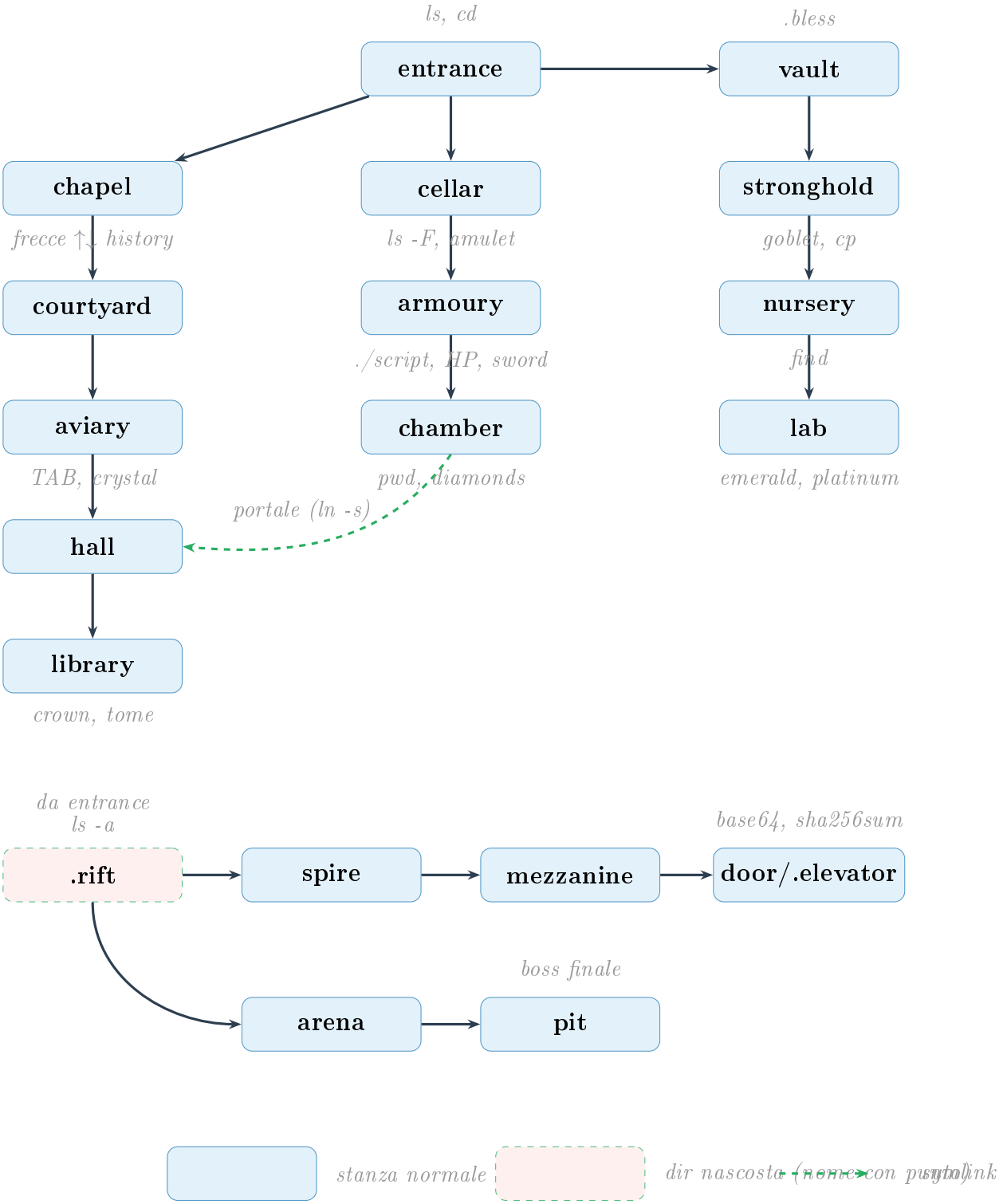
BASHCRAWL

Guida Completa per il Docente

Basata sull'esplorazione reale del dungeon — A.S. 2025/2026

1. Mappa del dungeon



2. Walkthrough stanza per stanza

entrance — Ingresso

Come arrivarci: punto di partenza. `cd ~/Downloads/bashcrawl-master/entrance`

Contenuto: `scroll`, `scrap/`, `cellar/`, `.chapel/`†, `.vault/`†, `.rift/` †=nascoste all'inizio

```
cat scroll # leggi le istruzioni iniziali
ls # lista file e directory visibili
cd cellar # entra nella prima stanza
```

Comandi: `ls`, `cd <dir>`, `cat <file>` **Tesori:** nessuno

cellar — Cantina

Come arrivarci: `cd cellar` **Contenuto:** `scroll`, `treasure*`, `armoury/`

```
cat scroll # impara ls -F e alias
ls -F # mostra / per dir, * per eseguibili
alias ls='ls -F' # d'ora in poi ls usa ls -F in automatico
./treasure # raccoglie l'amuleto (sblocca anche chapel e vault)
export I=amulet,$I # aggiungi amulet all'inventario
echo $I # verifica inventario
```

Gate: `./treasure` rinomina `.chapel` in `chapel` e `.vault` in `vault`.

Comandi: `ls -F`, `alias`, `export I=item,$I`, `echo $I` **Tesori:** `amulet`

armoury — Armeria

Come arrivarci: `cd armoury` da cellar **Contenuto:** `scroll`, `potion*`, `treasure*`, `chamber/`

```
cat scroll # impara ./script e cd ..
./potion # scelta: y -> HP=15 (bevi); n -> HP=10 (rifiuti)
export HP=15 # (o HP=10 -- il gioco dice quale digitare)
echo $HP # controlla i punti vita
./treasure # raccoglie la spada (serve amulet in $I)
export I=sword,$I
```

Nota: anche `./potion` sblocca chapel/vault se `./treasure` in cellar non era stato eseguito.

Comandi: `./nomescript`, `export HP`, `echo $HP`, `cd ..` **Tesori:** `sword`

chamber — Sala Segreta

Come arrivarci: `cd chamber` da armoury **Contenuto:** `scroll`, `spell*`, `pieces*`, `treasure*`

```
cat scroll # impara pwd e cd ..
pwd # mostra il percorso assoluto corrente
./spell # crea symlink portale verso hall
              # comando eseguito: ln -fs ../../../chapel/courtyard/aviary/hall portal
cd portal # entra in hall tramite il portale
./pieces # combattimento con la statua (richiede sword in $I)
              # ogni colpo ricevuto: let "HP=HP-5"
./treasure # raccoglie i diamanti
export I=diamonds,$I
```

Comandi: `pwd`, `cd ..`, `ln -s`, `let "HP=HP-5"` **Tesori:** `diamonds`, crea `portal`

chapel — Cappella

Come arrivarci: `cd ../../chapel` da armoury; oppure `cd ../chapel` da cellar
Contenuto: `scroll`, `.curse/`, `altar/`, `courtyard/`

```
cat scroll # impara le frecce su/giu per la history
              # freccia UP / DOWN per scorrere i comandi precedenti
cd courtyard
```

Comandi: frecce ↑↓ per la history **Tesori:** nessuno

courtyard — Cortile

Come arrivarci: `cd courtyard` da chapel **Contenuto:** `scroll`, `fountain/`, `rags/`, `aviary/`

```
cat scroll # narrativa (fontana, piante morte)
cd aviary
```

Nota: `rags/` contiene `rags` e `fish`; `fountain/` penalizza HP se personaggio ha `.curse`.

aviary — Voliera

Come arrivarci: `cd aviary` da courtyard **Contenuto:** `scroll`, `crystal*`, `penguin/`, `hall/`

```
cat scroll # impara il TAB autocomplete
cd h<TAB> # premi TAB: autocompleta in "hall"
./crystal # ottieni il seed crystal
export I=crystal,$I
```

Comandi: tasto **TAB** per autocompletare nomi file/directory **Tesori:** `crystal`

hall — Sala

Come arrivarci: `cd hall` da aviary; oppure `cd portal` da chamber (symlink)
Contenuto: `carcass/`, `treasure/`, `library/`

```
cd treasure # nella dir treasure c'e' il combat e il premio
./treasure # ottieni la corona (richiede sword in $I)
export I=crown,$I
cd ../library
```

Tesori: `crown`

library — Biblioteca

Come arrivarci: `cd library` da hall **Contenuto:** `scroll`, `tome*`

```
cat scroll
./tome # 6 incantesimi avanzati (scelti a caso ad ogni esecuzione)
```

Il tome insegna 6 comandi avanzati (random tra questi 6):

`alias ls='ls -F -color=auto'`	output colorato per tipo di file
`pushd <dir>` / `popd` / `dirs`	stack LIFO di directory
`tree -F`	struttura ad albero dell'intero dungeon
`set -o emacs` + Ctrl-a/e, Meta-f/b, Ctrl-u/k/w/y	navigazione avanzata sul prompt
`tmux`	terminal multiplexer, sessioni persistenti
`for i in {00..99}; do gzip ${i}; done`	brace expansion e for loop

vault — Caveau

Come arrivarci: `cd vault` da entrance (dopo sblocco) **Contenuto:** `scroll`, `glass*`, `.bless/`, `stronghold/`

```
cat scroll
./glass # deposita il crystal -> crea .bless (stato persistente)
cd stronghold
```

Meccanica .bless: file nascosto che modifica gli script di combattimento. Avere `.bless` attivo dà bonus nei fight successivi (in particolare col boss Nyarlathotep nel pit).

stronghold — Fortezza

Come arrivarci: `cd stronghold` da vault **Contenuto:** `scroll`, `goblet*`, `orb1/`, `orb2/`†, `orb3/`†

† orb2 e orb3 non esistono inizialmente: vanno creati con cp

```
cat scroll # puzzle: servono orb1+orb2+orb3 per liberare il goblet
ls -F # vedrai solo orb1/ e goblet*
cp orb1 orb2 # copia orb1 e chiamalo orb2
cp orb1 orb3 # copia orb1 e chiamalo orb3
./goblet # ora funziona -> ottieni il goblet
export I=goblet,$I
cd nursery
```

Comandi: `cp <origine> <destinazione>` **Tesori:** `goblet`

nursery — Vivaio

Come arrivarci: `cd nursery` da stronghold **Contenuto:** `scroll`, `potion`, `spell*`, `lab/`

```
cat scroll
./spell # prima volta: impara il comando find
find . -name "potion" # cerca potion a partire da qui
find ../../../.. -name "potion" # cerca da entrance (percorso da capire con pwd)
cp <percorso_trovato>/potion . # copia la potion in questa dir
./spell # seconda volta: bevi la pozione (+5 HP)
cd lab
```

Comandi: `find . -name "pattern"` **Bonus:** +5 HP

lab — Laboratorio

Come arrivarci: `cd lab` da nursery **Contenuto:** `scroll`, `ghost*`, `corpse/`, `platinum/`, `treasure/`

```
cat scroll
./ghost # combattimento ghost (richiede sword in $I)
cd treasure
./treasure # ottieni l'emerald
export I=emerald,$I
cd ../platinum # esplora la dir platinum
export I=platinum,$I
```

Tesori: `emerald`, `platinum`

Percorso alternativo: la dimensione nascosta (.rift)

.rift — Dimensione Nascosta [directory nascosta]

Come arrivarci: `cd .rift` da entrance (deve saperla cercare)
Contenuto: `scroll`, `box/`, `arena/`, `spire/`

```
ls -a # mostra tutti i file nascosti (iniziano con .)
ls -aF # nascosti + suffissi tipo/eseguibile
cd .rift # entra nella dimensione alternativa
cat scroll # "You have entered the rift"
```

Comandi: `ls -a`, `ls -aF` **Biforcazione:** `arena/` (boss) oppure `spire/` (puzzle)

arena / pit — Arena e Pozzo (boss finale) [directory nascosta]

Come arrivarci: `cd arena` poi `cd pit`
Contenuto pit: `scroll`, `drummer*`, `nyarlathotep*`, `wizard-light/`, `platinum/`, `treasure/`, `end`

```
cat scroll
./drummer # suggerisce: usa mv o rm per togliere drum
rm drum # indebolisce il boss rimuovendo il drum
./wizard-light # attiva bonus +2 per ogni colpo
./nyarlathotep # affronta il boss (richiede sword; .bless da' ulteriore bonus)
export I=hair,$I
export I=bracelet,$I
export I=armour,$I
cat end # messaggio di fine gioco
```

Comandi: `rm <file>`, `mv <src> <dst>` **Tesori:** `hair`, `bracelet`, `armour`

spire / mezzanine / .elevator / .satellite — Puzzle Robot [directory nascosta]

Come arrivarci: `cd spire` da .rift, poi `cd mezzanine`

```
cat window # narrativa: vista sul rift
./button # crea symlink "door" -> .elevator
cd door # entra nell'ascensore tramite symlink

% In .elevator:
ls -aF # vedi .direction, .satellite/, display*
./display # animazione (ascensore conta fino a 666)
cd .satellite # entra nel satellite

% In .satellite -- puzzle robot:
./spell # impara ls -aF per trovare file nascosti
cat notebook # leggi istruzioni per assemblare il robot
cat alpha beta gamma delta | base64 -d > robot # decodifica i pezzi
sha256sum -c sha256sums # verifica integrita' del robot assemblato
chmod +x robot # rendi il file eseguibile
./robot # esegui il robot
```

Comandi: `ls -aF`, `base64 -d`, `sha256sum -c`, `chmod +x`

3. Riepilogo comandi

Stanza	Comando	Cosa fa	Sintassi esatta
entrance	ls	elenca file e directory	`ls`
entrance	cd	entra in una directory	`cd cellar`
entrance	cat	legge un file di testo	`cat scroll`
cellar	ls -F	suffissi: / dir, * eseg.	`ls -F`
cellar	alias	crea abbreviazione	`alias ls='ls -F'`
cellar	export	crea variabile inventario	`export I=amulet,$I`
cellar	echo	stampa valore variabile	`echo $I`
armoury	./script	esegue uno script	`./potion`
armoury	export HP	crea variabile punti vita	`export HP=15`
armoury	cd ..	torna alla dir precedente	`cd ..`
chamber	pwd	mostra percorso corrente	`pwd`
chamber	ln -s	crea symlink (portale)	`ln -fs ../path portal`
chamber	let	aritmetica su variabili	`let "HP=HP-5"`
chapel	frecce	scorre la history comandi	tasto ↑ / ↓
aviary	TAB	autocompleta nomi file/dir	`cd h<TAB>`
library	tome	6 comandi avanzati (random)	`./tome`
	pushd/popd	stack di directory (LIFO)	`pushd <dir>`
	tree -F	struttura ad albero	`tree -F`
stronghold	cp	copia un file	`cp orb1 orb2`
nursery	find	cerca file per nome	`find . -name "potion"`
.rift	ls -a	mostra file/dir nascosti	`ls -a`
pit	rm	cancella un file	`rm drum`
pit	mv	sposta/rinomina un file	`mv src dst`
.satellite	base64	decodifica file base64	`cat a b c d | base64 -d > out`
.satellite	sha256sum	verifica integrità	`sha256sum -c sha256sums`
.satellite	chmod +x	rende eseguibile un file	`chmod +x robot`

4. Inventario finale atteso

Variabile $I (oggetti raccolti):

- `amulet` — da `cellar/treasure`
- `sword` — da `armoury/treasure`
- `diamonds` — da `chamber/treasure`
- `crystal` — da `aviary/crystal`
- `crown` — da `hall/treasure`
- `goblet` — da `stronghold/goblet`
- `emerald` — da `lab/treasure`

- `platinum` — da `lab/platinum`
- `hair` — drop da `nyarlathotep`
- `bracelet` — drop da `nyarlathotep`
- `armour` — drop da `nyarlathotep`

Variabile $HP:

- Inizio: 15 (beve) o 10 (rifiuta pozione)
- −5 per ogni colpo in combat
- +5 da `nursery/spell` (se pozione copiata)
- `wizard-light` dà +2 per colpo nel pit

File/stati speciali:

- `entrance/portal` (symlink a hall)
- `vault/.bless` (bonus combat)
- `robot` (assemblato nel satellite)

5. Domande per l'interrogazione

Livello 1 — Comandi di base

1. Qual è il primo comando da eseguire entrando in una nuova stanza? Perchè?
2. Come si entra in una directory? Come si torna alla directory precedente?
3. Cosa significa il simbolo `/` dopo un nome nell'output di `ls -F`? E il simbolo `*`?
4. Come si legge il contenuto di un file di testo? Scrivi la sintassi esatta.
5. Cosa fa `alias ls='ls -F'`? Dopo averlo eseguito, cosa succede se digiti solo `ls`?

Livello 2 — Filesystem e shell

6. Come hai ottenuto l'`amulet`? Scrivi la sequenza di comandi dall'ingresso.
7. Cos'è una variabile di shell? Come si crea? Come si legge il suo valore?
8. Come hai creato la variabile `HP`? Cosa succedeva ai tuoi HP durante i combattimenti?
9. Cos'è un symlink? Dove ne hai incontrato uno? Come è stato creato?
10. Come hai ottenuto il `goblet` nella fortezza? Perchè `./goblet` non funzionava subito?
11. Cosa fa `find`? In quale stanza l'hai usato e con quale sintassi esatta?
12. Come si vede una directory nascosta? Dove hai trovato la prima directory nascosta?

Livello 3 — Avanzato

13. Spiega cosa fa questa riga: `cat alpha beta gamma delta | base64 -d > robot`
14. Cos'è `chmod +x`? Perchè era necessario prima di `./robot`?
15. Qual è la differenza tra `mv` e `cp`? In quale situazione del dungeon era utile ciascuno?
16. Hai letto il codice sorgente di qualche script? Cosa hai capito o cosa ti ha sorpreso?