

2011

Gioco online “UNO”

Progetto di Reti di Calcolatori e Sistemi Informatici

Il progetto consiste nella realizzazione di un'applicazione Web che permetta a più giocatori di fare una partita ad “UNO” online. Ogni giocatore avvia localmente la propria applicazione e sul server parte una prima fase in cui i vari giocatori si uniscono alla partita. Se entro un certo intervallo di tempo si collegano 4 giocatori, la partita inizia; altrimenti i giocatori mancanti vengono sostituiti dai giocatori virtuali AI. Il sistema a questo punto distribuirà le carte iniziali ai giocatori e si occuperà poi di abilitarli a giocare quando è il loro turno. Sarà sempre il sistema a gestire l'estrazione delle carte, il calcolo dei punteggi e il rispetto delle regole.

Sarabakha Andriy



Indice

1. Introduzione	2
1.1. Specifiche del progetto	2
1.2. Gioco “Uno”	3
1.3. Regole del gioco	7
1.3.1. Carte	7
1.3.2. Obiettivo del gioco	7
1.3.3. Procedimento del gioco	7
1.3.4. Calcolo del punteggio	8
1.3.5. Funzioni delle carte “Azione”	9
1.4. Carte	10
2. Modulo server	17
2.1. Analisi del modulo server	18
2.2. Interfaccia grafica del modulo server	23
2.3. Giocatori Bot	25
2.3.1. Bot v.1	26
2.3.2. Bot v.2	27
2.3.3. Bot v.3	29
2.3.4. Bot v.4	31
2.3.5. Bot v.5	33
2.3.6. Confronto tra i Bot	34
3. Modulo client	36
3.1. Interfaccia grafica del modulo client	37
4. Interazione tra client e server	46
5. Conclusione	48

1.1. Specifiche del progetto

Il progetto consiste nella realizzazione di un'applicazione Web che permetta a più giocatori di fare una partita ad “UNO” online. Ogni giocatore avvia localmente la propria applicazione e sul server parte una prima fase in cui i vari giocatori si uniscono alla partita. Se entro un certo intervallo di tempo si collegano 4 giocatori, la partita inizia; altrimenti i giocatori mancanti vengono sostituiti dai giocatori virtuali AI. Il sistema a questo punto distribuirà le carte iniziali ai giocatori e si occuperà poi di abilitarli a giocare quando è il loro turno. Sarà sempre il sistema a gestire l'estrazione delle carte, il calcolo dei punteggi e il rispetto delle regole.

Le applicazioni del client e del server saranno realizzate come Java Application.

Gioco “Uno”

“ [...] Infatti, la situazione sta in questi termini: come vediamo che gli uccelli costruiscono e modellano i propri nidi, ma poi, dopo aver compiuto il lavoro, svolazzano di qua e di là liberamente, liberi da lavoro, allo stesso modo, i nostri animi, liberi dagli impegni politici e dal lavoro cittadino, gioiscono e desiderano vagare liberi da preoccupazioni e fatiche. [...]” – Marco Tullio Cicerone.

Fin dalla notte dei tempi, una delle attività più importanti per l'uomo, come momento di ristoro, motivo di ritrovo o come stimolo per la creatività dei bambini, è il gioco.

Il gioco è un'attività ricreativa che coinvolge una o più persone (i giocatori), basata su:

- un obiettivo che i giocatori devono cercare di raggiungere (che può anche essere diverso per ciascun giocatore) nell'ambito dell'attività di gioco.
- un insieme di regole, che determinano ciò che i giocatori possono e non possono fare durante l'attività ludica; intraprendere un'azione al di fuori delle stesse costituisce generalmente un errore o fallo (e se quest'ultimo è intenzionale significa barare).

I giochi nascono da cultura popolare o da inventori (purtroppo non sempre identificabili) e seppure il loro scopo sia di fornire intrattenimento e divertimento, spesso riescono a raggiungere anche notevoli risultati educativi. Nella prima età con i giochi basati su forme, lettere e colori, nei periodi successivi con l'esercizio della memoria (es. giochi con le carte e i quiz), con l'invito al ragionamento (es. scacchi, giochi di strategia) e l'apprendimento di alcune realtà (es. programmi di simulazione).

L'importanza del gioco, soprattutto per gli infanti, è confermata dalla Ludologia (scienza che studia il gioco), che spesso coinvolge molti campi tecnici, inclusi psicologia, sociologia, calcolo delle probabilità,

statistica, economia, etnomatematica e teoria dei giochi, branca specialistica della matematica.

Il tipo di gioco più diffuso fra gli italiani è l'azzardo: possibilità di guadagno per i più fortunati o oggetto di dipendenza patologica, l'azzardo è praticato da circa l'80% degli abitanti dell'Italia, che nel 2007 ha raggiunto il primato mondiale per spesa pro-capite l'anno (44 miliardi di euro in tutto) nel gioco lecito.

Alcuni giochi possono coinvolgere un solo giocatore, che gioca in "solitario", ma, nella maggior parte dei casi, essi prevedono una competizione tra due o più persone.

Le origini delle carte da gioco si perdono nella notte dei tempi: le prime testimonianze risalgono, infatti, al X secolo in Cina e in India con il ganjifa. Alcuni storici legano l'origine delle carte alla nascita della carta moneta proprio nel continente asiatico: in questa visione il mezzo e la posta in gioco dell'azzardo si fondevano in un unico veicolo; altri storici sostengono invece che le carte derivino direttamente dai tasselli del domino. Le tradizioni Indiane o Cinesi passarono comunque attraverso la Persia fino a giungere alle coste mediorientali dominate dagli arabi che nel medioevo trasformarono le carte in un modello che divenne già molto simile a quello contemporaneo. Le carte arabe, seguendo i dettami del corano che vietava di ritrarre le persone, riportavano solo disegni astratti, ma avevano comunque una didascalia con il nome degli ufficiali dell'esercito. È probabile che le carte inizino la loro diffusione nel vecchio continente solo dal XIV con l'intensificarsi degli scambi commerciali con gli arabi egiziani. Le prime testimonianze storiche in Europa dei giochi di carte si ebbero in Italia e in Spagna da dove poi si diffusero cambiando forma e adeguandosi di regione in regione.

Durante il XIV secolo le carte si diffusero velocemente in tutta Europa e rappresentano uno dei capitoli più affascinanti della storia del folclore europeo: ci sono pervenuti mazzi di carte commissionati da ricchi signori ad artisti che hanno espresso tutta la loro creatività nella

creazione delle carte, minuziosamente dettagliate e ornate con particolari in materiali preziosi.

In Europa variano molto la struttura e l'aspetto delle carte da gioco iniziando con l'introduzione dei disegni che ne semplificavano notevolmente l'utilizzo; inoltre, dopo la scoperta di Gutenberg, si diffusero le carte stampate, utilizzate soprattutto dal popolo e molto più rozze delle carte fatte a mano per i nobili.

Storicamente gli italiani unirono alle carte di origine moresca 22 soggetti illustrati di origine locale ed aggiunsero alle tre figure maschili una figura femminile la “regina” dando così origine al mazzo per il gioco dei trionfi cioè i “tarocchi” che con alterne fortune sono arrivati fino a noi. I tarocchi furono utilizzati solo dai ceti ricchi, data la difficoltà di realizzazione e il complesso regolamento che legava il gioco dei Trionfi. Il termine Tarocco comunque fu introdotto solo dal XVI secolo e solo dal 1780 si sviluppò l'uso di queste carte per pratiche esoteriche.

Le carte da gioco continuano a evolversi e mutare anche se in nessun caso vengono stravolte le logiche che ne regolano la struttura.

Tra il Cinquecento e il Settecento cominciarono ad apparire i primi manuali che fissavano le regole dei vari giochi di carte.

Dal XIX secolo iniziarono a comparire le indicazioni del valore delle carte sui bordi, per permettere al giocatore di tenere le carte ravvicinate a ventaglio con una sola mano. L'innovazione successiva fu quella di disegnare le figure in modo simmetrico, in altre parole “a due teste”, così da non costringere il giocatore a girare la carta dando indicazione all'avversario delle carte che si possedevano.

La storia recente ci dice che i giochi di carte attualmente più diffusi sono anche i più recenti. Il poker nacque nel 1829 negli Stati Uniti, il Bridge si sviluppò tra fine '800 e gli anni 20, il baccarà nella prima metà dell'ottocento, la Canasta fu inventata a Montevideo durante la seconda guerra mondiale e la scala quaranta iniziò la sua diffusione dall'Ungheria

nel primo dopoguerra. I giochi tradizionali italiani hanno di solito origini più lontane ma la loro forma attuale ha ben poco a che vedere con quella originale: Barrica e Primavera risalirebbero al XVI secolo, la Scopa e la Briscola non vanno più in là dell'inizio 1800, il Tresette sarebbe nato agli inizi del 1700 ed il Settemezzo intorno al 1890.

Il gioco delle carte contemporaneo e i giochi di carte in generale sono legati a doppio filo con il computer e la rete.

I siti per giocare on line ai vari giochi di carte si sono numericamente moltiplicati aprendo alle generazioni più anziane la possibilità di avvicinarsi a questa realtà e, viceversa, facendo conoscere questi giochi ai più giovani. Per esempio, molto diffuso in rete è il gioco di carte UNO.

UNO è un gioco di carte non collezionabili creato da Merle Robbins nel 1971. Il numero dei giocatori può variare da 2 a 10. È venduto in 80 paesi ed ha raggiunto i 150 milioni di copie vendute al mondo. Ne sono state distribuite varie versioni speciali, associate a un tema preciso (esistono, ad esempio, il Barbie UNO, l'Harry Potter UNO, il The Simpsons UNO, il Spongebob UNO, il Cars UNO), e ne sono state fatte delle trasposizioni per computer, come una versione giocabile in chat su IRC, una versione open source "Duo", una versione per Windows Live Messenger e Facebook. La versione per Xbox Live Arcade di Microsoft, quella per WiiWare (console Nintendo Wii) e per iPhone, iPod touch e iPad è stata sviluppata da Gameloft.

1.2. Regole del gioco

1.2.1. Le carte

Il mazzo contiene 108 carte così distribuite:

- 19 carte di colore rosso che vanno dall'1 al 9 (2 serie) più uno 0,
- 19 carte di colore blu che vanno dall'1 al 9 (2 serie) più uno 0,
- 19 carte di colore giallo che vanno dall'1 al 9 (2 serie) più uno 0,
- 19 carte di colore verde che vanno dall'1 al 9 (2 serie) più uno 0,
- 8 carte “Inverti” o “Cambio Giro” dei quattro colori sopracitati,
- 8 carte “Salta” o “Stop” dei quattro colori sopracitati,
- 8 carte “Pesca Due” dei quattro colori sopracitati,
- 4 carte “Jolly” o “Cambio Colore”,
- 4 carte “Jolly Pesca Quattro” o “Cambio Colore - Pesca Quattro”.

Tutte le carte sono elencate nella tabella del Capitolo 1.3.

1.2.2. Obiettivo del gioco

L'obiettivo del gioco è di essere il primo giocatore ad accumulare 500 punti. È necessario anzitutto terminare le proprie carte, per procedere poi al conteggio dei punti conseguiti. Come si calcolano i punti è descritto nel Capitolo 1.2.4.

1.2.3. Procedimento del gioco

Le carte vengono mescolate e distribuite 7 a testa. La porzione del mazzo rimanente, detta “mazzo pesca”, viene posata sul tavolo a dorso coperto e la prima carta viene girata, andando così a formare il “mazzo scarti”. Il primo giocatore deve scartare una delle proprie carte a disposizione, compatibile con quella in cima al mazzo scarti per colore o

per numero, oppure può usare una carta Azione sempre compatibile con la carta del mazzo scarti. Se il giocatore non ha a disposizione una carta da scartare, è l'obbligato a pescarne una dal mazzo pesca; se la carta pescata può essere giocata, lo può fare immediatamente, altrimenti il turno passa al giocatore successivo. Il gioco procede in senso orario, finché non è cambiato dalla carta “Cambio Giro”.

Mentre un giocatore scarta la penultima carta rimasta, deve pronunciare “UNO!”. Se il giocatore non esclama “UNO！”, deve pescare 2 carte dal mazzo pesca.

Una volta che uno dei giocatori rimane senza carte, si procede al calcolo del punteggio che ha conseguito. Nel caso in cui l'ultima carta giocata era un “Pesca Due” o un “Jolly Pesca Quattro” il giocatore seguente deve comunque prendere il numero di carte corrispondenti che andranno poi a sommarsi con gli altri punti.

Per impedire che un giocatore rimane inattivo per lunghi periodi, è stata aggiunta un'altra regola che obbliga ogni giocatore di fare una mossa entro 10 secondi dal momento che il turno è passato al giocatore stesso. Se entro questo intervallo di tempo la mossa non è stata effettuata, questo equivale di aver passato il turno e il giocatore deve prendere una carta. Se, invece, la mossa era la scelta di un colore, viene scelto un colore casuale.

1.2.4. Calcolo del punteggio

Il primo giocatore che non ha più carte in mano vince e guadagna un certo numero di punti calcolati sulla base delle carte rimanenti agli altri giocatori. Il punteggio viene calcolato sommando i valori delle carte:

- le carte numeriche valgono il valore della carta,
- “Cambio Giro”, “Salta” e “Pesca Due” – 20 punti,
- “Cambio Colore” e “Pesca Quattro” – 50 punti.

1.2.5. Funzioni delle carte “Azione”

Le funzioni delle carte “Azione” sono le seguenti:

- La carta “Inverti” o “Cambio Giro” – inverte il senso di gioco da orario ad antiorario o viceversa.
- La carta “Salta” o “Stop” – fa saltare un turno al giocatore successivo nel senso del gioco.
- La carta “Pesca Due” – il giocatore successivo nel senso del gioco pesca due carte e passa il turno al suo successivo; ma, nel caso abbia anch'esso una carta “Pesca Due”, può giocarla costringendo al giocatore successivo a pescare due carte in più.
- La carta “Jolly” o “Cambio colore” – permette al giocatore di scegliere uno tra i quattro colori disponibili (rosso, verde, blu, giallo); può essere giocata in qualsiasi momento e su qualsiasi carta.
- La carta “Jolly Pesca Quattro” – è la carta più potente del gioco; costringe al giocatore successivo nel senso del gioco a pescare quattro carte e consente al giocatore che l'ha giocata di cambiare il colore del gioco.

1.3. Carte

La seguente tabella riporta le informazioni riguardanti le carte utilizzate nel gioco:

Nome	Colore	Valore	Punti	File	Immagine
“Cambio Colore”	Nero	0	50	00.png	
“Pesca Quattro”	Nero	1	50	01.png	
“0 Blu”	Blu	0	0	10.png	
“1 Blu”	Blu	1	1	11.png	
“2 Blu”	Blu	2	2	12.png	
“3 Blu”	Blu	3	3	13.png	
“4 Blu”	Blu	4	4	14.png	
“5 Blu”	Blu	5	5	15.png	

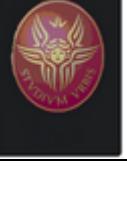
“6 Blu”	Blu	6	6	16.png	
“7 Blu”	Blu	7	7	17.png	
“8 Blu”	Blu	8	8	18.png	
“9 Blu”	Blu	9	9	19.png	
“Salta Blu”	Blu	10	20	110.png	
“Cambio Giro Blu”	Blu	11	20	111.png	
“Pesca Due Blu”	Blu	12	20	112.png	
“Blu” (carta finta)	Blu	13	-	113.png	
“0 Rosso”	Rosso	0	0	20.png	

“1 Rosso”	Rosso	1	1	21.png	
“2 Rosso”	Rosso	2	2	22.png	
“3 Rosso”	Rosso	3	3	23.png	
“4 Rosso”	Rosso	4	4	24.png	
“5 Rosso”	Rosso	5	5	25.png	
“6 Rosso”	Rosso	6	6	26.png	
“7 Rosso”	Rosso	7	7	27.png	
“8 Rosso”	Rosso	8	8	28.png	
“9 Rosso”	Rosso	9	9	29.png	

“Salta Rosso”	Rosso	10	20	210.png	
“Cambio Giro Rosso”	Rosso	11	20	211.png	
“Pesca Due Rosso”	Rosso	12	20	212.png	
“Rosso” (carta finta)	Rosso	13	-	213.png	
“0 Verde”	Verde	0	0	30.png	
“1 Verde”	Verde	1	1	31.png	
“2 Verde”	Verde	2	2	32.png	
“3 Verde”	Verde	3	3	33.png	
“4 Verde”	Verde	4	4	34.png	

“5 Verde”	Verde	5	5	35.png	
“6 Verde”	Verde	6	6	36.png	
“7 Verde”	Verde	7	7	37.png	
“8 Verde”	Verde	8	8	38.png	
“9 Verde”	Verde	9	9	39.png	
“Salta Verde”	Verde	10	20	310.png	
“Cambio Giro Verde”	Verde	11	20	311.png	
“Pesca Due Verde”	Verde	12	20	312.png	
“Verde” (carta finta)	Verde	13	-	313.png	

“0 Giallo”	Giallo	0	0	40.png	
“1 Giallo”	Giallo	1	1	41.png	
“2 Giallo”	Giallo	2	2	42.png	
“3 Giallo”	Giallo	3	3	43.png	
“4 Giallo”	Giallo	4	4	44.png	
“5 Giallo”	Giallo	5	5	45.png	
“6 Giallo”	Giallo	6	6	46.png	
“7 Giallo”	Giallo	7	7	47.png	
“8 Giallo”	Giallo	8	8	48.png	

“9 Giallo”	Giallo	9	9	49.png	
“Salta Giallo”	Giallo	10	20	410.png	
“Cambio Giro Giallo”	Giallo	11	20	411.png	
“Pesca Due Giallo”	Giallo	12	20	412.png	
“Giallo” (carta finta)	Giallo	13	-	413.png	
“Retro” (carta non giocabile)	-	-	-	back.png	

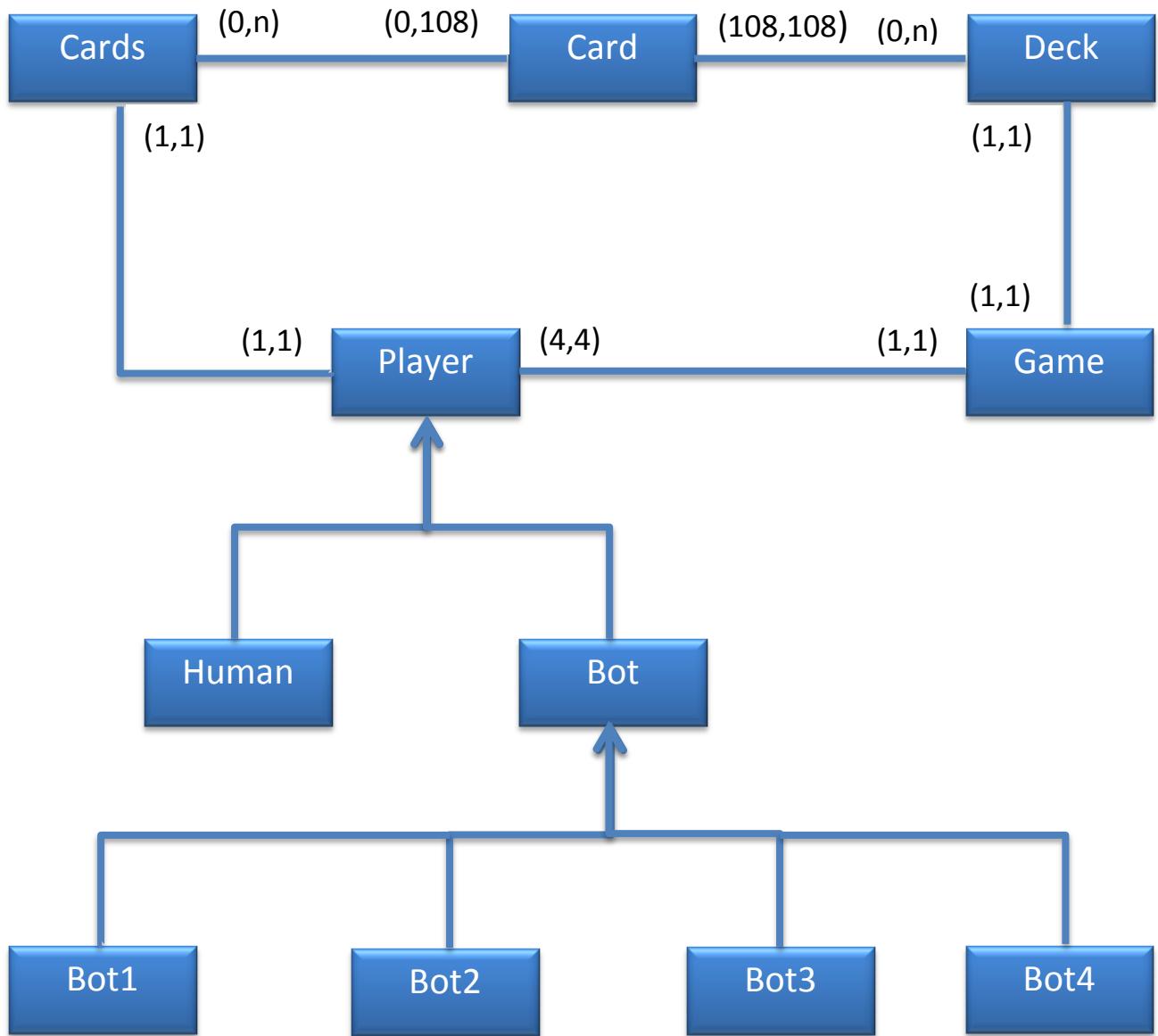
2. Modulo server

Il modulo server si occupa della gestione delle partite con più moduli client contemporaneamente. Per essere sempre operativo, è utilizzata una tipica architettura multithread: ogni nuova partita è servita da un nuovo thread. Ogni volta che c'è una partita in attesa di altri giocatori, il giocatore connesso è aggiunto alla partita. Se si connette un nuovo giocatore e non c'è nessuna partita, si crea un nuovo thread, che corrisponde ad una nuova partita.

Si suppone che la macchina sulla quale è eseguita l'applicazione server abbia notevoli risorse di calcolo per eseguire in tempi accettabili i compiti di elaborazione.

2.1. Analisi del modulo server

Il diagramma delle classi del server può essere schematizzato nel seguente disegno:



La classe `Card` (`card.Card`) rappresenta una carta da gioco con suo colore e valore. La classe `Cards` (`card.Cards`) è un insieme di Carte. La classe `Deck` (`card.Deck`) è un mazzo completo di carte, suddiviso in mazzo scarti e mazzo da giocare. La superclasse `Player` (`players.Player`) rappresenta un giocatore generico con nome, carte e numero di punti vinti. La classe `Human` (`players.Human`) rappresenta un giocatore umano. La superclasse `Bot` (`players.Bot`) rappresenta un giocatore automatico, che può essere `Bot1`, `Bot2`, `Bot3` o `Bot4` (i vari Bot sono descritti nel Capitolo 2.3).

Durante l'avvio, il server ha bisogno di 3 parametri:

- 1) *porta* – il numero di porta sulla quale il server sarà in ascolto per la connessione dei giocatori, per default è 11929 (le prime 5 cifre della mia matricola);
- 2) *bot* – la versione dei giocatori automatici, un valore fra 1 e 4, per default è 2;
- 3) *timer* – il numero di secondi che il server aspetta dalla connessione del primo giocatore e l'inizio della nuova partita, per default è 60 secondi.

Se, ad esempio, si vuole lanciare il server sulla porta 11930 con la versione 3 di Bot e con un tempo di attesa di 10 secondi, nel prompt dei comandi bisogna scrivere la seguente stringa:

```
java Server 11930 3 10.
```

Dopo l'avvio il server registra il suo indirizzo IP e la porta, sulla quale è in ascolto, su un server pubblico, che si trova all'indirizzo web <http://pacman.altervista.org>. Per fare questo bisogna semplicemente aprire una pagina PHP uno.php passandole due parametri:

- 1) password (per default è “server”)
- 2) numero di porta.

Se il numero di porta fosse 11929, allora la richiesta sarebbe:

```
http://pacman.altervista.org/uno.php?password=server&port=11929.
```

La pagina uno.php semplicemente memorizza la coppia indirizzo IP e numero di porta in un file di testo. Il metodo all'interno di uno.php è il seguente:

```
<?php  
if($password == server) {  
    $file = fopen('uno.txt', 'w');
```

```

        fwrite($file,
$_SERVER['REMOTE_ADDR']."\n".$port);

fclose($file);

}

$file = fopen('uno.txt', 'r');

$ip = fgets($file);

$port = fgets($file);

fclose($file);

echo $ip.$port;

?>

```

Poi è lanciato un nuovo thread della classe `threads.Waiting`, che si occupa della gestione degli utenti connessi. Ogni volta che si connette un nuovo giocatore parte un timer che aspetta un certo intervallo di tempo (per default, 60 secondi). Se prima dello scadere del timer si collegano altri 3 giocatori, la partita inizia; altrimenti, i giocatori mancanti vengono sostituiti dai giocatori virtuali automatici (Bot).

Dopo il completamento della fase d'inizializzazione parte il corpo del gioco situato nella classe `threads.Game`, che gestisce l'ordine delle azioni che devono/possono essere fatte da ogni giocatore. Nel gioco ci sono 3 possibili azioni base:

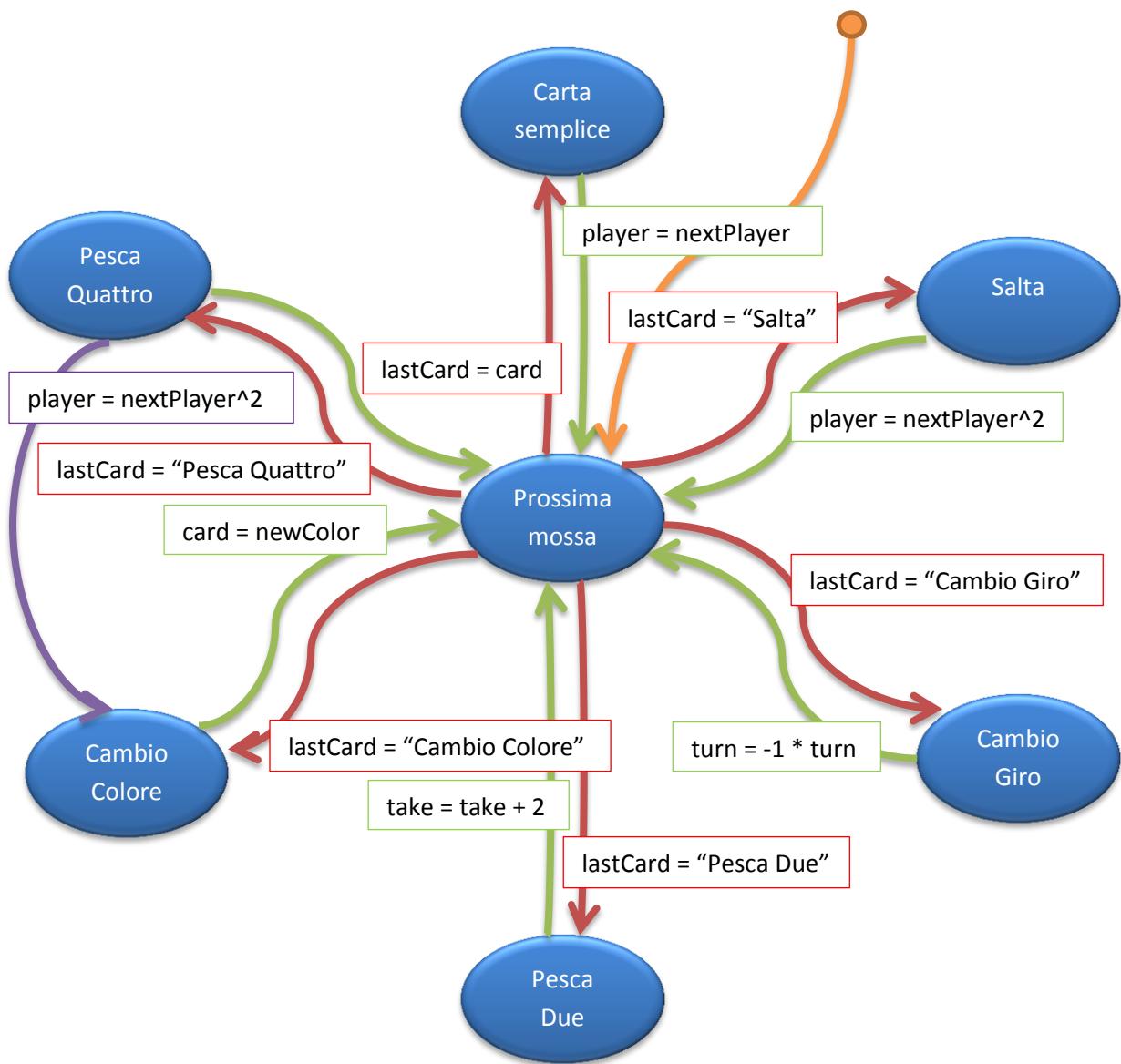
- 1) *prossima carta* – viene richiesta una carta tra le carte del giocatore coerente con l'ultima carta giocata;
- 2) *prossimo colore* – viene richiesto un colore tra rosso, blu, verde e giallo; questa azione è disponibile solo dopo aver giocato la carta “Pesca Quattro” oppure la carta “Cambia Colore”;
- 3) *più due* – viene richiesta la carta “Pesca Due”; questa azione è effettuata se il giocatore precedente ha giocato una carta “Pesca Due”.

Le fasi principali possono essere descritte nei seguenti punti:

- I. Si mischia il mazzo di carte.
- II. Ad ogni giocatore si inviano 7 carte estratte dal mazzo.
- III. Dal mazzo si estraе una carta, che sarà la prima carta del mazzo scarti.
- IV. Al giocatore, del quale è il turno, viene richiesta la prossima carta giocata.
- V. Se il giocatore non ha giocato nessuna carta, gli si invia una carta estratta dal mazzo e di nuovo si richiede una carta che il giocatore vuole giocare. Invece, se il giocatore ha giocato qualche carta, questa viene posta in cima al mazzo scarti e tutti gli altri giocatori vengono informati di questa mossa.
- VI. Lo scenario seguente dipende dal tipo della carta giocata:
 - i. se è una carta semplice con valore numerico da 0 a 9, il turno passa al giocatore successivo;
 - ii. se è una carta “Salta”, il giocatore successivo passa il turno e la prossima mossa sarà effettuata dal giocatore di fronte;
 - iii. se è una carta “Inverti giro”, si inverte il giro del gioco e la mossa successiva deve essere fatta dal giocatore che aveva il turno precedente;
 - iv. se è una carta “Pesca Due”, il giocatore successivo deve rispondere solo con un’altra carta “Pesca Due”, finché qualche giocatore non potrà rispondere e dovrà pescare dal mazzo pesca il totale delle carte accumulate;
 - v. se è una carta “Cambio Colore”, il giocatore sceglie il colore della prossima carta giocata e il turno passa al giocatore successivo;
 - vi. se è una carta “Pesca Quattro”, il giocatore sceglie il colore della prossima carta giocata e il giocatore successivo deve pescare 4 carte dal mazzo pesca e passare il turno.
- VII. Se il giocatore è rimasto con una sola carta e non ha detto “Uno！”, deve pescare 2 carte dal mazzo pesca.

- VIII. Se il giocatore è rimasto senza carte, si passa al calcolo del punteggio accumulato; tale punteggio sarà comunicato a tutti i giocatori.
- IX. Se nessun giocatore ha finito le carte, si passa al punto IV; altrimenti, si va al punto I.

L'insieme di queste fasi può essere descritto dal seguente diagramma degli stati:



2.2. Interfaccia grafica del modulo server

All'inizio si controlla se sul computer, dove si vuole far partire il server, non è già attivo un altro server del gioco. Per fare questa verifica si controlla la porta sulla quale il server deve essere in ascolto. Se la porta è occupata, viene visualizzato il messaggio di errore (in Figura 2.1).

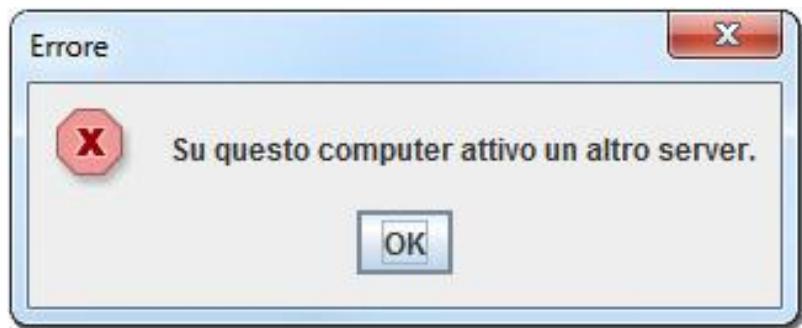


Figura 2.1: Messaggio di errore

Dopo la prima fase d'inizializzazione viene aperta una semplice finestra (implementata dalla classe `frames.FrameServer`) che visualizza le proprietà generali del server. In questa finestra si possono consultare le seguenti informazioni (Figura 2.2):



Figura 2.2: Finestra del server

- 1) *versione* – la versione del server;
- 2) *indirizzo* – l'indirizzo IP del computer sul quale sta girando l'applicazione server, può essere globale o locale;
- 3) *porta* – il numero di porta sulla quale il server è in ascolto per la connessione dei giocatori;
- 4) *bot* – la versione dei giocatori automatici;
- 5) *attivo* – la data e l'ora nelle quali è stato avviato il server;
- 6) *giocatori* – il numero di utenti che stanno giocando sul server;
- 7) *partite* – il numero di partite che si stanno svolgendo sul server.

2.3. Giocatori Bot

I giocatori automatici o Bot, che sono istanze di una classe astratta `player.Bot` (che implementa un'altra superclasse astratta `player.Player`), sono stati realizzati per permettere agli utenti di giocare senza dover aspettare tempi inaccettabili. Infatti, se entro un tempo prefissato (per default 60 secondi) non si connettono 4 giocatori remoti, allora i giocatori mancanti vengono sostituiti dai giocatori Bot. Ci sono 4 versioni di giocatori Bot che si distinguono per gli algoritmi della scelta della prossima mossa.

I giocatori automatici sono in grado di eseguire 3 azioni base:

- 1) *prossima carta* – viene richiesta una carta tra le carte del giocatore coerente con l'ultima carta giocata;
- 2) *prossimo colore* – viene richiesto un colore tra rosso, blu, verde e giallo; questa azione è disponibile solo dopo aver giocato la carta “Pesca Quattro” oppure la carta “Cambia Colore”;
- 3) *più due* – viene richiesta la carta “Pesca Due”; questa azione è effettuata se il giocatore precedente ha giocato una carta “Pesca Due”.

2.3.1. Bot v.1

Il primo Bot (implementato dalla classe `player.Bot1`) è il più semplice di tutti. Infatti, gioca sempre la carta con il valore più alto tra quelle che ha in mano. In questo modo cerca di rimanere con un numero di punti più basso possibile. Così, se la partita dovesse terminare, fa guadagnare meno punti all'avversario. Anche nella scelta del colore seleziona il colore della carta con valore più alto. Quest'algoritmo è “pessimistico” perché non costruisce una strategia che lo porti verso la vittoria, ma semplicemente cerca di ritardare la vittoria degli altri giocatori.

Ad esempio, se Bot1 possiede queste carte:



e l'ultima carta del mazzo scarti è



La carta giocata da Bot1 sarà



2.3.2. Bot v.2

Questo Bot (implementato dalla classe `player.Bot2`) è basato sull'algoritmo che costruisce una catena di carte tra quelle in mano, dove le carte adiacenti sono coerenti tra di loro, in modo tale che la somma dei valori delle carte di questa catena sia il più grande possibile. Quest'algoritmo si basa sulla speranza che nessuno degli avversari cambi il colore o il valore dell'ultima carta giocata, e dopo il giro completo, quando il turno ritorna a questo bot, continua a giocare la prossima carta della catena. Anche la scelta del prossimo colore utilizza quest'algoritmo, prendendo semplicemente il colore della seconda carta nella catena.

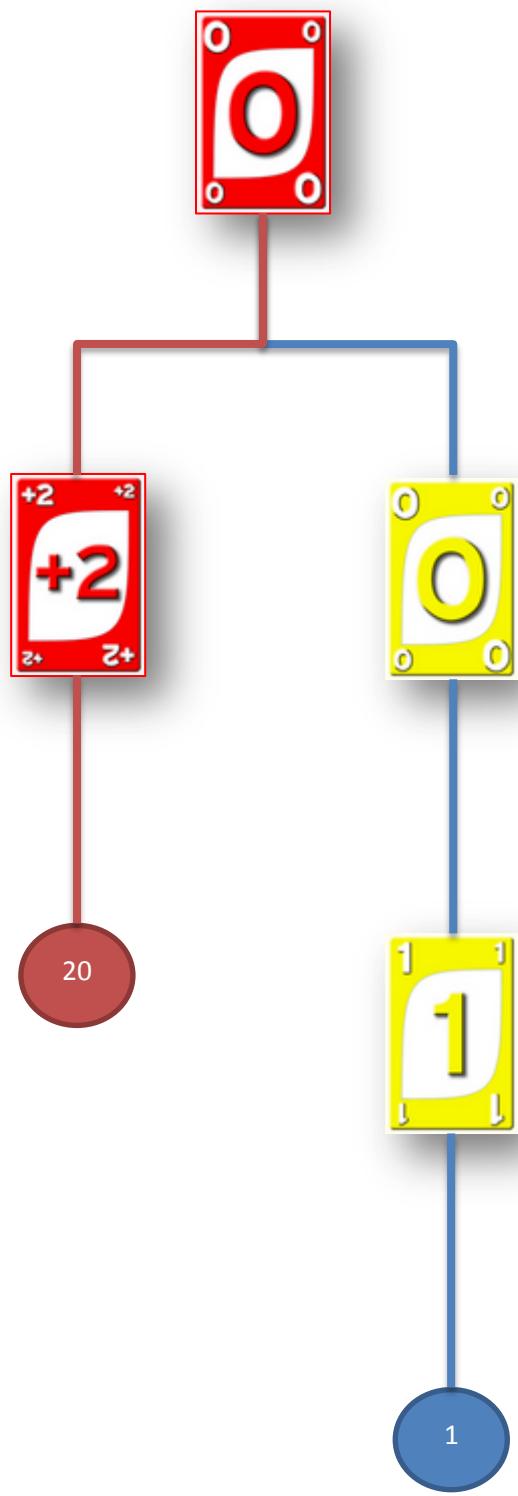
Ad esempio, se Bot2 possiede queste carte:



e l'ultima carta del mazzo scarti è



Il Bot2 costruisce un albero, dove la radice è l'ultima carta del mazzo scarti, i nodi intermedi sono le carte che Bot ha in mano e le foglie contengono il numero di punti di ogni ramo. In questo caso l'albero sarà:



La carta giocata da Bot2 sarà



2.3.3. Bot v.3

Questo Bot (implementato dalla classe `player.Bot3`) è fratello del Bot2. Anch'esso si basa sull'algoritmo che costruisce una catena di carte tra quelle in mano, dove le carte adiacenti sono coerenti tra di loro, però in modo tale che il numero delle carte di questa catena sia il più grande possibile. Anche quest'algoritmo si basa sulla stessa speranza che nessuno degli avversari cambi il colore o il valore dell'ultima carta giocata, e dopo il giro completo, quando il turno ritorna a questo bot, continua a giocare la prossima carta della catena. Analogamente avviene la scelta del prossimo colore, si prende il colore della seconda carta nella catena.

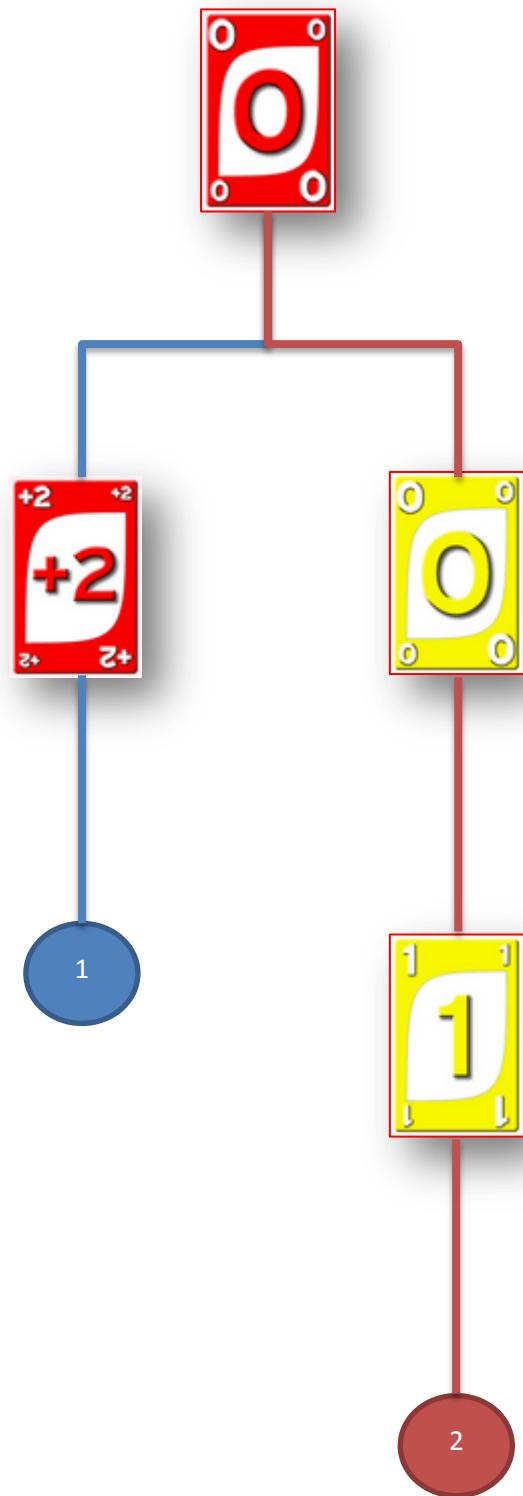
Ad esempio, se Bot3 possiede queste carte:



e l'ultima carta del mazzo scarti è



Il Bot3 costruisce un albero, dove la radice è l'ultima carta del mazzo scarti, i nodi intermedi sono le carte che Bot ha in mano e le foglie contengono il numero di carte di ogni ramo. In questo caso sarà:



La carta giocata da Bot3 sarà



2.3.4. Bot v.4

L'ultimo Bot (implementato dalla classe `player.Bot4`), a differenza di tutti i precedenti, utilizza una memoria aggiuntiva. Questa memoria gli serve per ricordare le carte che sono già uscite. Infatti, questo gli permette di scegliere la prossima carta da giocare con la probabilità relativa più alta, in modo tale che nessun avversario possa rispondere ad essa. Per calcolare questa probabilità si usa la seguente formula:

$$\text{probabilità}(carta) = 13 \cdot \text{colore}(carta) + 4 \cdot \text{valore}(carta)$$

dove $\text{colore}(carta)$ è il numero di carte uscite che hanno lo stesso colore della *carta* e $\text{valore}(carta)$ è il numero di carte uscite che hanno lo stesso valore della *carta*.

Ad esempio, se le carte del Bot4 sono:



le carte che sono uscite sono:

Colore	Blue			Verde			Rosso			Giallo			
Carte uscite	6			3			2			2			
Valore	0	1	2	3	4	5	6	7	8	9	S. ¹	C. ²	+2 ³
Carte uscite	2	0	1	0	1	0	1	0	2	1	3	1	1

l'ultima carta del mazzo scarti è



Calcoliamo la probabilità per ogni carta:

$$\text{probabilità(5)} = 13 \cdot \text{colore(5)} + 4 \cdot \text{valore(5)} = 13 \cdot 6 + 4 \cdot 0 = 78$$

$$\text{probabilità(S.)} = 13 \cdot \text{colore(S.)} + 4 \cdot \text{valore(S.)} = 13 \cdot 2 + 4 \cdot 3 = 38$$

$$\text{probabilità(C.)} = 13 \cdot \text{colore(C.)} + 4 \cdot \text{valore(C.)} = 13 \cdot 2 + 4 \cdot 1 = 30$$

$$\text{probabilità(0)} = 13 \cdot \text{colore(0)} + 4 \cdot \text{valore(0)} = 13 \cdot 3 + 4 \cdot 2 = 47$$

$$\text{probabilità(0)} = 13 \cdot \text{colore(0)} + 4 \cdot \text{valore(0)} = 13 \cdot 2 + 4 \cdot 2 = 34$$

La carta che ha la probabilità più alta che nessun avversario possa rispondere è la carta “5 Blu”, ma questa carta non può essere giocata secondo le regole del gioco. In seconda posizione c’è la carta “0 Verde” che può essere giocata. Di conseguenza la carta giocata dal Bot4 sarà:



¹ S. indica la carta “Salta”.

² C. indica la carta “Cambio Giro”.

³ +2 indica la carta “Pesca Due”.

2.3.5. Bot v.5

Questo Bot doveva essere un giocatore automatico senza un algoritmo delle mosse prestabilito, ma che fosse in grado d'imparare durante il suo funzionamento, memorizzando tutte le mosse migliori in un database. Ogni volta che il Bot doveva effettuare una mossa, consultava il database cercando la mossa più adatta.

Ma è sorto un problema: la quantità dei dati sarebbe stata enorme. Per verificare questo, bastano alcuni semplici calcoli e considerazioni. Una carta è caratterizzata da 2 informazioni: colore e valore. Il colore può assumere 5 valori: blu, verde, rosso, giallo e nero; di conseguenza per rappresentare questa informazione ci servono 3 bit ($2^3 = 8 > 5$). Il valore di una carta può essere di 14 tipi: da 0 a 9, "Salta", "Cambio Giro", "Pesca Due" e una carta vuota; perciò per memorizzare questo dato abbiamo bisogno di 4 bit ($2^4 = 16 > 14$). Allora una carta può essere rappresentata da un byte (in realtà, una carta è rappresentata da 2 int che sono 8 byte). Mediamente¹ un giocatore ha in mano contemporaneamente 5 carte. Una mossa è caratterizzata dall'ultima carta giocata (quella in cima al mazzo scarti), dall'insieme delle carte che il giocatore ha a disposizione e dalla carta giocata. Allora una mossa media può essere descritta da 7 carte oppure, in termini di memoria, da 7 byte. Mediamente¹ in una partita un giocatore fa 121 mosse. Da questo segue che per memorizzare tutte le mosse in una sola partita di un solo giocatore è necessario quasi un kilobyte di spazio. Per accumulare abbastanza informazioni, bisognerebbe salvare molte e molte partite, il che richiederebbe molta memoria.

Un altro problema è quello riguardante il tempo per trovare nel database la mossa ottima.

Per questi motivi (per ora) si è deciso di abbandonare questa idea. E Bot5 non è stato implementato.

¹ I valori sono stati calcolati su 1000 partite simulate.

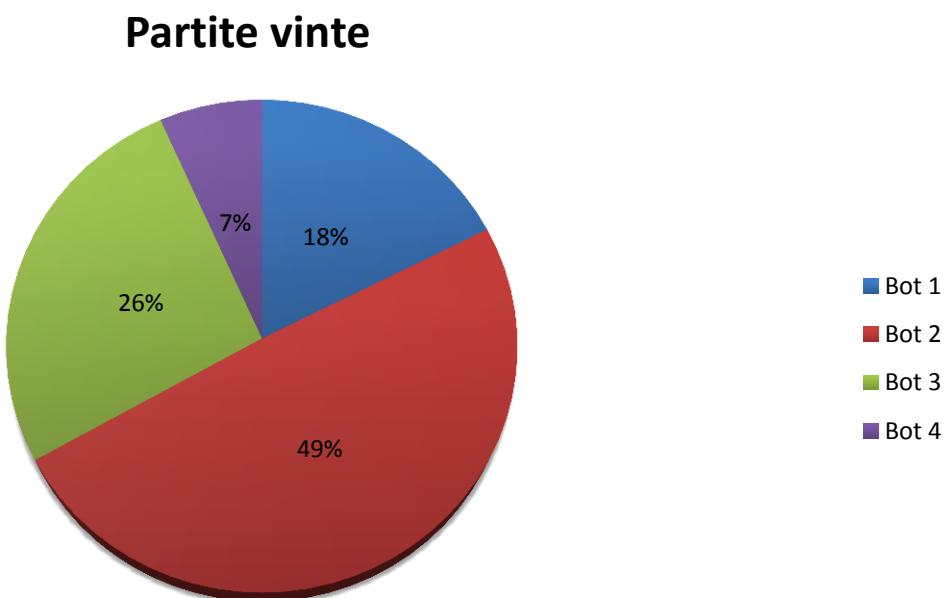
2.3.6. Confronto tra i Bot

I quattro Bot sono implementati da diversi algoritmi, di conseguenza le loro caratteristiche sono differenti. Per mettere a confronto i vari Bot si è organizzato un torneo composto di una serie di partite. Ogni giocatore era rappresentato da un Bot differente. Nella seguente tabella sono riassunte le proprietà principali di ogni Bot:

Proprietà	Bot 1	Bot 2	Bot 3	Bot 4
<i>Costo in tempo</i>	$O(n)$	$O(n!)$	$O(n!)$	$O(n)$
<i>Costo in spazio</i>	$O(1)$	$O(n!)$	$O(n!)$	$O(1)$
<i>Memoria aggiuntiva</i>	-	-	-	$O(1)$
<i>Tempo di risposta medio¹</i> (ms)	≈ 0	0,1	78	≈ 0
<i>Tempo di risposta massimo¹</i> (ms)	1	12724	163991 ²	1

n indica il numero di carte possedute dal giocatore Bot.

Il seguente diagramma indica la percentuale delle partite vinte³ da ogni Bot:



Il Bot2 si è dimostrato molto efficiente, ma può avere dei tempi di risposta inaccettabili. Per ovviare a questo problema si deciso di imporre un time-out di 8 secondi. Il Bot2 è il giocatore virtuale di default.

² La situazione si è verificata con 34 carte in mano.

³ I valori sono stati calcolati su 1000 partite simulate.

3. Modulo client

Il modulo client è l'applicazione con la quale interagisce il giocatore. Il programma client non mantiene lo stato del gioco, ma semplicemente interpreta le istruzioni ricevute dal server e le mostra all'utente. D'altra parte, il modulo client riceve i comandi dal giocatore e utilizzando un protocollo di comunicazione (per maggiori informazioni consultare Capitolo 4) li invia al server.

3.1. Interfaccia grafica del modulo client

Dopo l'avvio del programma “Uno” client, si apre la finestra (implementata dalla classe `frames.FrameMain`) con menu principale, dove si può scegliere fra 3 possibili azioni cliccando sui corrispettivi pulsanti (Figura 3.1):

- 1) “*Inizia*” – inizia la nuova partita;
- 2) “*Informazioni*” – visualizza informazioni sul gioco;
- 3) “*Esci*” – esce dal gioco.



Figura 3.1: Menu principale

Cliccando sul pulsante “Esci” il programma termina.

Cliccando sul pulsante “Informazioni” si apre una nuova finestra (implementata dalla classe `frames.FrameInformations`) dove sono riportate le informazioni generali sul programma (Figura 3.2).



Figura 3.2: Finestra delle informazioni

Dalla finestra di informazioni è possibile visualizzare anche le regole del gioco, cliccando sul pulsante “Visualizza le regole”.

Cliccando sul pulsante “Inizia” si apre una nuova finestra (implementata dalla classe `frames.FrameInit`) di caricamento del gioco. Immediatamente dopo si apre una finestra di dialogo, dove bisogna inserire il proprio nickname (Figura 3.3).



Figura 3.3: Dialogo di inserimento del nickname

A questo punto viene cercato un server disponibile per giocare. Ci sono 4 metodi di ricerca, se un metodo fallisce, si passa a quello successivo:

- 1) Sul server globale <http://pacman.altervista.org> si legge il file `uno.txt` che contiene una lista di tutti i server disponibili con relativi indirizzi IP e porte. Il client cerca di connettersi a uno di questi server.
- 2) Un host scanner cerca un server del gioco nella rete locale `192.168.0.0/23` (cioè dall'indirizzo `192.168.0.1` all'indirizzo `192.168.1.254`) sulla porta di default `11929`. Se viene identificato un tale server, ci si connette.
- 3) Si verifica se il server non è attivo su `localhost` all'indirizzo `127.0.0.1` sulla porta `11292`.
- 4) Si chiede all'utente di inserire l'indirizzo IP e la porta del server del gioco.

Se la connessione al server avviene con successo, nella finestra di caricamento viene visualizzato lo stato di caricamento (Figura 3.4). In realtà, non è un vero e proprio caricamento, ma solo un'attesa di altri giocatori. Infatti, la percentuale del gioco caricato viene calcolata con la formula

$$\text{caricato}(p, t) = \frac{100}{3} \cdot (p - 1) + \frac{1}{10} \cdot \frac{t}{\bar{T}} \cdot \left(1 - \frac{p - 1}{3}\right)$$

dove p è il numero di giocatori che si sono uniti al gioco, t è il tempo trascorso dal collegamento del primo giocatore e \bar{T} è il periodo di attesa di altri giocatori.

Il caricamento è suddiviso in 10 fasi, così denominate:

- ricerca dello spazio libero;
- edificazione della sala;
- costruzione del tavolo;
- invito dei partecipanti;
- sistemazione delle sedie;
- stampa delle carte;
- accoglienza dei giocatori;

- collocazione degli spettatori;
- addestramento degli avversari;
- mescolamento del mazzo.



Figura 3.4: Finestra di caricamento

Quando il caricamento è terminato, si passa alla fase principale del gioco (in Figura 3.5). Viene visualizzata la finestra (implementata dalla classe `frames.FrameClient`) che rappresenta il tavolo del gioco. Anche questa finestra, come tutte le altre, è stata costruita con l'ausilio della classe `GroupLayout`. Questa finestra si colloca sempre al centro dello schermo e si adatta automaticamente alle dimensioni del monitor, occupando 90% dello spazio in altezza e larghezza. Se, ad esempio, la risoluzione dello schermo è 1024x600 pixel, la dimensione della finestra sarà 922x540 pixel. La dimensione minima della finestra potrà essere 540x540 pixel, cioè la risoluzione dello schermo per supportare il gioco deve essere almeno 600x600 pixels. Se la finestra fosse più piccola, non ci

sarebbe abbastanza spazio per disporre in modo corretto e elegante le carte.



Figura 3.5: Finestra principale del gioco sullo schermo con risoluzione 1024x600

Si può paragonare la visualizzazione della finestra principale su due monitor con risoluzione diversa: uno che ha la risoluzione 1024x600 pixel è mostrato in Figura 3.5 e uno con la risoluzione 1920x1080 è mostrato in Figura 3.6.

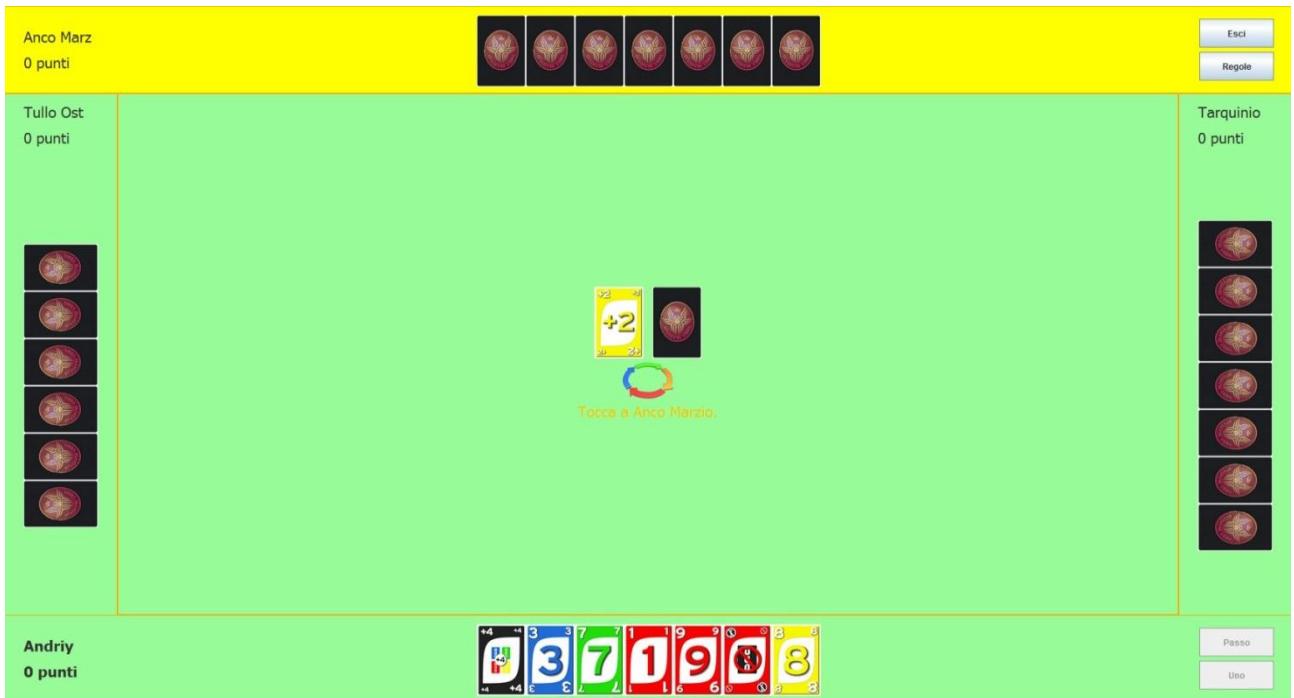


Figura 3.6: Finestra principale del gioco sullo schermo con risoluzione 1920x1080

La finestra è suddivisa in 5 aree mostrate in Figura 3.7:

- 1) area nord – contiene le informazioni (nome e punti guadagnati) dell'avversario di fronte e la quantità di carte che lui possiede, inoltre ci sono due pulsanti “Esci” e “Regole”;
- 2) area ovest – contiene le informazioni (nome e punti guadagnati) dell'avversario a sinistra e la quantità di carte che lui possiede;
- 3) area centro – contiene l'ultima carta giocata, il pulsante per pescare una carta (equivalente al pulsante “Passo”), l'indicatore del senso di gioco (orario o antiorario) e le informazioni sullo stato del gioco;
- 4) area est – contiene le informazioni (nome e punti guadagnati) dell'avversario a destra e la quantità di carte che lui possiede;
- 5) area sud – contiene le informazioni (nome e punti guadagnati) del giocatore e le carte che lui possiede, inoltre ci sono due pulsanti “Passo” e “Uno”.



Figura 3.7: Suddivisione della finestra principale in aree

Cliccando sul pulsante “Esci” il gioco finisce e si ritorna nel menu principale.

Cliccando sul pulsante “Regole” appare una serie di finestre (implementate dalle classi frames.FrameRules1,

frames.FrameRules2,
 frames.FrameRules3,
 frames.FrameRules4, frames.FrameRules5) che riassumono i principi e l'obiettivo del gioco (Figura 3.8, Figura 3.9, Figura 3.10, Figura 3.11, Figura 3.12):



Figura 3.8: Regole del gioco – Le carte



Figura 3.9: Regole del gioco – Obiettivo del gioco

Procedimento del gioco

Le carte vengono mescolate e distribuite 7 a testa. La porzione del mazzo rimanente, detta "mazzo pesca", viene posata sul tavolo a dorso coperto e la prima carta viene girata, andando così a formare il "mazzo scarti". Il primo giocatore deve scartare una delle proprie carte a disposizione, compatibile con quella in cima al mazzo scarti per colore o per numero, oppure può usare una carta Azione sempre compatibile con la carta del mazzo scarti. Se il giocatore non ha a disposizione una carta da scartare, è l'obbligo a pescarne una dal mazzo pesca; se la carta pescata può essere giocata, lo può fare immediatamente, altrimenti il turno passa al giocatore successivo. Il gioco procede in senso orario, finché non è cambiato dalla carta "Cambio Giro".

Mentre un giocatore scarta la penultima carta rimasta, deve pronunciare "UNO!". Se il giocatore non esclama "UNO!", deve pescare 2 carte dal mazzo pesca.

Una volta che uno dei giocatori rimane senza carte, si procede al calcolo del punteggio che ha conseguito. Nel caso in cui l'ultima carta giocata era un "Pesca Due" o un "Pesca Quattro" il giocatore seguente deve comunque prendere il numero di carte corrispondenti che andranno poi a sommarsi con gli altri punti.

[Precedente](#) [Indietro](#) [Successivo](#)

Figura 3.10: Regole del gioco – Procedimento del gioco

Calcolo del punteggio

Il primo giocatore che non ha più carte in mano vince e guadagna un certo numero di punti calcolati sulla base delle carte rimanenti agli altri giocatori. Il punteggio viene calcolato sommando i valori delle carte:

- le carte numeriche valgono il valore della carta,
- "Inverti", "Salta" e "Pesca Due" - 20 punti,
- "Cambio Colore" e "Pesca Quattro" - 50 punti.

The screenshot shows a Windows-style application window titled "Regole del gioco". The main content area is titled "Calcolo del punteggio". It contains a list of scoring rules and two examples of how to calculate scores based on the cards remaining in play.

Scoring Rules:

- le carte numeriche valgono il valore della carta,
- "Inverti", "Salta" e "Pesca Due" - 20 punti,
- "Cambio Colore" e "Pesca Quattro" - 50 punti.

Example Calculations:

Two sets of cards are shown with their calculated scores:

- Top Row:** A black card with four "+4" symbols, a blue card with a "Skipped" symbol, a green card with a "5", and a red card with a "Turn Around" symbol. The calculation is: $= 50 + 20 + 5 + 20 = 95$.
- Bottom Row:** A blue card with two "0"s, a blue card with a "+2" symbol, and two yellow cards, each with a "9". The calculation is: $= 0 + 20 + 9 + 9 = 38$.

[Precedente](#) [Indietro](#) [Successivo](#)

Figura 3.11: Regole del gioco – Calcolo del punteggio

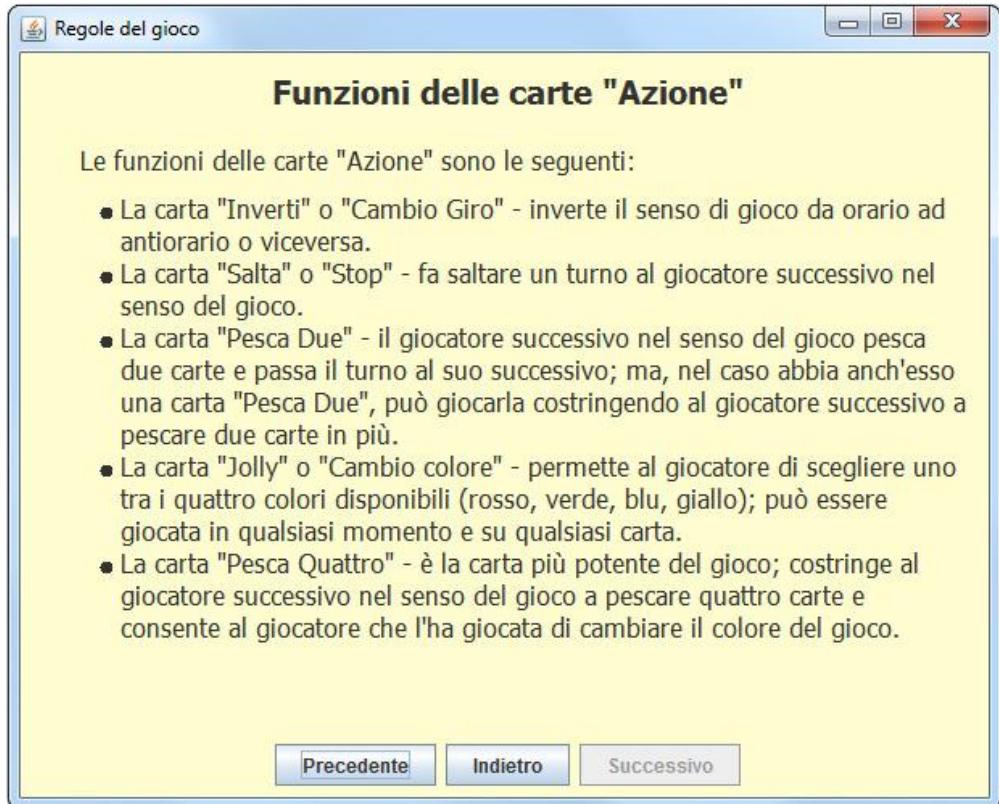


Figura 3.12: Regole del gioco – Funzioni delle carte “Azione”

Se durante la partita si verifica qualche problema con la connessione al server, la partita termina visualizzando il messaggio di errore (Figura 3.13).



Figura 3.13: Messaggio di errore

4. Interazione tra client e server

Il server e il client devono continuamente “parlare” in tempo reale. Per permettere questa comunicazione è stato realizzato un protocollo. Ogni messaggio di questo protocollo di comunicazione è composto da due parti:

- 1) head – contiene il numero dell’operazione da eseguire;
- 2) parametri – informazioni necessarie per eseguire una certa operazione.

Nella seguente tabella sono descritte le operazioni che possono essere richieste con relativi parametri e risultato dell’operazione:

Operazione	Head	Parametri	Risultato
<i>Prossima carta</i>	1	-	carta : Card – carta giocata
<i>Prossimo colore</i>	2	-	colore : int – colore scelto
<i>Prossima carta “Pesca Due”</i>	3	-	carta : Card – carta “Pesca Due” giocata
<i>Uno</i>	4	-	uno : boolean – indica se giocatore ha detto “Uno”
<i>Invia carta</i>	5	• carta : Card – carta da inviare	-
<i>Invia punti</i>	6	• giocatore : int – posizione ¹ del giocatore che ha guadagnato i punti • punti : int – numero di punti guadagnati	-
<i>Carta giocata</i>	7	• carta : Card – carta giocata	-
<i>Invia prossimo</i>	8	• giocatore : int – posizione ¹ del prossimo giocatore	-

<i>Aggiungi carte</i>	9	<ul style="list-style-type: none"> • giocatore : int – posizione¹ del giocatore che riceve le carte • carte : int – numero di carte ricevute 	-
<i>Cambia turno</i>	10	<ul style="list-style-type: none"> • verso : int – verso² del gioco 	-
<i>Gioco finito</i>	11	<ul style="list-style-type: none"> • giocatore : int – posizione¹ del giocatore che ha vinto 	-
<i>Invia caricato</i>	12	<ul style="list-style-type: none"> • percentuale : int – percentuale del gioco caricato 	
<i>Invia nome</i>	13	<ul style="list-style-type: none"> • nome : String – nome dell'avversario 	
	14	<ul style="list-style-type: none"> • 	

¹ Le posizioni vengono contate iniziando dal giocatore stesso e procedendo in senso orario. Di conseguenza, 0 è il giocatore stesso, 1 – l'avversario a sinistra, 2 – l'avversario di fronte, 3 – l'avversario a destra.

² 1 indica il verso orario e -1 indica il verso antiorario.

5. Conclusione

I videogiochi sono una famiglia di programmi molto importanti nel mondo virtuale. Oltre ad essere uno stimolo per il nostro cervello, sono anche un modo per intrattenere contatti con altri esseri umani appartenenti a culture e mentalità diverse dalle nostre; in questo senso il gioco può essere visto come collante per le comunità, come mezzo per scoprire e capire paesi molto lontani da noi.

Anche dal punto di vista economico, specialmente il gioco d'azzardo, è una delle principali fonti di guadagno per le città come Las Vegas e Montecarlo, che, purtroppo, però favoriscono anche il diffondersi delle pratiche illegali nell'ambito del ludo.

Infine non va mai dimenticato che il gioco è una medaglia a due facce: assieme a tutte le sue caratteristiche positive, vanno considerate anche quelle negative, come la dipendenza (assai diffusa tra i giovani d'oggi) o i problemi economici e le difficoltà nei rapporti interpersonali che potrebbe causare. Sta soprattutto ai genitori educare i propri figli in modo che non si facciano ingannare dalla realtà virtuale dei giochi e farsi assorbire da essi: è sempre più facile immedesimarsi nei personaggi dei Videogames grazie alla grafica perfezionata e alle accattivanti trame e dimenticare i problemi e le difficoltà reali per vivere in un mondo d'illusioni che, per di più, non insegna niente di buono.

Non è forse vero che il gioco perde sempre più pregi per acquisire solo difetti? Forse la tecnologia interferisce e contamina i benefici naturali del gioco... In questo caso dovremmo fare un passo indietro e ritornare ai buoni vecchi giochi di una volta...