



Esercitazione 7 (11-04)

7.1 Moon Patrol, test

- Creare degli *unit test* per il gioco Moon Patrol
 - Sono consigliati *test parametrizzati*
- Cercare di testare ciascun personaggio separatamente
 - Chiamare direttamente `move`, `collide` ecc., in varie situazioni
- Es. testare il rover simulando varie sequenze di comandi
 - Percorre la traiettoria prevista? (Non può saltare se non è a terra...)
 - Sceglie sempre il simbolo corretto?
 - Reagisce correttamente alle diverse collisioni?



7.2 Moon Patrol, estensioni

- Estensioni *opzionali* del gioco
 - Secondo rover, per giocare in coppia, con l'uso di tasti diversi
 - Cannoni ed altri avversari sul percorso
 - Punteggio, livelli, bonus ed estensioni *a fantasia*



7.3 Moon Patrol, gioco e GUI

- Classe `MoonPatrolGame`, per gestire il gioco *Moon Patrol*
 - Inizializzare i personaggi, prendendo i dati da un file di configurazione
 - Incapsulare personaggi ed arena
 - Controllare la conclusione della partita, positiva o negativa
 - Guardare l'esempio nei commenti in `bounce`
- Classe `MoonPatrolGui` per la rappresentazione del gioco
 - Disegno immagini e funzionalità legate a `g2d`
 - Metodi `tick` etc.
 - Prendere spunto da `BoardGameGui`

7.4 Hitori, righe e colonne

- Nell'inizializzazione, leggere la matrice di numeri da un file (*parametro*)
- Nel metodo **finished**, controllare ogni riga e colonna
 - Nessun valore ripetuto, tra quelli non anneriti
 - Celle annerite non adiacenti
- Trascurare per ora il controllo di contiguità tra tutte le celle bianche

	8		6	3	2		7
3	6	7	2	1		5	4
	3	4		2	8	6	1
4	1		5	7		3	
7		3		8	5	1	2
	5	6	7		1	8	
6		2	3	5	4	7	8
8	7	1	4		3		6



7.5 Hitori, test

- Creare degli *unit test* per il gioco Hitori
- Verificare il gioco dopo varie chiamate a `play_at` e `flag_at`
 - Usare `value_at` per osservare lo stato
- Verificare il funzionamento di `finished` rispetto alle seguenti due regole
 - Nessun valore ripetuto, tra quelli non anneriti
 - Celle annerite non adiacenti
- Trascurare per ora il controllo di contiguità tra tutte le celle bianche

7.6 Hitori, GUI

- Utilizzare il gioco tramite la GUI fornita con gli esempi

	8		6	3	2		7
3	6	7	2	1		5	4
	3	4		2	8	6	1
4	1		5	7		3	
7		3		8	5	1	2
	5	6	7		1	8	
6		2	3	5	4	7	8
8	7	1	4		3		6

7.7 Progetti in bella copia

- **Codice leggibile**

- *Costanti*, anzichè numeri “*magici*” sparsi nel codice
- *Nomi esplicativi* e semplici
- *Regole di stile*: `variable_name`, `function_name`, `ClassName`, `CONSTANT_NAME`
- *Commenti*, quando utili: *function annotation*, *docstring*



- **Codice ben organizzato**

- *No copia&incolla* del codice: funzioni parametrizzate e/o cicli
- *Programmazione strutturata*: preferibilmente `return` a fine funzione, cicli senza `break`
- *OOP*: *incapsulamento*, *ereditarietà*, *polimorfismo*