



# TombolaStudio

## Istruzioni per l'uso

Versione 1.1.00

Versione del documento: 1.1.00

Data di Pubblicazione: 21 Giugno 2020

*A chi piaceva molto  
giocare a Tombola,  
ha giocato tanto,  
ha fatto giocare divertendo.*

*Bisogna avere il caos dentro di sé per partorire una stella danzante.*

## Sommario

Introduzione ( <i>versione breve</i> ).....	4
Informazioni sul documento .....	7
Riepilogo delle versioni .....	7
Contenuto dei prossimi capitoli.....	7
1. Prima di iniziare: prerequisiti ed installazioni.....	9
Prerequisiti .....	9
Installazione di Java.....	9
Installazione di TombolaCards e TombolaPrint .....	10
2. TombolaCards: generare cartelle.....	14
Anatomia di una Cartella .....	14
Le Serie di cartelle.....	15
I numeri Jolly! .....	15
Modalità “a riga di comando” .....	16
Riepilogo dei parametri “a riga di comando” .....	20
I formati dei file di cartelle .....	26
Modalità “interattiva” .....	29
Riepilogo dei comandi in modalità interattiva .....	33
Riepilogo delle variabili di ambiente in modalità interattiva .....	35
Il “problema” delle vincite concomitanti .....	39
Il momento della verità .....	44
3. TombolaPrint: stampare cartelle .....	47
Principi di funzionamento .....	47
Finalmente si stampa .....	48
Riepilogo dei parametri a “riga di comando” .....	52
I modelli di stampa visti da vicino .....	53
Pasticciamo un po’ .....	54
Tipi di template disponibili .....	54
Elenco dei tag supportati.....	54
4. TombolaWeb: L’ombelico di TombolaStudio (Coming soon) .....	55
5. TombolaLib: per programmatori o aspiranti tali.....	56

## Introduzione (*versione breve*)

Benvenuto! Questa è la documentazione di un progetto personale, un classico *pet project* realizzato in pochi momenti liberi ed in un periodo di tempo molto lungo, *indecentemente* lungo.

**TombolaStudio** è una raccolta di diversi *strumenti software* che supportano il tradizionale gioco della Tombola<sup>1</sup> e permettono di:

- **Preparare le cartelle**, ossia ‘generare’ in modo casuale disposizioni di numeri valide per formare le cartelle della Tombola. Tale funzionalità è particolarmente utile quando si vogliono organizzare tombolate cui partecipa un elevato numero di giocatori, per cui si rendono necessarie tante cartelle, *il più possibile diverse tra loro* (riprenderemo più avanti questo tema). Una volta generate, le cartelle possono essere salvate su file e quindi archiviate, inviate via e-mail, passate su chiavetta usb, etc.
- **Stampare su carta le cartelle generate**. Per giocare a Tombola *in modo tradizionale* occorre avere le cartelle stampate, su carta o cartoncino, per distribuirle fisicamente ai vari giocatori. È utile poi poter stampare le cartelle con una veste grafica gradevole, personalizzando aspetti come loghi ed immagini, il numero di cartelle stampate su ciascun foglio, il tipo di caratteri utilizzato, i bordi, i colori...
- **Rendere più divertenti le partite a Tombola**, quando a giocare si è in tanti. Questo è il pezzo forte della vicenda. Un modulo software di TombolaStudio consente infatti di:
  - **Aiutare** gli organizzatori di “tombolate” con molti giocatori a gestire le varie fasi del gioco: vendita/distribuzione delle cartelle, estrazione dei numeri e visualizzazione del tabellone, verifica delle vittorie, arbitraggio di vincite concorrenti, ...
  - **Contribuire al divertimento** dei partecipanti, visualizzando informazioni e curiosità di vario tipo durante il gioco, come ad esempio il numero di giocatori che, ad un certo punto, *stanno aspettando un solo numero per fare tombola, oppure da quante estrazioni qualcuno sta aspettando un solo numero per far tombola...* cose così
  - **Archiviare e consentire l’analisi** di dati ed informazioni in relazione a più “tombolate” organizzate ed a più partite giocate (chi è il giocatore più fortunato delle vacanze di Natale 2020? Quale è la cartella che stasera ha vinto di più? E la cartella “maglia nera”? quella sulla quale in assoluto sono stati segnati meno numeri di tutte le altre?)

In realtà, non tutti i moduli software che servono per raggiungere questi obiettivi sono pronti. Alcuni sono disponibili, almeno in una loro prima versione, altri sono ancora troppo incompleti per essere resi pubblici. Come tutti i progetti software che si rispettino, anche **TombolaStudio** avrà aggiornamenti e rilasci progressivi nel tempo. Al paragrafo “*Riepilogo delle versioni*”, poco più avanti, sono indicate le componenti ancora mancanti ma pianificate per essere rilasciate a breve ed in genere le evoluzioni previste.

Ecco invece introdotti brevemente i moduli già disponibili:

- **TombolaCards**: è il programma da utilizzare per *generare cartelle*. In Internet si possono trovare tantissimi software che fanno questo mestiere, perché in effetti creare una cartella della tombola con 15 numeri casuali è una cosa molto semplice. Anche creare gruppi di 6 cartelle (e cioè *serie* di cartelle) utilizzando tutti e 90 i numeri della Tombola è cosa molto semplice. TombolaCards ha una funzionalità in più: *empiricamente* cerca di creare *insiemi numerosi* di cartelle che abbiano il

---

<sup>1</sup> Se non conosci il gioco della Tombola, consulta Wikipedia: <https://it.wikipedia.org/wiki/Tombola>

minor numero possibile di elementi uguali tra loro, *sperando*<sup>2</sup> così di *ridurre la probabilità di vincite concorrenti*. Avete presente quando si gioca in tanti e, ad esempio, all'estrazione di un numero ci sono tre persone che urlano insieme "TERNO!"? E poi bisogna trovare un modo, senza creare ulteriore confusione, per "dirimere la contesa" ed assegnare il premio? Ecco, TombolaCards prova in qualche misura ad affrontare questo tipo di problema.

È un programma "a riga di comando", di quelli che vanno eseguiti in una vecchia console a caratteri di Microsoft Windows™ oppure in una shell di Linux. Una volta generate, le cartelle sono salvate in semplici file di testo, così da poter essere utilizzate sia con gli altri moduli di questo progetto sia esternamente.

TombolaCards può essere utilizzato:

- **In modalità batch:** i parametri che si desiderano, come ad esempio il numero di serie di cartelle da generare, il nome del file in cui salvarle, etc., vengono aggiunti alla riga di comando. Dopodiché si batte [INVIO] sulla tastiera e... si aspetta che TombolaCards abbia finito.
- **In modalità interattiva:** avviando TombolaCards si entra in un *ambiente di lavoro* in cui l'utente può digitare comandi, aspettare i risultati prodotti ed agire di conseguenza. È possibile ad esempio iniziare a generare un primo insieme di cartelle, verificare che ci piacciono, eliminarne alcune o aggiungerne delle altre, salvarle in uno o più file, etc.
- **TombolaPrint:** è il modulo che consente di stampare su carta le cartelle della tombola, una volta che queste ultime sono state generate e salvate su file con TombolaCards. Anche TombolaPrint è un programma a riga di comando. A partire da un file di cartelle, TombolaPrint genera **file HTML** pronti per essere stampati con un qualsiasi moderno browser per Internet (Microsoft Edge, Google Chrome, Apple Safari, Firefox, ...). I file utilizzano la diffusissima tecnologia dei fogli di stile CSS per definire l'aspetto estetico delle cartelle che si vogliono stampare, come i font utilizzati, la dimensione ed il colore dei bordi, il posizionamento all'interno delle pagine, etc. TombolaPrint ha con sé una serie di "modelli" già pronti per stampare le cartelle senza doversi preoccupare di altro. I modelli sono anche un ottimo punto di partenza per realizzare, con poche modifiche, stampe personalizzate secondo le proprie esigenze e gusti estetici.
- **TombolaLib.** TombolaLib è un oggetto particolare, non è pensato per essere utile ad un *utente finale*, a qualcuno cioè che ha bisogno di generare cartelle, stamparle o vuole giocare a Tombola, ma è qualcosa che può interessare *i programmatori di computer*. Come è facile immaginare, infatti, sia TombolaCards che TombolaPrint utilizzano le stesse strutture dati (cartelle, serie di cartelle, etc.) e molte funzionalità in comune. Tutte queste funzionalità sono state raccolte insieme realizzando una *libreria di oggetti*, TombolaLib. In TombolaLib vi sono *classi* per gestire cartelle, serie di cartelle, insiemi di serie di cartelle, classi per simulare partite a Tombola, e tanto altro ancora. Con TombolaLib si vuole:
  - Creare una base comune di componenti software, che tutti i programmi del progetto possano utilizzare in modo coerente ed omogeneo, secondo i buoni propositi della programmazione ad oggetti;
  - Consentire anche ad altri di sviluppare programmi nuovi (e migliori!) e nuove funzionalità a partire dalle strutture dati e dagli algoritmi di base presenti in TombolaLib.

---

<sup>2</sup> Di sicuro l'utilizzo del termine "*sperando*" non sarà sfuggito alla tua attenzione. Qualche dettaglio in più sul tema è riportato nel box *Il "problema" delle vincite concomitanti*, a pagina 39.

Manca ancora all'appello **TombolaWeb**, il pezzo forte del progetto, ovvero l'applicazione web pensata come strumento di supporto durante il gioco della tombola. Ancora qualche settimana di pazienza...

Il diagramma seguente illustra sinteticamente i moduli del progetto, le relazioni tra di essi e lo stato di quanto completato ad oggi.

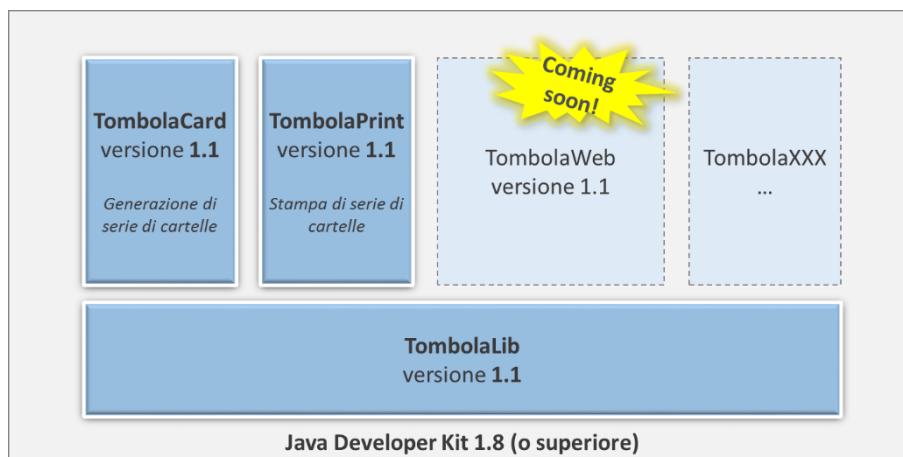


Figura 1 – Schema dei moduli che compongono il progetto.

Tutto il software è stato realizzato utilizzando il linguaggio di programmazione ad oggetti Java™. Java è il linguaggio che ho utilizzato di più all'università, per cui ci sono molto affezionato, anche se ultimamente sta attraversando un periodo non proprio facile... ma questa è un'altra storia. Mi piace poi l'idea che il software realizzato in Java possa essere utilizzato su più piattaforme (anche se dubito che **TombolaStudio** andrà molto in giro).

Programmo per hobby, mi guadagno da vivere di più con PowerPoint, Excel e Word che con editor e compilatori. Ho deciso però di pubblicare su GitHub il codice scritto e questo manuale perché mi piace pensare che possa esser utile anche a qualcun altro, possa aver senso in qualche modo *condividerlo*. È per questo stesso motivo che mi sono impegnato a commentarlo con un minimo di attenzione ed anche a cercare di seguire un po' di buone pratiche di programmazione ad oggetti, in modo che, se non ad altro, possa almeno servire come esercizio per chi si avvicina a questa stupenda forma di arte ed artigianato.



Salerno (Italy), settembre 2017

*Harry G. Potter*

## Informazioni sul documento

### Riepilogo delle versioni

La tabella riporta le versioni di questo documento già rilasciate su GitHub ed almeno una *prossima versione*, ovvero la versione del documento su cui si sta lavorando e quindi prossima al rilascio.

Versione	Stato	Data di Pubblicazione	Note
1.1.00	 <b>Pubblicato</b> (Versione Corrente)	21 giugno 2020	Prima versione di questo documento resa pubblica su Github. Accompagna il primo “rilascio pubblico” del software, che include: <ul style="list-style-type: none"><li>• TombolaLib 1.1</li><li>• TombolaCards 1.1</li><li>• TombolaPrint 1.1</li></ul>
1.1.50	 <b>Pianificato</b>	30/11/2020	<ul style="list-style-type: none"><li>• Revisione e completamento del capitolo dedicato a TombolaPrint</li><li>• Rilascio di TombolaWeb 1.1</li><li>• Rilascio della prima versione in inglese di questo manuale</li></ul>

### Contenuto dei prossimi capitoli

Questo “PDF” è da intendersi come compendio delle informazioni necessarie per utilizzare i “programmi” a supporto del gioco della Tombola di questo progetto. La documentazione è stata realizzata

- per supportare gli *utenti finali*, e cioè chi vorrà usare questi strumenti per giocare a Tombola, e quindi avrà bisogno di generare e stampare cartelle, organizzare le partite, etc.;
- per supportare qualche *informatico o programmatore* più curioso, che invece vorrà anche capire come sono stati realizzati questi programmi, vederne il codice sorgente e magari estenderli, migliorarli, utilizzarli per altre applicazioni.

Il [primo capitolo](#) è una guida all’installazione del software del progetto **TombolaStudio** su di un comune personal computer. Occorrerà assicurarsi di avere installato dei tool indispensabili (ad esempio una versione appropriata del Java Development Kit, JDK) e di aver scaricato da GitHub ed installato i vari moduli disponibili (TombolaCards, TombolaPrint, etc.). Come riferimento si prende un pc con sistema operativo Microsoft Windows 10, ma se sei un minimo smaliziato nulla vieta di installare gli analoghi componenti anche sulla distribuzione Linux che preferisci o anche su un Apple Mac™.

Al termine del primo capitolo sarai pronto per far funzionare gli strumenti di TombolaStudio senza alcun problema!

Il [secondo capitolo](#) invece è la guida all’utilizzo dello strumento **TombolaCards**. Con TombolaCards si generano file contenenti serie di cartelle. Sono possibili alcune varianti per farlo, e ci sono diverse opzioni che è possibile specificare: la scelta del formato dei file, il modo di etichettare le singole cartelle, ... Una volta letto il secondo capitolo sarai in grado di salvare in versatili file di testo tutte le cartelle che vorrai per giocare a Tombola.

Il [terzo capitolo](#) è dedicato a **TombolaPrint**, ovvero dello strumento che consente di generare file in formato html (quello delle pagine internet, per intenderci) contenente le cartelle preparate con TombolaCards. Una volta in formato html, le cartelle per la tombola possono facilmente essere stampate su carta utilizzando un qualsiasi browser web (Microsoft Edge, Internet Explorer, Google Chrome, Safari, etc.). Avrai anche la possibilità di personalizzare il *layout* (lo stile) delle tue cartelle.

Al termine del capitolo, avrai pronte, finalmente, fisicamente su carta, le cartelle per la Tombola!

Il [quarto capitolo](#) è bello grosso ([e per questo non ancora disponibile!](#)). È il manuale per utilizzare la componente Web di questo progetto, ovvero **TombolaWeb**, lo strumento che ti supporterà durante le partite a Tombola e ti permetterà di rendere “digitale” l’esperienza di gioco di intere tombolate. Con TombolaWeb è possibile organizzare comunità di giocatori ed eventi di gioco, gestire l’assegnazione (o la vendita) delle cartelle per le varie partite, monitorare l’andamento di una partita, mantenere statistiche, etc. etc. tantissima roba... tutta da scoprire.

Al termine del quarto capitolo potrai definirti Gran Maestro di **TombolaStudio**.

Cos’altro aggiungere? Buona lettura!

Se hai **feedback**, osservazioni, correzioni o integrazioni da proporre... scrivimi! Sarò felice di confrontarmi ed accettare ogni suggerimento per migliorare e far evolvere questo progetto (ehm... sempre a tempo perso ovviamente).

[harry.g.potter@gmail.com](mailto:harry.g.potter@gmail.com)



# 1. Prima di iniziare: prerequisiti ed installazioni

## Prerequisiti

Ecco quello che serve per utilizzare gli strumenti già disponibili oggi, TombolaCards e TombolaPrint:

- 1) Un **personal computer**. In questo manuale si farà riferimento (comandi, screenshot, etc.) al mio pc, su cui è installato il sistema operativo Microsoft Windows 10. I programmi di TombolaStudio, essendo realizzati con Java, possono funzionare anche su altre piattaforme: sistemi Linux, Unix, computer Apple con sistema operativo Mac OS. Non posso far tutte le prove necessarie, ma se tu vorrai sperimentare e farmi sapere... basterà adattare di poco i comandi che sono riportati qui a seguire.
- 2) Un qualsiasi **browser web**, purché sia abbastanza recente da supportare lo standard HTML5. Tutti i browser degli ultimi 2-3 anni supportano HTML5, sicuro. Il browser ti servirà per navigare la documentazione e, soprattutto, per stampare le cartelle su carta con TombolaPrint. Come fatto per il sistema operativo, anche per il browser, in questo manuale, se ne citerà uno solo per riferimento, Google Chrome, ma sei libero di provare ed utilizzare il browser che più preferisci.
- 3) Avere installato l'ambiente Java, ovvero un **Java Development Kit**. In questo manuale si farà riferimento al Java Development Kit (JDK) versione 1.8 di Oracle ma TombolaCards e TombolaPrint non hanno requisiti particolari, funzionano bene anche con altri JDK, sempre però successivi alla versione 1.8, come ad esempio Open JDK 10, 11, 12.
- 4) Infine, ovviamente, servirà **installare TombolaCards e TombolaPrint**, che puoi scaricare dal *repository* sulla piattaforma di condivisione software GitHub. Se stai leggendo questo manuale... vuol dire che conosci già GitHub e ci sei già stato!

## Installazione di Java

Effettuare il download del Java Development Kit 8 e del Java Runtime Engine 8 (JDK 8 e JRE 8) è davvero facile e ben documentato in Internet. Ti fornisco qui soltanto il link alla pagina di documentazione ufficiale per la versione che ho utilizzato.

- [https://docs.oracle.com/javase/8/docs/technotes/guides/install/install\\_overview.html](https://docs.oracle.com/javase/8/docs/technotes/guides/install/install_overview.html)

Quando avrai completato l'installazione, potrai aprire una *"shell di comando"*, ovvero una finestra interattiva del sistema operativo in cui i comandi vengono digitati uno alla volta sulla tastiera invece di cliccare a destra e a manca su bottoni e menu a video.

Per aprire una shell di comando puoi utilizzare la casella di ricerca che c'è in basso a sinistra: scrivi *"cmd"* (abbreviazione di *command*), come vedi in figura.

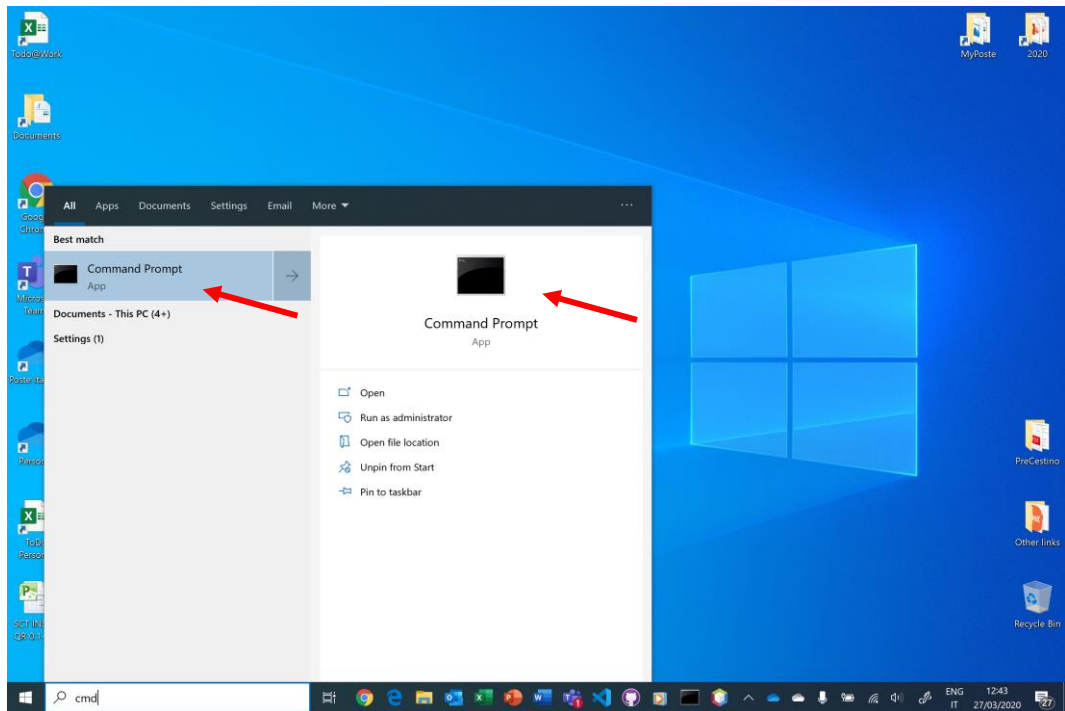


Figura 2 – L'avvio di una shell dei comandi in Windows 10.

Ora, nella shell di comando, posizionandoti in una directory qualsiasi, digita il comando di prova **Java -version** seguito da <INVIO>. Dovresti vedere qualcosa di molto simile a quanto riportato qui a seguire.

```
C:\>Java -version

java version "1.8.0_211"
Java(TM) SE Runtime Environment (build 1.8.0_211-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.211-b12, mixed mode)
```

É così? Fantastico, non devi fare altro, Java è installato e funzionante sul tuo pc. Se invece, malauguratamente, vedi un messaggio del tipo

```
C:\>java -version
"java" non è riconosciuto come comando interno o esterno,
un programma eseguibile o un file batch.
```

Allora vuol dire che qualcosa è andato storto nell'installazione. Mi spiace, devi riprovarci seguendo le istruzioni riportate ai link di cui sopra. **Per poter proseguire è indispensabile avere Java che funziona alla perfezione sul tuo pc.**

### Installazione di TombolaCards e TombolaPrint

È possibile effettuare il download di TombolaCards dallo spazio ad esso dedicato su GitHub. GitHub è, per quanto ci riguarda al momento, uno spettacolare *repository* mondiale di progetti *open source*, ovvero di progetti software in cui gli sviluppatori decidono di condividere più o meno liberamente il

codice sorgente che realizzano. È questo il caso anche del progetto **TombolaStudio**. Accedi dunque a GitHub<sup>3</sup> e raggiungi il seguente link:

- <https://github.com/HarryGPotter/TombolaStudio>

Sulla destra, abbastanza in alto sulla pagina web, dovresti vedere un tasto verde che ti invia a “clonare” o ad “effettuare il download” del progetto:

- Se sei uno sviluppatore e vorrai contribuire a questo progetto, allora con ogni probabilità saprai utilizzare anche un po’ GitHub e quindi puoi scegliere “Clona”;
- Se invece sei un utilizzatore interessato al gioco della Tombola, scegli pure “Download ZIP”, come indicato in figura.

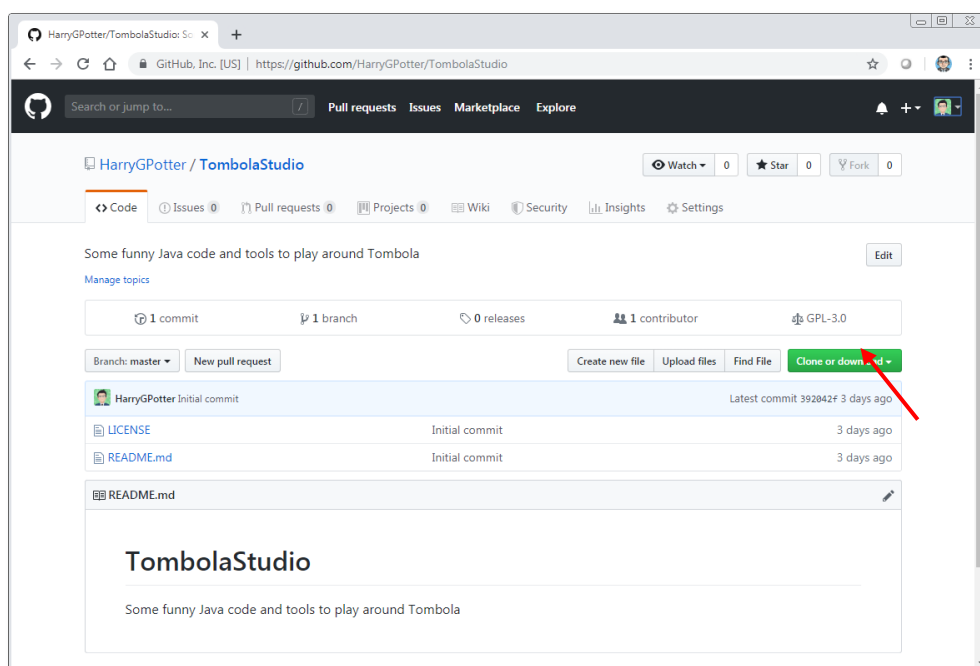


Figura 3 – La home page del progetto TombolaStudio su GitHub.

Bene, ci siamo quasi. Ora non ti resta che spaccettare il file ZIP nella cartella che preferisci. Io, ad esempio, ho spaccettato il tutto nella mia cartella **C:\progetti** e quindi ora ho tutti i file in **C:\progetti\TombolaStudio\**. D’ora in avanti farò riferimento a questa cartella, tu ovviamente dovrai adeguare i comandi e tutto il resto al “percorso” di cartelle che avrai scelto.

Vediamo insieme cosa trovi in questo progetto:

Cartella	Contenuto
- /TombolaStudio	È il nostro punto di partenza. Qui trovi il classico file README (“leggimi”) in cui è riassunto lo scopo del progetto, il file di licenza e soprattutto questo documento, in formato PDF.

<sup>3</sup> Per accedere a GitHub occorre essere registrati ed effettuare quindi il login. Presumo tu sia registrato e vado spedito, ad ogni modo se non lo fossi... ti basta avere un indirizzo e-mail valido per farlo. Al momento in cui scrivo non costa nulla.

Cartella	Contenuto
- /TombolaStudio/release	È questa la cartella in cui sono disponibili tutti i file “direttamente utilizzabili” del progetto, ovvero i file eseguibili (in realtà, trattandosi di programmi Java, sono file “.jar”) delle ultime versioni dei vari tool (TombolaCards, TombolaPrint) ed i file a loro supporto, come ad esempio file di configurazione, etc. <b>Questa cartella, insomma, è il punto di partenza per chi vuole utilizzare i programmi disponibili di TombolaStudio.</b>
- /TombolaStudio/release/print_templates	Questa cartella contiene i modelli ( <u>templates</u> ) utilizzati da TombolaPrint per preparare le stampe delle cartelle. Per i più tecnici, i modelli sono in formato HTML5 ed utilizzano i fogli di stile CSS3. Il modello da utilizzare di volta in volta va specificato con TombolaPrint, per cui è utile conoscere i nomi dei file presenti in questa cartella.
- /TombolaStudio/TombolaCards, - /TombolaStudio/TombolaPrint, - /TombolaStudio/TombolaLib	In queste cartelle sono contenute i file di “progetto” veri e propri, ovvero i file di <b>codice sorgente Java</b> . Sono pertanto cartelle di interesse per informatici e programmatori che hanno voglia di curiosare sui meccanismi interni di questi strumenti o magari vogliono evolverli o personalizzarli. Lo strumento per gestire questi progetti è <b>Apache Maven</b> , per cui chi vorrà troverà tutto quanto serve per “compilare”, “fare il deploy”, “collaudare”, “installare”, ...
- /TombolaStudio/samples	Questa cartella contiene materiale vario aggiuntivo, che può comunque esserti utile. Ad esempio, puoi trovarci file di cartelle prodotti con TombolaCards, file HTML5 contenenti le relative stampe, etc.

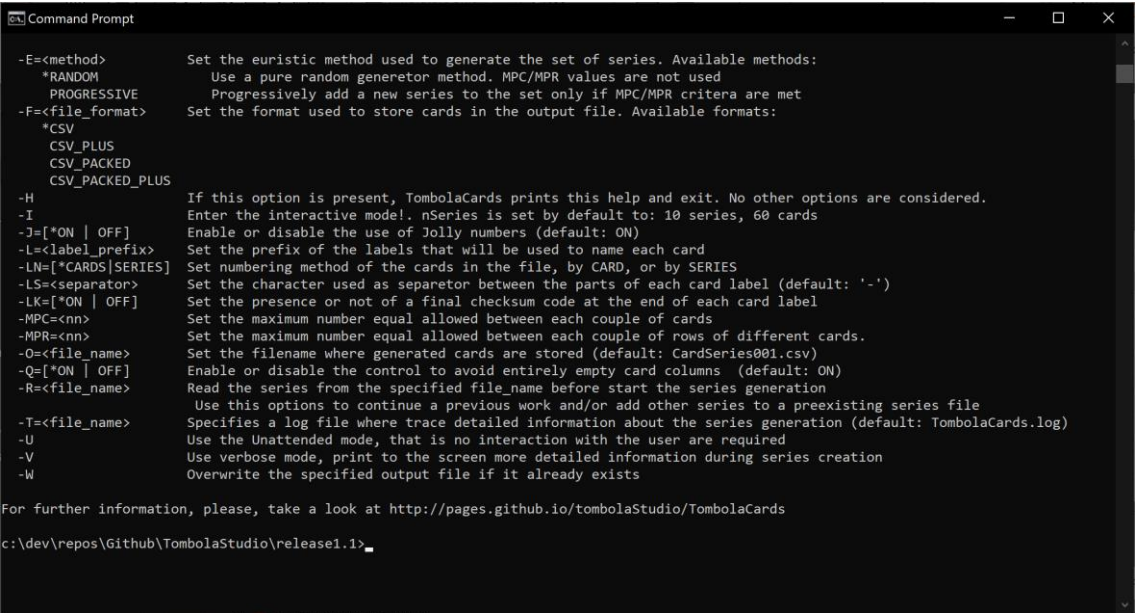
*Tabella 1 – Riepilogo delle principali sottocartelle del repository TombolaStudio su Github.*

Tutto chiaro? No? Non ti preoccupare, man mano che inizierai ad utilizzare i tool di **TombolaStudio** tante cose ti diventeranno naturalmente familiari.

Facciamo solo un’ultima prova per vedere se tutto è installato per il meglio. Apri una nuova shell dei comandi, portati nella directory in cui hai spaccettato il tutto. Ora prova a dare il seguente comando:

```
java -jar TombolaCards.jar -version
```

Dopo qualche frazione di secondo dovresti vedere una sbrodolata di testo scorrere nella tua shell di comando. Dovresti riuscire a leggere la versione di TombolaCards e poi l’elenco di tutti i *parametri* che dovrebbero o potrebbero essere specificati affinché il programma faccia il suo dovere, qualcosa di molto simile a quanto riportato nella seguente figura.



```
Command Prompt

-E=<method>          Set the heuristic method used to generate the set of series. Available methods:
  *RANDOM             Use a pure random generator method. MPC/MPR values are not used
  *PROGRESSIVE        Progressively add a new series to the set only if MPC/MPR criteria are met
-F=<file_format>     Set the format used to store cards in the output file. Available formats:
  *CSV
  CSV_PLUS
  CSV_PACKED
  CSV_PACKED_PLUS
-H                  If this option is present, TombolaCards prints this help and exit. No other options are considered.
-I                  Enter the interactive model. nSeries is set by default to: 10 series, 60 cards
-J=[*ON | OFF]      Enable or disable the use of Jolly numbers (default: ON)
-L=<label_prefix>    Set the prefix of the labels that will be used to name each card
-LN=[*CARDS|SERIES] Set numbering method of the cards in the file, by CARD, or by SERIES
-LS=<separator>      Set the character used as separator between the parts of each card label (default: '-')
-LK=[*ON | OFF]      Set the presence or not of a final checksum code at the end of each card label
-MPC=<nn>            Set the maximum number equal allowed between each couple of cards
-MPR=<nn>            Set the maximum number equal allowed between each couple of rows of different cards.
-O=<file_name>        Set the filename where generated cards are stored (default: CardSeries001.csv)
-Q=[*ON | OFF]       Enable or disable the control to avoid entirely empty card columns (default: ON)
-R=<file_name>        Read the series from the specified file_name before start the series generation
                     Use this options to continue a previous work and/or add other series to a preexisting series file
-T=<file_name>        Specifies a log file where trace detailed information about the series generation (default: TombolaCards.log)
-U                  Use the Unattended mode, that is no interaction with the user are required
-V                  Use verbose mode, print to the screen more detailed information during series creation
-W                  Overwrite the specified output file if it already exists

For further information, please, take a look at http://pages.github.io/tombolaStudio/TombolaCards

c:\dev\repos\Github\TombolaStudio\release1.1>
```

Figura 4 – Il risultato dell'avvio di TombolaCards senza parametri.

Se invece tutto ciò non accade, e continui a leggere messaggi del tipo

```
"java" non è riconosciuto come comando interno o esterno,
un programma eseguibile o un file batch.
```

Oppure

```
Error: Unable to access jarfile TombolaCards-1.1.jar
```

Allora vuol dire che qualcosa della tua installazione non è andata a buon fine. Coraggio, non è roba così difficile, armati di un po' di pazienza, torna all'inizio di questo capitolo e verifica tutti i passi che descritti. Non ci vorrà molto.

## 2. TombolaCards: generare cartelle

Se sei arrivato fin qui vuol dire che hai un pc con un browser ed hai scaricato e installato **Java**, **TombolaCards** e **TombolaPrint**. Molto bene. Possiamo iniziare ad utilizzare TombolaCards per elaborare *ex novo* cartelle per il gioco della Tombola.

Prima di continuare, però, solo per chi ha qualche minuto in più e voglia di leggere, ricordiamo come è fatta una *cartella* della tombola e cosa si intende per *serie di cartelle*, così sappiamo poi cosa aspettarci dal lavoro di **TombolaCards**.

### Anatomia di una Cartella

Una cartella per giocare a tombola può essere schematizzata come una *griglia* formata da 27 posizioni, disposte su 3 righe e 9 colonne. Per capirci, in figura la seconda riga è cerchiata in rosso scuro, la seconda colonna in verde. Ogni posizione può essere vuota, oppure contenere un numero che va da 1 a 90.

Ci sono altre regole che devono essere sempre rispettate:

- **Su ogni riga devono esserci sempre 5 numeri**, e quindi le altre 4 posizioni delle nove in totale restano vuote. Su ogni cartella ci sono dunque sempre 15 numeri e 12 posizioni vuote.
- **In ciascuna colonna vi sono solo numeri appartenenti ad una specifica decina**, ed in particolare: in colonna 1 ci possono essere solo numeri che vanno da 1 a 9, in colonna 2 solo numeri che vanno da 10 a 19, in colonna tre numeri che vanno da 20 a 29, e così via, fino alla colonna 9 in cui possono esserci soltanto numeri che vanno da 80 a 90.

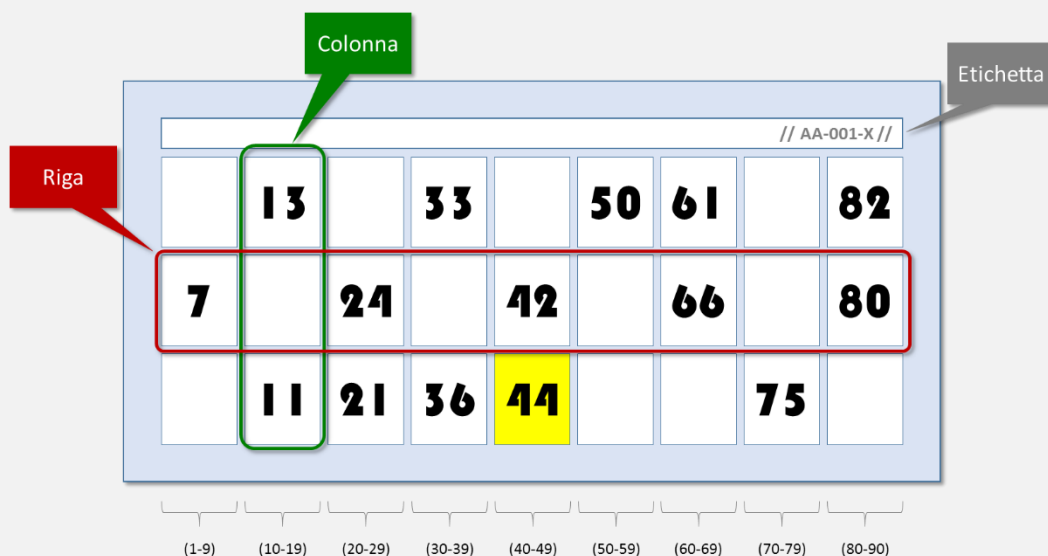


Figura 5 – Anatomia di una cartella della Tombola

Inoltre, anche se è tecnicamente possibile e la cosa non inficerebbe in nessun modo il gioco, è raro trovare cartelle in cui siano presenti colonne totalmente vuote, e cioè con tre posizioni senza nessun numero.

Infine, di solito, **le cartelle hanno una etichetta**, un nome o un codice progressivo che permette di identificarle univocamente in modo pratico e semplice. L'etichetta della cartella in figura è riportata in alto a destra, "AA-001-X".

## Le Serie di cartelle

Abbiamo detto che i numeri del gioco della tombola sono in tutto 90 e che ogni cartella è formata da 15 numeri. Bene, utilizzando tutti i 90 numeri della tombola una ed una sola volta è possibile realizzare 6 cartelle, ovvero una **serie di cartelle**:

***Una serie di cartelle è un gruppo di 6 cartelle che non hanno nessun numero in comune, e cioè dispongono nelle loro posizioni tutti i 90 numeri della tombola una ed una sola volta.***

// TT-001-JJ //														
2	17				58	65		88						
		26	31		59		76	81						
		25	30	43	50	67								

// TT-004-93 //														
		21	38	49			78	83						
9	14		39			61		82						
1	10				57	64	74							

// TT-002-FD //														
			36	42		60	71	90						
5	16	29			56		79							
7		27		46	55			84						
// TT-005-NF //														
4		24	37	40		69								
	19	20		45	54			80						
3			34	44	52		77							
// TT-003-IT //														
	11			41	53	62		87						
	13	28	35				73	86						
8	15			48		66		89						
// TT-006-89 //														
	18	23	33		51		72							
6	12			47		63	70							
		22	32			68	75	85						

Figura 6 – Una serie dispone, sulle sue 6 cartelle, tutti i 90 numeri del gioco, senza ripetizioni

Puoi controllare se vuoi. Tutti i numeri, dall'Uno al Novanta, sono presenti in una delle sei cartelle della figura. Inoltre, tutti i numeri sono utilizzati una volta soltanto. 15 numeri su 6 cartelle.  $15 \times 6 = 90$ .

## I numeri Jolly!

Non ti sarà sfuggito, osservando sia *Figura 5* che *Figura 6*, che su ogni cartella c'è sempre un numero evidenziato in modo differente. TombolaCards ha la possibilità, infatti, di scegliere, per ciascuna cartella, un numero particolare, detto **numero Jolly**. In *Figura 5* il numero Jolly è il 44, in *Figura 6* sono invece Jolly i numeri 65 per la prima cartella, 7 per la seconda cartella, 8 per la terza, 61 per la quarta, 54 per la quinta e 47 per la sesta cartella.

I numeri Jolly non hanno nulla di particolare e non esistono nemmeno nel gioco della Tombola tradizionale. È un'aggiunta che fa TombolaCards quando genera le cartelle, come spiegato anche nei paragrafi successivi. I numeri Jolly possono essere utilizzati a discrezione degli utenti, ad esempio per assegnare premi aggiuntivi nel caso in cui un numero Jolly venga coinvolto in un terno, una quintina... oppure nel caso in cui il numero

Jolly di una cartella sia proprio l'ultimo estratto per fare Tombola. Qualche esempio:

- Se un giocatore vince la Quintina utilizzando anche il numero Jolly, allora ha diritto ad una cartella gratis in gioco in più per la prossima manche;
- Se un giocatore vince la Tombola coprendo, come ultimo numero, proprio il numero Jolly della sua cartella... beh, allora avrà diritto ad una bottiglia di champagne extra, o ad un super panettone!

TombolaCards è un programma che può essere utilizzato sia in modalità a *“riga di comando”*, sia in modalità *“interattiva”*. In entrambi i casi vi sono molte opzioni e possibilità di personalizzare il comportamento di TombolaCards durante la generazione delle cartelle. Procederemo dunque in questo modo:

- Un primo paragrafo illustra il modo più semplice per utilizzare ciascuna delle due modalità, a riga di comando o interattiva. Vengono mostrati soltanto i parametri più importanti per ottenere i risultati più comuni e desiderati. La semplicità prima di tutto.
- Un secondo paragrafo elenca tutte le opzioni disponibili, utilizzando tabelle riepilogative sintetiche ed esaustive. In questo modo, chi vuole, può addentrarsi in tutte le sfaccettature di TombolaCards.

### Modalità “a riga di comando”

Il modo più semplice per utilizzare TombolaCards è quello di lanciarlo come un qualsiasi tool a riga di comando. Posizioniamoci nella cartella (d'ora in poi utilizzeremo anche il termine *directory*, proprio per non confonderci con le cartelle della Tombola) dove è installato TombolaCards:

```
> cd c:\progetti\TombolaStudio\release1.1
```

Se hai seguito le istruzioni di installazione del capitolo precedente, hai già provato a lanciare l'esecuzione di TombolaCards senza specificare nessun parametro in input. L'output che ottieni è riportato in Figura 4: viene stampato a video una guida sintetica all'utilizzo e l'elenco dei parametri ammessi. Andiamo avanti e proviamo a dare il comando più semplice possibile con cui ottenere nuove cartelle della Tombola. Digittiamo il seguente comando:

```
> java -jar TombolaCards.jar 5
```

Dopo qualche istante a video compare quanto segue:

```
Hello Tombola World!
TombolaCards version: 1.1
Library: TombolaLib, version 1.1
-----

Requested 5 series, 30 cards.
Generating: ..... Done! 5 have been generated.
File [CardSeries001.csv] has been saved.
```



-----  
Bisogna avere il caos dentro di sé per partorire una stella danzante.

È vero, alcuni messaggi a video sono un po'... estemporanei, ma non importa. Prima di iniziare a lavorare, **TombolaCards** stampa un messaggio di benvenuto ("*Hello again Tombola World! ...*") ed informazioni circa la versione utilizzata della libreria TombolaLib. Poi inizia la generazione delle cartelle e stampa a video un puntino per ogni serie di cartella generata (ricorda, ogni serie contiene sei cartelle). Avevamo chiesto 5 serie di cartelle, indicandolo come primo ed unico parametro sulla riga di comando, ed infatti, se contiamo i puntini, ne ritroviamo 5. 5 Serie, 6 cartelle per ciascuna serie, 30 cartelle in tutto.

Una volta generate, le cartelle vengono salvate in un file nella directory corrente. Il nome del file predefinito è **CardSeries001.csv**. Se usi il comando `dir` nella cartella corrente dovresti vederlo.

```
>dir

Volume in drive C has no label.
Volume Serial Number is 7A54-A664

Directory of c:\progetti\TombolaStudio\releases\1.1

25/03/2020  12:00    <DIR>          .
25/03/2020  12:00    <DIR>          ..
25/03/2020  12:13             4.408 CardSeries001.csv
25/03/2020  12:00        124.722 TombolaCards.jar
25/03/2020  12:13           918 TombolaCards.log
```

Il file prodotto è un normalissimo file di testo, in cui su ogni riga vi sono i numeri scelti per ogni cartella, separati dal carattere punto e virgola. Possiamo usare qualsiasi utility per gestire file di testo (Notepad, Wordpad, Visual Studio Code) per dare un'occhiata più da vicino al contenuto di questo file.

```
>notepad CardSeries001.csv
```

```

####TombolaLib;1.1;CSV_PLUS## TombolaLib 1.1
# File description: Series list freshly generated by TombolaCards
# Creation timestamp: 2020/03/25 20:01:12
# File Format: CSV_PLUS. Use of Jolly numbers: ON
# 1 Series (6 cards)
# Card generation technique: PerfectSwap_1.1
# Card set generation heuristic: Pure Random Factory 1.0
#
CardLabel1;NumR0C0;NumR0C1;NumR0C2;NumR0C3;NumR0C4;NumR0C5;NumR0C6;NumR0C7;NumR0C8;NumR1C0;NumR1C1;
NumR1C2;NumR1C3;NumR1C4;NumR1C5;NumR1C6;NumR1C7;NumR1C8;NumR2C0;NumR2C1;NumR2C2;NumR2C3;NumR2C4;NumR2C5;
NumR2C6;NumR2C7;NumR2C8;JollyIndex;MaxEqualPerCard;MaxEqualPerRow;

"TT-001-8B";02;00;00;34;00;00;66;77;82;00;13;27;00;00;57;65;00;83;00;00;00;30;48;00;68;74;80;(11);00;00;
"TT-002-3J";07;00;25;35;00;00;62;79;00;00;19;28;00;00;53;00;76;90;03;00;29;38;46;00;00;70;00;(07);00;00;
"TT-003-U9";00;16;21;00;43;51;00;00;89;09;14;23;00;00;00;67;00;87;01;00;00;36;41;54;00;75;00;(08);00;00;
"TT-004-VJ";05;11;26;37;00;59;00;00;08;15;00;00;49;00;60;78;00;00;00;24;00;45;00;61;73;81;(05);00;00;
"TT-005-4V";00;00;00;39;47;56;63;72;00;00;10;00;31;40;00;69;00;85;06;00;20;32;44;50;00;00;00;(14);00;00;
"TT-006-GB";04;12;00;00;00;52;64;00;86;00;18;00;33;00;58;00;71;88;00;17;22;00;42;55;00;00;84;(01);00;00;

```

Figura 7 – Un esempio di file di cartelle prodotto da TombolaCards. La descrizione di tutti i formati possibili è riportata al successivo paragrafo “I formati dei file di cartelle”.

Le prime righe del file iniziano tutte con il carattere “#” (*cancelletto*, o *hashtag*). In genere questo carattere identifica delle righe che non contengono dati fondamentali (come ad esempio i numeri delle cartelle) ma piuttosto commenti o informazioni aggiuntive destinate a chi, come noi in questo momento, legge il file. La prima riga è un po’ criptica, perché serve a programmi come TombolaCards e TombolaPrint per ‘capire’ come è stato generato il file. Le altre righe che iniziano con l’hashtag contengono informazioni facilmente comprensibili: la versione della libreria software utilizzata, una breve descrizione del file, la data e l’orario in cui è stato creato, se sono utilizzati o meno i numeri jolly.

Poi iniziano le righe che contengono le cartelle vere e proprie. Su ogni riga c’è una cartella. La prima cosa che si legge, racchiusa tra doppi apici è una etichetta, ovvero un nome che identifica univocamente ciascuna cartella all’interno del file.

Dopo l’etichetta, su ogni riga ci sono 27 numeri: sono i numeri della cartella, nell’ordine in cui compaiono su di essa. 15 numeri sono diversi da zero, 12 sono uguali a zero e rappresentano le caselle che devono restare vuote.

Infine, in coda ad ogni riga ci sono altre informazioni, sempre relative alla cartella: l’indice del numero jolly, alcuni dati statistici.

Il file è chiuso con altre righe di commento, che iniziano sempre con il carattere cancelletto e che riportano altre informazioni generali relative al file.

Il dettaglio di queste informazioni, e le variazioni di formato che sono supportate da TombolaCards, sono descritte con dovizia di particolari al paragrafo “I formati dei file di cartelle” più avanti in questo capitolo.

**Potremmo fermarci qui.** L’obiettivo principale di TombolaCards è stato raggiunto: generare un file di testo contenente cartelle per il gioco della Tombola. Se vuoi, puoi già saltare al capitolo

*“TombolaPrint: stampare cartelle”* per imparare a stampare su carta le cartelle ed iniziare a giocare.

Per i più curiosi... continuiamo a giocare con la nostra riga di comando e vediamo come si usa qualche altra opzione tra quelle più utili e comuni.

Proviamo a creare un file di cartelle più grande. Diciamo... 100 serie di cartelle, quindi 600 cartelle in tutto. Diamo quindi il seguente comando:

```
> java -jar TombolaCards.jar 100
```

Avremo una piccola delusione (tranquillo, l'abbiamo fatto di proposito, a scopi didattici), TombolaCards con ogni probabilità risponderà così:

```
Hello Tombola World!  
TombolaCards version: 1.1  
Library: TombolaLib, version 1.1  
-----  
<ERROR!> Chosen output file [CardSeries001.csv] already exist.  
Change name or use option -W to overwrite it.
```

Quello che succede è che TombolaCards si ferma prima ancora di generare le cartelle, perché si accorge che il file CardSeries001.csv, il file in cui dovrebbe scrivere le nuove cartelle, esiste già e non vuole dunque sovrascriverlo per errore.

Abbiamo due possibilità (così facciamo la conoscenza di due ulteriori parametri):

- Possiamo decidere che le vecchie cartelle possono essere sovrascritte, quindi possiamo salvare le nuove in un file che sostituisce il precedente. Per farlo basta aggiungere consapevolmente l'opzione -W (overWrite) alla riga di comando precedente.

```
> java -jar TombolaCards.jar 100 -w
```

- Possiamo decidere di salvare le nuove cartelle in un file con un altro nome a nostra scelta. L'opzione da utilizzare per farlo è l'opzione -O (Output file), seguita da un uguale e dal nome che desideriamo per il nostro file.

```
> java -jar TombolaCards.jar 100 -O=NuovoNome.csv
```

Tutto qui. Direi che possiamo passare al prossimo paragrafo, dove tutte le opzioni sono scrupolosamente elencate e descritte. E non dimenticare che in qualsiasi momento puoi lanciare TombolaCards senza opzioni per stampare a video una guida sintetica delle opzioni disponibili.

## Riepilogo dei parametri “a riga di comando”

In questo paragrafo vengono riportate tutte le opzioni che è possibile utilizzare sulla riga di comando di TombolaCards. La tabella che segue, riporta, per ognuna di esse, l'obbligatorietà o meno, i valori e le impostazioni predefinite utilizzate nel caso in cui l'opzione è omessa, una breve descrizione di come tali opzioni modificano il comportamento di TombolaCards stesso.

Tutte le opzioni devono essere precedute da un trattino “-” e separate tra loro almeno da uno spazio.

Possono essere scritte sia in maiuscolo che in minuscolo, non fa differenza. Nemmeno l'ordine in cui sono riportate sulla riga di comando è significativo.

Tipicamente le opzioni possono funzionare in due modi:

- **Presenti / Non Presenti:** sono le opzioni più semplici e sono quelle che modificano il comportamento di TombolaCards semplicemente in base alla loro presenza o meno sulla riga di comando. Funzionano così, ad esempio, le opzioni -H, -I, -V, -W, etc.
- **Impostazione di un valore:** sono le opzioni che consentono di specificare il valore di un parametro utilizzato da TombolaCards, come ad esempio, il nome del file in cui scrivere le cartelle generate. In questo caso l'opzione, sempre preceduta dal trattino, va fatta seguire dal simbolo di “=” e poi dal valore che si vuole impostare. Per stabilire che le cartelle debbano essere scritte nel file “CartelleDiPippo.csv”, ad esempio, occorre dunque aggiungere sulla riga di comando l'opzione -O=CartelleDiPippo.csv.

Opzione	Valori ammessi	Obbligatorietà e valori predefiniti	Descrizione
nseries	1..100	<b>Parametro obbligatorio</b>  Può essere omesso solo se si vuole avviare la modalità <i>interattiva</i> e quindi si utilizza l'opzione “-I” (vedi più avanti).	Indica il <b>numero di serie di cartelle che si desidera generare</b> . Questo parametro è l'unico ad essere obbligatorio, deve essere il primo dei parametri inseriti sulla riga di comando e non deve essere preceduto dal trattino “-” che invece viene utilizzato come prefisso per tutti gli altri parametri opzionali.
-E	RANDOM, PROGRESSIVE	<b>Parametro facoltativo</b>  (Valore predefinito: RANDOM)	Indica il metodo (euristico) utilizzato per generare le cartelle. Al momento è possibile scegliere tra due metodi diversi: <ul style="list-style-type: none"><li>• <b>RANDOM:</b> genera le cartelle scegliendo i numeri da disporre nelle varie caselle in modo del tutto casuale, senza nessuna verifica particolare. È il metodo più “veloce” per generare le cartelle.</li><li>• <b>PROGRESSIVE</b> questo metodo genera le cartelle cercando di evitare che nel file ve ne siano con troppi numeri in comune. In pratica, ogni volta che genera una nuova serie, prima di aggiungere quest'ultima al file, controlla che non contenga cartelle con <i>troppi</i> numeri uguali alle cartelle già presenti nel file stesso. Per determinare se i numeri in comune sono troppi si utilizzano i valori impostati utilizzando le opzioni -MPR e -MPC descritte più avanti. La logica di questo metodo ed i vantaggi che si possono avere utilizzandolo sono più ampiamente descritti nello</li></ul>

Opzione	Valori ammessi	Obbligatorietà e valori predefiniti	Descrizione
			spazio di approfondimento al paragrafo “Il problema” delle vincite” più avanti a pag. 39.
-F	CSV, CSV_PLUS, CSV_PACKED, CSV_PACKED_PLUS	<b>Parametro facoltativo</b> (valore predefinito: CSV_PLUS)	Specifica il formato con cui vengono salvate le cartelle nel file. Ad oggi sono supportati quattro modalità di salvataggio leggermente diverse tra loro, tutte però che basate sulla consuetudine CSV, <i>Comma Separated Value</i> . I formati differiscono tra loro perché alcuni memorizzano solo i 15 numeri di ciascuna cartella, omettendo il salvataggio delle posizioni vuote, altri invece inseriscono zeri anche per le caselle vuote. Alcuni formati aggiungono informazioni statistiche, altri no. Una descrizione completa e dettagliata di ciascun formato è riportata al paragrafo “I formati dei file di cartelle” a pag. 26.
-H	-	<b>Parametro facoltativo</b>	Se questo parametro è presente sulla riga di comando, TombolaCards ignora tutti gli altri parametri specificati e stampa a video una sintetica guida all’utilizzo (help), come riportato in <i>Figura 4</i> a pag. 13. Il programma poi termina senza né generare cartelle, né compiere altre operazioni. Lo stesso risultato si ottiene lanciando TombolaCards senza specificare nessuna opzione aggiuntiva sulla riga di comando.
-I	-	<b>Parametro facoltativo</b>	Questo parametro consente di avviare TombolaCards in modalità interattiva, modalità di lavoro descritta al paragrafo Modalità “interattiva” più avanti. Se presente questa opzione, TombolaCards non inizia a generare cartelle, ma entra in una modalità di lavoro in cui l’utente può dare comandi specifici ed interagire con i risultati che questi comandi man mano producono. Se utilizzato insieme ad altre opzioni, come, ad esempio, il numero di serie che si vogliono generare, oppure le opzioni -E, -MPC, -MPR, -R, -O, -T, tali opzioni aggiornano i valori dei rispettivi parametri prima di avviare la modalità interattiva.
-J	ON, OFF	<b>Parametro facoltativo</b> (valore predefinito: ON)	Attiva o disattiva la predisposizione dei numeri Jolly per le cartelle generate (per maggiori dettagli sui numeri Jolly consulta il paragrafo “I numeri Jolly!” a pag. 15). Il valore predefinito è ON la scelta dei numeri Jolly non costa nulla e può essere ignorata successivamente. Ad ogni modo, se si vuole disattivare la generazione, è sufficiente aggiungere alla riga di comando l’opzione -J=OFF.
-L	<prefisso_etichetta>	<b>Parametro facoltativo</b> (valore predefinito: “TT”)	TombolaCards associa sempre ad ogni cartella una etichetta che consente di identificarla univocamente all’interno del file. Le etichette sono formate da: <ul style="list-style-type: none"> <li>• Un <b>prefisso</b>, sempre uguale per tutte le cartelle del file.</li> <li>• Un <b>numero progressivo</b>, che invece è diverso per ogni cartella. Vedi l’opzione “LN” più avanti per conoscere come viene generato questo progressivo.</li> <li>• Un <b>codice di controllo</b>, detto anche codice <i>checksum</i>, utilizzato per ridurre il rischio di</li> </ul>

Opzione	Valori ammessi	Obbligatorietà e valori predefiniti	Descrizione
			<p>contraffazioni o modifiche non autorizzate ai numeri della cartella. Il codice di controllo è opzionale, vedi l'opzione <b>LK</b> per ulteriori informazioni.</p> <p>Prefisso, numero progressivo e codice di controllo possono essere intervallati con un carattere separatore, che può essere scelto con l'opzione -LS</p> <p>L'opzione -L consente di specificare il prefisso che si vuole per le etichette dell'intero file di cartelle. Puoi specificare la parola che vuoi. Alcuni esempi:</p> <ul style="list-style-type: none"> <li>-L = "Natale2012"</li> <li>-L = "AAA"</li> <li>-L = "LaMiaTombola"</li> </ul>
-LN	CARDS, SERIES	<p><b>Parametro facoltativo</b> (valore predefinito: CARDS)</p>	<p>Questa opzione consente di specificare il modo in cui le cartelle vengono numerate progressivamente. Il numero di ciascuna cartella è poi utilizzato per formare l'etichetta della cartella stessa (vedi opzione -L più sopra). Sono possibili due tipi di numerazione: "per cartella" (valore dell'opzione CARDS) e "per serie" (valore dell'opzione SERIES). Nel primo caso il numero progressivo assegnato ad ogni cartella è un progressivo incrementato di uno dalla prima all'ultima cartella del file. Nel secondo caso si numerano progressivamente le serie, non le cartelle. Le cartelle vengono invece numerate da 1 a 6 all'interno di ciascuna serie. Vediamo un po' di etichette di esempio nei due casi (per semplicità usiamo come prefisso "AA" ed omettiamo il checksum):</p> <p>-LN=CARDS → AA.001, AA.002, AA.003, AA.004, ..., AA.nnn</p> <p>-LS=SERIES → AA.S001.C1, AA.S001.C2, AA.S001.C3, ... AA.S001.C6, AA.S002.C1, AA.S002.C2, AA.S002.C3, ..AA.S002.C6, ..., AA.Snnn.C1, ..., AA.Snnn.C6.</p>
-LS	<carattere_separatore>	<p><b>Parametro facoltativo</b> (valore predefinito: "-")</p>	<p>Questo parametro consente di specificare il carattere separatore da utilizzare nella formazione delle etichette. Se l'opzione non è presente, viene utilizzato un trattino, "-", come nel seguente esempio: AA-001-XY. Alcuni esempi:</p> <ul style="list-style-type: none"> <li>-LS=. → AA.001.XY</li> <li>-LS=% → AA%001%XY</li> <li>-LS → AA00XY</li> </ul> <p>Come illustrato dall'ultimo esempio, Se non si vuole utilizzare alcun carattere separatore, si può specificare l'opzione -LS senza farla seguire da altri caratteri.</p>
-LK	ON, OFF	<p><b>Parametro facoltativo</b> (valore predefinito: ON)</p>	<p>Consente di disattivare l'aggiunta di un codice di controllo (<i>checksum</i>) di due caratteri alfanumerici in coda alle etichette di ciascuna cartella (tale campo è previsto sempre a meno che non si espliciti per LK il valore di OFF).</p> <ul style="list-style-type: none"> <li>parametro LK non usato → AA.001.XY</li> </ul>

Opzione	Valori ammessi	Obbligatorietà e valori predefiniti	Descrizione
			<ul style="list-style-type: none"> <li>• -LK=ON → AA.001.XY</li> <li>• -LK=OFF → AA.001</li> </ul> <p>Questo codice di controllo è generato applicando un algoritmo a tutti i numeri che compongono la cartella, per cui è possibile utilizzarlo, in fase di gioco, ad esempio con TombolaWeb, per controllare eventuali alterazioni apportate alle cartelle dopo la loro generazione.</p>
-MPC	<max uguali per cartella>	<b>Parametro facoltativo</b> (valore predefinito: 15)	<p>Questo parametro è preso in considerazione solo quando l'algoritmo utilizzato per la creazione delle cartelle è quello denominato PROGRESSIVE (vedi opzione -E più sopra).</p> <p>Consente di specificare il valore massimo consentito di numeri che una cartella può avere in comune con un'altra cartella dello stesso file.</p> <p>Il valore minimo consentito è 5, il valore massimo che ha senso è 14: valori superiori sono consentiti, ma equivalgono a non applicare alcun controllo.</p> <p>Se ad esempio aggiungiamo alla riga di comando l'opzione -MPC=7, quando TombolaCards genera una nuova serie di cartelle, prima di aggiungerle al file, verifica che nessuna delle sue 6 cartelle ha più di 7 numeri in comune con tutte le altre cartelle precedentemente generate e presenti nel file. In caso di verifica negativa la serie viene scartata, non viene aggiunta al file, se ne genera una nuova e si riprova.</p> <p>Il modo ottimale di utilizzare ed assegnare valori a questo parametro, peraltro senza rendere il processo di generazione delle cartelle estremamente lungo ed oneroso, è raccontato con maggiore dettaglio al successivo paragrafo // <i>"problema" delle vincite concomitanti</i> a pag. 39.</p>
-MPR	<max uguali per riga>	<b>Parametro facoltativo</b> (valore predefinito: 5)	<p>Questo parametro è preso in considerazione solo quando l'algoritmo utilizzato per la creazione delle cartelle è quello denominato PROGRESSIVE (vedi opzione -E più sopra).</p> <p>Consente di specificare il valore massimo consentito di numeri uguali che possono esserci tra due righe qualsiasi di due cartelle dello stesso file.</p> <p>Il valore minimo consentito è 2, il valore massimo che ha senso è 4: valori superiori sono consentiti, ma equivalgono a non applicare alcun controllo.</p> <p>Se ad esempio aggiungiamo alla riga di comando l'opzione -MPR=3, quando TombolaCards genera una nuova serie di cartelle, prima di aggiungerle al file, verifica che nessuna delle sue 6 cartelle abbia una riga con più di 3 numeri in comune con un'altra riga in una delle altre cartelle precedentemente generate e presenti nel file. In caso di verifica negativa la serie viene interamente scartata, non aggiunta al file, se ne genera una nuova e si riprova.</p> <p>Il modo ottimale di utilizzare ed assegnare valori a questo parametro, peraltro senza rendere il processo di</p>

Opzione	Valori ammessi	Obbligatorietà e valori predefiniti	Descrizione
			generazione delle cartelle estremamente lungo ed oneroso, è raccontato con maggiore dettaglio al successivo paragrafo II <i>“problema” delle vincite concomitanti</i> a pag. 39.
-O	<nome output file>	<b>Parametro facoltativo</b> (valore predefinito: CardSeries001.csv)	Questa opzione consente di specificare il nome del file in cui si vogliono salvare le serie di cartelle generate. Se questa opzione viene omessa viene utilizzato un nome di file predefinito riportato di fianco. Se il file esiste già, TombolaCards si interrompe e stampa a video un messaggio di errore per evitare di sovrascrivere (e quindi cancellare) un file precedentemente creato. File preesistenti possono essere sovrascritti utilizzando esplicitamente l'opzione -W, descritta più avanti in questa tabella.
-Q	ON, OFF	<b>Parametro facoltativo</b> (valore predefinito: ON)	Quando vengono generate le cartelle, TombolaCards controlla che, su ciascuna cartella, non vi siano mai colonne interamente vuote: delle tre posizioni disponibili in ogni colonna (vedi Figura 5 a pag. 14) almeno in una ci deve essere un numero. Tuttavia, cartelle con colonne vuote sono perfettamente regolari e non inficiano in alcun modo il gioco della tombola. Per cui questo controllo può essere disattivato aggiungendo sulla riga di comando l'opzione -Q=OFF.
-R	<nome file in input>	<b>Parametro facoltativo</b> (valore predefinito: nessun file)	<p>Questo parametro consente di specificare un file da cui leggere serie di cartelle prima ancora di iniziare a generarne di nuove. È una opzione utile per continuare ad aggiungere cartelle a file generati in precedenza. Supponiamo di avere un file, contenente già 10 serie di cartelle e vogliamo aggiungerne altre 10, per avere un file che complessivamente ne contenga 20 (120 cartelle in tutto). Supponiamo che il file contenente le cartelle già disponibili si chiami GiaPronte.csv.</p> <p>Per ottenere il nostro scopo possiamo usare il seguente comando:</p> <pre>java -jar TombolaCards.jar 20 -R=GiaPronte.csv -O=TutteInsieme.csv</pre> <p>Al termine dell'esecuzione di TombolaCards, avremo il file <b>TutteInsieme.csv</b> contenente 20 cartelle, le 10 già presenti nel file <b>GiaPronte.csv</b> più le 10 appena create ed aggiunte. Attenzione: le cartelle nel file finale sono sempre riordinate in ordine crescente dell'attributo "MAXEPC" ricalcolato per ogni cartella, quindi sarà difficile distinguere le vecchie dalle nuove (più informazioni su questo attributo, MAXEPC, sono fornite più avanti, a partire dal paragrafo I <i>formati dei file di cartelle</i> a pag. 26.</p> <p>L'opzione -R può essere utilizzata anche in combinazione con il metodo euristico "PROGRESSIVE" (vedi opzione -E). In questo caso, la creazione di cartelle con pochi</p>



Opzione	Valori ammessi	Obbligatorietà e valori predefiniti	Descrizione
			numeri in comuni tra loro può impiegare molto tempo, per cui può far comodo avere la possibilità di interrompere il lavoro, salvare le cartelle ottenute con un primo lancio per poi aggiungerne altre successivamente.
-S	<seme_numeri_casuali>	<b>Parametro facoltativo</b> (valore predefinito: <i>generato in modo casuale in base a valore dell'orologio interno del pc</i> )	<p>Questo parametro consente di specificare il seme con cui inizializzare il generatore di numeri pseudocasuali utilizzato nel processo di generazione delle cartelle.</p> <p>Se non si specifica questo parametro, il valore del seme è inizializzato sfruttando un metodo a sua volta pseudocasuale, come ad esempio la lettura dei millisecondi dell'orologio interno del pc. In questo modo ci si assicura la generazione di cartelle diverse tra loro ad ogni esecuzione di TombolaCards.</p> <p>Se però, per qualche motivo, si vuole rigenerare esattamente le stesse cartelle già generate in precedenza, è possibile utilizzare questo parametro per forzare uno stesso seme, in modo da ottenere la stessa sequenza di cartelle.</p>
-T	<nome output file>	<b>Parametro facoltativo</b> (valore predefinito: TombolaCards.log)	<p>Questa opzione consente di specificare il nome del file di log utilizzato da TombolaCards durante la generazione delle cartelle.</p> <p>Un file di log è un file in cui vengono scritti uno dopo l'altro messaggi che tracciano le operazioni eseguite, consentendo così l'analisi a posteriori di quanto successo, ad esempio per individuare errori, fasi di esecuzione troppo lunghe, etc.</p> <p>Se l'opzione -T non è presente, viene utilizzato il nome del file di log predefinito riportato accanto;</p> <p>Se l'opzione -T è presente senza specificare un preciso nome di file, la scrittura nel file di log è disattivata e durante la generazione delle cartelle non viene salvato nessun messaggio di log.</p>
-U	-	<b>Parametro facoltativo</b> (valore predefinito: OFF)	<p>Se questa opzione è presente sulla riga di comando, TombolaCards:</p> <ul style="list-style-type: none"> <li>• Evita di 'fare domande' all'utente finale durante la generazione di cartelle;</li> <li>• Riduce il numero di messaggi ed informazioni stampati a video.</li> </ul> <p>L'uso di -U agevola l'utilizzo di TombolaCards in modalità <i>unattended</i>, ovvero senza la supervisione ed il controllo di un utente umano. Si pensi ad esempio ai casi in cui si vuole richiamare TombolaCards all'interno di altri scripts facilitando così l'automazione di processi ripetitivi.</p>
-V	-	<b>Parametro facoltativo</b> (valore predefinito: OFF)	<p>Se questa opzione è presente, vengono scritti a video molti più messaggi e forniti molti più dettagli durante la generazione delle cartelle (V sta per <i>Verbose</i>). Può essere utilizzata per avere più dettagli in merito alle logiche di generazione delle cartelle o per investigare casi di errore o comportamenti di TombolaCards poco chiari.</p>

Opzione	Valori ammessi	Obbligatorietà e valori predefiniti	Descrizione
-W	-	<b>Parametro facoltativo</b> (valore predefinito: OFF)	Se presente, questa opzione forza TombolaCards a scrivere il file di output contenente le cartelle generate anche se un file con lo stesso nome esiste già, per cui deve essere sovrascritto (W come in overWrite). Se l'opzione non è presente, TombolaCards non sovrascrive file esistenti, ma si ferma visualizzando un messaggio di errore. Utilizza questa opzione se sei sicuro che non ti importa perdere file precedentemente salvati.

Tabella 2 – Riepilogo delle opzioni a riga di comando di TombolaCards.

## I formati dei file di cartelle

TombolaCards può salvare le cartelle generate in file con formati differenti, per consentire flessibilità di utilizzo con tutti i tool di TombolaStudio e con altri strumenti software anche al di là di questo progetto<sup>4</sup>.

Ad oggi sono supportate quattro varianti di file, non molto dissimili tra loro. Sono file di tipo “csv”, *comma separated value*, in cui i vari elementi informativi presenti su una stessa riga sono intervallati sempre dal carattere punto e virgola “;”.

La struttura generale dei file è sempre la stessa. Di seguito se ne riportano le varie sezioni.

- Un primo gruppo di righe costituiscono l'intestazione del file e contengono alcune informazioni generiche. Il primo carattere di ogni riga è sempre il carattere *cancellato* “#”, generalmente utilizzato per distinguere nei file csv le righe di commento dalle righe che invece contengono propriamente dati (cartelle, nel nostro caso). Troviamo, in questo ordine:
  - Una prima riga, un po' criptica, che contiene una serie di caratteri codificati in modo da consentire a TombolaCards e/o TombolaPrint di “capire” quale è il formato utilizzato e quindi di continuare a leggere e ad interpretare le righe successive;
  - Una riga con nome e versione della libreria software (TombolaLib) utilizzata;
  - Un breve descrizione del file;
  - La data e l'orario in cui il file è stato creato;
  - L'indicazione dell'attivazione o meno dei numeri jolly, per cui sulle righe che contengono i dati di ciascuna cartella è riportato l'indice del numero jolly scelto o meno.
- Seguono le cartelle della tombola, ogni riga contiene una cartella.

<sup>4</sup> Ad esempio, si potrebbe importare le cartelle in un programma *database*, come Microsoft Access™, ed utilizzare le sue funzionalità di generazione report per disegnare graficamente modelli di cartelle e poi stamparle come report di database. Oppure potresti utilizzare la funzionalità “Stampa Unione” di Microsoft Word o Microsoft Publisher.

- Il primo campo è sempre l'**etichetta** della cartella, il nome di battesimo che la identifica univocamente all'interno del file, costruito da TombolaCards secondo quanto descritto in merito alle opzioni -L, -LN, -LK, -LS al paragrafo precedente;
  - Seguono i **numeri della cartella**. I formati CSV e CSV\_PLUS riportano per ogni cartella, ventisette numeri: 15 sono diversi da zero, 12 sono uguali a zero e rappresentano le caselle vuote della cartella. I formati CSV\_PACKED e CSV\_PACKED\_PLUS riportano invece solo i 15 numeri diversi da zero, omettendo quindi del tutto di rappresentare con degli zero le caselle che restano vuote. I numeri sono sempre riportati in "ordine di apparizione" sulle cartelle, quindi sempre da sinistra a destra, prima la prima riga in alto, poi quella centrale, poi la terza in basso.
  - Dopo i numeri della cartella, solo se in fase di generazione del file si è lasciato attivo l'utilizzo dei numeri Jolly, segue, tra parentesi tonde, l'**indice del numero jolly**, ovvero la sua posizione nell'elenco dei numeri della cartella diversi da zero, iniziando a contare da zero. Più difficile a dirsi che a farsi... la Figura 8 che segue aiuterà a capire più di tante parole.
  - Infine, solo per i formati "PLUS" (CSV\_PLUS e CSV\_PACKED\_PLUS), chiudono la riga di ciascuna cartella altri due numeri, denominati in breve MAXEPC e MAXEPR. Il numero **MAXEPC** indica il numero massimo di numeri uguali tra questa cartella ed un'altra cartella appartenente allo stesso file di cartelle. Un valore di 6, ad esempio, vuol dire che all'interno di questo file c'è almeno un'altra cartella che ha 6 numeri in comune con questa cartella. Nessuna cartella invece all'interno del file ha 7 o più numeri in comune con questa cartella. Il numero **MAXEPR** indica il numero massimo di numeri uguali tra una riga di questa cartella ed una riga di un'altra cartella appartenente allo stesso file. Un valore di 3, ad esempio, vuol dire che all'interno di questo file c'è almeno un'altra cartella con una riga che ha 3 numeri in comune con una riga in questa cartella. Non vi son righe invece con 4 o più numeri uguali ad una riga di questa cartella.
- In coda, infine, vi è un ultimo gruppo di righe di commento, sempre precedute dal carattere cancelletto, che contengono altre informazioni di dettaglio e statistiche per le cartelle contenute nel file.
    - Viene riportato il numero intero (generalmente molto grande, che può essere positivo o negativo) che è stato usato come **seme per il generatore di numeri pseudo-casuali** utilizzato dall'algoritmo di generazione delle cartelle stesse. Questo numero viene generalmente scelto in modo pseudo-casuale, ma può essere impostato manualmente utilizzando l'opzione a riga di comando -S. Se si lancia TombolaCards scegliendo lo stesso algoritmo, gli stessi parametri di MAXEPC e MAXEPR e lo stesso seme per il generatore di numeri pseudo casuali, si ottiene la generazione delle stesse cartelle della tombola, per cui può essere utile conoscere il seme utilizzato per la generazione di un file proprio per riprodurre la stessa sequenza di cartelle.

Se il file contiene cartelle che non sono state generate direttamente da TombolaCards, ma che ad esempio sono state lette da un altro file, allora il seme per il generatore di numeri casuali non è riportato.

  - Il tempo speso per generare il file. Questa indicazione è particolarmente interessante soprattutto per i file che contengono cartelle "ottimizzate", e cioè generate con il metodo euristico "PROGRESSIVE" che riduce i numeri in comune tra le cartelle e che per questo motivo potrebbe impiegare tempi lunghi (ore o addirittura... giorni!) nella

generazione dei file (vedi anche quanto raccontato al paragrafo *Il “problema” delle vincite concomitanti* a pag. 39).

- L'indicazione dell'ordinamento in cui le serie di cartelle sono state disposte nel file. TombolaCards, infatti, prima di salvare un file dispone le serie di cartelle in ordine crescente del valore di MAXEPC<sup>5</sup>. In questo modo, le cartelle che hanno meno numeri in comune con le altre, sono le prime nel file.
- Seguono alcune righe di sintesi che riportano quante cartelle ci sono nel file che hanno al massimo N numeri uguali tra loro, con N che assume tutti i casi realmente esistenti nel file.
- Analogamente a quanto descritto al punto precedente, seguono le righe di sintesi che indicano quante cartelle ci sono nel file che hanno righe con al massimo N numeri uguali ad un'altra riga di una cartella del medesimo file, con N che assume tutti i casi realmente esistenti nel file.

La tabella seguente riepiloga le 4 varianti supportate, evidenziandone le principali differenze.

Tipologia File	Valori rappresentati per ogni cartella	Valori statistici MAXEPC, MAXEPR in coda ad ogni riga/cartella
CSV	27	NO
CSV_PLUS	27	SI
CSV_PACKED	15	NO
CSV_PACKED_PLUS	15	SI

*Tabella 3 – Riepilogo dei formati file disponibili in TombolaCards*

Le due figure seguenti illustrano ancor meglio come sono composte le righe di ciascun file per le due classi di tipologie di file (*non-PACKED* e *PACKED*).

---

<sup>5</sup> Come valore MAXEPC di una serie di cartelle si considera il valore massimo MAXEPC tra le 6 cartelle della serie. Analogo discorso per il valore MAXEPR di una serie.

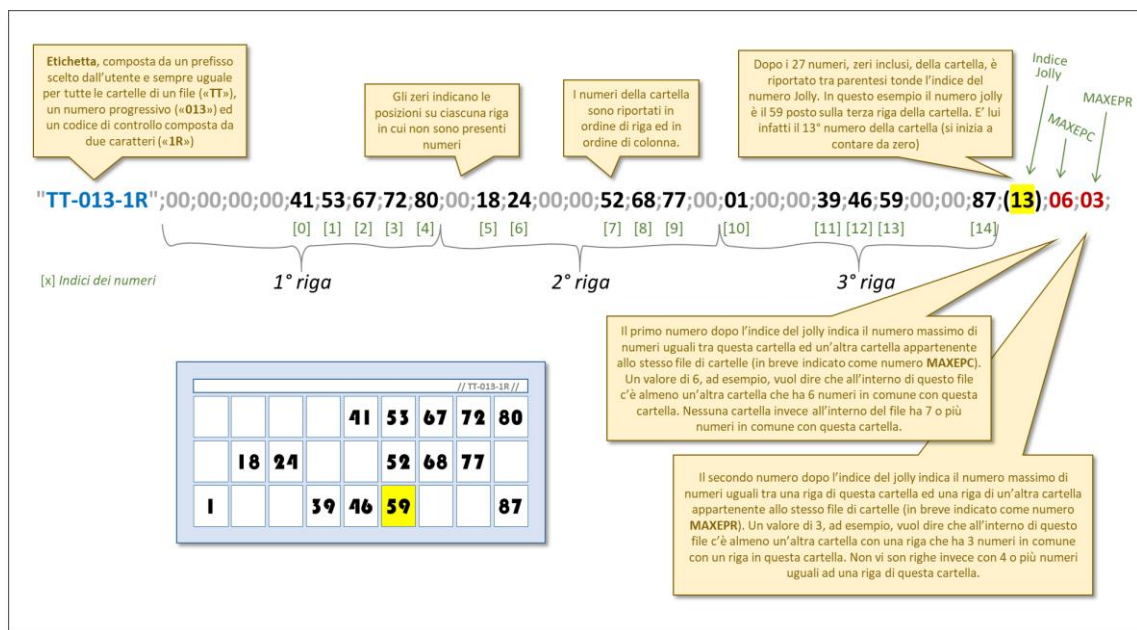


Figura 8 – Descrizione del contenuto di una riga di un file di cartelle in formato CSV\_PLUS (il formato CSV differisce solo per l'assenza degli ultimi due valori statistici, MAXEPC e MAXEPR, in coda ad ogni riga).

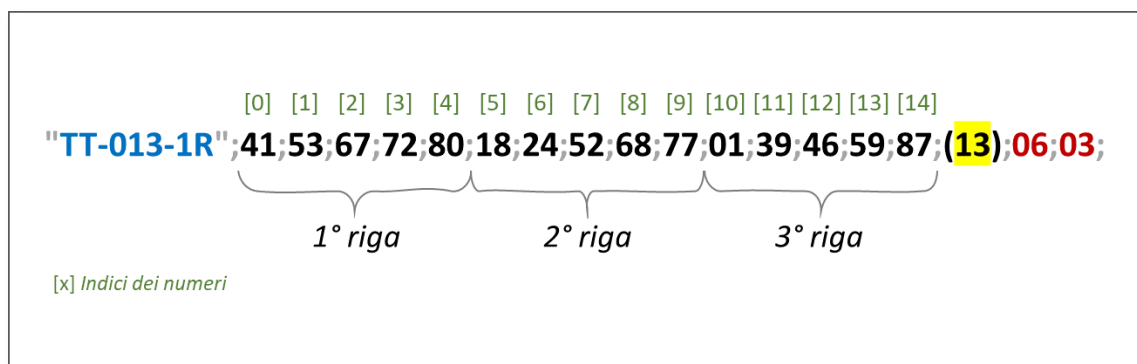


Figura 9 – Contenuto della stessa cartella della figura precedente, questa volta salvata in un file in formato CSV\_PACKED\_PLUS (il formato CSV\_PACKED differisce solo per l'assenza degli ultimi due valori statistici, MAXEPC e MAXEPR, in coda ad ogni riga).

## Modalità “interattiva”

TombolaCards può essere utilizzato anche in un modo più interattivo. Se sulla riga di comando aggiungiamo l'opzione “-i”, TombolaCards non inizia subito a generare cartelle, ma presenta all'utente un messaggio di benvenuto e si mette in ascolto di comandi interattivi che possono essere dati per procedere un passo alla volta nel lavoro che si vuole portare a termine.

```
> java -jar TombolaCards.jar -i
```

```

Hello Tombola World!
TombolaCards version: 1.1
Library: TombolaLib, version 1.1
-----

Interactove mode - Type HELP to get hints on available interactive commands.

>>

```

Come già fatto in precedenza, in questo paragrafo si introducono i comandi più usati in modo da poter rapidamente lavorare e sperimentare in autonomia. Nel prossimo paragrafo, invece, una tabella riepilogherà tutti i possibili comandi per soddisfare la curiosità degli utenti più esigenti.

Il primo comando introdotto è quello che stampa la guida a video di questa modalità interattiva, ed in particolare stampa l'elenco di tutti i comandi disponibili.

```

>> help
-----
Here we are! The HELP command you have just typed is here to give you hints
on how to use the TombolaCards interactive mode. But you should provide the
command for which you need help just after the HELP keyword.
Available commands are (it doesn't matter if you write in lower or
uppercase):
CLEAR, CLS, DEL or DELETE, ENV, EXIT or QUIT, HELP or ?, READ, RUN, SAVE,
SET,
SHOW, STATUS, STOP.
-----

```

I comandi possono essere digitati sia in maiuscolo che in minuscolo, non c'è differenza, e possono eventualmente essere seguiti da "parametri", ovvero altre informazioni aggiunte dopo il nome del comando stesso. Il comando help, ad esempio, consente di specificare come parametro il nome di un altro comando, in modo da stampare a video un messaggio di aiuto specifico per tale comando. Se scriviamo `help env`, ad esempio:

```

>> help env
-----
Display on screen one, more or all parameters present in memory and their
respective values.

Usage: ENV | ENV <paramName> | ENV <param first letters>

Use just env to list all parameters, use a param name to check just its
value,
use few characters to check values for all parameter starting with those
chars.
-----

```

**Env** è un comando fondamentale. Proviamolo subito, senza specificare ulteriori parametri:

```
>> env

Current environment parameters:
avoidEmptyColumn:      true
cardLabelChecksum:     true
cardLabelMode:         BYCARDS
cardLabelPrefix:       TT
cardLabelSeparator:    -
defaultSeriesTitle:    CardSeries001
desiredSeries:         10
fileFormat:            CSV_PLUS
fileName:              CardSeries001.csv
fileOverwrite:         false
interactionLimit:      500000000
maxepc:               15
maxepr:               5
method:               RANDOM
randomSeed:           null
timeLimit:            172800000
traceLogFileName:     TombolaCards.log
unattended:           false
useJolly:             true
verbose:              false

There are 0 series in memory.
Series factory engine has not been initialized yet.
Use the RUN command to start the engine.
```

Come vedete, **env** stampa a video una serie di variabili di ambiente, seguite dal loro corrispettivo valore. Le variabili di ambiente rappresentano gli elementi che utilizza TombolaCards per generare le cartelle quando si è in modalità interattiva. Ad esempio, il parametro **desiredSeries** ha un valore di 10: vuol dire che se iniziamo a generare le cartelle, TombolaCards genererà 10 serie, 60 cartelle. Le cartelle verranno generate con etichette che iniziano con “TT”, giacché al momento è “TT” il valore della variabile di ambiente **cardLabelPrefix**. E se decidiamo di salvare le cartelle, il file che verrà creato si chiamerà **CardSeries001.csv** perché questo è il valore della variabile **fileName**. È così via per tutte le variabili di ambiente che vedete in figura. Per modificarne il valore si utilizza un altro comando, il comando **set**. Portiamo a 20 il numero di serie che vogliamo generare ed impostiamo a PROGRESSIVE il metodo che vogliamo usare.

```
>> set desiredSeries 20
<< <OK!> Desired Number of Series has been set to 20 (120 cards).
>> set method PROGRESSIVE
<< <!> Set Factory method changed to [PROGRESSIVE].
```

Molto facile. **Set** si utilizza facendolo seguire dal nome della variabile di cui vogliamo modificare il valore e poi dal nuovo valore che vogliamo assegnarle. Come altro esempio, avendo scelto il metodo progressive per generare le cartelle, e cioè il metodo che verifica se le cartelle generate hanno troppi numeri in comune, modifichiamo anche i valori delle variabili di controllo MAXEPC e MAXEPR.

```
>> set maxepc 7
<< <OK!> maxepc set to 7
>> set maxepr 3
<< <OK!> maxepr set to 3
```

A questo punto direi che possiamo avviare la generazione delle cartelle. Il comando da usare è **run**.

```
>> run
<< Preparing the series set factory...
<< Process Started. Hit enter or use the STATUS command to check progress
    Good Luck!
```

**Run** avvia internamente il processo che genera le cartelle. Per vedere a che punto è tale processo basta battere invio sulla riga di comando, oppure usare il comando **status**.

```
>> <ENTER>
<< There are 20 series (120 cards) in memory. Set Factory is COMPLETED
<< Generation algorithm has come to an end.
```

```
>> status
<<
<< <-> Generation algorithm Control & Add Factory 1.0 is COMPLETED.
<< <-> There are 20 series (120 cards) in memory, versus 20 series (120
    cards) to be generated (100,0%).
<< <-> Max equal number per card limit: 7
<< <-> Max equal number per row limit: 3
<<
<< <!> Attention! There are unsaved cards in memory!
```

Non ci resta che salvare sul file le cartelle generate. Il comando da usare è **save** e se lo lanciamo senza altri parametri otteniamo il salvataggio nel file **CardSeries001.csv**. Possiamo però specificare un nuovo nome per il file che creiamo semplicemente specificandolo dopo il comando **save**.

```
>> save CartelleNatale2020.csv
<< Updating output filename from [CardSeries001.csv] to
    [CartelleNatale2020.csv]
<< Going to save 20 series (120 cards) on file CartelleNatale2020.csv
    using CSV_PLUS format.
<< <OK!> File has been properly saved.
```

Abbiamo visto sommariamente un flusso di lavoro tipico di TombolaCards in modalità interattiva. Il bello, appunto, è dato proprio dalla possibilità di continuare a sperimentare in modo incrementale rispetto ad un primo run di generazione delle cartelle. Sono disponibili comandi per eliminare dalle cartelle prodotte una specifica serie e sostituirla, oppure è possibile provare etichette differenti, o aggiungere ulteriori serie a quelle precedentemente create, etc.

Nel seguente paragrafo sono riportate le tabelle di sintesi che riepilogano tutti i comandi e tutte le variabili di ambiente con i quali è possibile agire in modalità interattiva.



## Riepilogo dei comandi in modalità interattiva

Le tabelle che seguono riportano, in ordine alfabetico, tutti i comandi e tutte le variabili di ambiente che sono disponibili nella modalità interattiva di TombolaCards. Ricordate che in modalità interattiva è sempre disponibile il comando **help** per rileggere direttamente a video le indicazioni più essenziali senza dover ricorrere a questo lungo manuale per poter proseguire.

Comando	Sintassi	Descrizione
CLS	-	Pulisce lo schermo (o la finestra a caratteri entro la quale è in esecuzione TombolaCards) e riporta il cursore di inserimento in alto a sinistra. Utile se si vuole, ad esempio, 'fare spazio' prima di visualizzare l'elenco delle variabili con <code>env</code> , oppure le cartelle in memoria, etc.  Si utilizza senza specificare altri parametri sulla riga di comando.
DELETE	<code>DEL [nSeries   ALL   firstSeries lastSeries]</code>	Elimina, una, più o tutte le serie presenti in memoria. DEL (o Delete) seguito da un solo numero N elimina la N-esima serie in memoria, iniziando a contare da zero. Se vengono specificati 2 numeri, A e B, DEL elimina tutte le serie da quella in posizione A a quella in posizione B (sempre iniziando a contare da zero). Il valore A deve essere minore o uguale al valore B. Se si specifica come parametro "ALL", tutte le serie in memoria vengono eliminate.  Il comando DEL chiede sempre conferma prima di eliminare le cartelle (occorre rispondere Yes per confermare, No per annullare).  Una volta eliminate, le cartelle non possono essere più recuperate (a meno, ovviamente, siano state salvate precedentemente in un file).  Se lanciato senza ulteriori parametri, DEL stampa a video un messaggio di errore che visualizza la sintassi corretta.
ENV	<code>ENV   ENV paramName   ENV param first letters</code>	Visualizza i nomi ed i corrispettivi valori correnti di tutte le variabili di ambiente utilizzate da TombolaCards in modalità interattiva.  Se lanciato senza ulteriori parametri, ENV visualizza l'intera lista di variabili, seguita dai valori correntemente configurati. Se lanciato seguito dal nome di un parametro, ENV visualizza soltanto il valore corrente di quel parametro.  ENV può essere utilizzato anche specificando come parametro le prime lettere di eventuali variabili di ambiente: in questo modo vengono visualizzati i valori per tutte le variabili che iniziano con le lettere specificate (ENV m, ad esempio, visualizza i valori delle variabili MAXEPC, MAXEPR, METHOD).
EXIT	-	Consente di terminare l'esecuzione interattiva di TombolaCards e di riportarsi alla console di comando del sistema operativo da cui si era partiti.  Se vi sono cartelle in memoria, viene richiesta conferma con un messaggio al quale occorre rispondere YES per uscire o NO per annullare. Si utilizza senza specificare altri parametri sulla riga di comando. È possibile utilizzare anche il comando QUIT come sinonimo.
HELP	<code>HELP   HELP commandName</code>	Visualizza una sintetica guida a video per i comandi disponibili in modalità interattiva. Lanciato senza altri parametri fornisce indicazioni generali e poi l'elenco di tutti i comandi disponibili.  Lanciato seguito da un nome di comando (ad es. HELP RUN) fornisce una sintetica guida all'uso del comando specificato, comprensiva della sintassi con cui il comando va utilizzato (parametri richiesti, etc.).
LABEL	-	Rigenera le etichette identificative e le riapplica alle cartelle in memoria, utilizzando i valori definiti dalle variabili d'ambiente <code>cardLabelMode</code> , <code>cardLabelPrefix</code> , <code>cardLabelSeparator</code> e <code>cardLabelChecksum</code> (cfr. anche informazioni riportate nella seguente Tabella 5 per tali variabili di ambiente).

Comando	Sintassi	Descrizione
		<p>Lanciando questo comando si sovrascrivono eventuali etichette precedentemente assegnate alle cartelle, senza possibilità di recuperarle.</p> <p>Il comando LABEL è implicitamente richiamato quando si generano nuove cartelle, prima del salvataggio del file. Va quindi utilizzato esplicitamente quando si cambia idea rispetto all'etichettatura iniziale, o quando si "fondono" serie di cartelle provenienti da file diversi (comando READ).</p>
READ	READ <i>fileName</i>	<p>Legge il file di cartelle specificato, controlla la validità formale delle serie di cartelle e le carica in memoria, in modo che possano essere utilizzate come punto di partenza per ulteriori successive elaborazioni (cfr. comando RUN).</p> <p>Il parametro <i>fileName</i> specifica il <i>path</i>, assoluto o relativo alla cartella corrente, del file da caricare (ad es. CartelleNuove.csv, C:\tombola\CartelleNuove.csv).</p>
RESET	-	<p>Riporta tutte le variabili di ambiente ai loro valori iniziali e cancella, se presenti, eventuali cartelle in memoria. Il comando può essere lanciato solo se non vi è un processo di generazione di cartelle in corso.</p>
RUN	-	<p>Questo comando avvia il processo di generazione delle cartelle. RUN genera nuove serie di cartelle se il numero di serie che sono correntemente in memoria (cfr. comando STATUS) è inferiore al numero di serie che si desidera (cfr. variabile di ambiente <i>desiredSeries</i>). Per la generazione delle cartelle vengono utilizzati i valori correntemente impostati per diverse variabili di ambiente, come ad es. <i>method</i>, <i>maxpec</i>, <i>maxper</i>, <i>allowEmptyColumn</i>, etc.</p> <p>Il comando RUN non blocca l'interattività di TombolaCards, e cioè consente all'utente di continuare ad interagire ed inserire ulteriori comandi anche durante la generazione di cartelle. Per visualizzare lo stato del processo di generazione delle cartelle è possibile utilizzare il comando STATUS oppure battere semplicemente invio (comando vuoto) per avere informazioni sintetiche sul processo in corso.</p> <p>Il comando RUN si utilizza senza specificare altri parametri sulla riga di comando.</p>
SAVE	SAVE   SAVE <i>fileName</i>   SAVE <i>filename</i> <i>fileFormat</i>	<p>Salva su file le cartelle che sono correntemente in memoria. Se lanciato senza parametri, utilizza i valori correnti delle variabili di ambiente <i>filename</i> e <i>fileFormat</i>. Diversamente, è possibile specificarle come parametri aggiuntivi sulla riga di comando.</p> <p>Il comando SAVE può essere utilizzato solo quando il processo di generazione delle cartelle non è in esecuzione, quindi o è terminato oppure è stato messo in pausa con il comando STOP.</p>
SET	SET <i>paramName</i> <i>newParamValue</i>	<p>Consente di modificare il valore di una delle variabili di ambiente utilizzate in modalità interattiva. Per farlo occorre specificare, sulla riga di comando, sia il nome della variabile che si vuole modificare, sia il nuovo valore che le si vuole assegnare.</p> <p>Il comando SET può essere utilizzato solo quando il processo di generazione delle cartelle non è in esecuzione, quindi o è terminato oppure è stato messo in pausa con il comando STOP.</p> <p>Se non si specifica un nuovo valore, ma soltanto il nome di una variabile di ambiente, il comando SET si comporta un po' come il comando ENV e ne visualizza quindi il valore corrente.</p>
SHOW	SHOW ALL   S<nSeries>   S<nStart> S<nEnd>   C<nCard>   C<nStart> C<nEnd>	<p>Stampa a video le serie e/o le cartelle attualmente in memoria.</p> <p>Se lanciato senza ulteriori parametri, SHOW mostra un messaggio di errore e si ferma.</p> <p>È possibile specificare le cartelle da visualizzare sia facendo riferimento ad una numerazione progressiva per cartella, sia ad una numerazione progressiva per serie, anteponendo ai numeri indici da visualizzare una C per la numerazione per cartella o una S per una numerazione per serie:</p> <ul style="list-style-type: none"> <li>SHOW S1 visualizza le 6 cartelle appartenenti alla seconda serie in memoria (la prima serie è S0);</li> </ul>

Comando	Sintassi	Descrizione
		<ul style="list-style-type: none"> <li>• SHOW S2 S4 visualizza le serie 2,3,4 in memoria.</li> <li>• SHOW C0 visualizza la prima cartella in memoria.</li> <li>• SHOW C13 visualizza la quattordicesima cartella in memoria, ovvero la seconda cartella della terza serie di cartelle in memoria</li> <li>• SHOW C5 C9 visualizza l'ultima cartella della prima serie in memoria e le prime 4 cartelle della seconda serie.</li> <li>• SHOW ALL visualizza tutte le cartelle in memoria.</li> </ul>
STATUS	-	Visualizza lo stato corrente del processo di generazione delle cartelle
STOP	-	<p>Con questo comando puoi fermare (mettere in pausa) l'esecuzione dell'algoritmo che genera le cartelle.</p> <p>È un comando utile soprattutto quando le cartelle da generare sono tante e magari i parametri MAXEPC e MAXEPR impostati per il metodo progressive richiedono tempi lunghi per generare tutte le serie di cartelle desiderate.</p> <p>Una volta messo in pausa l'algoritmo di generazione, è possibile applicare le etichette alle cartelle generate (comando LABEL), salvarle su file (comando SAVE), modificare il numero massimo di cartelle che si vuole creare (comando SET desiredSeries), etc.</p> <p>Il processo di generazione delle cartelle può riprendere lanciando nuovamente il comando RUN. Il comando STOP si utilizza senza specificare altri parametri sulla riga di comando.</p>

Tabella 4 – I Comandi di TombolaCards disponibili in modalità Interattiva.

## Riepilogo delle variabili di ambiente in modalità interattiva

Variabile	Valore predefinito	Descrizione
avoidEmptyColumn	ON	Se lasciato ad ON, le cartelle che vengono generate da TombolaCards non presenteranno mai una colonna della griglia totalmente vuota. Se invece questo valore viene posto ad OFF, tale controllo viene disabilitato e durante la generazione di cartelle è possibile che ne vengano preparate con colonne vuote. Si tratta di una questione solo estetica, ai fini del gioco l'importante è che su ogni riga ci siano sempre 5 numeri ben disposti, condizione che è sempre verificata sia con colonne vuote che senza.
cardLabelChecksum	ON	Se lasciato al valore di ON, una coppia di caratteri di controllo ( <i>checksum</i> ) viene aggiunta in coda ad ogni etichetta di cartelle generata (cfr. quanto riportato relativamente al parametro a line a di comando -LK).
cardLabelMode	BYCARDS	<p>Questa variabile specifica il modo in cui le cartelle vengono numerate progressivamente. Il numero di ciascuna cartella è poi utilizzato per formare l'etichetta della cartella stessa (vedi opzione -L più sopra). Sono possibili due tipi di numerazione: "per cartella" (valore dell'opzione CARDS) e "per serie" (valore dell'opzione SERIES). Nel primo caso il numero progressivo assegnato ad ogni cartella è un progressivo incrementato di uno dalla prima all'ultima cartella del file. Nel secondo caso si numerano progressivamente le serie, non le cartelle. Le cartelle vengono invece numerate da 1 a 6 all'interno di ciascuna serie.</p> <p>Vediamo un po' di etichette di esempio nei due casi (per semplicità usiamo come prefisso "AA" ed omettiamo invece il checksum):</p> <p>BYCARDS → AA.001, AA.002, AA.003, AA.004, ..., AA.nnn</p>

Variabile	Valore predefinito	Descrizione
		BYSERIES → AA.S001.C1, AA.S001.C2, AA.S001.C3, ... AA.S001.C6, AA.S002.C1, AA.S002.C2, AA.S002.C3, ..AA.S002.C6, ..., AA.Snnn.C1, ..., AA.Snnn.C6.
cardLabelPrefix	TT	Questa variabile d'ambiente consente di specificare il prefisso che si vuole per le etichette dell'intero file di cartelle (vedi anche l'opzione "-L" nella tabella delle opzioni per la modalità a riga di comando).
cardLabelSeparator	"_"	Questa variabile d'ambiente consente di specificare il prefisso che si vuole per le etichette dell'intero file di cartelle (vedi anche l'opzione "-L" nella tabella delle opzioni per la modalità a riga di comando).
defaultSeriesTitle	CardSeries001	Contiene il "nome" dell'intero set di cartelle che viene generato. Questo nome è scritto all'interno del file che viene generato con il comando SAVE, nelle prime righe di intestazione.
desiredSeries	10	Consente di impostare il numero di serie totali che si vogliono generare con il comando RUN (ricorda, ogni serie contiene 6 cartelle). Va quindi impostato al valore desiderato quando l'algoritmo di generazione delle cartelle non è in esecuzione, per cui prima di lanciare il comando RUN o dopo che una prima esecuzione è terminata o è stata volontariamente messa in pausa con il comando STOP.
fileFormat	CSV_PLUS	Questa variabile consente di specificare il formato con cui vengono salvate le cartelle all'interno del file. Ad oggi sono supportate quattro modalità di salvataggio leggermente diverse tra loro, tutte però che basate sulla consuetudine CSV, <i>Comma Separated Value</i> . I formati differiscono tra loro perché alcuni memorizzano solo i 15 numeri di ciascuna cartella, omettendo il salvataggio delle posizioni vuote, altri invece inseriscono zeri anche per le caselle vuote. Alcuni formati aggiungono informazioni statistiche, altri no. Una descrizione completa e dettagliata di ciascun formato è riportata al paragrafo "I formati dei file di cartelle" a pag. 26 I valori possibili per questa variabile sono CSV, CSV_PLUS, CSV_PACKED, CSV_PACKED_PLUS.
fileName	CardSeries001.csv	Contiene il nome nel file nel quale verranno salvate le cartelle utilizzando il comando SAVE, a meno che non si specifichi un nome diverso proprio sulla riga del comando SAVE.
fileOverwrite	OFF	Se valorizzata ad ON, questa variabile forza TombolaCards a scrivere il file di output contenente le cartelle generate anche se un file con lo stesso nome esiste già, per cui deve essere sovrascritto. Se la variabile ha valore "OFF", TombolaCards non sovrascrive file esistenti, ma si ferma visualizzando un messaggio di errore. Utilizza dunque questa opzione se sei sicuro che non ti importa di perdere file precedentemente salvati.
interactionLimit	500.000.000	Questo parametro controlla il numero massimo di serie di cartelle che possono essere generate (e scartate) in un ciclo di RUN. È un parametro "di sicurezza", che impedisce processi di generazione di cartelle troppo lunghi o addirittura che non riescono a terminare. Tale eventualità si può verificare quando si utilizza il metodo progressive con parametri MaxEPC e MaxEPR troppo esigenti.
Maxepc	15	Questo parametro ha effetto solo quando l'algoritmo utilizzato per la creazione delle cartelle è quello denominato PROGRESSIVE (vedi l'uso della variabile METHOD più avanti). Specifica il valore massimo consentito di numeri che una cartella può avere in comune con un'altra cartella dello stesso file.

Variabile	Valore predefinito	Descrizione
		<p>Il valore minimo consentito è 5, il valore massimo che ha senso è 14: valori superiori sono consentiti, ma equivalgono a non applicare alcun controllo.</p> <p>Il modo ottimale di utilizzare ed assegnare valori a questo parametro, peraltro senza rendere il processo di generazione delle cartelle estremamente lungo ed oneroso, è raccontato con maggiore dettaglio al successivo paragrafo <i>Il “problema” delle vincite concomitanti</i> a pag. 39.</p>
Maxper	5	<p>Questo parametro è preso in considerazione solo quando l’algoritmo utilizzato per la creazione delle cartelle è quello denominato PROGRESSIVE (vedi l’uso della variabile METHOD più avanti).</p> <p>Specifica il valore massimo consentito di numeri uguali che possono esserci tra due righe qualsiasi di due cartelle dello stesso file.</p> <p>Il valore minimo consentito è 2, il valore massimo che ha senso è 4: valori superiori sono consentiti, ma equivalgono a non applicare alcun controllo.</p> <p>Il modo ottimale di utilizzare ed assegnare valori a questo parametro, peraltro senza rendere il processo di generazione delle cartelle estremamente lungo ed oneroso, è raccontato con maggiore dettaglio al successivo paragrafo <i>Il “problema” delle vincite concomitanti</i> a pag. 39.</p>
Method	RANDOM	<p>Indica il metodo (euristico) utilizzato per generare le cartelle. Al momento è possibile scegliere tra due metodi diversi:</p> <ul style="list-style-type: none"> <li>• <b>RANDOM</b>: genera le cartelle scegliendo i numeri da disporre nelle varie caselle in modo del tutto casuale, senza nessuna logica o controllo particolare. È il metodo più “veloce” per generare le cartelle.</li> <li>• <b>PROGRESSIVE</b> questo metodo genera le cartelle cercando di evitare che nel file ve ne siano con troppi numeri in comune. In pratica, ogni volta che genera una nuova serie, prima di aggiungere quest’ultima al file, controlla che non contenga cartelle con <i>troppi</i> numeri uguali alle cartelle già presenti nel file stesso. Per determinare se i numeri in comune sono troppi si utilizzano i valori impostati utilizzando le opzioni -MPR ed MPC descritte più avanti. La logica di questo metodo, i vantaggi che se ne possono avere meritano di essere descritti con un po’ di spazio in più. Trovi questo approfondimento al paragrafo <i>Il “problema” delle vincite concomitanti</i> più avanti a pag. 39.</li> </ul>
randomSeed	<casuale>	<p>Questa variabile di ambiente contiene il “seme” che inizializza il generatore di numeri pseudocasuali utilizzato dal processo di generazione delle cartelle.</p> <p>Tipicamente, questo valore è inizializzato con un metodo a sua volta pseudocasuale, come ad esempio la lettura dei millisecondi dell’orologio interno del pc. Questo succede di norma quando non si assegna esplicitamente a randomSeed alcun valore, in modo da generare sequenze di cartelle sempre diverse ad ogni esecuzione di TombolaCards.</p> <p>Se, per qualche motivo, si vuole rigenerare esattamente la stessa sequenza di cartelle già generate in precedenza, è possibile utilizzare questa variabile per forzare l’utilizzo di uno preciso seme.</p>
timeLimit	172.800.000 (millisecondi, ovvero 48 ore, 2 giorni)	<p>Un po’ come il parametro iterationLimit descritto più sopra, questa variabile fissa un tempo massimo oltre il quale un singolo processo di RUN viene interrotto anche se non ha completato la generazione del numero di cartelle desiderate.</p> <p>È un parametro “di sicurezza”, che impedisce processi di generazione di cartelle troppo lunghi o addirittura che non riescono a terminare. Tale eventualità si può verificare quando si utilizza il metodo PROGRESSIVE con parametri MaxEPC e MaxEPR troppo esigenti.</p>

Variabile	Valore predefinito	Descrizione
traceLogFileName	TombolaCards.log	È il nome del file di log che viene salvato da TombolaCards per tracciare informazioni di dettaglio relativi alla generazione delle cartelle.
UseJolly	ON	Questa variabile consente di attivare (ON) o disattivare (OFF) la predisposizione dei numeri Jolly per le cartelle generate (per maggiori dettagli sui numeri Jolly consulta il paragrafo “I numeri Jolly!” a pag. 15). Il valore predefinito è ON, i numeri Jolly vengono sempre scelti per le cartelle, l’operazione non costa nulla e successivamente, ad esempio durante l’uso con TombolaPrint o TombolaWeb, possono sempre essere ignorati.
verbose	OFF	Questa variabile aumenta il numero di informazioni e di dettagli che vengono presentati nei messaggi a video durante l’elaborazione delle cartelle. È utile attivarla (valore ON) quando non si comprende appieno il perché dei risultati ottenuti e si vuole capire meglio la logica di funzionamento del programma.

*Tabella 5 – Tutte le variabili di ambiente di TombolaCards disponibili in modalità Interattiva.*

## Il “problema” delle vincite concomitanti



**Attenzione! Questo paragrafo è consigliato solo ai veri *Tombola addicted*, i più curiosi in merito ai meccanismi interni di TombolaCards. Se vuoi usare TombolaCards *solo* per generare cartelle e TombolaPrint per stamparle... non è assolutamente necessario leggere questo paragrafo.**

C'è un aspetto molto importante da considerare quando il numero di giocatori per una partita a Tombola aumenta. Ed è quello dell'aumento delle probabilità che si verifichino vincite *concomitanti*, ovvero che più giocatori, con cartelle diverse, facciano contemporaneamente, e cioè all'estrazione del medesimo numero dal bussolotto, ambo, terno... un premio qualsiasi.

Se organizziamo una tombolata di quartiere, in cui sono in gioco diverse decine di cartelle, magari centinaia... la probabilità che vi siano 3 o 4 giocatori che esclamano “AMBO!” all'unisono già all'estrazione del secondo numero... non sono per nulla trascurabili.

Perché si verifica questo fenomeno? Semplice: perché le cartelle del gioco della tombola **devono avere numeri in comune**. Complessivamente abbiamo 90 numeri a disposizione ed ogni cartella è composta da 15 numeri. Quindi sono possibili solo sei cartelle (una serie!) senza dover riusare due volte lo stesso numero. **Se al gioco aggiungiamo una settima cartella, quest'ultima avrà di sicuro numeri uguali ai numeri su qualcuna delle altre cartelle.**

Le due figure a seguire mostrano, per esser ancor più chiari, due cartelle che non hanno numeri in comune, e due cartelle che invece ne hanno un bel po'.

// AA-001-X //														
1		22			52	69	71							
9	11		36		55		77							
	19	24		46	64								87	

// AA-002-Y //														
	15		30		57		74	84						
	16	27		49		65		88						
2	18		34		58		75							

Figura 10 – Esempio di due cartelle che non hanno numeri in comune.

// AA-034-X //														
	16	21	33		51		75							
3		25		45		64		81						
7	19		39		55		78							

// AA-046-Y //														
	16		33		51		72	85						
7		26		45		63		89						
	19		39		57	64	77							

Figura 11 – Due cartelle con 8 numeri in comune, segnati con cerchietti colorati: La prima riga della prima cartella ha 3 numeri in comune con la prima riga della seconda cartella; la seconda riga della prima cartella ha un numero in comune con la seconda riga ed un numero in comune con la terza riga; la terza riga della prima cartella ha un numero in comune con la seconda riga e due numeri in comune con la terza riga della seconda cartella.

Ed ecco la domanda da cento milioni di dollari: **Esiste un modo per generare insiemi di cartelle che riducano il più possibile la probabilità che vi siano vincite concorrenti?** Questa domanda mi ha accompagnato per un bel po' di tempo e... non ho saputo trovarvi risposta in modo scientifico, matematico. Sono troppo ignorante... e troppo pigro anche.

L'unica idea, deboluccia per la verità, è stata quella di provare a generare serie di cartelle che non avessero "troppi" numeri in comune, sia tra una cartella ed un'altra, sia tra una riga di una cartella ed una riga di un'altra cartella. In altri termini ho pensato che più i numeri fossero "distribuiti uniformemente" più la probabilità che ci fossero vincite concorrenti sarebbe stata contenuta. È solo una "sensazione", una ipotesi molto debole... che, se ci pensate bene anche voi, regge poco anche solo da un punto di vista intuitivo.

Ed in tutto questo tempo non c'è stata ricerca su Google, domanda su forum, fortuito paragrafo ritrovato in un libro di statistica o altro che sia riuscito ad illuminarmi.

Come verificare questa ipotesi? Come avere almeno un qualche "conforto numerico" che cartelle ben assortite possano permettere tombolate di quartiere o di parrocchia più serene, con meno trambusti, meno spargere estemporanei e speranze di vittoria tradite all'ultimo momento per l'esito di un sorteggio?

Ho pensato ad una scorciatoia: **usare il computer per fare simulazioni di partite di Tombola**, fare tante simulazioni utilizzando insiemi di cartelle diversi e... vedere cosa succede in termini di vincite concomitanti (per brevità, d'ora in avanti, potrei chiamare le vincite concomitanti utilizzando anche il termine "conflitti").

La libreria **TombolaLib** di cui vi parlavo al primo capitolo contiene una classe, **TGame**, che consente di simulare al computer una partita di Tombola. In pratica fa tutto il computer: estrae i numeri in modo pseudocasuale da un bussolotto elettronico, controlla tutte le cartelle in gioco ed assegna i premi, fino a quando su qualche cartella non si fa l'ultima Tombola<sup>6</sup> in palio e la partita finisce.

Iniziamo a fare delle prove. Facciamo simulare al computer 10.000 partite di tombola. Usiamo per tutte e 10.000 le partite le stesse 48 cartelle, e cioè 8 serie di cartelle, copiate da un gioco della tombola tradizionale, fisico, quello che c'è a casa dei miei genitori almeno da una quarantina d'anni. **TGame** ci fa scegliere con quali premi vogliamo giocare e, per tenere le cose semplici, giochiamo usando un insieme di cinque premi base: un ambo, un terno, una quaterna, una quintina ed una tombola.

Tanto per iniziare a vedere come funziona *la legge dei grandi numeri*, per ognuna delle 10.000 partite giocate, possiamo registrare l'estrazione alla quale viene assegnato ciascun premio. Il grafico che segue illustra questi dati, ovvero illustra come si distribuiscono mediamente le varie vincite man mano che i numeri vengano estratti. Facciamo qualche considerazione osservando il grafico, così prendiamo dimestichezza con questo strumento:

- Dopo il primo numero estratto, ovviamente, non ci sono vincite. È infatti impossibile vincere con un solo numero estratto (il premio "Umbo" non esiste 😊);
- Con due numeri estratti, ci sono quasi 3.000 "Ambi" vinti. Considerando che stiamo vedendo le somme di 10.000 partite... e che quindi ci sono 10.000 ambi da fare... si tratta del 30% circa degli ambi;
- Al terzo numero estratto si aggiunge un altro bel 37% di ambi e... già qualche primo terno! (pochi per la verità, rappresentati dalla fettina sottile arancione in cima alla alta barra blu degli ambi);
- Tra il quinto numero estratto ed il sedicesimo si stende la macchia arancione dei 10.000 terni, il grosso dei quali viene vinto tra la nona e la decima estrazione. Intanto, qualche altra cartella inizia a fare quaterne, occhio alle barrette grigie in cima a quelle arancioni;
- Con un andamento più distribuito sulle varie estrazioni seguono le grigie quaterne e le gialle quintine... queste ultime possono considerarsi tutte vinte dalla quarantottesima estrazione.

---

<sup>6</sup> Ti starai chiedendo come si comporta il simulatore in caso di vincite concomitanti. Bene, TombolaLib prevede diverse possibilità: o il premio viene automaticamente assegnato tra le cartelle contendenti, scegliendone una a caso, oppure la partita si interrompe e viene chiesto ad un utente di indicare a quale cartella assegnare il premio.



- Dopo una valle di silenzio... che tanto mi fa venire in mente la fase più noiosa delle partite a tombola, quella in cui tutti sbadigliano, sonnecchiano, senza vincere nulla, ecco emergere la collinetta delle tombole. Fatta salva qualche eccezione statisticamente poco rilevante, il grosso delle tombole viene vinto tra il cinquantatreesimo ed il settantottesimo numero estratto. Il “picco” di vincite lo si ha intorno al settantaduesimo numero estratto. Ovviamente le nostre sono considerazioni statistiche... e non mancano le eccezioni, come tombole vinte già dalla 42-esima estrazione.

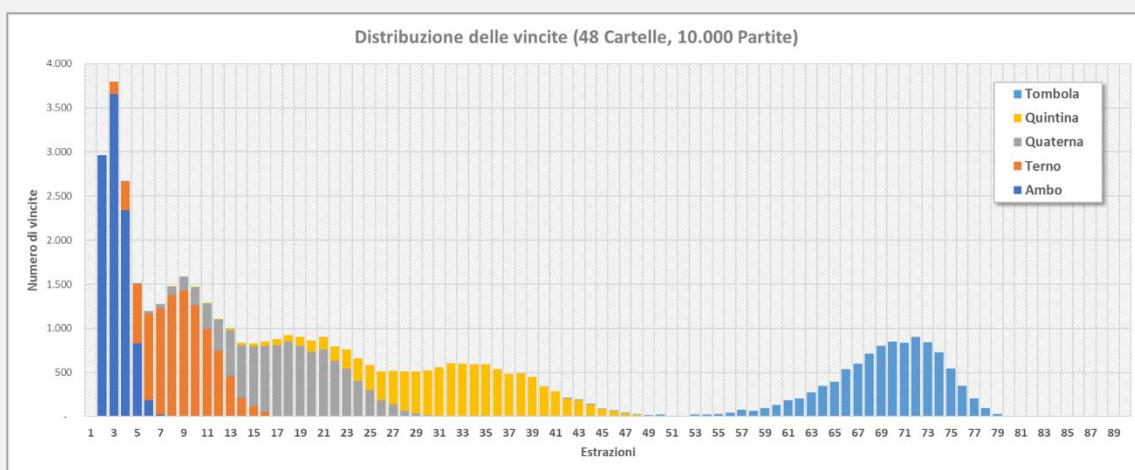


Figura 12 – Distribuzione delle vincite di 10.000 partite giocate con 48 cartelle e 5 premi disponibili per ogni partita (1 ambo, 1 terno, 1 quaterna, 1 quintina, 1 tombola).

Spero tu sia riuscito ad orientarti sul grafico di Figura 12 e ad aver capito in che modo sono valorizzate le barrette verticali. Prenditi il tempo che serve nel caso e... proseguiamo solo quando è tutto chiaro.

Tutto chiaro? Bene. Rifacciamo allora la stessa identica cosa, simuliamo 10.000 nuove partite, mettendo in gioco non più 48 cartelle, ma un numero significativamente più alto, diciamo 240 cartelle (40 serie di cartelle). Questa volta le cartelle sono state generate casualmente con TombolaCards, in fondo è proprio per questo che è stato scritto TombolaCards! E poi in giro non si trovano facilmente giochi della tombola in scatola con 240 cartelle già pronte. I premi in palio sono sempre i soliti cinque, uno per categoria.

Il grafico che illustra questi nuovi test è quello della seguente Figura 13.

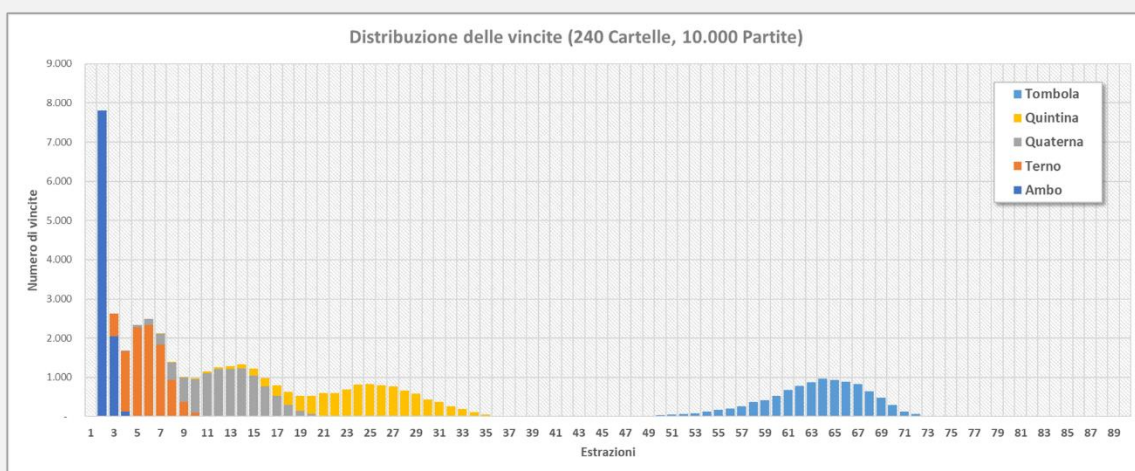


Figura 13 – Distribuzione delle vincite di 10.000 partite giocate con 240 cartelle e 5 premi disponibili per ogni partita (1 ambo, 1 terno, 1 quaterna, 1 quintina, 1 tombola).

Osserviamo anche questo grafico un po' più da vicino e, soprattutto, confrontiamolo con quello precedente:

- Anche in questo caso non ci sono vincite con un solo numero estratto. Niente umbi anche in questa volta, bene, il mondo ha una sua razionalità;
- Il numero di ambi vinti alla seconda estrazione è... enorme! Quasi l'80% dei 10.000 in palio! In fondo c'era da aspettarselo... più cartelle in gioco significano più combinazioni vincenti possibili... quindi più probabilità di fare ambo con un minor numero di numeri estratti (eh sì, il bisticcio di parole non si può evitare);
- Anche la macchia arancione dei terni si stiracchia verso l'alto e di conseguenza si ritira sull'asse orizzontale, di fatto esaurendosi già tra la nona e la decima estrazione (nella simulazione precedente, con 48 cartelle, si vincevano terni fino alla sedicesima estrazione);
- Analogo comportamento è quello di quaterne e quintine: alla ventesima estrazione tutte le quaterne sono praticamente vinte, alla trentacinquesima estrazione lo sono anche tutte le quintine.
- Ad allungarsi invece, un pochetto, è la valle degli sbadigli... prima che qualcuno, intorno alla cinquantesima estrazione, inizi a far tombola.
- La collinetta delle tombole inizia poco o niente prima rispetto alle giocate con 48 cartelle (quarantanovesima estrazione), ma finisce un po' di numeri prima, alla settantaduesima estrazione anziché alla settantanovesima del grafico precedente. Facile corollario: **in più si gioca, prima si fa tombola, prima finisce la partita** 😊

Ora... faccio fatica a credere che ci sia qualcuno qui che stia ancora leggendo... ma andiamo avanti. Questi due grafici in realtà sono solo "di riscaldamento", servono solo per introdurre e farci capire lo strumento di analisi. Il nostro obiettivo, se ricordiamo, non è studiare le vincite, ma i conflitti di vincite, ovvero le vincite concomitanti. Bene, possiamo farlo, producendo in modo analogo dei grafici simili a quelli appena visti, dove però non contiamo e sommiamo le vincite, bensì solo le vincite in cui ci sia più di una cartella a raggiungere il medesimo premio all'estrazione di uno stesso numero. In pratica contiamo i conflitti. **TGame** riesce a fare anche questo. Nuovo giro, nuovo grafico... quello in Figura 14. Diecimila nuove partite, con 48 cartelle in gioco ed i soliti 5 premi.

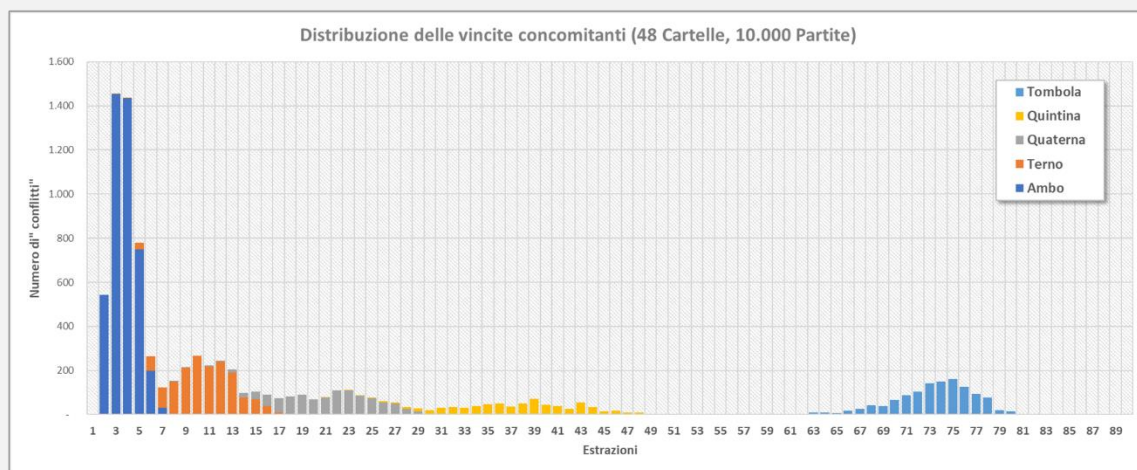


Figura 14 – Distribuzione dei conflitti di 10.000 partite giocate con 48 cartelle e 5 premi disponibili per ogni partita (1 ambo, 1 terno, 1 quaterna, 1 quintina, 1 tombola).

Riprendiamo la nostra lente di ingrandimento e procediamo nuovamente con l'analisi:

- Niente umbi anche in questo caso (e lo so, potrei non scriverlo più, ma oramai è diventata una

scaramanzia... chi scrive codice mi capirà);

- Al secondo numero estratto contiamo già poco meno di 600 conflitti per la vincita degli ambi. A voler essere precisi, è bene dirlo, questa volta sono conteggiate le cartelle che totalizzano un premio contemporaneamente, quindi se ci sono due cartelle che fanno ambo insieme sommiamo due al valore del grafico, se ce ne dovessero essere tre sommiamo tre e così via.
- Guardiamo tutte le barrette blu relative ai conflitti per gli ambi e proviamo a sommarle: poco meno di 600 al secondo numero estratto, più di 1.400 al terzo numero estratto, altri 1.400 dopo il quarto numero estratto, 700 e passa al quinto estratto... ci sono in tutto 4.410 cartelle che hanno fatto ambo insieme ad almeno un'altra cartella, su 10.000 ambi disponibili. Non è un numero piccolo... già con 48 cartelle in gioco.
- Le barrette arancione per i terni, grigie per le quaterne, gialle per le quintine ed azzurre per le tombole si leggono allo stesso modo. Le concomitanze diminuiscono statisticamente e si distribuiscono più a lungo sulle varie estrazioni, del resto ce lo si poteva aspettare.

E adesso? Oramai abbiamo capito il gioco. Adesso generiamo lo stesso grafico, 10.000 partite, 5 premi, ma facciamo scendere in campo più cartelle, le 240 già utilizzate in precedenza. Eccolo, nella seguente in Figura 15. Non farti ingannare dal fatto che le barrette sembrano quasi tutte molto più basse (e quindi indicherebbero meno vincite concomitanti). La scala di valori sull'asse delle Y a sinistra cresce e rappresenta un intervallo molto più ampio, per cui le concomitanze sono in realtà molto di più.

- Le cartelle che fanno ambo in concomitanza ad esempio, ora sono ben 16.713<sup>7</sup>. Quasi quattro volte quelli che si sono verificati quando in campo c'erano 48 cartelle.
- E lo stesso si può dire per il numero di cartelle che fanno terno insieme ad almeno un'altra cartella: quando in gioco c'erano 48 cartelle in tutto questo numero era 1.668, ora siamo a 3.453, praticamente più del doppio.

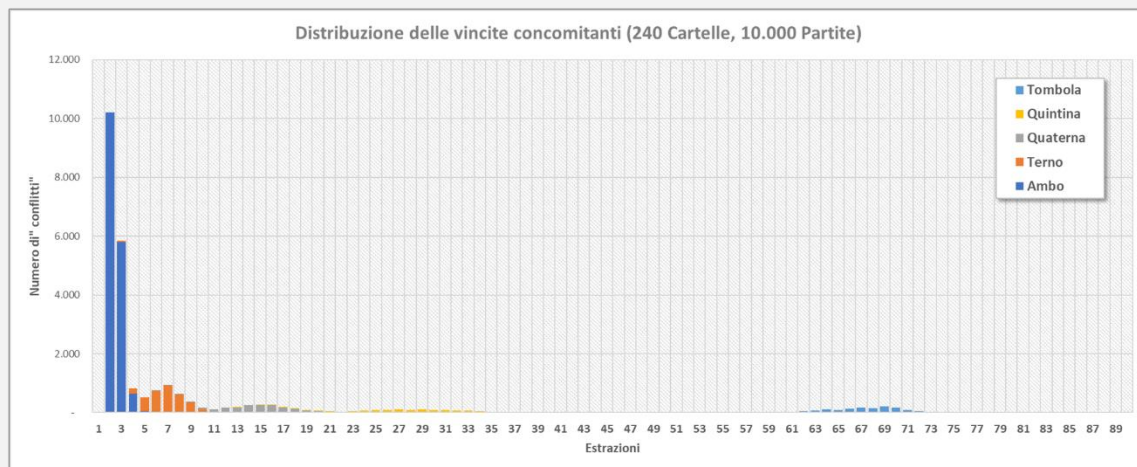


Figura 15 – Distribuzione dei conflitti di 10.000 partite giocate con 240 cartelle e 5 premi disponibili per ogni partita (1 ambo, 1 terno, 1 quaterna, 1 quintina, 1 tombola).

Volendo farla breve... come era del resto intuibile, **più cartelle abbiamo in gioco, più è alta la probabilità che vi siano vincite concomitanti**. La tabella seguente riporta le somme di vincite concomitanti per ogni categoria di premio, risultanti dalle 10.000 simulazioni con 48 cartelle in gioco e 10.000 simulazioni con 240 cartelle in gioco.

<sup>7</sup> Se ti stai chiedendo come faccio ad essere così preciso con questi numeri... no, non ho la super vista e non lo sto deducendo guardando le figure sul grafico. Semplicemente... li sto leggendo sui valori precisi che **TGame** ha fornito in output dalle varie simulazioni e che sono alla base dei grafici realizzati con Microsoft Excel.

	48 Cartelle in gioco	240 Cartelle in gioco	Differenze (%)
<i>Conflitti x Ambi</i>	<b>4.410</b>	<b>16.713</b>	+379%
<i>Conflitti x Terni</i>	<b>1.668</b>	<b>3.453</b>	+207%
<i>Conflitti x Quaterne</i>	<b>1.045</b>	<b>1.737</b>	+166%
<i>Conflitti x Quintine</i>	<b>710</b>	<b>1.166</b>	+164%
<i>Conflitti x Tombole</i>	<b>1.196</b>	<b>1.396</b>	+117%
<b>Totali</b>	<b>9.029</b>	<b>24.465</b>	<b>+271%</b>

*Tabella 6 – Confronto del numero di “conflitti” tra partite giocate con 48 cartelle e partite giocate con 240 cartelle.*

Magari in una prossima versione di questo manuale, se proprio non siete ancora contenti, aggiungo un altro po’ di dati, simulando, ventimila partite con cinquecento cartelle in gioco, o mille... l’andamento sarà più che confermato.

\*\*\*\*\*

Ci siamo quasi. Non abbiamo che un ultimo passo da fare. Abbiamo capito che il fenomeno delle vincite concomitanti è inevitabile e che più cartelle abbiamo in gioco e più si manifesta ed è fastidioso. E allora? Che armi abbiamo per provare a ridurlo? **Considerando che le possibili combinazioni di numeri valide per formare le cartelle sono ben più di 48... o 240... possiamo scegliere le cartelle che mettiamo in gioco tra tutte quelle possibili in modo da contenere questo increscioso fenomeno?**

## Il momento della verità

L’ultimo test che ci resta da fare è... verificare se cambia qualcosa utilizzando il metodo “Progressive” di TombolaCards per generare le cartelle. Eh, sì. Se hai letto il capitolo che racconta come usare il tool per creare le cartelle, saprai già che TombolaCards, nella sua versione corrente, consente di scegliere tra due algoritmi per la generazione degli insiemi di cartelle:

- Uno totalmente causale, denominato, per l’appunto, **Random**;
- L’altro, detto **Progressive**, che invece crea sempre serie di cartelle in modo casuale, ma le aggiunge all’insieme che si sta generando solo se non hanno troppi numeri in comune con le serie di cartelle che del medesimo insieme fanno già parte. L’aggiunta delle nuove serie è, per l’appunto, progressiva e consentita solo alle cartelle che non sono troppo simili a cartelle già precedentemente inserite.

In tutte le simulazioni fin qui fatte abbiamo utilizzato sempre cartelle, diciamo così, generate casualmente, ovvero non ottimizzate secondo i criteri del metodo *Progressive*. Nelle simulazioni con 48 cartelle, le cartelle sono state letteralmente copiate a mano da quelle del gioco che avevo a casa, e che è ritratto anche dalla foto di copertina. Una volta copiate, sempre utilizzando TombolaCards, è stato possibile valutarne i numeri uguali... ed ho potuto scoprire così che:

- Due cartelle hanno tra loro 8 numeri in comune;
- 8 cartelle hanno tra loro 7 numeri in comune;
- 16 cartelle hanno tra loro 6 numeri in comune
- 21 cartelle ne hanno 5 in comune

E così via. Ci sono poi, sempre tra queste 48 cartelle, anche due cartelle che hanno almeno una riga con 4 numeri dei 5 presenti uguali a 4 numeri su un’altra riga di un’altra cartella<sup>8</sup>. Se proprio dovessi esprimere un

<sup>8</sup> Su GitHub, nella cartella /samples del repository TombolaStudio, trovi il file TombolissimaClementoni.csv che riporta queste 48 cartelle e tutti gli altri file di cartelle utilizzati in queste sperimentazioni.



giudizio, direi che chi ha confezionato quel gioco tanti anni fa non ha fatto particolare attenzione alla distribuzione dei numeri ed alla diversità delle cartelle che metteva nella scatola.

Stesso ragionamento per l'insieme di 240 cartelle utilizzato per le simulazioni dei grafici precedenti. In questo caso non sono "copiate" da un gioco tradizionale, ma sono state generate con TombolaCards utilizzando l'algoritmo *Random*. Il risultato è un insieme in cui i numeri in comune tra le cartelle non sono pochi: 22 cartelle hanno 8 numeri in comune con altrettante cartelle, 119 cartelle hanno 7 numeri in comune, 95 ne hanno 6, etc. Anche in questo caso ci sono 14 cartelle che hanno righe con 4 numeri su cinque uguali ad un'altra riga su un'altra cartella dell'insieme.

In termini di riduzione dei numeri uguali tra cartelle, con il metodo progressive di TombolaCards ed un po' di pazienza è possibile ottenere di meglio. Giocando per qualche ora in qualche remoto weekend ho ottenuto due nuovi insiemi di cartelle, chiamiamoli "Selezionate", o "Ottimizzate", con le seguenti caratteristiche:

- 48 Cartelle: con al massimo 5 numeri uguali tra cartelle (e quindi non 8) e righe aventi al massimo 2 numeri uguali con un'altra riga (e non 4)
- 240 Cartelle: con al massimo 6 numeri uguali tra cartelle (e quindi non 8) e righe aventi al massimo 3 numeri uguali con un'altra riga (e non 4)

Ed eccoci... al verdetto finale. Non ci resta che simulare 10.000 partite anche con questi due nuovi insiemi, contarne le vincite concomitanti che si verificano e... confrontare questi risultati ottenuti con cartelle "ottimizzate" con quelli riportati poco più su ottenuti con cartelle non ottimizzate, "casuali".

Vi lascio guardare i grafici che seguono prima di aggiungere ulteriori considerazioni.

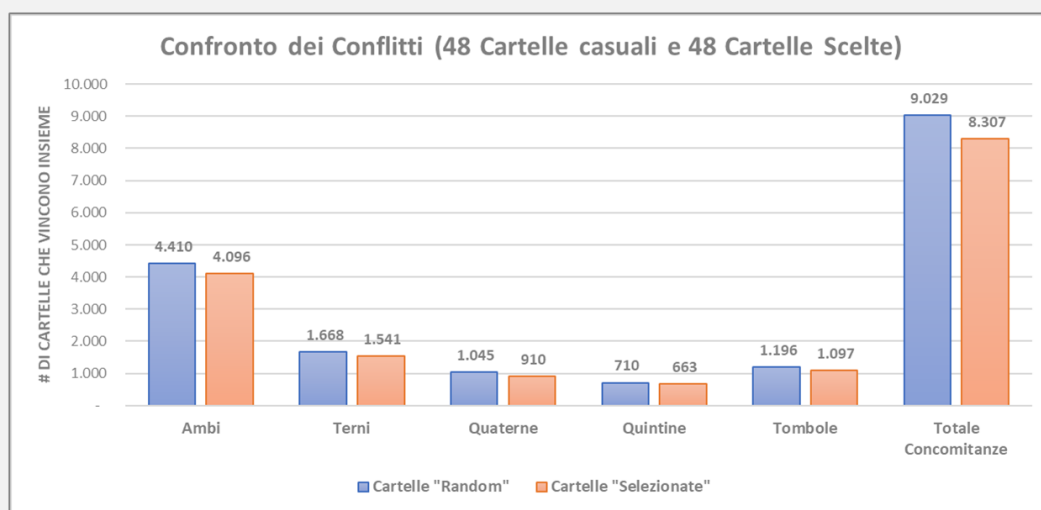
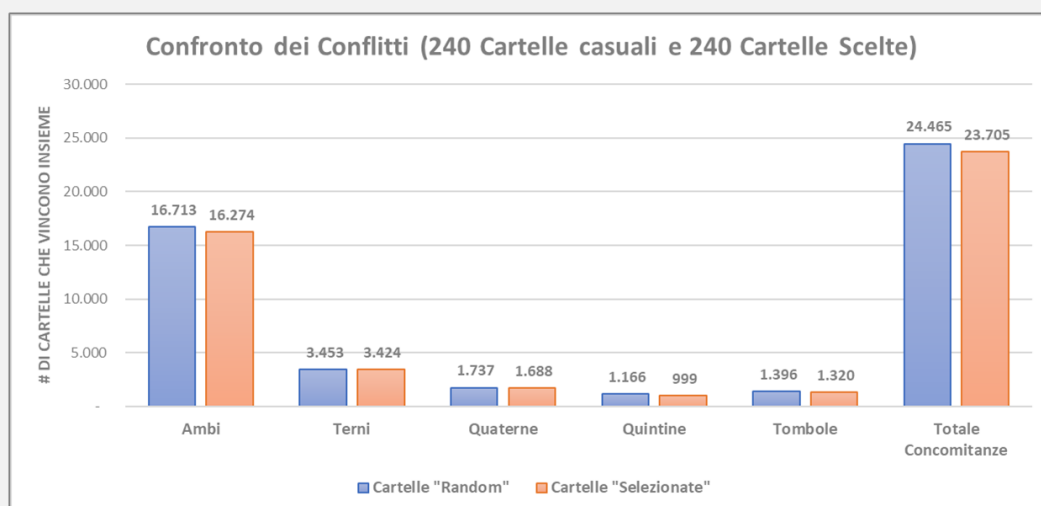


Figura 16 – Confronto del numero totale di conflitti che si verificano simulando 10.000 partite con 48 cartelle "random" e 48 "ottimizzate".



*Figura 17 – Confronto del numero totale di conflitti che si verificano simulando 10.000 partite con 240 cartelle "random" e 240 "ottimizzate".*

Visto? La (piccola) buona notizia è che in entrambi i casi i conflitti diminuiscono sempre quando si gioca con cartelle ottimizzate. Ambi, terni, quaterne, quintine, tombole... le barrette arancioni sono sempre più basse di quelle blu. In entrambi i grafici.

La brutta notizia è che... questa differenza è oggettivamente poca roba. Un 8% complessivo quando si gioca con 48 cartelle, poco più del 3% complessivo quando le cartelle in gioco son 240.

Certo, **giocare con cartelle ottimizzate è sempre meglio che giocare con cartelle che non lo sono, ma mi sarei aspettato una differenza più marcata.** Oggettivamente il numero di vincite concomitanti, il numero di conflitti, resta, a malapena scalfito. Magari andrebbero fatti più test, effettuate ulteriori verifiche con altri insiemi di cartelle... ma il tempo speso su questo tema è già stato per me... *tanto*.

Lascio l'ardua fatica di proseguire la ricerca ad altri... e qui metto il punto a questa storia.

### 3. TombolaPrint: stampare cartelle

#### Principi di funzionamento

Prima di lanciare TombolaPrint per ottenere un file di cartelle pronte per essere stampate, spendiamo qualche parola per illustrare il principio generale alla base del funzionamento di questo tool. In buona sostanza, TombolaPrint ha bisogno di due oggetti in input:

- **Il file contenente le cartelle da stampare.** È un file, tipicamente con estensione csv, prodotto da TombolaCards. Sono accettati i formati CSV, CSV\_PLUS, PACKED, PACKED\_PLUS
- **Un file “modello” che contiene,** in formato HTML + CSS<sup>9</sup>, le informazioni necessarie per comporre le cartelle della tombola sulle pagine da stampare: il numero di cartelle che vanno stampate su ogni foglio, i caratteri (font) da utilizzare e la loro dimensione, l’eventuale presenza di bordi, griglie, immagini o tonalità di sfondo, l’orientamento stesso del foglio, etc.

TombolaPrint lavora in questo semplice modo: per ogni cartella che legge dal file in input, “applica” le regole di formattazione che trova nel file “modello” e produce un file di output, anch’esso in formato HTML+CSS. Tale lavoro è reso semplice dal modo stesso in cui sono realizzati i file modello: in essi infatti sono presenti *blocchi* pre-formatati all’interno dei quali vi sono *segnaposto (tag)* che vengono sostituiti con i numeri delle cartelle. L’immagine che segue illustra meglio di tante parole il procedimento.

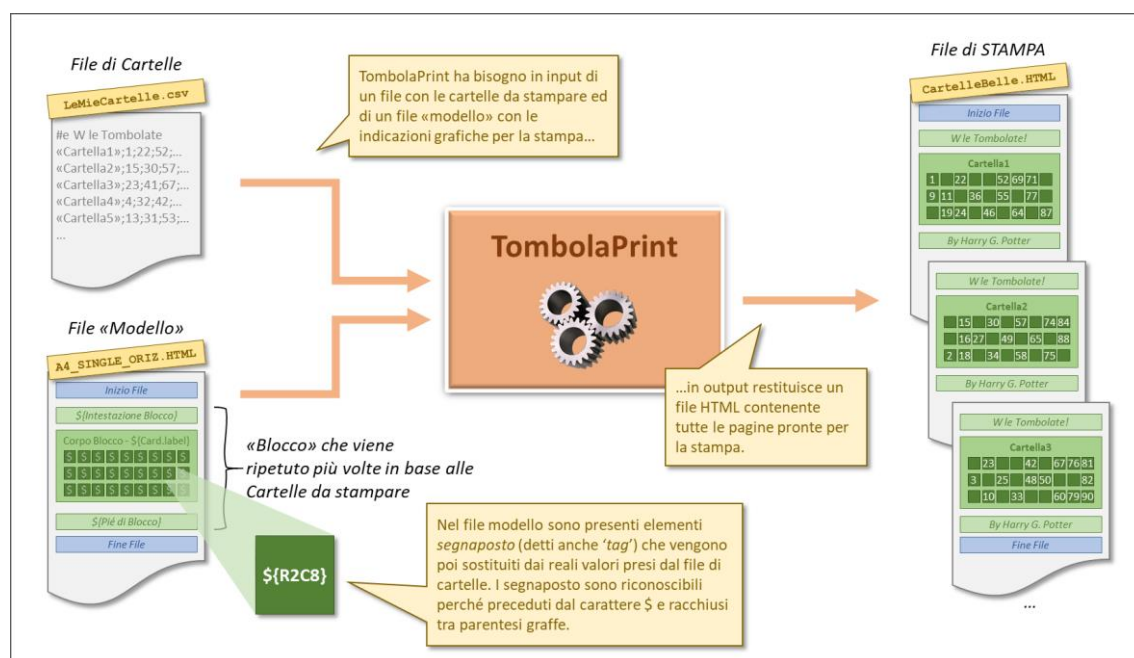


Figura 18 – Schematizzazione del principio di funzionamento di TombolaPrint

<sup>9</sup> HTML è l’abbreviazione di Hyper Text Markup Language, il linguaggio universalmente utilizzato per le pagine dei siti web. CSS è l’abbreviazione di Cascaded Style Sheets, un ulteriore linguaggio utilizzato per dare stile e forma al contenuto delle pagine HTML. I file che adottano questi linguaggi sono fruibili utilizzando un qualsiasi browser web, come Microsoft Edge o Internet Explorer, Firefox, Google Chrome, Safari. Anche su computer e sistemi operativi differenti: Microsoft Windows, Apple Mac OS, Linux.

Al momento sono stati predisposti 2 modelli base, uno per stampare una cartella per ogni foglio di dimensioni A4 (orientamento “orizzontale”), ed uno per stampare due cartelle per ogni foglio A4, con orientamento in verticale. I file modello sono nella directory `/print_templates` e per praticità sono riassunti nella seguente tabella.

Nome del file HTML	Descrizione
SIMPLE_1Card_x_Page_A4L.html	Una cartella per pagina, su foglio A4 orientato orizzontalmente. Su ogni pagina vi è una intestazione ed un piè di pagina che possono essere personalizzati. I numeri delle cartelle sono in una griglia con bordo.
SIMPLE_2Card_x_Page_A4P.html	Due cartelle per pagina, su foglio A4 orientato verticalmente. Su ogni pagina vi è una intestazione ed un piè di pagina che possono essere personalizzati. I numeri delle cartelle sono in una griglia con bordo.

*Tabella 7 – File “modello” disponibili per stampare con TombolaPrint.*

Tali modelli possano essere utilizzati anche come punto di partenza per ulteriori personalizzazioni. Puoi ad esempio inserirvi il nome del gruppo di amici con cui giochi a Tombola, oppure cambiare il font di caratteri utilizzato, oppure dare un tocco di colore alle intestazioni, ai bordi, ai numeri... oppure tutte queste cose insieme! Puoi anche inserire immagini di sfondo, fotografie, loghi... quello che occorre è un po’ di conoscenza dei linguaggi HTML e CSS.

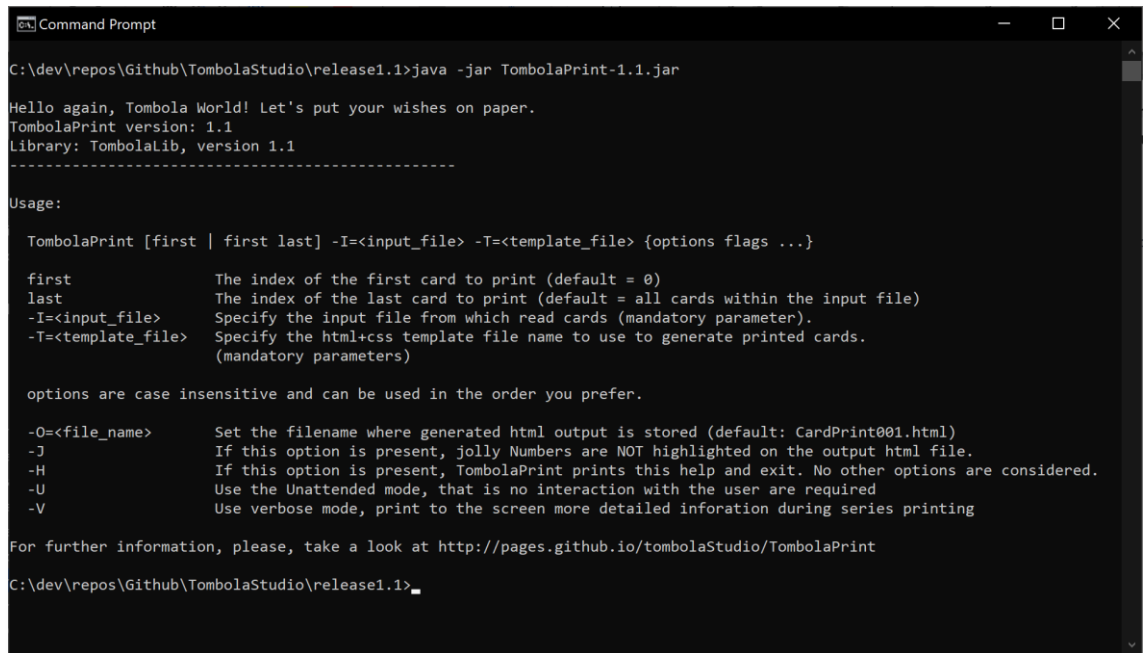
Ma andiamo per gradi. Proviamo dapprima a stampare le cartelle utilizzando i modelli che abbiamo, poi dedicheremo un po’ di tempo a studiare l’anatomia di questi modelli ed a capire quindi come modificarli rapidamente.

### Finalmente si stampa

Corriamo a stampare le cartelle! TombolaPrint si utilizza in modo del tutto simile a TombolaCards: se lo si lancia senza alcun parametro aggiuntivo, si ottiene la stampa a video di un sintetico help, come riportato in figura.

```
> cd c:\progetti\TombolaStudio\release1.1
> java -jar TombolaPrint-1.1.jar
```





```
C:\dev\repos\Github\TombolaStudio\release1.1>java -jar TombolaPrint-1.1.jar

Hello again, Tombola World! Let's put your wishes on paper.
TombolaPrint version: 1.1
Library: TombolaLib, version 1.1
-----
Usage:

  TombolaPrint [first | first last] -I=<input_file> -T=<template_file> {options flags ...}

first          The index of the first card to print (default = 0)
last           The index of the last card to print (default = all cards within the input file)
-I=<input_file> Specify the input file from which read cards (mandatory parameter).
-T=<template_file> Specify the html+css template file name to use to generate printed cards.
                  (mandatory parameters)

options are case insensitive and can be used in the order you prefer.

-O=<file_name>  Set the filename where generated html output is stored (default: CardPrint001.html)
-J             If this option is present, jolly Numbers are NOT highlighted on the output html file.
-H             If this option is present, TombolaPrint prints this help and exit. No other options are considered.
-U            Use the Unattended mode, that is no interaction with the user are required
-V            Use verbose mode, print to the screen more detailed information during series printing

For further information, please, take a look at http://pages.github.io/tombolaStudio/TombolaPrint
C:\dev\repos\Github\TombolaStudio\release1.1>
```

Figura 19 – Il messaggio di help stampato a video da TombolaPrint quando lo si lancia senza parametri.

I primi due parametri, *first* e *last*, riportati tra parentesi quadre, possono essere omessi. Servono per indicare la prima ed ultima cartella da stampare tra quelle presenti nel file, qualora non si volesse stampare l'intero file. Gli altri due parametri invece, -I e -T, sono strettamente necessari:

- -I consente di specificare il nome del file di cartelle in input, prodotto con TombolaCards, che si vuole stampare;
- -T consente di specificare il nome del file "modello" (o, anche, file "template") da utilizzare per produrre la stampa. Maggiori informazioni su cosa sono i modelli e come si usano a brevissimo.

Supponiamo dunque di avere pronto un file di cartelle realizzato con TombolaCards, e questo file si chiama **LeMieCartelle.csv**. Vogliamo stampare una cartella per foglio, bella grande, che riempia la maggior parte dello spazio A4 a disposizione. Il file modello da usare è dunque **SIMPLE\_1Card\_x\_Page\_A4L.html**, presente nella cartella **/print\_templates**. Infine, vogliamo chiamare il nostro file di "cartelle stampate" **CartelleDaStampare.HTML**.

Siamo pronti per dare il nostro comando TombolaPrint completo:

```
> java -jar TombolaPrint-1.1.jar -I=LeMieCartelle.csv
-T=print_templates/SIMPLE_1Card_x_Page_A4L.html
-O=CartelleDaStampare.html
```

Se tutto va per il verso giusto... a video leggerai quanto segue:

```
Reading card series file: LeMieCartelle.csv... [OK!]
30 cards just loaded.
Start printing from card 0 to card 29.
Using template file: print_templates\SIMPLE_1Card_x_Page_A4L.html
Jolly numbers will be NOT highlighted.
```

Hic sunt leones! Just printed 30 cards on file CartelleDaStampare.html.  
Best regards!

Molto bene. Se ora lanci il comando dir in questa cartella... dovresti vedere, tra gli altri, il file di cartelle in formato html appena creato da TombolaPrint.

```
> dir

Volume in drive C has no label.
Volume Serial Number is 7A54-A664

Directory of c:\progetti\TombolaStudio\release1.1

05/06/2020  12:30    <DIR>          .
05/06/2020  12:30    <DIR>          ..
05/06/2020  12:30             76.540 CartelleDaStampare.html
05/06/2020  12:28             4.358 LeMieCartelle.csv
25/03/2020  16:13    <DIR>          print_templates
14/04/2020  19:33       126.395 TombolaCards-1.1.jar
05/06/2020  12:28           366 TombolaCards.log
27/03/2020  11:58       76.701 TombolaLib-1.1.jar
03/06/2020  14:28       86.078 TombolaPrint-1.1.jar
               6 File(s)          370.438 bytes
               3 Dir(s)  316.598.022.144 bytes free
```

Ci siamo. Ora non resta che aprire il file appena generato con il tuo browser preferito. Di solito, su un pc Windows, basta digitarne il nome sulla riga di comando e dare INVIO. L'altra possibilità è aprire una finestra del browser e copiare, nella barra dell'indirizzo in alto, il percorso completo del file. Nel mio caso, ad esempio, il percorso completo è

```
c:\progetti\TombolaStudio\release1.1\CartelleDaStampare.html
```

```
> CartelleDaStampare.html + <INVIO>
```

Il tuo browser predefinito viene aperto ed il file HTML visualizzato in tutto il suo splendore! Ispezionalo tutto scorrendo verso il basso: troverai tutte le cartelle del file originario. Con i numeri disposti correttamente nelle griglie di ogni cartella.

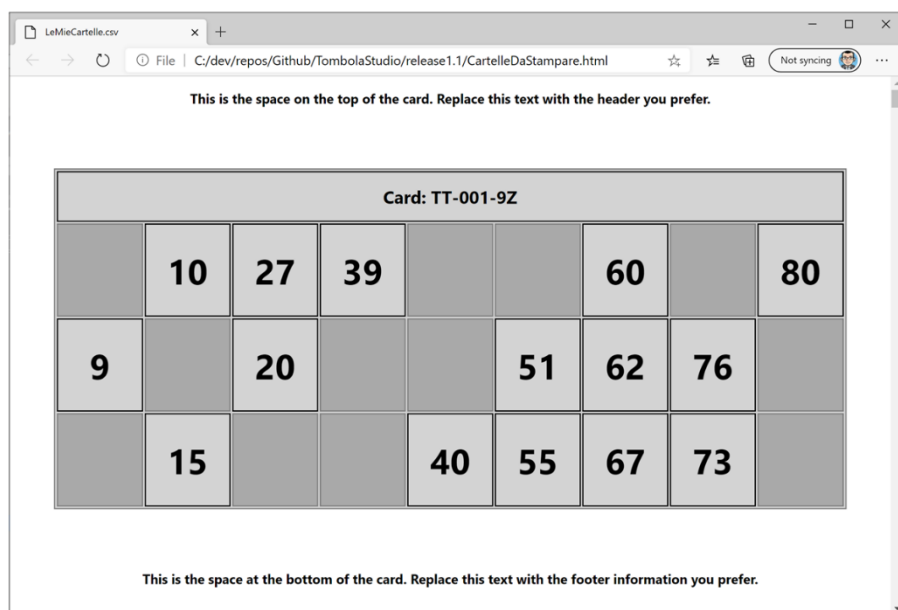


Figura 20 – Finalmente il file di cartelle in formato HTML pronto per essere stampato.

So bene a cosa stai pensando: il look di queste tabelle non è proprio il massimo... cartelle grigine, squadrate... con dello strano testo in inglese che precede e segue ogni cartella... È vero. I template di adesso sono un po' rudimentali ma la notizia buona è che possono essere personalizzati: colori, fonts, bordi, sfondi... il paragrafo *"I modelli di stampa visti da vicino"* a pag.53 ti dirà come si fa.

Intanto, per stampare, usa le funzionalità proprie del tuo del browser. Nella figura seguente è riportata la finestra di "anteprima di stampa" del mio browser. Come stampante ho scelto "stampa in PDF", così verrà prodotta non una stampa su carta, ma un file, appunto, in formato PDF che potrò stampare quando vorrò.

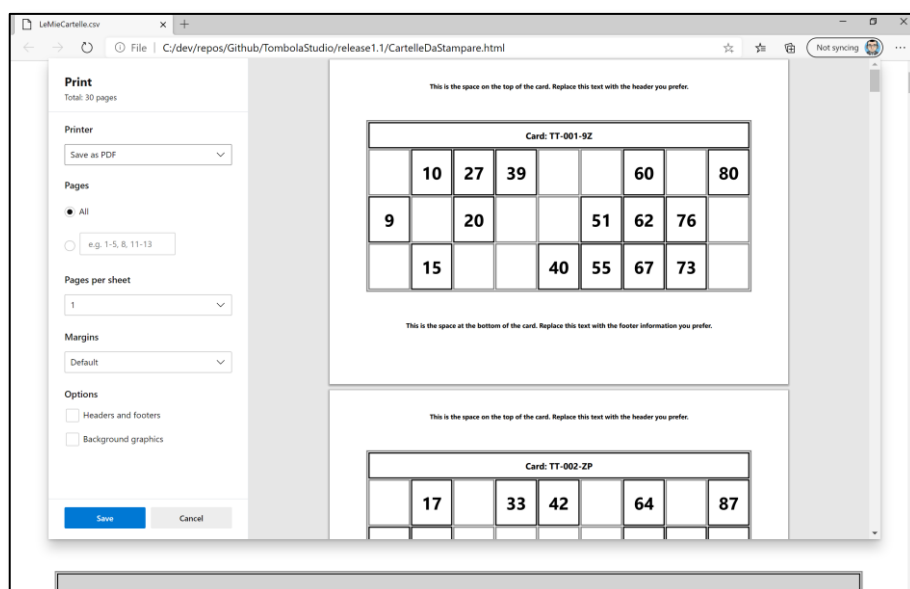


Figura 21 – La visualizzazione della "anteprima di stampa" del file di cartelle.

Facile, non è vero? Ora puoi stampare le tue cartelle su fogli A4, cartoncini colorati o anche fogli su cui hai precedentemente stampato o fatto stampare motivi comuni, immagini... puoi dare libero sfogo alla tua fantasia.

### Riepilogo dei parametri a “riga di comando”

Come al solito, in questo paragrafo, anche per TombolaPrint riportiamo in un'unica tabella la sintesi di tutti i parametri che sono possibili a riga di comando.

Opzione	Valori ammessi	Obbligatorietà e valori predefiniti	Descrizione
First	Da 0 al numero di cartelle presenti nel file indicato con l'opzione -I	<b>Parametro Facoltativo</b> (valore predefinito: 1)	Questo parametro opzionale consente di iniziare a stampare le cartelle a partire da una cartella diversa dalla prima. Occorre immaginare le cartelle come se fossero numerate in ordine crescente a partire da zero. Per stampare dalla seconda cartella in poi, quindi, occorre indicare “1” come prima cartella da considerare.
Last	Da 0 al numero di cartelle presenti nel file indicato con l'opzione -I	<b>Parametro Facoltativo</b> (valore predefinito: l'ultima cartella nel file)	Specificando un secondo valore numerico sulla riga di comando è possibile ridurre l'intervallo delle cartelle da inserire nel file di stampa a partire dalla coda del file. Sempre immaginando le cartelle come numerate progressivamente a partire da 0, questo parametro specifica l'indice di cartella a cui fermarsi, per cui anche se nel file di input sono presenti altre cartelle verranno ignorate.
-I	<nome file in input>	<b>Parametro obbligatorio</b>	Questo parametro consente di specificare il file da cui leggere le serie di cartelle che si vogliono stampare. Il file è un file prodotto da TombolaCards, in uno qualsiasi dei formati CSV supportati.
-T	<nome file modello>	<b>Parametro obbligatorio</b>	Questo parametro consente di specificare il file modello (o, anche, il file “template” da utilizzare per applicare il giusto formato (font, colori, impaginazione, bordi, immagini, etc.) alle cartelle che si stampano.
-O	<nome file di output>	<b>Parametro obbligatorio</b>	Specifica il nome del file nel quale vengono salvate le cartelle pronte per la stampa. È un file in formato HTML + CSS.
-J	-	<b>Parametro facoltativo</b>	Se presente, questa opzione DISABILITA l'utilizzo di un formato particolare per evidenziare i numeri jolly sulle cartelle quando viene prodotto il file di stampa.
-H	-	<b>Parametro facoltativo</b>	Se questo parametro è presente sulla riga di comando, TombolaPrint ignora tutti gli altri parametri specificati e stampa a video una sintetica guida all'utilizzo (help), poi il programma termina. Lo stesso risultato lo si ottiene lanciando TombolaPrint senza specificare nessuna opzione aggiuntiva sulla riga di comando.
-U	nessuno	<b>Parametro facoltativo</b> (valore predefinito: OFF)	Se questa opzione è presente sulla riga di comando, TombolaPrint: <ul style="list-style-type: none"> <li>• Evita di ‘fare domande’ all'utente finale per preparare i file di stampa delle cartelle;</li> <li>• Riduce il numero di messaggi ed informazioni stampati a video.</li> </ul> È pensata per facilitare l'utilizzo di TombolaPrint in modalità <i>unattended</i> , ovvero senza la supervisione ed il controllo di un utente umano. Si pensi ad esempio ai casi in cui si vuole richiamare

Opzione	Valori ammessi	Obbligatorietà e valori predefiniti	Descrizione
			TombolaPrint all'interno di altri scripts facilitando così l'automazione di processi ripetitivi.
-V	-	<b>Parametro facoltativo</b> (valore predefinito: OFF)	Se presente questa opzione, vengono scritti a video molti più messaggi e fornite molti più dettagli durante la predisposizione dei file di stampa (V sta per Verbose).

Tabella 8 – Riepilogo delle opzioni a riga di comando di TombolaPrint.

## I modelli di stampa visti da vicino

In questo paragrafo trovi indicazioni su come sono fatti i file modello così da poter capire dove mettere le mani nel caso tu voglia modificarli per ottenere dei file di stampa dal layout più accattivante o comunque più rispondenti alle tue esigenze. Ribadiamo un po' di concetti che sono fondamentali per capire i principi di funzionamento di TombolaPrint e quindi come ottenerne il meglio.

- 1) I file modello altro non sono che dei file in formato HTML + CSS. I vari elementi che li compongono (intestazioni di pagina, cartelle, righe nelle cartelle, numeri nelle righe, etc.) altro non sono elementi HTML a cui, mediante istruzioni CSS, sono applicati appositi stili. **Se modifichi le istruzioni CSS che sono presenti all'inizio dei file, puoi personalizzare il layout delle tue cartelle.**
- 2) I file modello includono una riga di inizio blocco ed una riga di fine blocco per identificare la parte di modello che deve essere ripetuta più volte affinché il file di stampa contenga tutte le cartelle che si è scelto di stampare. Tutto quanto precede la riga di inizio blocco è riportato, nel file di output, una sola volta, così come tutto quanto segue la riga di fine blocco. Quanto è riportato all'interno del blocco, invece, è ripetuto N volte, con N sufficiente a riprodurre tutte le cartelle previste per la stampa. N dipende dunque sia da quante cartelle ci son da stampare, sia da quante cartelle sono previste all'interno di ciascun blocco. I blocchi infatti possono contenere 1, 2, 3 o anche 6 cartelle. **Pensa al blocco come al corpo di una pagina stampata.** La possibilità di prevedere una o più cartella all'interno di un blocco ci consente di predisporre file modello che stampano 1 o più cartelle su ciascun foglio. Il numero di cartelle che è previsto per ciascun blocco è indicato nella riga di definizione del template.
- 3) **All'interno dei file modello sono previsti elementi detti segnaposto o tag.** Tali elementi sono sostituiti da valori presi dai file di cartelle al momento di generare i file di stampa. I tag sono riconoscibili perché preceduti dal carattere dollaro "\$" e perché racchiudono ciascuno il proprio nome in una copia di parentesi graffe, "{" e "}". Ad esempio, il tag "\${CurrentDate}", in fase di stampa verrà sostituito con la data odierna, per cui al suo posto nel file di stampa in output ci troverai qualcosa come "21 giugno 2020". Vi sono segnaposti per le etichette di cartella e per ciascun loro numero: posizionando opportunamente i segnaposti nei file modello è possibile realizzare le stampe in modo flessibile e personalizzato. **Trovi l'elenco completo di tutti i tag supportati nel paragrafo successivo.**

**!!! WORK STILL IN PROGRESS...**

Pasticciamo un po'

**!!! WORK STILL IN PROGRESS...**

Tipi di template disponibili

**!!! WORK STILL IN PROGRESS...**

Identificativo	Numero di cartelle per blocco	Orientamento del foglio
A4_ONECARD_L	1	Orizzontale
A4_TWOCARDS_P	2	Verticale
A4_THREECARDS_P	3	Verticale
A4_FOURCARDS_L	4	Orizzontale
A4_SIXCARDS_P	6	Verticale

*Tabella 9 – Elenco dei tag, ad oggi supportati, che identificano un “tipo di template” di TombolaPrint.*

Elenco dei tag supportati

**!!! WORK STILL IN PROGRESS...**

Il tag	viene sostituito, in fase di stampa, da...
<code>\${Author}</code>	
<code>\${SeriesListName}</code>	
<code>\${SeriesListTitle}</code>	
<code>\${CurrentDate}</code>	
<code>\${CurrentTimeStamp}</code>	
<code>\${CreationTimeStamp}</code>	
<code>\${GenerationTotalCounter}</code>	
<code>\${SeriesGenerationMethod}</code>	
<code>\${GenerationSeed}</code>	
<code>\${GenerationMethod}</code>	
<code>\${GenerationDuration}</code>	
<code>#{C[n].L}</code>	L’etichetta della cartella n-esima all’interno del blocco.
<code>#{C[n].N[riga, colonna]}</code>	Il numero in colonna col, riga row, della ennesima cartella del blocco.
<code>#{C[n].Style[riga, colonna]}</code>	number-normal, number-empty, number-marked, jolly-cell, jolly-marked

#### 4. TombolaWeb: L'ombelico di TombolaStudio (Coming soon)

TombolaWeb è davvero divertente... ed è anche in uno stato di sviluppo *quasi accettabile*... penso davvero di pubblicarlo a breve.

Torna qui ogni tanto... prima di Natale troverai di sicuro una piacevole sorpresa.



## 5. TombolaLib: per programmatori o aspiranti tali

Anche per un capitolo introduttivo su TombolaLib occorre aspettare un pochetto. Intanto però puoi consultare la documentazione dell'API prodotta con l'utilità standard Javadoc. La trovi qui:

- <https://github.com/HarryGPotter/TombolaStudio/TombolaLib/apidoc>

