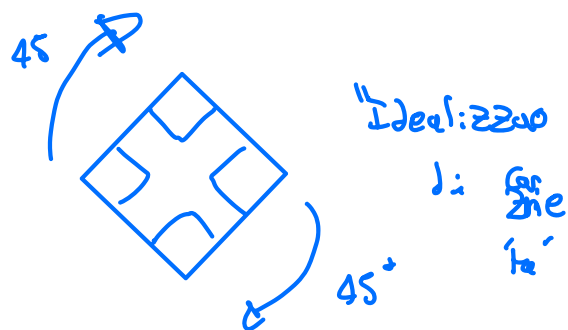
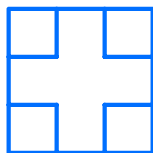
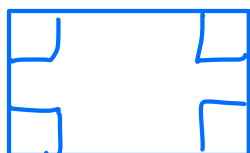


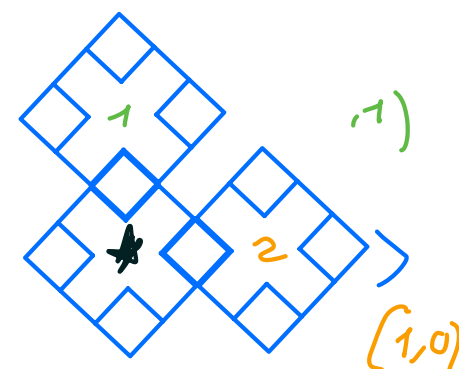
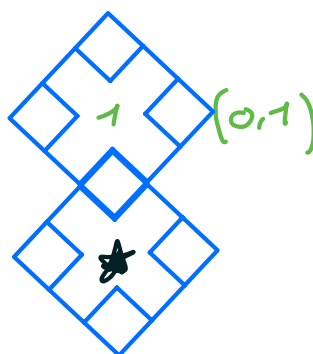
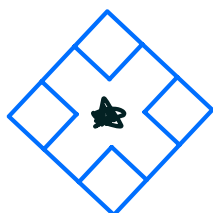
Playable Card



gluterCard

1, 2, ... 21

(0,0)



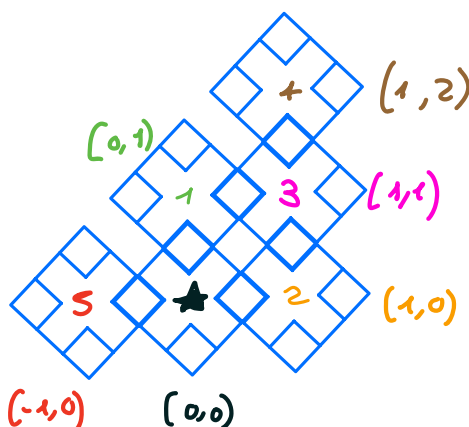
• La carta iniziale è a coordinate (0,0)

• La carta 1 è a coordinate (0,1)

• La carta 2 ha coordinate (1,0)

(aggiunta alla List)

<List>		
C	X	Y
0	0	0
1	0	1
2	1	0
3	1	1
4	1	2
5	-1	0



Ipotesi: Guardare le "distanze di Hamming" delle

coordinate per trovare le carte vicine tra loro.

Se il valore assoluto della differenza tra coordinate di 2 carte
concordi $(x, 0 y)$ risulta ≤ 1 allora le carte sono adiacenti

Inoltre, se $D > 2$ allora le 2 carte distano $D-1$

"carte" di distanza l'una dall'altra

<List>		
C	X	Y
0	0	0
1	0	1
2	1	0
3	1	1
4	1	2
5	-1	0

...

Procedura:

1) si copia il vettore (per mantenere l'ordine di graduatoria)

e si mettono in ordine crescente le

carte (X ha priorit  su Y)

<List> ord.		
C	X	Y
5	-1	0
0	0	0
1	0	1
2	1	0
3	1	1
4	1	2

,

2) Per ogni carta si fa la differenza delle coordinate

$$\Phi(C_1, C_2) = (D_x, D_y)$$

$$(5, 0) = (1, 0) \quad (5, 1) = (1, 1) \quad (5, 2) = (2, 0) \quad (5, 3) = (2, 1) \quad (5, 4) = (2, 2)$$

$$(0,1) = (0,1) \quad (0,2) = (-1,0) \quad (0,3) = (-1,-1) \quad (0,4) = (-1,-2)$$

$$(1,2) = (1,1) \quad (1,3) = (1,0) \quad (1,4) = (1,-1)$$

$$(2,3) = (0,1) \quad (2,4) = (0,2)$$

$$(3,4) = (0,1)$$

Per esempio le
carte vicino alla
Starting Card sono
5, 1 e 2

[distanza Max 1]

In questo modo abbiamo le distanze fra tutte le carte
(le implementazioni possono sfruttare questa caratteristica
per facilitare i molti calcoli. Per esempio per gli
obiettivi "schiera")

La lista è una lista di CardData
che contiene la carta stessa e le sue
coordinate tramite la classe coords (2 int)

CardData
<ul style="list-style-type: none"> - card: PlayerWeCard - coords: Coord
<ul style="list-style-type: none"> + setCoords (int x, y) void + getCard() Card + getCoords(): Coord

*Ogni giocatore possiede questa struttura.