

## Come si Gioca:

L'obiettivo del gioco è semplice: non bisogna esplodere!

All'inizio del gioco, si prende il mazzo di carte, si rimuovono le carte "Exploding Djanni" e "Meoooow", si mischiano le carte e si danno 4 carte ad ogni giocatore. Ad ogni giocatore viene data una carta "Meoooow", e le restanti carte (comprendenti di "Meoooow" avanzati e "Exploding Djanni") vengono mescolate nuovamente. A questo punto il gioco può cominciare.

Ad ogni turno, ogni giocatore potrà giocare nessuna o più carte e per concludere il proprio turno, a meno di carte speciali giocate dal giocatore stesso, dovrà pescare una carta. Una volta pescata la carta, se questa non è un "Exploding Djanni", la conserverà e il suo turno sarà concluso passando dunque al giocatore successivo. Pescare una carta è ovviamente un'operazione pericolosa perché si potrebbe esplodere (man mano che si va avanti, ci saranno sempre meno carte nel mazzo e quindi sarà più alta la probabilità di esplodere).

Nel caso in cui invece un giocatore estragga un "Exploding Djanni", ci sono due casi:

- lo sfortunato giocatore possiede una carta "Meoooow": la gioca (ossia viene scartata) e sopravvive. La carta "Exploding Djanni" dovrà quindi riposta casualmente nel mazzo, e il turno passa al giocatore successivo.
- lo sfortunato giocatore non possiede una carta "Meoooow": la carta "Exploding Djanni" viene scartata e con essa tutte le carte possedute dal giocatore esplosivo, che uscirà dunque dal gioco. Un giocatore e un exploding djanni in meno! Il turno passa al giocatore successivo.

Durante il proprio turno un giocatore potrà giocare tutte le carte che vorrà (non c'è un limite sul numero di carte che può possedere o giocare un giocatore: può tenerle sempre in mano e non giocarle mai, oppure può sempre giocarle tutte e rimanere con zero carte in mano). Ecco elencati i comportamenti nel caso in cui si giochino le seguenti carte:

- NOPE: non succede niente di particolare (ma la carta poteva essere giocata in modo più furbo). Va specificato che il NOPE può essere giocato su un altro NOPE, annullando l'effetto, questo per un numero indefinito di volte. Ipotizziamo che un giocatore usi un ATTACK, il giocatore successivo (nelle funzionalità medie può essere un qualsiasi giocatore) usa un NOPE, a questo punto se viene scartato un altro NOPE la carta ATTACK diventa nuovamente valida perché un NOPE ha annullato l'altro.
- MEOOOOW: non succede niente di particolare (ma la carta poteva essere giocata in modo più furbo, tipo evitare un'esplosione)...
- SEE THE FUTURE: il giocatore che la gioca (solo lui!) può guardare il contenuto delle prime tre carte nel mazzo, e quindi successivamente comportarsi di conseguenza (per esempio: gioco un SEE THE FUTURE e vedo che la prima carta è un "Exploding Djanni"... mi conviene evitare in modo da non essere io il prossimo a pescare una carta!). Questa azione può essere fermata da una NOPE;
- SHUFFLE: il giocatore che la gioca può mischiare il mazzo. Questa azione può essere fermata da una NOPE;
- FAVOR: il giocatore che la gioca può chiedere una carta ad un altro dei giocatori in gioco. Sarà il giocatore a cui è stato chiesto il favore a scegliere la carta da cedere all'avversario. Questa azione può essere fermata da una NOPE;
- ATTACK: il giocatore conclude il proprio turno con questa carta, e non dovrà pescare dal mazzo. Il giocatore successivo dovrà tuttavia eseguire due turni (il giocatore successivo sarà sotto "attack"). Questa azione può essere fermata da una NOPE;
- SKIP: il giocatore conclude il proprio turno con questa carta, e non dovrà pescare dal mazzo; Questa azione può essere fermata da una NOPE;

- DJANNI CARD (singola): non succede niente di particolare (ma la carta poteva essere giocata in modo più furbo).
- DJANNI CARDS (coppia): se il giocatore possiede due DJANNI CARD identiche (es: Djanni Grasso e Djanni Social non sono due Djanni Cards identiche), le può giocare contemporaneamente. In questo caso però sarà il giocatore che ha giocato le Djanni Cards a scegliere una delle carte dell'avversario (senza vederle). Questa azione può essere fermata da una NOPE;
- DJANNI CARDS (tripla) (funzionalità medie e avanzate): se il giocatore possiede tre DJANNI CARD identiche, le può giocare contemporaneamente. Anche in questo caso sarà il giocatore che ha giocato le Djanni Cards a scegliere una delle carte dell'avversario, che però potrà vederle. Questa azione può essere fermata da una NOPE.

NOPE: La NOPE è l'unica carta che può essere giocata da un giocatore *al di fuori dal proprio turno*. Per esempio:

- Giocatore A gioca SEE THE FUTURE;
- Giocatore B decide di giocare NOPE;
- Giocatore A non può vedere le prime tre carte del mazzo, e continua il proprio turno.

In altre parole, il giocatore B si è "intromesso" durante il turno del giocatore A e gli ha impedito di eseguire l'azione relativa alla carta giocata. Il turno del giocatore A si concluderà comunque quando estrarrà una carta o quando giocherà una SKIP/ATTACK (se non verrà bloccata da una NOPE)! Nel caso un giocatore riceva una NOPE dopo aver giocato una SKIP o una ATTACK, dovrà pescare la sua carta e il giocatore successivo non sarà eventualmente costretto a fare due turni. In alternativa, il giocatore può a sua volta giocare un'altra NOPE ed annullare l'effetto della NOPE giocata dall'avversario. Esempio:

1. Giocatore A gioca una ATTACK;
2. Giocatore B gioca una NOPE;
3. Giocatore A gioca un'altra NOPE (ora è come se le ultime due NOPE non fossero mai state giocate, il giocatore B dovrebbe fare due turni);
4. Giocatore B gioca un'altra NOPE;
5. Giocatore A gioca un'altra NOPE;
6. Giocatore B non ha più NOPE, si rassegna e gioca i suoi due turni...

Notare che tutte le mosse dalla 1 alla 5 sono state svolte durante il turno del Giocatore A. Solo quando non viene più giocata alcuna NOPE, viene eseguita l'azione della ATTACK e il turno passa al Giocatore B.

Nelle modalità base, solamente il giocatore immediatamente successivo al giocatore che sta giocando il suo turno potrà giocare una NOPE; Nella modalità media e avanzata invece, chiunque tra i giocatori in gioco potrà giocare una nope.

ATTACK: nel momento in cui un giocatore riceve una ATTACK (non respinta con una NOPE), dovrà giocare due turni, ossia giocare quante carte preferisce ma alla fine concludere due turni. Come visto prima, un turno si può concludere sia pescando una carta che giocando una SKIP o una ATTACK. Nel caso in cui il giocatore sotto ATTACK giochi a sua volta una ATTACK al primo dei due turni, non sarà tenuto ad eseguire il secondo turno. Il giocatore successivo ad esso invece subirà la ATTACK.

# Specifiche

I giocatori sono sempre e solo 4, e devono essere gestiti su un array statico.

Per ogni giocatore, le carte possedute vengono memorizzate in un array dinamico. L'array deve cambiare dimensione ogni volta che viene scartata o pescata una carta.

Il mazzo deve essere gestito:

- mediante un array dinamico per la modalità base (anche in questo caso deve cambiare dimensione ogni volta che viene pescata una carta);
- mediante una lista lineare semplice per le modalità media e avanzata.

All'avvio del gioco si deve poter scegliere se iniziare una nuova partita o caricare una partita già esistente.

Se si inizia una nuova partita:

- Si deve chiedere il nome dei 4 giocatori in gioco;
- Si deve caricare il mazzo di carte da file (dopo aver scelto la modalità di gioco);
- Si deve mischiare il mazzo e distribuire le carte seguendo le regole del gioco. ATTENZIONE: non mettete informazioni che stanno sul file in modo EMBEDDED nel codice, ossia la lettura da file deve funzionare in un qualsiasi file dove il numero di EXPLODING DJANNI e di MEOOOOW è variabile.
- Scelta casuale del giocatore che inizia;

Se si carica una partita:

- Nella modalità base si carica il file "partita.sav", nelle modalità media e avanzata si caricherà dal file inserito in input;
- Si riprende dal punto in cui si è salvato.

## Consigli di Implementazione

Per una migliore giocabilità, fare in modo che ad ogni turno l'utente possa scegliere di fare una di queste tre azioni:

- Pescare una carta: il turno passa al giocatore successivo che potrà scegliere se pescare, giocare una carta o salvare la partita;
- Giocare una carta: eseguo l'azione relativa alla carta giocata e richiedo all'utente se vuole giocare un'altra carta, pescare oppure salvare la partita;

- Salvare la partita: la partita viene salvata, e successivamente viene chiesto se pescare una carta, giocare una carta o (inutilmente) salvare di nuovo la partita.

Quando un giocatore gioca una carta che può essere annullata da una NOPE, chiedere se qualcuno (nella modalità base solo il giocatore successivo) vuole giocare una NOPE solo se c'è effettivamente qualcuno che può giocare una NOPE. Per esempio: è inutile chiedere "Giocatore A, vuoi giocare una NOPE?" se il Giocatore A non ha in mano una NOPE.

Nel caso in cui ci teneste a fare un gioco davvero giocabile, l'ideale sarebbe che le carte possedute dal giocatore A venissero visualizzate solamente quando è il turno del Giocatore A. Tuttavia, durante lo sviluppo è bene tener sotto controllo le carte possedute da tutti i giocatori ad ogni turno per essere sicuri che tutto funzioni correttamente. Questo è un caso perfetto per l'utilizzo della compilazione condizionale vista durante l'ultima lezione di laboratorio!

## IA:

Provate a scrivere il codice in modo tale che tutte le interazioni necessarie con l'utente siano gestite da funzioni. In questo modo, quando si inizierà l'implementazione della IA il codice relativo alla gestione del gioco cambierà poco: se il giocatore è un utente, chiamo la funzione che acquisisce il dato che mi serve, altrimenti chiamo la funzione gestisce la IA.

Durante il gioco è perfettamente legittimo contare quante carte ci sono ancora all'interno del mazzo (una stampa di tale numero ad ogni turno è consigliata), ed è legittimo contare quanti sono i giocatori ancora in gioco (e di conseguenza conoscere quanti "EXPLODING DJANNI" sono presenti nel mazzo). Conoscendo questi dati, è possibile calcolare la probabilità di pescare un EXPLODING DJANNI:

$p = n_{\text{ExplodingDjanni}} / n_{\text{Carte}}$ ; (p è un valore da 0 a 1, dove 0 significa "0%" e 1 significa "100%")

questo valore dovrebbe essere utilizzato dal computer per capire quando conviene giocare determinate carte!

Per quanto riguarda il giocare la NOPE, capire in modo automatico quando conviene e quando non conviene usare una NOPE non è banale. Se non avete idee particolari, anche una semplice strategia del tipo "genero un numero casuale per stabilire se giocare la nope o no" andrà bene.

## Consegne:

Per chi vuole consegnare solo la versione base, il consiglio è di consegnare il prima possibile. Consegnare un progetto versione base il 31 maggio significa consegnare un progetto che potrà avere come voto massimo 15, e su questo voto poi dovranno essere tolti 4.14 punti per malus di consegna in ritardo, ossia voto massimo 10.86. Tuttavia la sufficienza sul progetto è 10, e prendendo meno di 10 il progetto risulta insufficiente e bisognerà risostenere tutti gli esami (teoria e pratica) di Programmazione 1.