

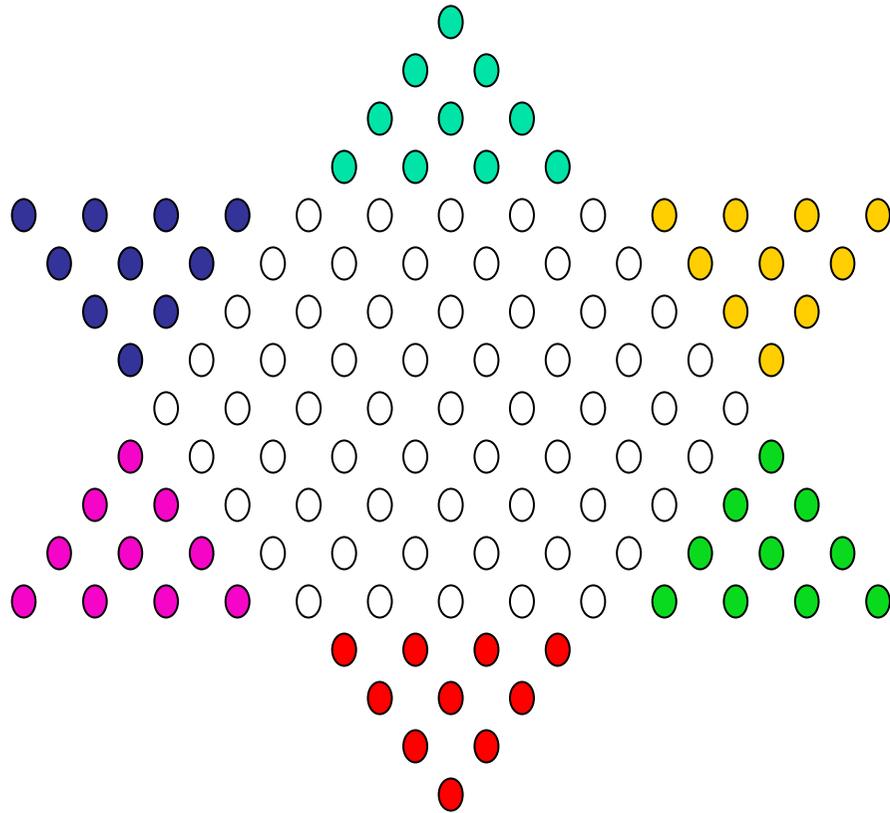
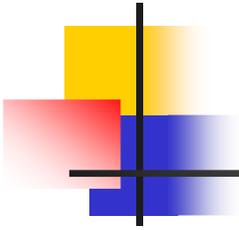


CES-11

---

Laboratório LAB01

Carlos Henrique Q. Forster 2010



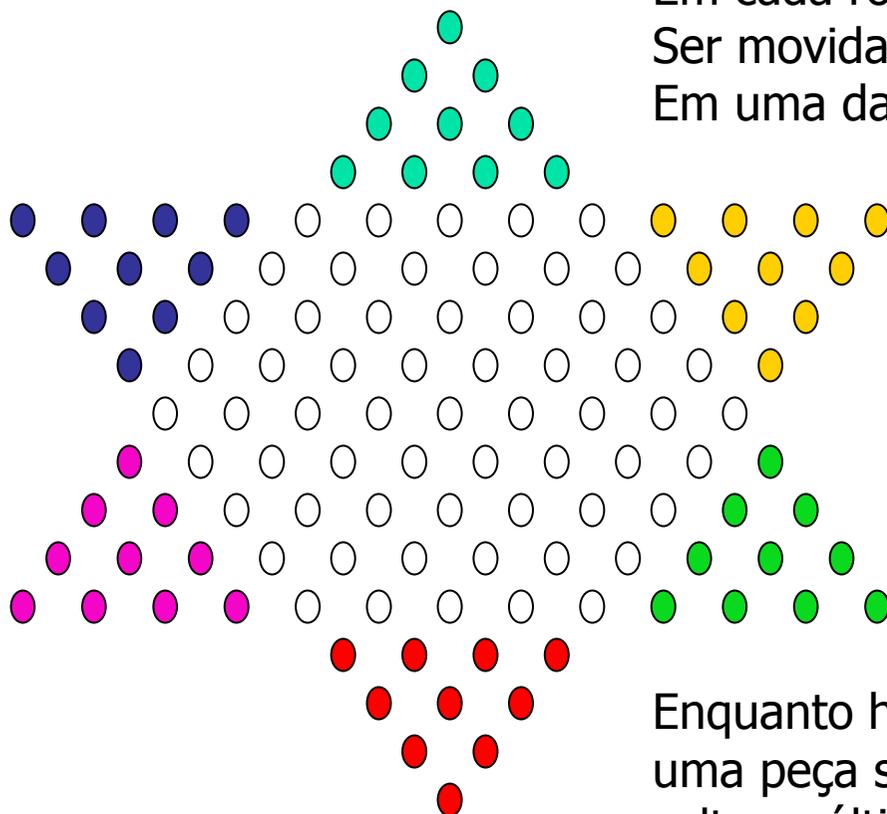
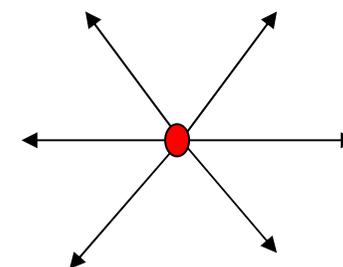
***Xadrez***

***Xinez***

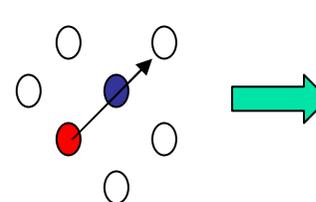
# Regras do jogo

Joga-se em rodadas.

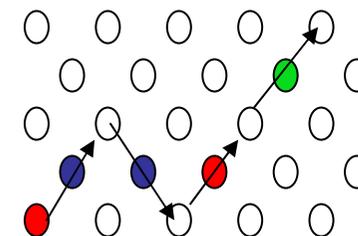
Em cada rodada, uma peça pode  
Ser movida para uma casa adjacente  
Em uma das seis direções possíveis.

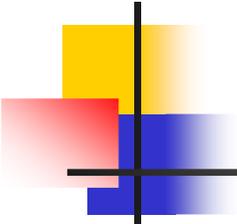


Porém, se a casa adjacente estiver  
Ocupada, mas a seguinte não estiver, é  
Possível saltá-la.

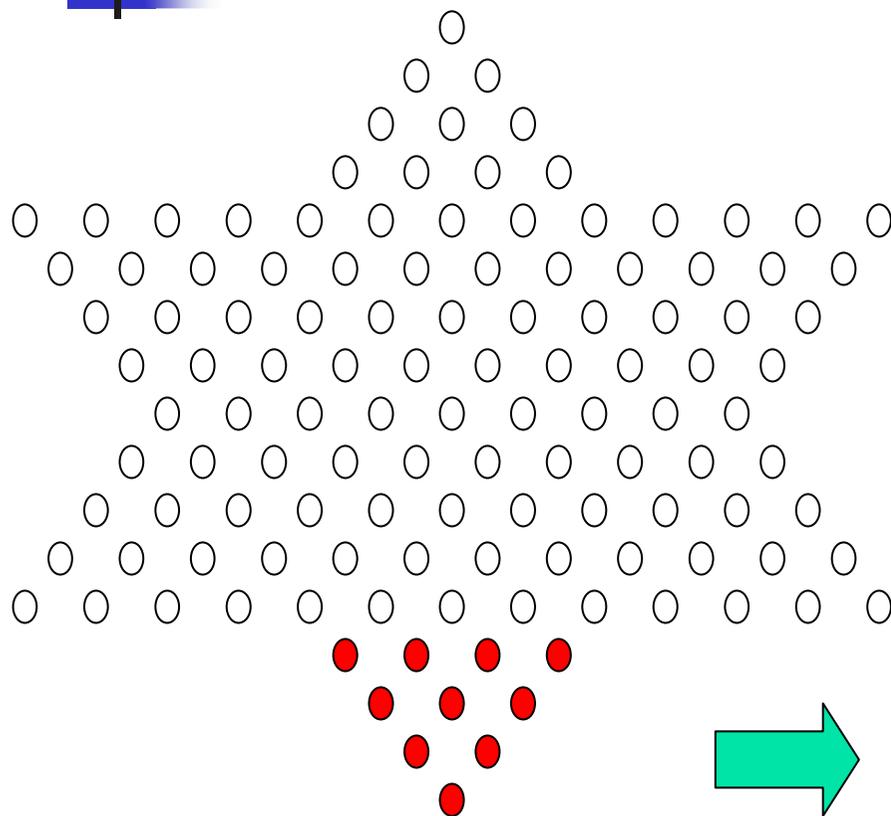


Enquanto houver espaço para  
uma peça saltar, ela poderá  
saltar múltiplas vezes na  
mesma rodada.



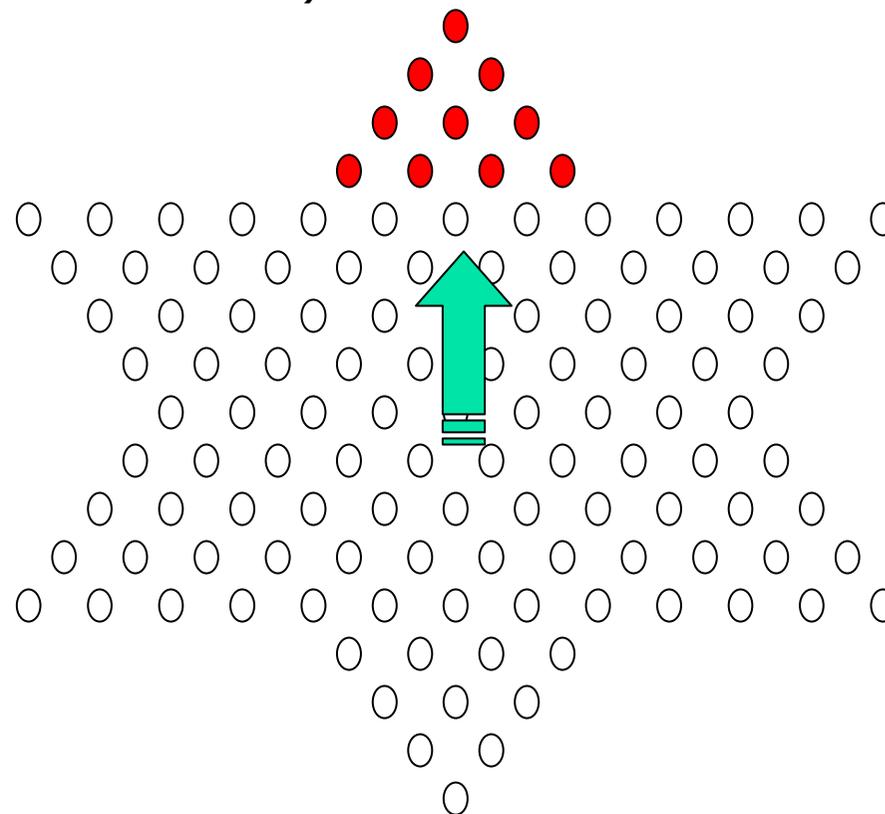


# Objetivo do jogo

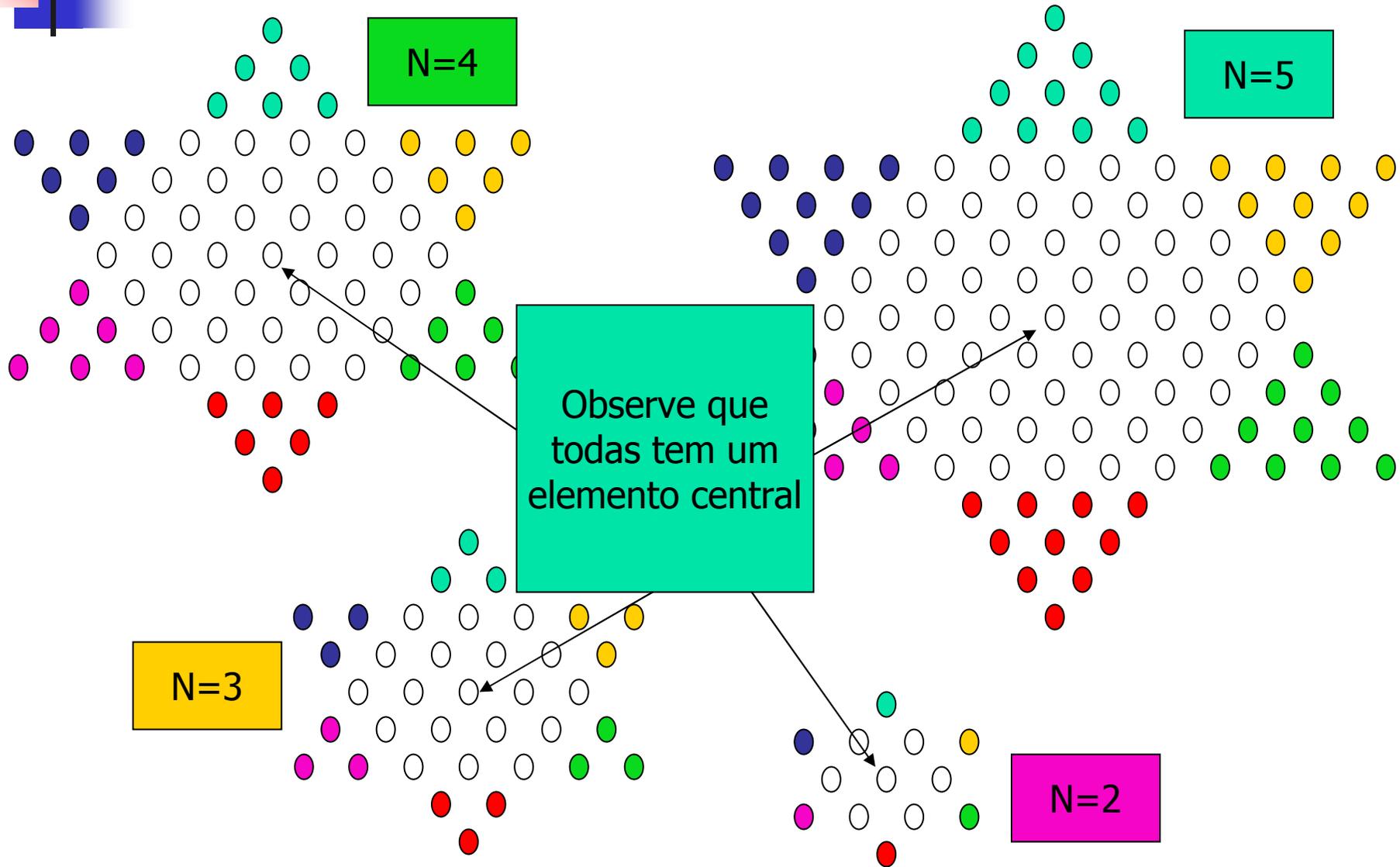


Até seis jogadores podem jogar!!

Mover todas as peças para o canto oposto no menor número de rodadas possível (ou em menos rodadas que o adversário)



# Variações do jogo

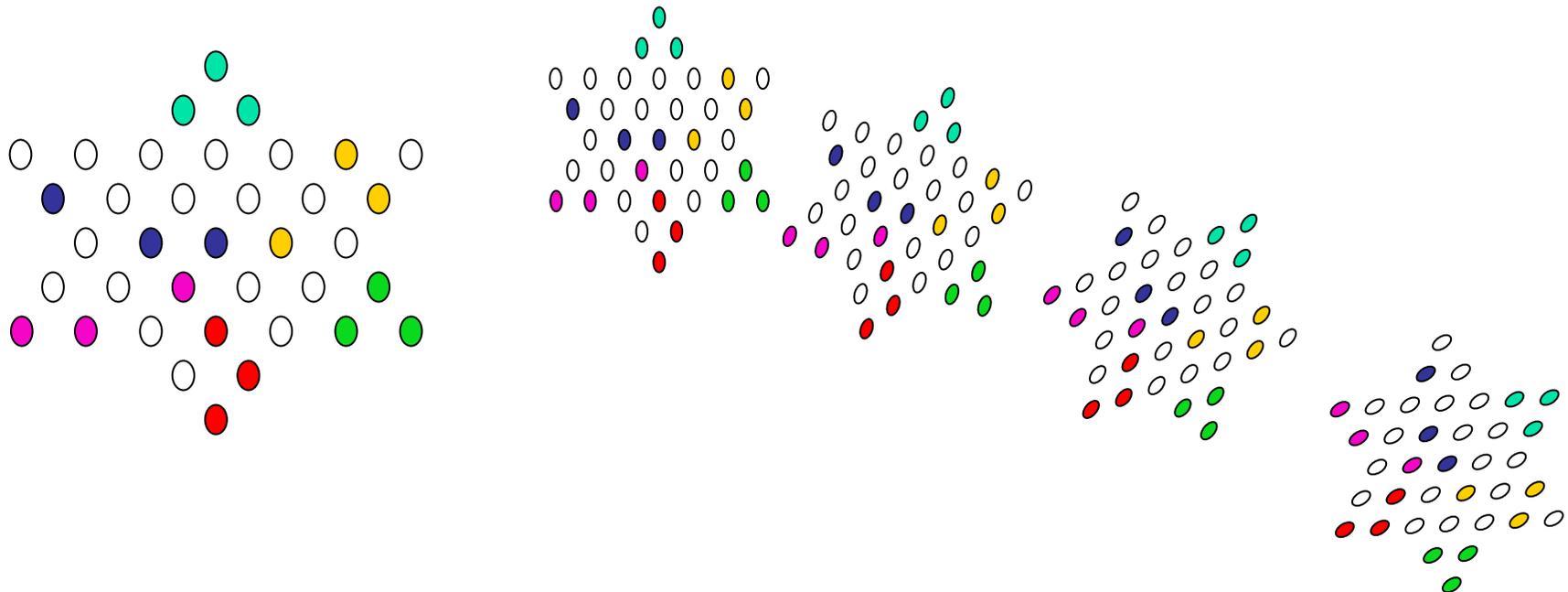




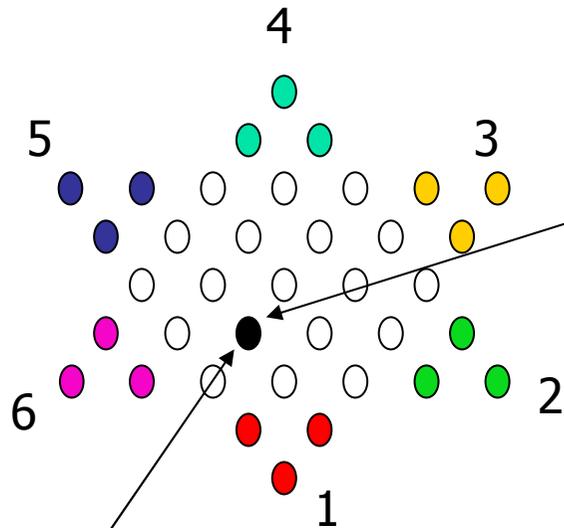
# Propriedades

Observe que o objetivo do jogo corresponde a maximizar o somatório das coordenadas  $w$  de todas as peças de um determinado jogador!!

Podemos calcular as coordenadas em relação ao ponto de vista de outro jogador. Seria como rotacionar o tabuleiro!!



# Rotação



Para o jogador verde, as coordenadas são  
 $u=-1$   
 $v=-1$   
 $w=0$

Para rotacionar ao contrário, basta rotacionar 5 vezes!!

Para o jogador vermelho, as coordenadas são  
 $u=0$   
 $v=-1$   
 $w=-1$

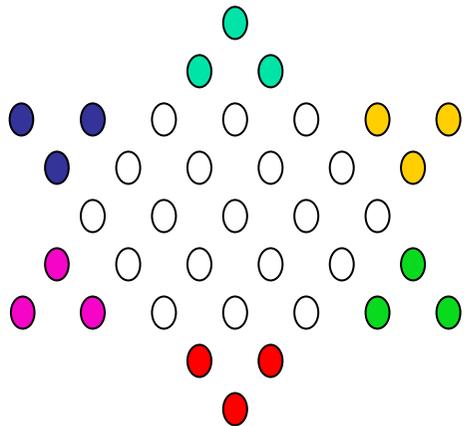
Os jogadores são numerados no sentido anti-horário

A coordenada  $w$  é então a medida da situação de cada jogador em seu ponto de vista.

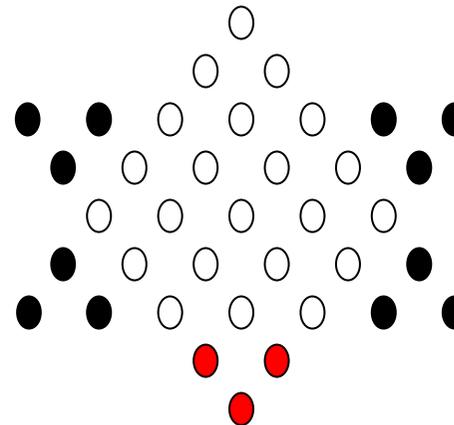
Reservamos o Zero para representar as casas vazias do tabuleiro.

# Jogadas Válidas

Uma jogada válida consiste na movimentação para uma casa adjacente não-ocupada ou numa seqüência de saltos.

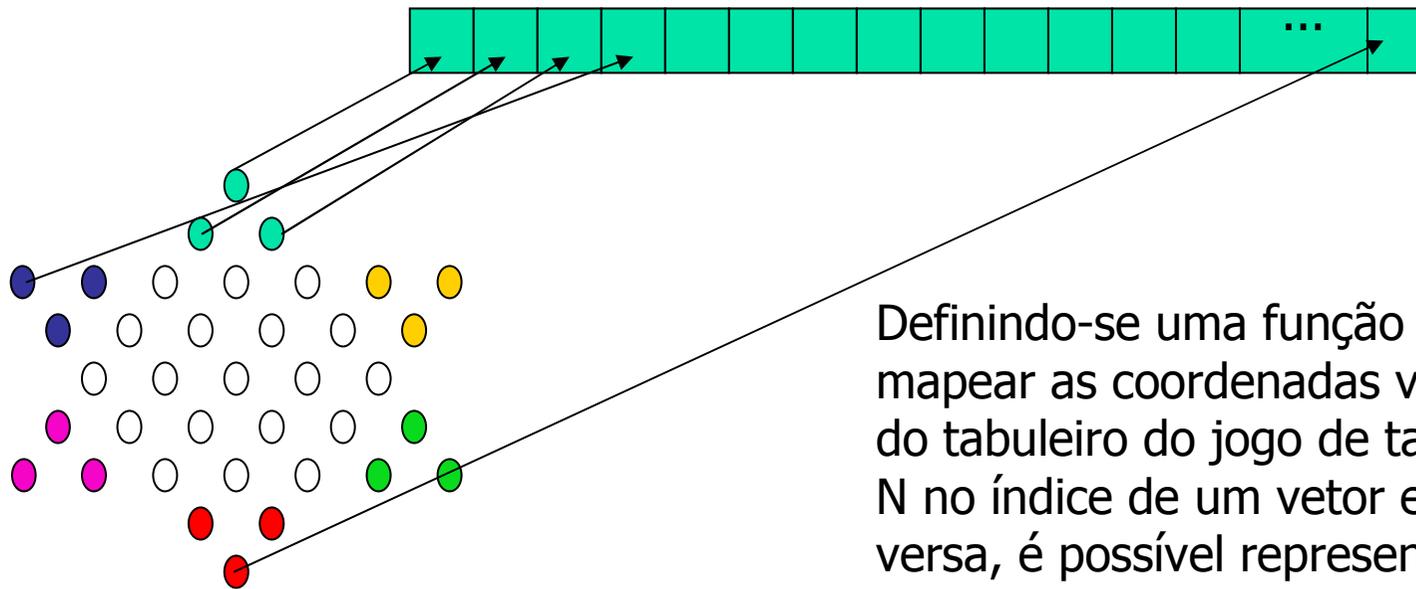


No caso ao lado, as peças vermelhas só podem ocupar suas casas originais, suas casas de destino ou o hexágono central.



Usando o sistema de coordenadas definido, fica fácil determinar se alguma jogada sairia para fora da área permitida.

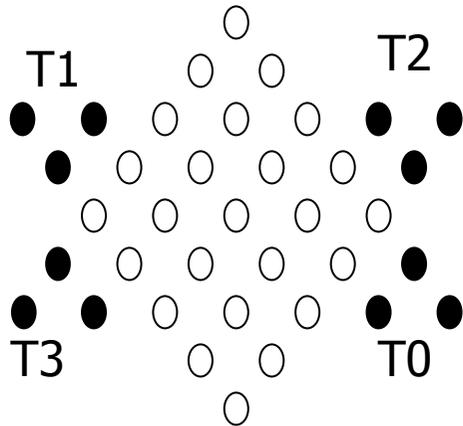
# Representação linear



Definindo-se uma função para mapear as coordenadas válidas do tabuleiro do jogo de tamanho  $N$  no índice de um vetor e vice-versa, é possível representar o posicionamento das peças no tabuleiro em um vetor de tamanho fixo.

É possível representar uma jogada através de apenas 2 valores inteiros: o índice do ponto de partida e o índice do ponto de chegada.

# Representação Linear



Uma peça está dentro do tabuleiro se duas destas três condições forem cumpridas:

$$-2 \leq u \leq 2$$

$$-2 \leq v \leq 2$$

$$-2 \leq w \leq 2$$

Se cumprir as três, estará no hexagono central.

Observe que a região central é um "quadrado" facilmente linearizável com coordenadas:

$$-2 \leq u \leq 2$$

$$-2 \leq v \leq 2$$

O índice de uma peça nesse quadrado será:

$$\text{pos} = u + 2 + 5 * (v + 2)$$

O triângulo externo pode ser determinado testando as condições:

$$u > 2 : t = 0$$

$$u < -2 : t = 1$$

$$v > 2 : t = 4$$

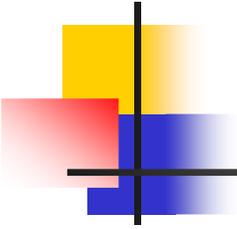
$$v < -2 : t = 3$$

O índice inicial do triângulo é

$$25 + 3 * t + x$$

Onde 25 é o tamanho do quadrado e 3 é o tamanho de cada triângulo.

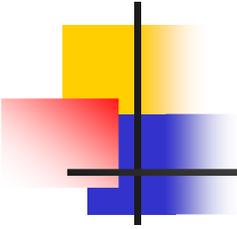
E,  $x < 3$  depende de  $u$ ,  $v$  e  $t$ .



# Tarefa

---

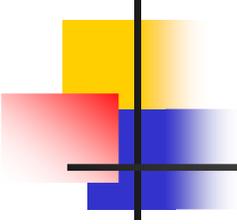
- Pedem-se Funções para...
  - Calcular um índice inteiro dadas as coordenadas  $u,v,w$
  - Obter  $u,v,w$  dado o índice inteiro da posição
  - Gerar uma configuração inicial do tabuleiro definidos os jogadores (de 1 a 6) que vão participar.
  - Imprimir o tabuleiro no console (formato legível).
  - Rotacionar uma tripla de coordenadas  $(u,v,w)$  do tabuleiro dado o jogador que deve ficar ao sul.
  - Gerar todas jogadas válidas para uma peça.
  - Verificar se uma jogada é válida.
  - Medir o valor da soma das coordenadas  $w$  para um jogador.
  - Determinar se as peças de um jogador estão todas no destino.
- E um programinha para
  - Iniciar um jogo dados quais jogadores são humanos e quais são robots.
  - Para jogadores humanos: receber uma jogada do console, verificar se é válida e efetuar a jogada.
  - Para jogadores automáticos: efetuar a jogada que mais aumenta a soma de  $w$ .



# Sugestões de funções

---

- `int posicao(int u, int v, int w);`
- `int u_coord(int pos); int v_coord(int pos); int w_coord(int pos);`
- `tabuleiro *cria_tabuleiro(jog1, jog2, jog3, jog4, jog5, jog6);`
- `void imprime_tabuleiro(tabuleiro *t);`
- `struct coord rotaciona(struct coord uvw, int r);`
- `int valida_jogada(int pos_origem, int pos_destino);`
- `void gera_jogadas(int pos_peca);`
- `int somaw(tabuleiro *t, int jogador);`
- `int terminou_jogo(tabuleiro *t, int jogador);`



# Recomendações gerais

---

- Usar o tamanho N como variável global
- Alocar o tabuleiro dinamicamente
- Criar tipos adicionais conforme a necessidade, usando struct, enum, union. Deixar claro o significado da informação.