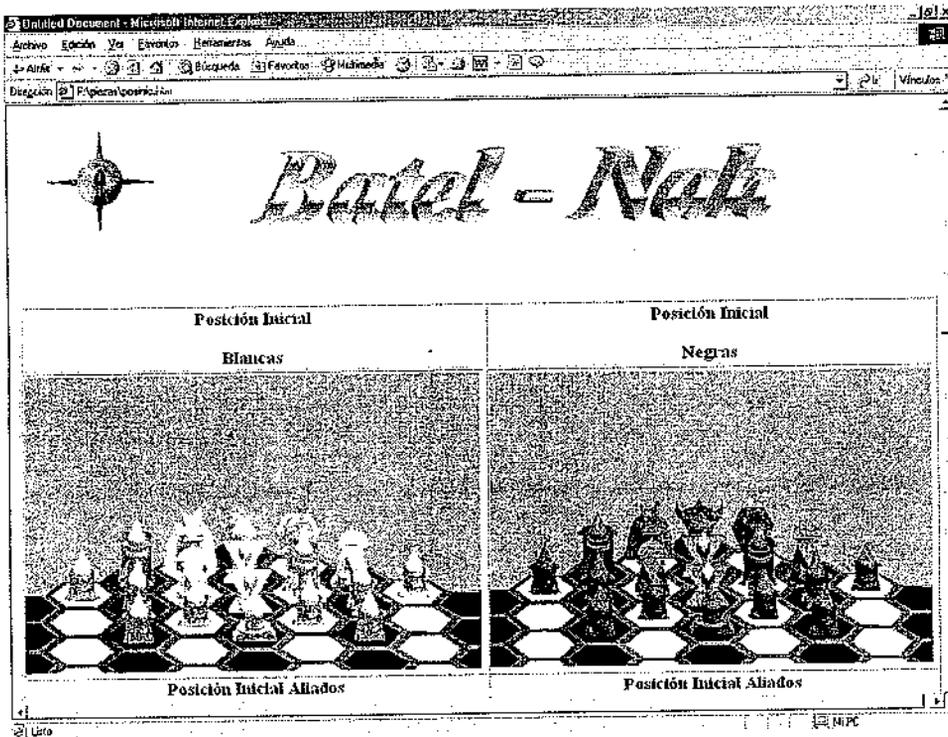
Botón Movimientos. Direcciona hacia la página donde se muestran animaciones de las reglas de movimientos válidos para cada pieza (mas adelante explicaré cómo realicé estas animaciones), esta página será de gran utilidad, ya que aquí los jugadores podrán revisar las veces que sea necesario cómo se mueven las piezas.

Botón eliminar pieza. Al igual que en los movimientos creé unas animaciones para las reglas de eliminación de las piezas.



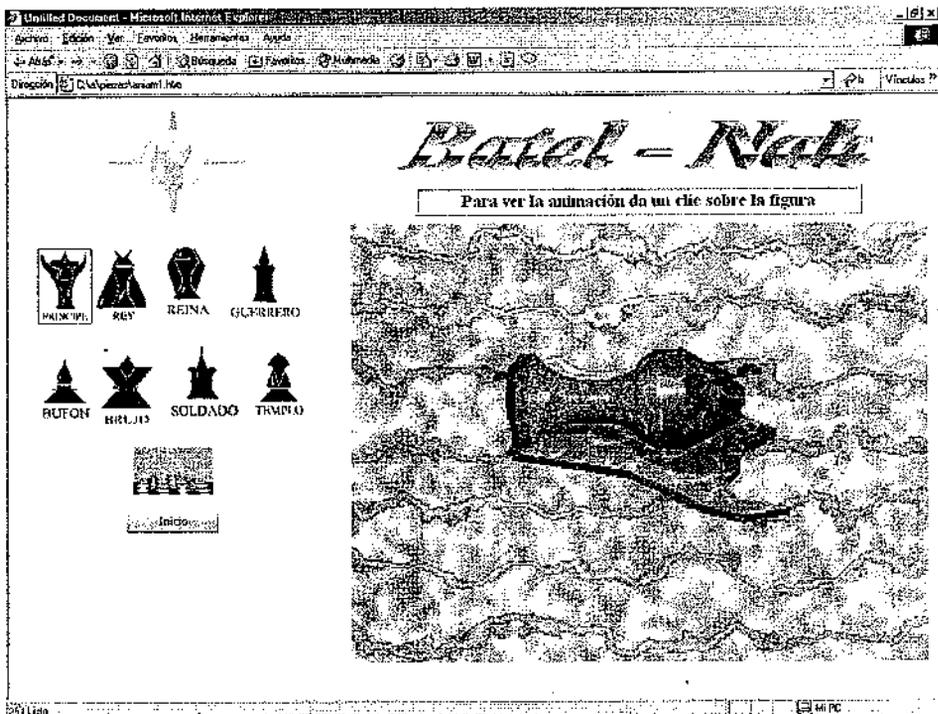
De igual forma que en la página anterior, sólo es necesario dar un clic en cada figura al lado izquierdo de la pantalla para poder ver la animación correspondiente.

Botón posición inicial. Este botón muestra la manera en la que las piezas deben ir colocadas al momento de empezar una partida, esto se refiere a cuando el juego se comercialice de manera física, y haya que acomodarlas manualmente.



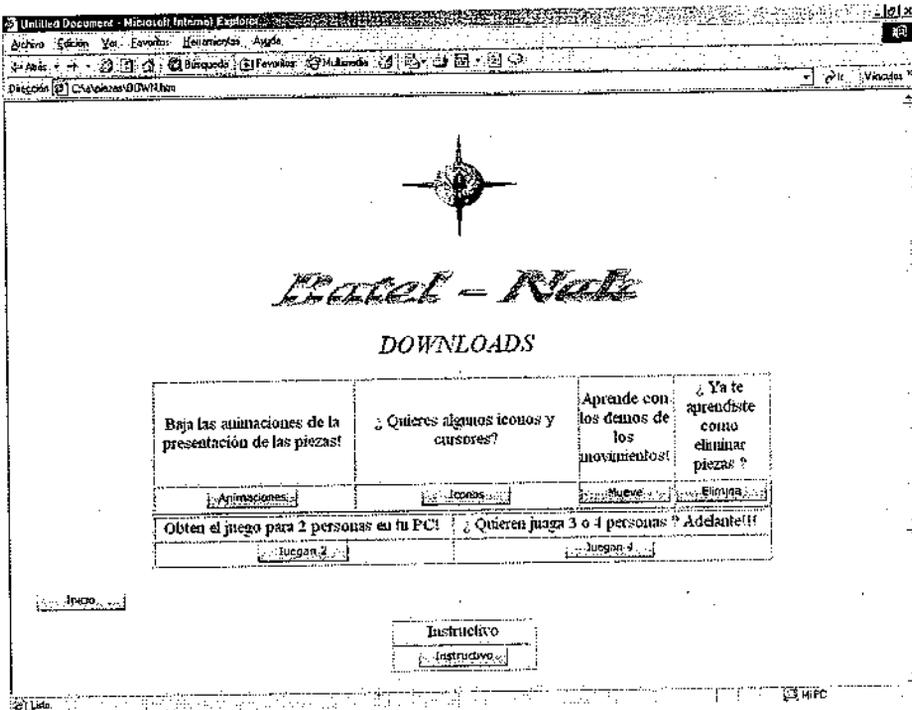
Se muestran las posiciones para los cuatro jugadores, obviamente aplicarán según los jugadores que participen.

Botón animaciones. Para hacer un poco más divertido el aprendizaje del reconocimiento visual de las piezas hice unas animaciones donde éstas se mueven, giran, se transforman, etc.



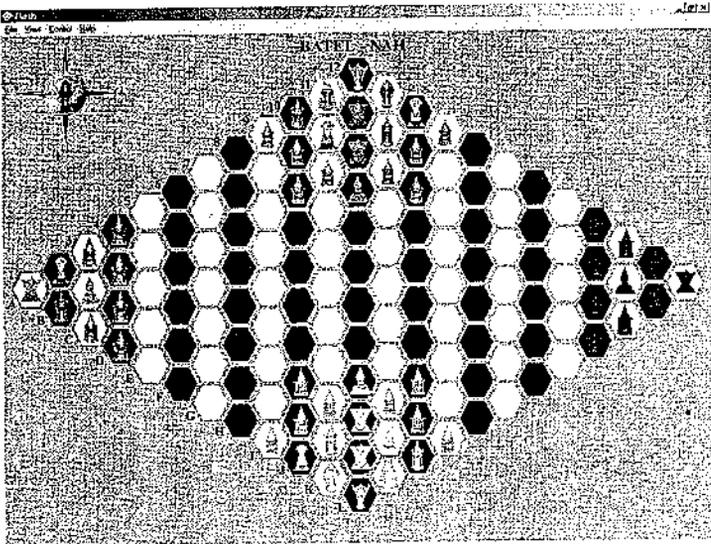
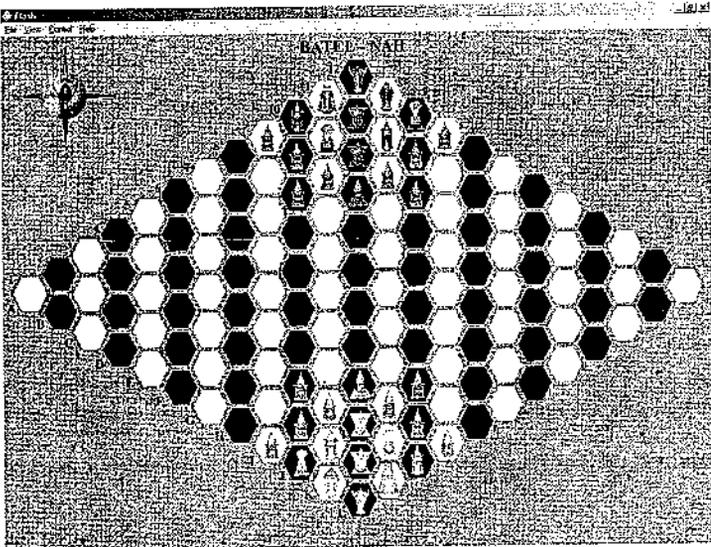
Con éstas animaciones el usuario identifica los nombres con las figuras, para que al momento de jugar sepa cuáles son las piezas y sus movimientos.

Botón. Downloads. En esta página el usuario podrá descargar todo lo antes visto, animaciones, instrucciones, el juego, etc. para así tenerlo en su computadora y no tener que estar necesariamente conectado a Internet para poder practicar el juego.

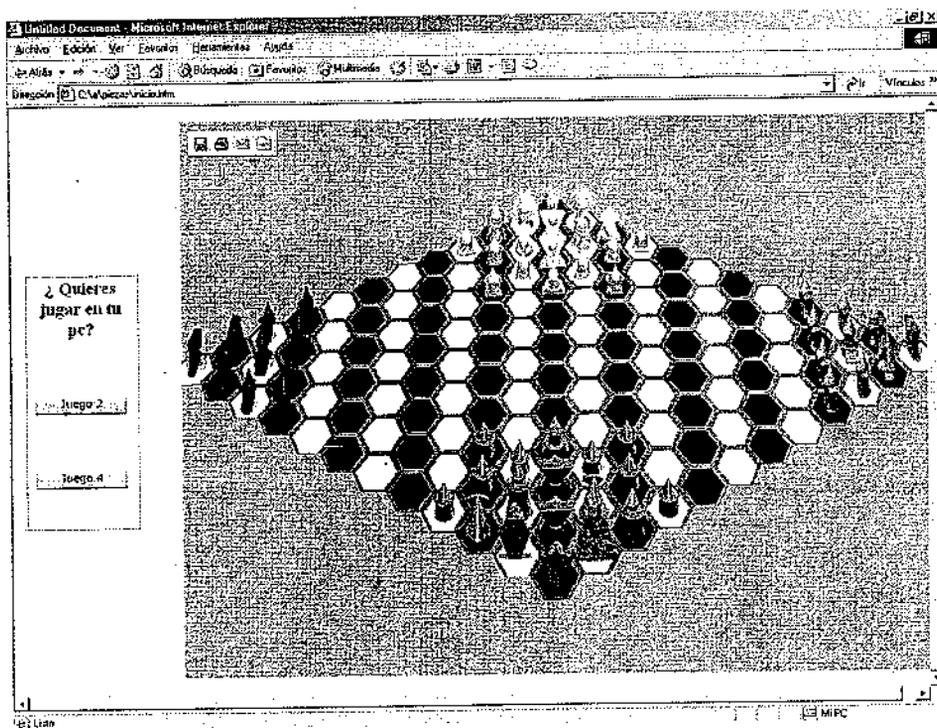


Además también podrá descargar cursores, iconos, protectores de pantalla, así como las animaciones..

Los juegos que el sistema descarga son aplicaciones para que dos personas jueguen en una misma computadora, estas aplicaciones fueron creadas con Director y convertidas a un formato ejecutable.

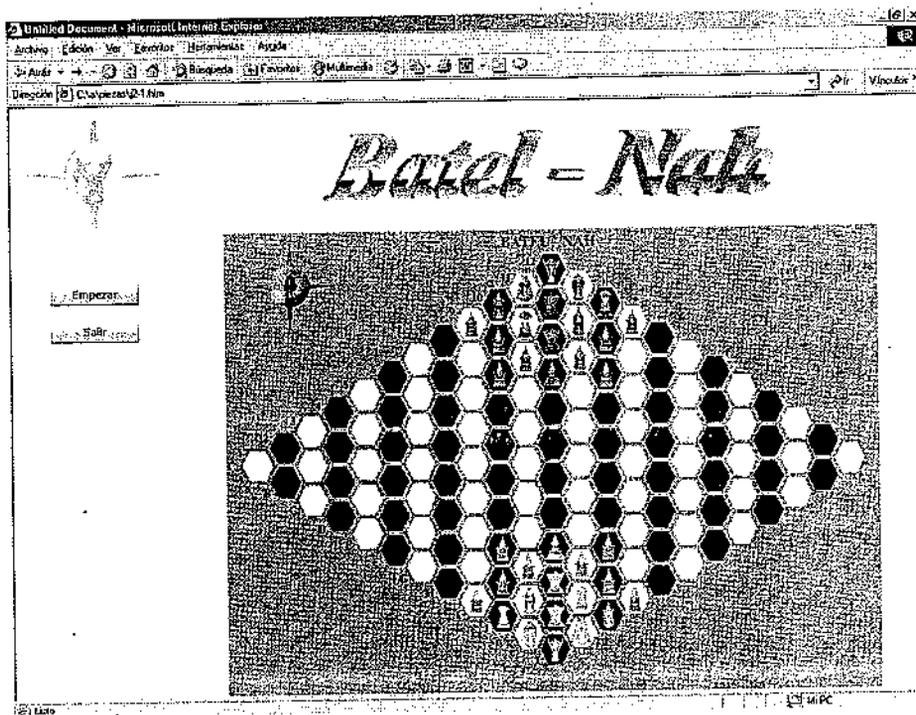


En la pantalla de inicio, en la parte inferior se encuentran 2 botones que dan acceso al juego en sus dos modalidades, estas aplicaciones se ejecutarán sobre otra página web, aunque en la sección de downloads los juegos que se descargan son ejecutables.



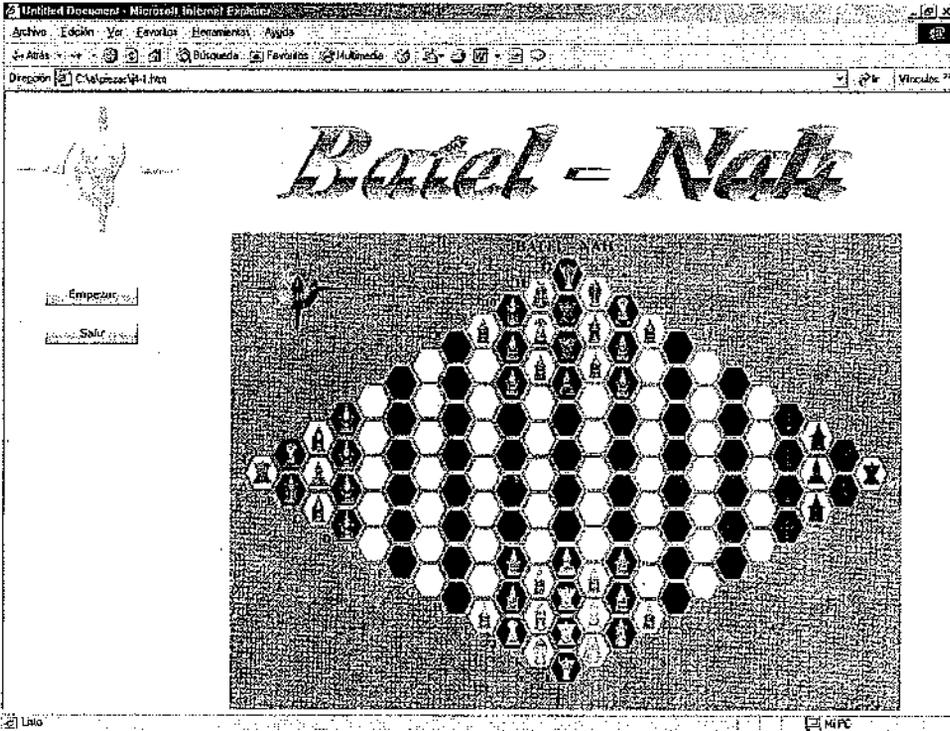
La imagen que se muestra sólo es para dar un mejor aspecto a la página, aunque así es como se vería un tablero completo.

Botón Juego 2. Este botón habilita la página en donde se podrá jugar la versión de 2 jugadores Reino vs. Reino.



En esta página ya se puede interactuar con el juego!

Botón Juego 4. Habilita el juego para 3 o 4 personas, y al igual que la página anterior ya es posible tomar las piezas y empezar a jugar .



Estas páginas son para que dos personas que físicamente están juntas puedan jugar una partida en una PC.

Todas las páginas cuentan con un botón de "Inicio", que al presionarlo nos envía al inicio.

5.2 Creación de animaciones

Internet es una interfase de comunicación gráfica, por lo que es muy importante que sea vistosa o llamativa hacia el usuario, ya que los colores, formas, dibujos, gráficas, juegan un papel esencial dentro del diseño de una página Web.

Debido a esto he creado una serie de animaciones, como se mostró en el capítulo anterior, estas animaciones son en tercera dimensión, son representativas de cada pieza del juego y del tablero, su función es que el usuario se familiarice con ellas.

El programa que utilicé fue el Moray, el cual genera desde las formas básicas de las figuras, hasta movimientos, efectos e iluminación, es un programa especializado, que requiere de ciertos conocimientos para lograr los objetivos deseados.

En Pov-Ray se manejan los objetos que hemos creado, cada objeto tiene varias características, como son: textura, color, forma y tamaño, otras variables son la iluminación y su posición dentro del espacio tridimensional de nuestro ambiente.

Para ilustrar un poco el concepto de ambiente tridimensional, lo explicaré brevemente con algunas gráficas:

Un espacio bidimensional está formado por dos planos, y tridimensional como su nombre lo indica, está formado por tres planos, aplicando esto a las figuras, en la Fig. 1 tenemos un rectángulo, el cual está en un plano bidimensional, ya que sólo tiene alto y ancho, en la figura 2 tenemos el mismo rectángulo, sólo que ahora le agregamos un plano más, que es el de la profundidad y de esta manera obtenemos la imagen tridimensional.

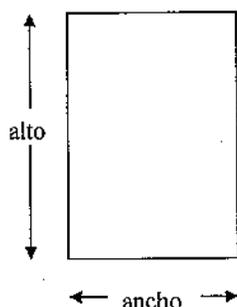


Fig.1

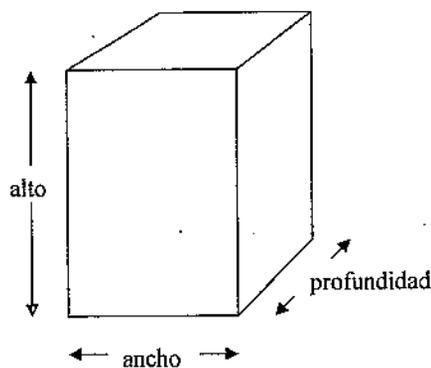


Fig.2

Ahora este objeto tridimensional se tiene que ubicar en un espacio, un lugar, una posición, en la Fig.3 lo vemos ubicado en este espacio, por lo que en este momento ésta posición estará

dada por la distancia que tiene con respecto a cada uno de los planos, quedando así su posición tridimensional como $pos = (dist. \text{ en plano } x, dist. \text{ en plano } y, dist. \text{ en plano } z)$

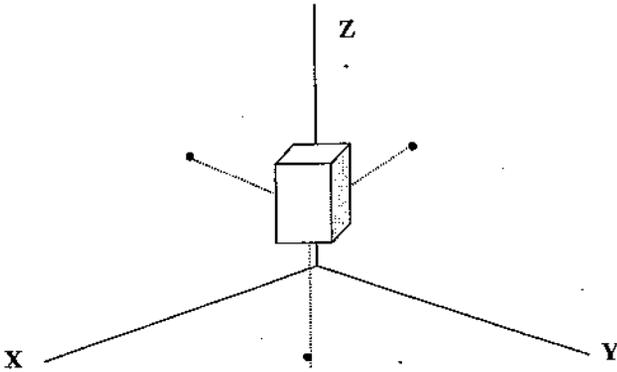


Fig.3

En este momento tenemos un objeto ubicado en un plano tridimensional, pero debemos nombrar a estos planos para que podamos distinguirlos, comúnmente se toman las letras X, Y, Z para nombrar a los ejes de estos planos. El objeto tiene más características distintivas, como lo son textura y color, por lo que en la figura agregaremos estas últimas variantes Fig.4.

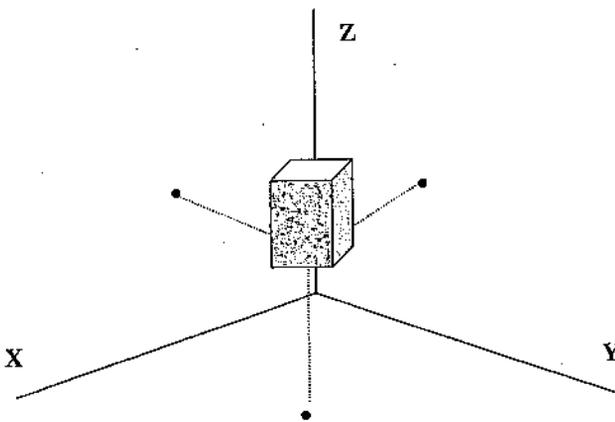


Fig.4

Como este objeto está "flotando" en el espacio que hemos definido, podemos trasladarlo, o rotarlo como queramos Fig.5

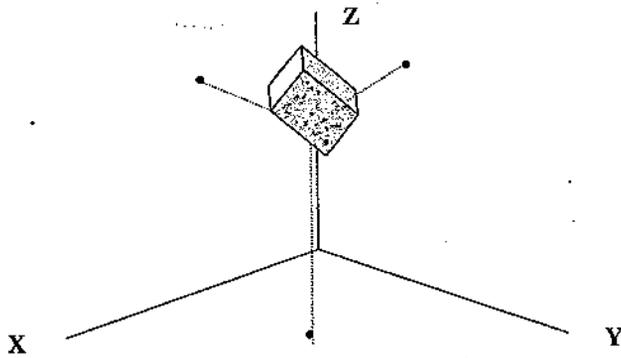


Fig.5

En la figura anterior al objeto lo desplazamos hacia arriba, o sea lo desplazamos sobre el eje Z, y además lo rotamos hacia la derecha o sea hacia el eje X.

De igual forma podemos modificar a la figura, y mantenerla en las mismas coordenadas con respecto al espacio definido Fig. 6.

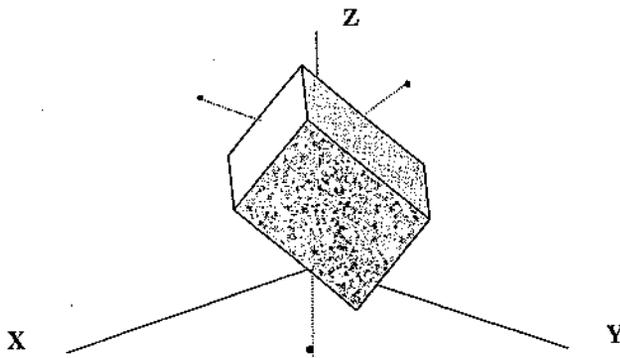


Fig.6

También es posible deformar al objeto modificando sus valores particulares fig.7

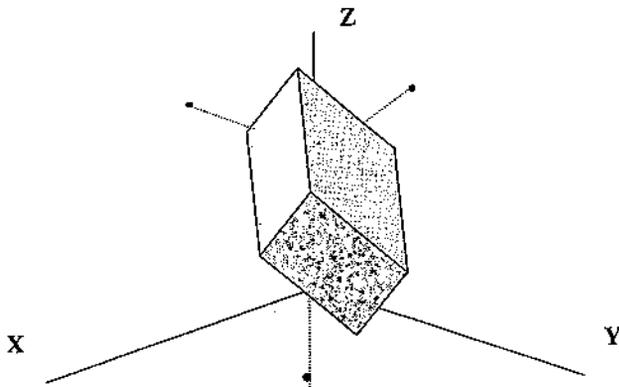


Fig.7

Una vez que entendimos dimensiones de objeto y de espacio tenemos que tomar en cuenta otro factor, la luz, sin ella no tendríamos nada, ya que aunque estuviera ahí, no lo podríamos ver, en la Fig. 8 tenemos una esfera, la cual está alumbrada.

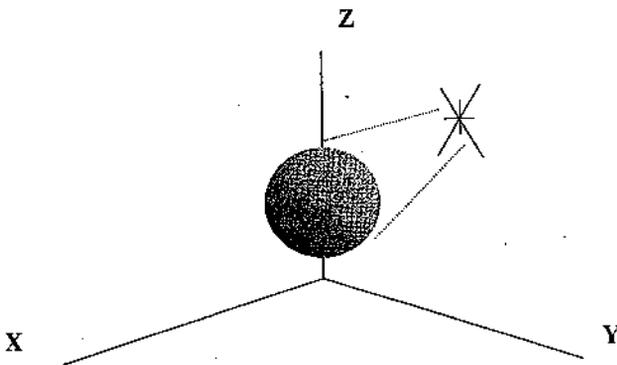
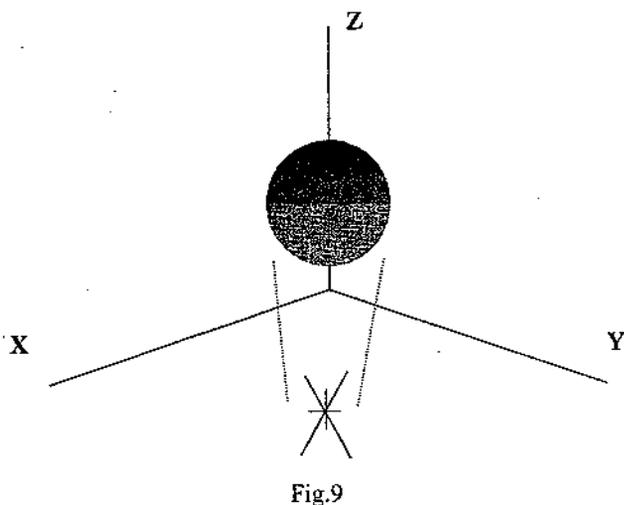
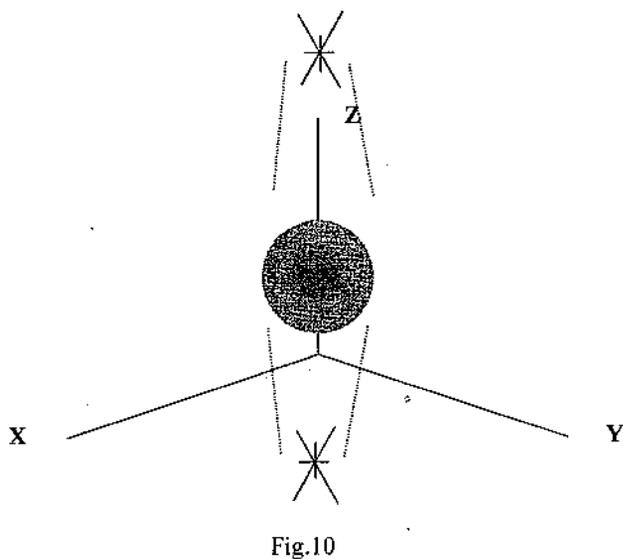


Fig.8

En este caso la luz incide sobre la esfera en la parte superior derecha, por lo que en el lado opuesto se ve más oscuro, debido a la sombra, la luz es también un objeto al que podemos modificar, su tamaño, intensidad y dirección. Fig.9



Si queremos que toda la figura sea visible de igual forma, podemos agregar más luces fig10.



Recordemos que estamos en un espacio tridimensional, por esto en la figura vemos que el centro de la esfera está un poco obscurecido, y esto es debido a que pusimos luz en el eje Z arriba y abajo del objeto, por lo que si quisiéramos que estuviera totalmente iluminada tendríamos que colocar una luz "enfrente", o sobre el eje X para que se iluminara toda.

El último elemento que debemos tomar en cuenta, es la cámara, o el punto desde donde estamos colocados para ver el objeto, en la Fig.11 vemos los objetos de frente, vemos los

vértices de los planos, la esfera del lado izquierdo y el cilindro del lado derecho. Si movemos la cámara lo que veremos será diferente, aunque no hayamos movido los objetos.

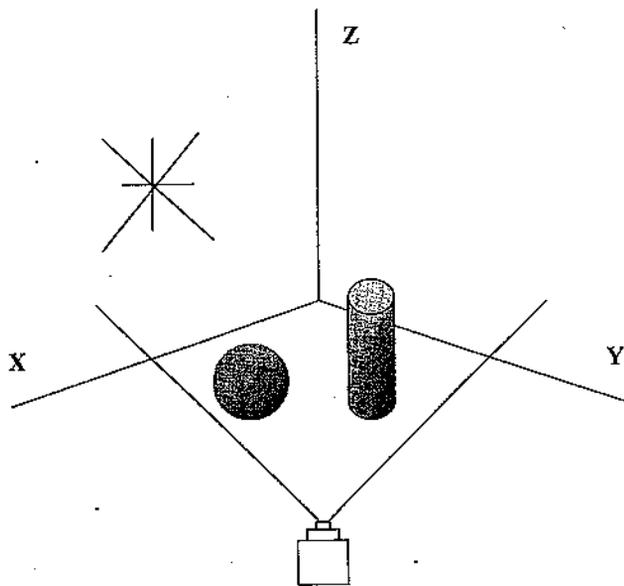


Fig.11

Moveremos la cámara hacia la izquierda Fig. 12 y Fig. 13

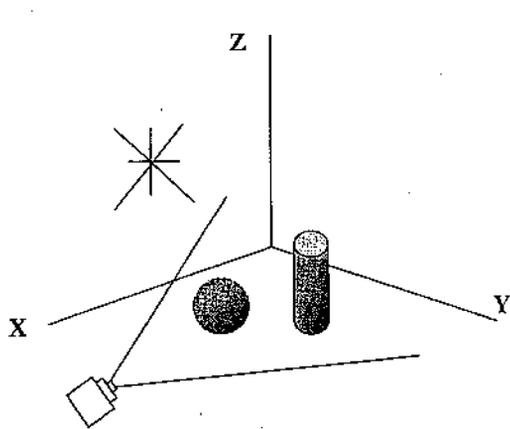


Fig.12

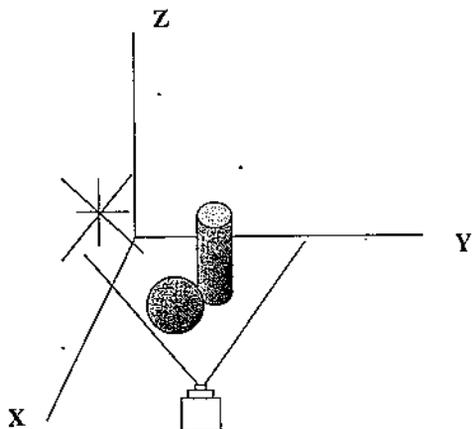
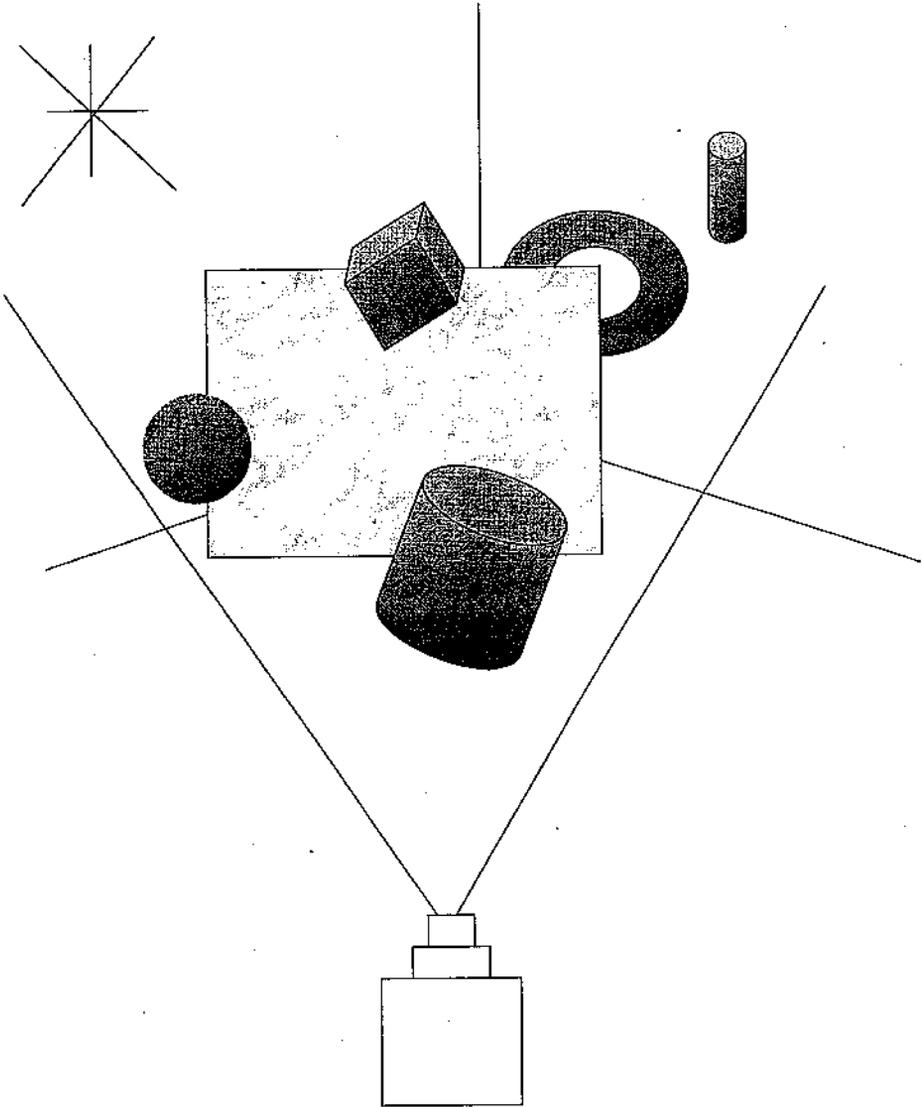


Fig.13

En la figura 13 vemos como realmente se vería desde la cámara una vez que la hemos movido, y como la podemos mover hacia cualquier dirección, si la hacemos hacia atrás los objetos se verían más pequeños.

Ahora que hemos visto las modificaciones que podemos realizar tanto en la forma del objeto como en su posición en el espacio, podemos asumir que esto es aplicable a cualquier cosa, objetos, planos y luces.

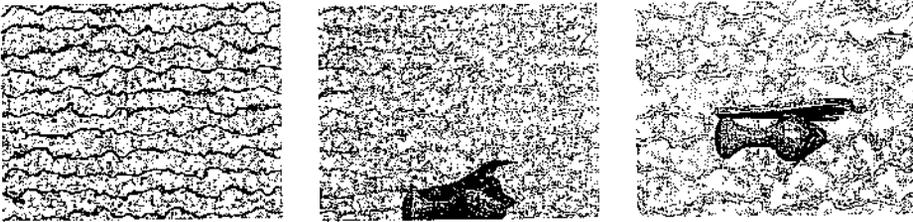


Animación del Príncipe.

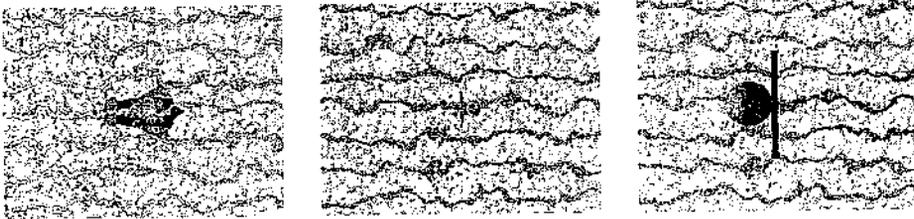
Sería muy complicado explicar una a una cada animación que hice de las piezas, por lo que únicamente explicaré la animación del príncipe.

La idea principal es que el Príncipe es tan versátil en sus movimientos que es como si volara, por lo que la animación refleja esto.

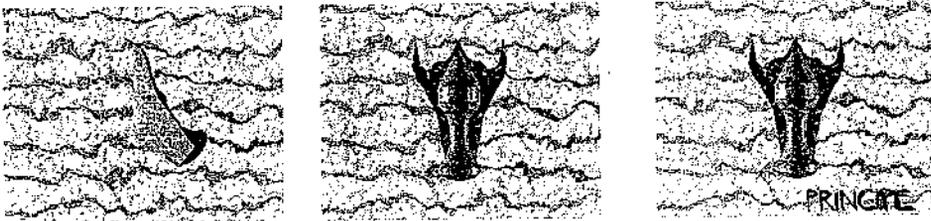
Inicia con el movimiento que asemeja un vuelo horizontal, lo que logré poniendo un fondo de cielo con nubes, y encima un fondo que asemeja rayos de luz, después hice que la cámara se desplazara horizontalmente hacia el lado derecho, posteriormente incorporé al príncipe de abajo hacia arriba girando sobre su eje.



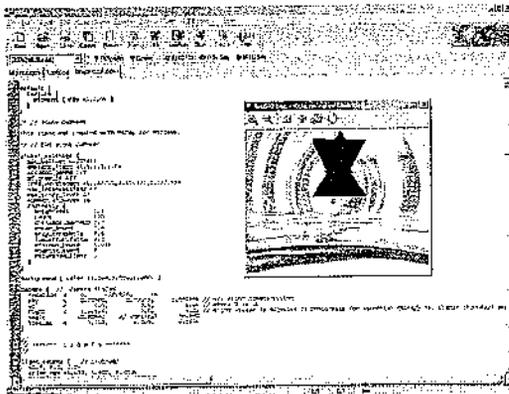
Luego hace un movimiento como si volara hacia abajo y hacia arriba, esto lo hice desplazando la figura hacia el eje que apunta en dirección perpendicular con respecto al plano



Finalmente hace un giro y queda de frente posesionándose en el centro de la animación, aparece el nombre de la pieza con un desplazamiento horizontal de izquierda a derecha. Estos movimientos se logran modificando de manera conjunta las 3 coordenadas de la pieza.



De igual forma que en el capítulo 2, primero se crea la imagen que vamos a trabajar con Moray, al realizar el "render" de la figura que es cuando visualizamos la piezas en su forma real, se abre automáticamente Pov Ray, que es el programa que realmente crea esta imagen.



El modo en la que el programa genera las imágenes para la animación es por medio de un contador o reloj, el cual realiza cambios en base a los datos que nosotros le suministramos. Estos datos se colocan directamente sobre el archivo fuente de la imagen a modificar, quedando de la siguiente manera:

Primero creamos un objeto conjuntando las diversas partes que forman la pieza, esto se hace con una unión, "# declare príncipe = unión { conjunto de elementos que forman la figura}", así , ya tenemos un solo elemento.

Dependiendo de qué es lo que queramos obtener como animación, son las variables que vamos a modificar, esto se hace con las siguientes instrucciones:

- Rotate Rota la figura
- Translate Transporta la figura a otro punto
- Scale Cambia el tamaño de la figura

Como se ha mencionado, al ser 3 dimensiones, cada una de las instrucciones anteriores puede alterar a cada una de estas dimensiones.

- Rotate {x,y,z}
- Translate {x,y,z}
- Scale {x,y,z}

Por lo que si tenemos una posición inicial del objeto de {5,30,2} y queremos hacer la pieza un poco más hacia arriba sólo tendremos que modificar el valor de z, quedando, {5,30,4}. Sería muy complicado hacer estos cambios uno a uno según la animación, por esto Pov Ray utiliza el reloj.

Si queremos que nuestro objeto realice un giro de 360° sobre el eje Z tenemos que incluir este reloj en ese plano, esto sería:

Tomando en cuenta una posición inicial de {0,0,0}

Rotate {0,0,clock}

Con esta instrucción la variable irá tomando diferentes valores para cada imagen. Ahora, este reloj se define en un archivo .ini que es con el que le diremos a Pov-Ray los cambios que queremos realizar, el archivo debe de contener:

```
initial_clock = 0  
final_clock = 360
```

```
initial_frame = 1  
final_frame = 32
```

```
input_file_name= templo azul.pov  
output_file_type = s  
output_file_name = templ
```

initial_clock nos define el valor inicial del reloj.
final_clock nos define el valor final del reloj.

Esto es, el reloj realizará 360 cambios sobre el plano que especificamos anteriormente, con lo que creará imágenes que reflejan a la imagen rotando.

Initial_frame define a partir de que imagen empezará a crear.
Final_frame define el número final de imagen a crear.

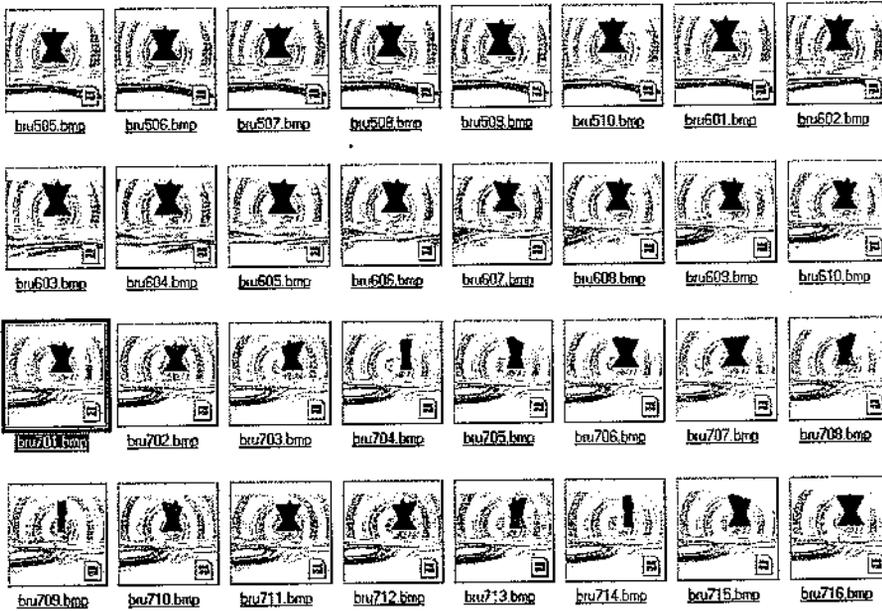
Esto es, que el programa creará un total de 10 imágenes, dentro de las cuales tomará la imagen correspondiente al reloj.

Input_file_name es el archivo fuente de la imagen.

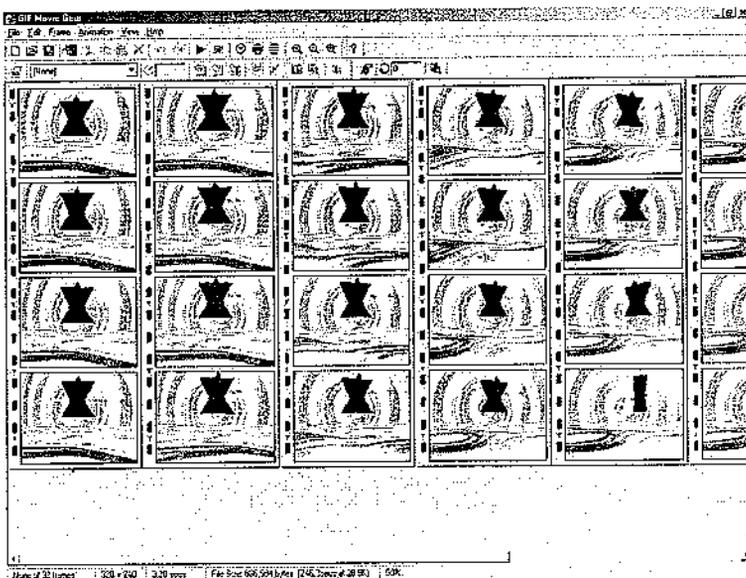
Output_file_type da el tipo de formato en que queremos que nos genere las imágenes, la s significa archivos de sistema, que el caso de Windows son los BMP.

Output_file_name es el nombre base con el que generará las imágenes.

Las imágenes que nos generaría serian las siguientes:

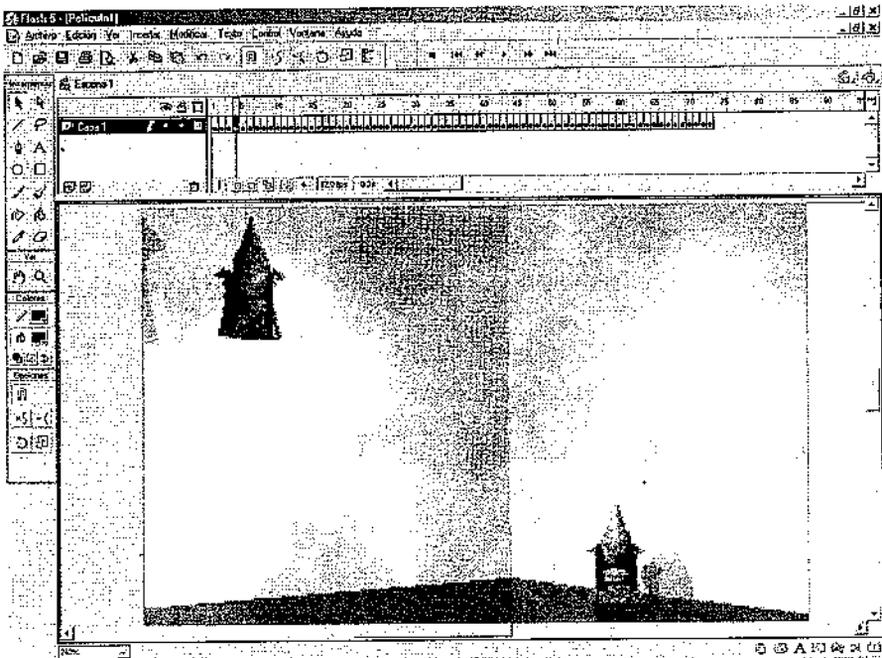


Después con Gif Movie Gear (programa que ya se usó en el capítulo 2) unimos las imágenes individuales, creando la secuencia animada, esta secuencia se puede guardar en un archivo gif, o avi.

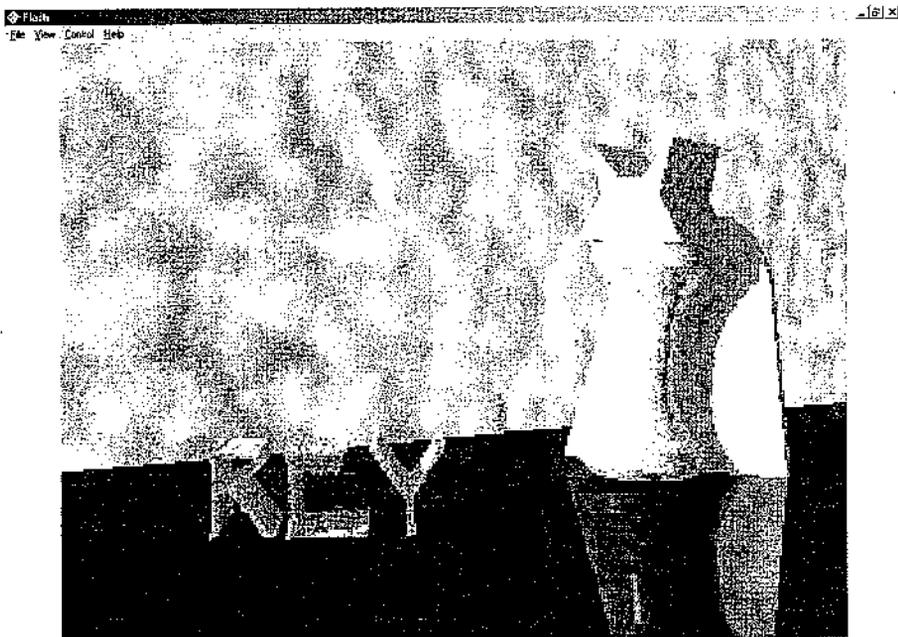


5.3 Creación de Protectores de pantalla

Una vez que se ha creado el archivo .gif, se procede a importarlo en Macromedia Flash, con el cual se generará el archivo .swf, este archivo es muy útil, ya que se puede usar en las páginas web, y también sirve para generar otro tipo de archivos como los screen savers, o protectores de pantalla. De igual forma una vez creadas la imágenes, cualquiera de ellas se puede utilizar como wallpaper o fondo de pantalla.

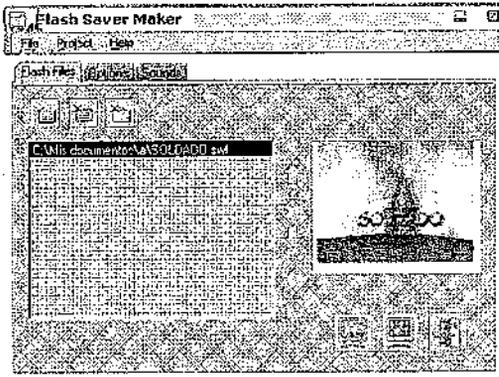


Con Flash también se pueden crear archivos ejecutables, lo cual es muy útil cuando se desea hacer un demos de algo sin necesidad de tener una plataforma ya corriendo para poder visualizar este demo.

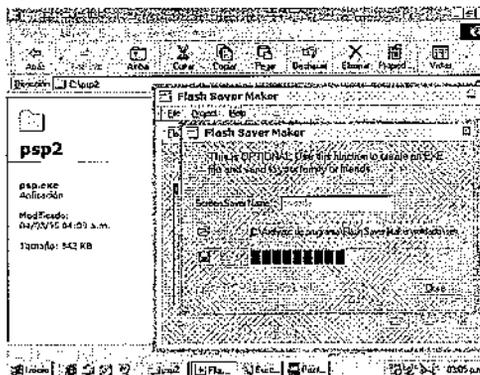
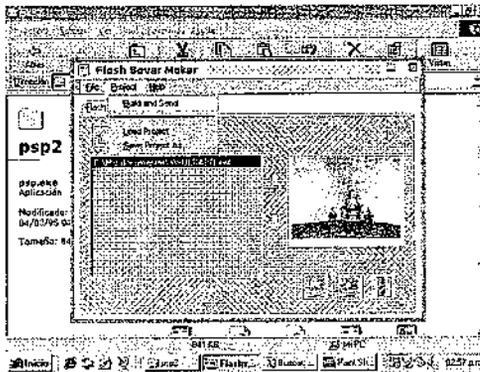


DERECHOS RESERVADOS

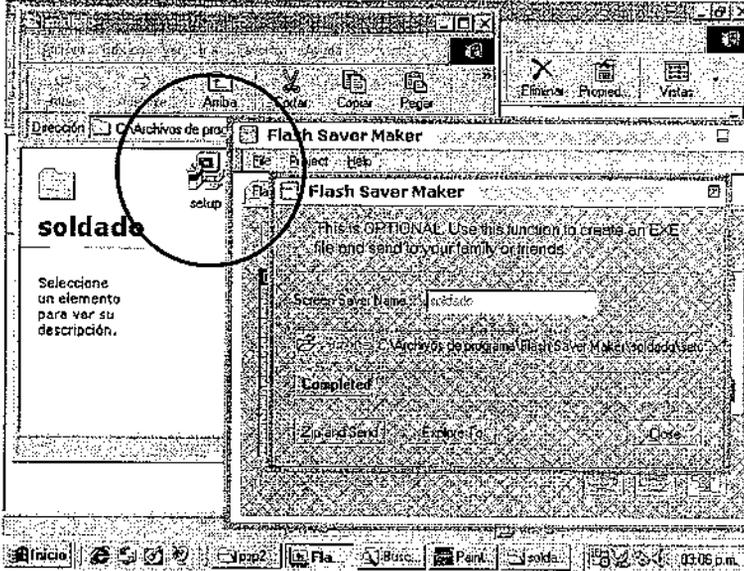
El programa que usé para crear los screen savers fue el Flash Saver Maker, con éste hay que importar el archivo .swf.



Después es necesario pasarnos al apartado de Project y creamos el archivo que instalará el Protector de pantalla en la computadora.



Una vez que hemos creado el archivo Setup, sólo hay que ejecutarlo y automáticamente instala el protector, éste quedará como el protector definido de inicio.



CONCLUSIONES

Cuando empecé a realizar este trabajo, sólo tenía la idea de crear un juego como el ajedrez, pero que tuviera más "acción", más variantes, que las piezas fueran más "audaces", y que no sólo fuera para 2 personas. Pero cuando empecé a desarrollarlo, todo se fue complicando, porque a medida que quería hacerlo más versátil, se iba generando más y más trabajo, al ser tantas piezas y tantos espacios, al momento de realizar validaciones entre todos ellos el código se volvió muy complejo, ya que mientras la pieza más sencilla solo tiene que validar 6 posiciones sin validación de espacios de trayectoria, la pieza más compleja tiene que validar hasta 88 posiciones con sus respectivas validaciones de la trayectoria.

Como el desarrollo del software a nivel mundial ha crecido de una manera impresionante, existen una infinidad de programas y herramientas que facilitan el trabajo para la creación de proyectos de cómputo, esto, aunado al gran avance tecnológico, que ahora permite el manejo de información de una manera más eficaz y rápida, permite a los desarrolladores crear cualquier tipo de programa multimedia.

Flash, Director y Dreamweaver de Macromedia, son un ejemplo de dichos programas, es muy sencillo crear películas, animaciones, presentaciones, y programas completos, ya que con su lenguaje de programación, que es Lingo, se pueden realizar complejos y extensos códigos de programación.

Para crear la interfaz gráfica utilicé 10 diferentes software, no todos los sabía usar, por lo que tuve que aprender a usarlos sobre la marcha, claro que es importante tener las bases para comprender mejor su funcionamiento, así como una buena bibliografía.

La creación de este programa fue de forma exhaustiva, tanto en el desarrollo de conceptos como en su codificación, de esta manera, todas las personas que vean este juego, sabrán que lo que se necesita para crear algo sólo es imaginación y un poco de conocimientos técnicos, que no es necesario ser un experto para poder hacerlo, aunque seguramente para los programadores avanzados, el uso de la Inteligencia Artificial habría simplificado las cosas.

El juego como tal, es un tablero con piezas, similar a otros, y de igual forma lo pueden jugar 2 personas, pero ¿qué pasa cuando hay más personas?, pues con Batel-Nah las invitas a jugar, ya que pueden jugar hasta 4 personas. También pueden darse otras combinaciones, puede jugar uno sólo contra otras 2 personas, o sólo dos personas manejando los 4 equipos, esto permite crear un mayor nivel de dificultad, para esas personas que buscan algo nuevo y tal vez mas complejo que el ajedrez.

Para que el juego en software funcione hay muchas cosas que se tienen que validar, como las reglas individuales de movimiento para cada pieza, definir el área del tablero para que el sistema sepa cuando la pieza se quiere colocar fuera de éste, controlar las piezas activas y las eliminadas, ubicar en cada espacio su estado, ya sea que esté libre u ocupado, y por quién está ocupado, si dentro de la trayectoria de movimiento de una pieza existe otra que estorbe el paso, validar que cuando se quiere eliminar a un enemigo éste realmente lo sea, y no sea un compañero, controlar los turnos de tiro, para que los demás no puedan tirar, validar que si se elimina a un grupo de aliados completamente, el turno saite al siguiente,

contar el tiempo por turno, para que si pasan más de 5 min. sin que el jugador tire, éste pierda su turno, y muchas otras cosas más.

Si se quiere que un programa sea exitoso, es necesario distribuirlo al mayor número de usuarios, para ello es necesario buscar las mejores plataformas de distribución, por lo que Internet es una gran ventana hacia millones de usuarios es todo el mundo, así que partiendo de este hecho es necesario crear la interfaz para implementar Batel-Nah en un portal de Internet.

En el último capítulo de mi tesis, que es un proyecto futuro, desarrollo un portal, una ventana de cómo podría implementarse, aunque para poder realmente implementarlo en un servidor y liberarlo, es necesario hacerlo con otras herramientas y toda una infraestructura de hardware, pero lo más importante, es estar protegido legalmente para que no plagien la idea.

A corto plazo espero contar con los elementos necesarios para que este último capítulo se vuelva una realidad.

Al realizar este trabajo demuestro cómo con un poco de creatividad, los conocimientos y las herramientas indicadas, se puede crear cualquier tipo de programa, en este caso un juego, un juego que aunque sus bases están dadas ya por otro, ofrece variantes muy interesantes, además de que fomenta el trabajo en equipo, tan necesario en nuestros días para todas las estructuras organizacionales de cualquier empresa, así como para la unión familiar.