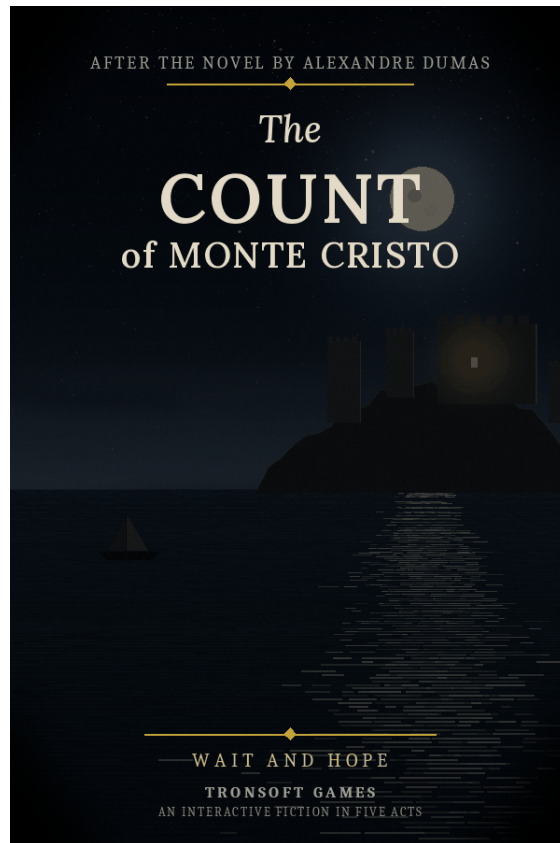




The Count of Monte Cristo

The manual

Version 2.0.0 · Tronsoft Games · for TIFE Core 2.3.0
adapted from the novel by Alexandre Dumas, 1844

Contents

Chapter	What is in it
1	What this is <i>The package, the engine, and how the two fit together.</i>
2	How to play <i>Beginning, the controls, the turn, and movement.</i>
3	The story, act by act <i>Every room, in order, with the ways out.</i>
4	The people <i>Eight speaking parts, and why three of them stay behind.</i>
5	The four names <i>The mechanic the second half is built on.</i>
6	Errands and scoring <i>Where all 150 points are.</i>
7	The six ways to die <i>And what to do about having no undo.</i>
8	Two measured routes <i>A perfect game and a lean one.</i>
9	Porting to Core 2.3.0 <i>What a smaller engine cost, and what it gave.</i>
10	What is patched, and why <i>Eleven corrections, none in the engine.</i>
11	Sound and accessibility <i>Six profiles, and captions for everything.</i>
12	Verification <i>What was tested, and against what.</i>
13	Reference tables <i>Everything shipped, listed.</i>

Chapter one

What this is

The Count of Monte Cristo is a parser adventure in five acts, adapted from Alexandre Dumas's novel of 1844. It is played by typing short commands, or by clicking, and it takes about two hours the first time.

This package is a complete, self-contained game for the Tronsoft Games Portal, built to the Tronsoft Games Package Specification v1.4. It contains three separable things, and the separation is the point:

- **The engine.** TIFE Core 2.3.0, in `js/core`, nine files, included exactly as published in the TIFE Core native installation package and not modified in any way. Their SHA-256 sums are printed in chapter ten and in the patch note, so the claim can be checked rather than believed.
- **The game.** `games/monte-cristo/game.js`, a single object of data: 20 rooms, 66 objects, 8 people, 110 story rules, 49 hint ladders and about 10,082 words of prose. It contains no logic. Everything it describes is interpreted by the engine.
- **The patch layer.** `js/interface.js`, `js/sound.js`, `js/map.js` and `css/theme.css`, plus the entry file. The engine's own shell adapter is shipped verbatim as well, so anything this game needs beyond it is done by patching prototypes from outside, after the engine has loaded and before the shell runs. Chapter ten lists every patch and why it is there.

Engine purity

Eleven files ship byte-for-byte as published: the nine services in `js/core`, the shell adapter `js/game.js`, and `css/game.css`. Their digests are in `docs/engine-purity.json`, the package validator rechecks every one of them, and the external verifier checks them a third time against the uploaded TIFE Core package itself. This is the house rule for this line, and it is what makes the next paragraph safe: nothing in this game can have been achieved by quietly editing the engine.

Two editions, both current

There is also a version 1.0.0 of this game, which is a data-only `.tife-game` package for the TIFE Developer Specification v3.0.1. That edition contains no executable file at all, because v3.0.1 forbids it. This one contains an engine, because Core 2.3.0's game format is a JavaScript object and a v1.4 package is a playable bundle. Neither supersedes the other; they are different formats for different engines.

What is in the package

Path	Contents
<code>index.html</code>	The entry file the portal launches.
<code>manifest.json</code>	Package metadata, v1.4 schema.
<code>games/monte-cristo/game.js</code>	The whole game, as data.
<code>js/core/</code>	TIFE Core 2.3.0, nine files, unmodified.
<code>js/game.js</code>	The engine's shell adapter, unmodified.
<code>js/interface.js</code>	The patch layer: seven corrections, and this game's panel.
<code>js/sound.js</code>	The cue player, and the engine's audio redirected to it.
<code>js/map.js</code>	The map panel repainted as a five-act chart.
<code>css/game.css</code>	The engine's stylesheet, unmodified.
<code>css/theme.css</code>	This game's skin, and the six profiles made structural.
<code>docs/engine-purity.json</code>	The digests of all eleven verbatim files.
<code>sounds/</code>	8 cue files, 22 kHz mono WAV, generated.
<code>assets/</code>	Cover, icon, banner, logo, loading plate.
<code>screenshots/</code>	Four plates from a real browser playthrough.
<code>docs/</code>	This manual, the tutorial, the description, the credits.
<code>patches/</code>	The patch note for this version, and the change log.
<code>README.txt</code> , <code>LICENSE.txt</code>	The short version, and the terms.

Table 1.1 — the package, as shipped.

Nothing to install, nothing sent anywhere

The package makes no network request of any kind. It has no external font, script, image or stylesheet. It sets no cookie. The only thing it writes is your saved games and your accessibility preferences, both in this browser's local storage, on this device, and both declared in the manifest.

Chapter two

How to play

You are Edmond Dantès. It is February 1815, you are nineteen, you are bringing a merchant ship into Marseilles because her captain died at sea, and you are to be made captain and married in the same week. Two men on that quay would rather you were not.

The controls

Type a command and press Enter. Small words are ignored, so *read the log* and *read log* are the same. Capitals do not matter.

```
> read the log
> take packet
> talk to danglars
> become the count
> down
```

Under the transcript are four rows of buttons: actions, directions, information, and the things you can act on here. Choose an action and its possible targets appear. The whole story can be finished with the mouse alone, and the whole story can be finished with the keyboard alone.

Command	What it does
LOOK	Describe the room again. Free.
EXAMINE x	Look closely at one thing. Do this often.
TAKE x / DROP x	Pick up, or put down.
USE x	Do the obvious thing to something.
OPEN x	A lid, a wardrobe, a chest.
SEARCH x	Turn over, look under, go through.
READ x	Read something written.
TALK TO x	Speak. Ask again; people say more.
WAIT	Let time pass. A real action in this story.
INVENTORY, I	What you are carrying.
ERRANDS	What is being asked of you.
JOURNAL	What has happened, in order.
SCORE, STATS	Where you stand. Free.
ACHIEVEMENTS	What you have managed. Free.
HINT	Three graded rungs for where you stand.
SAVE name	Keep a game. As many as you like.
RESTORE name	Bring one back, from anywhere.
RESTART	Begin again from the deck of the Pharaon.
TRANSCRIPT	The whole game so far, as a text file.
HELP	The engine's own summary.

Table 2.1 — the standard verbs.

The verbs this story adds

Seven, and every one of them is an alias of USE. That is not a compromise so much as a discovery: this engine lets a game name its own synonyms for the verbs it already has, so a story verb costs nothing and requires no change to the parser. DIG WALL and USE WALL are the same command.

Verb	Also	What it is for
BECOME	ASSUME, WEAR	Put on one of the names you have earned.

Verb	Also	What it is for
DIG	EXCAVATE, DELVE	Work at stone or ground with whatever edge you have.
HIDE	CONCEAL	Put yourself where you will not be seen.
CUT	SLASH, SEVER	Use a blade on what is holding you.
DIVE	PLUNGE, SWIM	Go into the water. Think first.
SIGNAL	HAIL, SHOW	Send word, by hand or by lantern.
HOPE	WAIT, Z, ENDURE	The second of the two words.

Table 2.2 — from the parser table in *game.js*.

What happens in a turn

- **Before.** Entering a room, using a thing, talking to somebody or taking something can each set off a story rule, and if one fires you are told about it first.
- **The action.** Then the plain meaning of the command is carried out.
- **After.** Then the consequences: an errand ticked on, a point, a journal entry, a way opening.

Which is why you are always told that you have left a place before you are told where you have arrived. LOOK, INVENTORY, JOURNAL, ERRANDS, SCORE, STATS, ACHIEVEMENTS and HELP are free and cost no move. HINT is not: see chapter ten.

Movement

Compass directions, the four diagonals, and up or down. There are 31 ways between the 20 rooms, and 9 of them start shut. None of them needs a key: each is opened by something you do, and each says in plain words why it will not open yet. If a way is shut, what to do about it is in the room you are standing in.

Chapter three

The story, act by act

Every room in the game, in the order the story visits them, with how many ways out it has and how much is in it. Room names are as they appear in play.

I. Marseilles, 1815

Room	Ways out	At first	Things	People
The Deck of the Pharaon	1	—	3	1
The Quai des Belges	4	—	1	—
Pere Pamphile's Tavern	1	—	3	1
The Arbour at La Reserve	2	1 shut	2	1

Table 3.1 — I. Marseilles, 1815.

II. The Denunciation

Room	Ways out	At first	Things	People
The Village of the Catalans	1	—	2	1
The Landing Stair, Chateau d'If	1	—	2	—
The Dungeon of Edmond Dantes	1	1 shut	3	—
The Deputy Procureur's Cabinet	1	1 shut	3	1

Table 3.2 — II. The Denunciation.

III. The Chateau d'If

Room	Ways out	At first	Things	People
The Tunnel	2	—	1	—
The Cell of the Abbe Faria	2	1 shut	6	1
The Sea Below the Rampart	1	1 shut	3	—
The Isle of Tiboulén	1	1 shut	3	—

Table 3.3 — III. The Chateau d'If.

IV. The Treasure and the Count

Room	Ways out	At first	Things	People
The Tartan Jeune Amelie	1	1 shut	1	—
The Grotto of Monte Cristo	1	1 shut	2	—
No. 30, Champs-Elysees	6	1 shut	2	1
The House at Auteuil	1	—	3	1

Table 3.4 — IV. The Treasure and the Count.

V. The Reckoning

Room	Ways out	At first	Things	People
The Hotel de Morcerf	1	—	2	—
The Chamber of Peers	1	—	2	—
The Banker's Study	1	—	3	—
Villefort's Garden	1	—	3	—

Table 3.5 — V. The Reckoning.

Two doors that do not come back

The stair down from the garden at La Réserve, and the road north from the island, are both one-way. Nothing required is ever left behind and every errand stays reachable, but two of the four keepsakes exist only in the first act. That is deliberate: the point of a keepsake is that you had to think of it at the time.

Chapter four

The people

Eight speaking parts, 37 things said between them. Everybody says more if asked again, and what they say last is nearly always what matters.

Person	Found in	Lines
Danglars	The Deck of the Pharaon	3
Caderousse	Pere Pamphile's Tavern	3
Mercedes	The Arbour at La Reserve	3
Fernand	The Village of the Catalans	3
Villefort	The Deputy Procureur's Cabinet	8
the Abbe Faria	The Cell of the Abbe Faria	8
Haydee	No. 30, Champs-Elysees	4
Bertuccio	The House at Auteuil	5

Table 4.1 — the cast, and where each of them is.

Danglars, Fernand and Villefort are all in the first two acts and none of them is in Paris. That is not an omission. Core 2.3.0 resolves a person's name against every person in the game at once, regardless of where they are standing, so no two people can answer to the same word — which rules out having a young Danglars on the Pharaon and a Baron Danglars in a study nine years later.

So the fifth act is fought on paper: four lines in a newspaper, a bond ledger, a semaphore almanac, a confession made to the wrong priest, and a box dug out of a garden at Auteuil. The Count barely speaks to the three of them, which is very nearly how Dumas does it, and the constraint improved the adaptation.

Chapter five

The four names

This is the mechanic the second half is built on, and it is the one thing in the adaptation that is wholly invented. In the novel Dantès moves through Paris as several people at once. Here that is a verb.

```
> become the count
> become busoni
```

Each name is earned from an object or from people, never handed over:

- **Sinbad the Sailor** is decided on by the crew of the *Jeune Amélie*, because a smuggler's deck does not ask for papers and a man pulled out of the sea has to be called something.
- **the Abbé Busoni** comes from Faria's habit, which you are carrying because you cut your way out of a sack wearing it.
- **Lord Wilmore** comes from a passport in a wardrobe, in a house bought furnished in a single morning.
- **the Count of Monte Cristo** comes from the deed to the island, which was in the chest under the slab.

Technically each name is an object you carry, and BECOME is USE. Wearing one clears the other three. Every door in the fifth act tests which is on:

Room	Opens to	And refuses
The Hôtel de Morcerf	the Count	Edmond and Sinbad — fatally

Room	Opens to	And refuses
The Chamber of Peers	the Count	<i>anybody without a witness</i>
The Banker's Study	the Count	<i>anybody without credit</i>
The House at Auteuil	the Abbé Busoni	<i>his master, but not his confessor</i>
Villefort's Garden	Lord Wilmore, then Edmond	<i>the Count, who is not believed</i>

Table 5.1 — which name each door in act five wants.

The one that kills you

In the Hôtel de Morcerf, becoming Edmond Dantès or Sinbad the Sailor ends the story. Fernand knew both men and believes one of them dead. Haydée tells you so in as many words, twice, before you can get into the house.

Chapter six

Errands and scoring

150 points. Eight required errands at fifteen each, one optional errand worth ten, and four keepsakes worth five each.

Errand	Points
Bring the Pharaon home	15
Learn who is writing	15
Be married	15
The abbe's legacy	15
Leave the Chateau d'If	15
The treasure of Spada	15
The four names	15
The reckoning	15
The house of Morrel (<i>optional</i>)	10
Errands together	130

Table 6.1 — the 9 errands.

Keepsake	Where it is	Points
Catalan ring	The Arbour at La Reserve	5
your father's letter	The Village of the Catalans	5
the abbe's treatise	The Cell of the Abbe Faria	5
the Spada signet	revealed in play	5
Keepsakes together		20

Table 6.2 — nothing depends on any of these.

A complete winning game — every required errand, no keepsakes, nothing optional — scores 120 in 93 moves. All 150 takes 115. The difference is not difficulty; it is attention.

Achievements

Achievement	For
Acting Master	<i>Brought the Pharaon in and answered for the captain's errand.</i>
Number 34	<i>Left the fortress by the only door it has.</i>
The Vine Trellis	<i>Stood in the shadow and heard all three names before the gendarmes did.</i>
His Father's Hand	<i>Kept the four lines your father wrote about your hands.</i>
Four Names	<i>Held Sinbad, Busoni, Wilmore and Monte Cristo at once.</i>
Thirteen Millions	<i>Opened the furthest angle of the second cave.</i>
The Reckoning	<i>Undid all three, each by the thing he built.</i>
A Thin Silver Band	<i>Carried Mercedes' ring out of Marseilles.</i>
The Abbe's Pupil	<i>Heard every one of Faria's lessons to the end.</i>
The Cardinal's Thumb	<i>Kept the one thing in the chest that was worth nothing.</i>
From Sinbad the Sailor	<i>Paid a debt that was never owed to you.</i>
Fifty Feet of Rock	<i>Dug through to the Abbe Faria.</i>
On the Monarchy of Italy	<i>Brought two thousand pages of shirt-linen out of the Chateau d'If.</i>
Wait and Hope	<i>Finished the story with every point it had to give.</i>

Table 6.3 — 14 of them, none worth a point.

Chapter seven

The six ways to die

Six commands end the story badly. Every one is described in the room before it can be typed, and none can be walked into blind.

Room	Command	What happens
The Landing Stair, Château d'If	DIVE WATER	<i>four fathoms, in irons</i>
The Sea Below the Rampart	WAIT	<i>the thirty-six pound shot goes first</i>
The Sea Below the Rampart	DIVE SEA	<i>swimming inside a sewn sack</i>
The Isle of Tiboulén	DIVE GALE	<i>back into the gale, having survived it once</i>
The Hôtel de Morcerf	BECOME EDMOND	<i>Fernand knew him, and has a drawer</i>
The Hôtel de Morcerf	BECOME SINBAD	<i>Fernand knew the voice</i>

Table 7.1 — each was reached along the route during verification, confirmed fatal and confirmed warned beforehand.

There is no undo on this engine, so save

The 3.0.1 edition could promise that dying cost a keystroke. Core 2.3.0 has no UNDO, so this edition cannot. What it has instead is unlimited named saves, and — because of a shim described in chapter ten — a save can be restored even after the story has ended. Every fatal ending in this game says so in the same breath, naming both RESTART and the SAVE MANAGER.

Chapter eight

Two measured routes

Both were measured by playing the built game through the stock engine, one command at a time, and the full one was then replayed in Chromium through this package's own shell by typing every command into the command form.

	A perfect game	A lean game
Moves	115	93
Score	150 of 150	120 of 150
Rooms seen	20 of 20	most of 20
Errands finished	9	8
Achievements	14	most
Ends in	the grotto	the grotto

Table 8.1 — as measured against the built game.

Chapter nine

Porting to Core 2.3.0

This game existed first for the TIFE Developer Specification v3.0.1, whose reference runtime offers about twenty-two rule conditions and twenty-eight rule effects. Core 2.3.0 offers eight and ten. It has no way for a rule to move the player, no way for a rule to move a person, no per-errand objectives and no undo. Porting was therefore not a translation; the machinery was rebuilt. The prose was carried over unchanged.

What the smaller engine cost, and what replaced it

What 3.0.1 had	What this edition does instead
<i>A rule effect that moves the player</i>	Locked exits. Every forced move — the arrest, the burial, the nine years — is a way that is shut with an in-character refusal and opened by the beat that earns it. The rooms beyond simply have no way back, so the one-way act boundaries survive intact.
<i>A rule effect that moves a person</i>	Nobody moves. Three of the eight people stay in the first two acts and the fifth act is fought on paper. See chapter four: this improved the story.
<i>Twenty-nine errand objectives</i>	Errand progress as a percentage, which is what this engine's quest log displays anyway.
<i>Eight custom story verbs</i>	Seven aliases of USE, declared in the game's own parser table. No engine change, and both spellings work.
<i>A condition that counts how many times you have talked to somebody</i>	Staged rules written in reverse. See below.
<i>UNDO, forty turns deep</i>	Unlimited named saves, restorable after an ending.
<i>Silence</i>	8 sound cues, every one of them also a caption. The engine has an audio service and an accessibility service; the 3.0.1 edition had neither, so this edition gained something the other cannot have.

Table 9.1 — the port, item by item.

Reverse-ordered staged rules

Core runs every rule matching a trigger in array order, and a flag set by an early rule is visible to a later one in the same pass. A counter written forwards therefore cascades: stage one sets its flag, stage two sees it and fires, stage three sees that and fires, and the whole ladder collapses into a single turn.

Written backwards it behaves. Stage three is tested before stage two can set the flag that would satisfy it, so exactly one stage fires per turn. Every counter in this game — the eight conversations with Faria, the five stages of digging out of cell 34, the five rungs of Bertuccio's confession — is built this way, and the harness asserts that exactly one stage fires per turn.

A trap worth naming

RuleService marks a rule as fired only after its whole effect list has run, and moveItem emits inventory.changed, which runs the inventory.changed rules again. So a rule triggered by inventory.changed that itself moves an item re-enters itself and recurses until the stack goes. Three rules in this game were written that way and crashed the engine on the first run. All take consequences now hang off the action instead, which also puts the consequence after “Taken.” rather than before it.

Chapter ten

What is patched, and why

Every correction below is in the engine, affects any game running on it, and is made from outside by patching prototypes in js/interface.js. Nothing in js/core, js/game.js or css/game.css is edited. Each is asserted in the harness in both directions, so neither the defect nor the correction can quietly change.

What the engine does	What is done about it
<i>execute() handles a restart by calling reset(), which sets lastAction to null; control returns to dispatch(), which reads lastAction.type and throws. The typed command and the header button both reach it.</i>	TIFECore.dispatch intercepts a restart, does it in order, and gives the engine a harmless last action first.
<i>HINT and RESTART are not in the list of actions that do not consume a turn, so asking for help costs the same as digging through fifty feet of rock.</i>	isTurnAction is replaced with a list that includes both.
<i>The save key is tife-core-saves-<id>, the engine's own, rather than one derived from the package slug as section 23 requires.</i>	An accessor on SaveService.prototype with both halves - the constructor assigns this.key, so a getter alone throws under strict mode. Saves live under tronsoft_monte_cristo_save.
<i>serialize() lists fourteen fields and gameOver is not one of them, so a save made before a fatal ending restores into a game that is still over and refuses every command.</i>	serialize records it and load clears it. The save also carries the version of the game it was made with.
<i>dispatch() refuses everything but a restart once the story has ended - including a typed RESTORE, which can leave a player with no way back at all.</i>	A restore is allowed through. This is what lets the fatal endings in this game honestly name the SAVE MANAGER as a way back.
<i>A restore reprints the room but never re-renders, so the score, the move count and every panel keep showing the game that was abandoned.</i>	load re-renders.
<i>use(), search() and read() act on a name that resolves to nothing, minting used.<word> flags and answering questions nobody asked.</i>	Presence checks, with one deliberate exception: the four names are passed through so that BECOME COUNT before the Count exists gets a real refusal rather than a shrug.
<i>open() looks only in the room, so nothing can be opened while it is being carried.</i>	Extended to the inventory.
<i>The shell narrates a hard-coded line about Colossal Cave Expedition over the cover of whatever game it is running.</i>	AccessibilityService.narrate is wrapped and the line is taken from the game data instead.

What the engine does	What is done about it
<i>The engine's audio service synthesises its own cave ambience.</i>	<code>startAmbient</code> , <code>click</code> and <code>success</code> are delegated to this game's cue files, so the sound of the game is one set and not two.
<i>The panel table and the action palette are closure-local in the shell, and the shell wipes their contents on every render.</i>	The names <code>panel</code> and the act chart are injected and repainted through a <code>MutationObserver</code> , keyed on a signature and on whether the marker is still attached. The story-verb buttons are marked <code>data-story-verb</code> so the shell's own click handler ignores them.

Table 10.1 — eleven corrections, none of them in the engine.

One trap that cost a crash

The injected panel must not carry the class `panel`. The shell's render walks every `.panel` element and looks its `data-panel` up in a table it keeps in a closure, so a panel it has never heard of throws on every single turn. It wears its own class and borrows the geometry in `css/theme.css`.

The engine, as shipped

These are the SHA-256 digests of the eleven files that ship byte-identical. They are written by `tools/core23_purity.py` from the uploaded package, recorded in `docs/engine-purity.json`, rechecked by the package validator, and checked once more by the external verifier against the upload itself.

File	SHA-256
<code>css/game.css</code>	47dc7e0eb7a03c54bb1ef033fbfd9de2228db9ec3b10eef613b60e6da9344e3a
<code>js/core/accessibility-service.js</code>	35135c7a095638c85680deb5babbe3ea78e571e0ce9b134594452d406ac9640a
<code>js/core/audio-service.js</code>	37a1f64f2800758d3aa668855a27c2eb58810caee91930700d48061b3304ae61
<code>js/core/event-bus.js</code>	c09e488fa6cfd89fa2af7c92f853b90fb9cec5820338fccba6b8103a92b318bb
<code>js/core/parser-service.js</code>	ac7462180e220e53983387882361d4bec59159859c98fed6b9c0fcdcd595a6426
<code>js/core/rule-service.js</code>	ada0a5ca8b4d3484722d5642f0bfe09d99aaabe88f1e9f7de63995da64f6ec80
<code>js/core/save-service.js</code>	5c1399bd9264d8a4fedff9fe05975750e8e69045a63f5ec2d8dfacdb2fec7748
<code>js/core/state-machine.js</code>	1fc58be4dacd30dc8e8a21ca1e499516c947c6a9cf32249a5926aa8e25246d1c
<code>js/core/tife-core.js</code>	876e983fc5c5daebb6f1d422ceb1429bb649a3df675bdd4c1094748041411132
<code>js/core/world-service.js</code>	fe8011075e5228a2b1e6502aace52623be80c9c2d78552a4121c05e17c74bf66
<code>js/game.js</code>	528f6cff214560bf2f2fce0b5cc289b3377cb987af4bc4fe40b1ed75f9874a56

Table 10.2 — the engine is unmodified, and this is how to check it.

Chapter eleven

Sound and accessibility

The engine ships an accessibility service with six presets and an optional spoken narration. This game makes those presets structural rather than cosmetic, and adds its own sound.

Profile	What changes
Standard	Two columns of panels either side of the transcript.
Low vision	Larger type, white on black, doubled borders and focus rings, larger controls.
Blind / screen reader	One linear column in reading order; the map panel and the cover art are removed; the transcript is a polite live region and the targets become a full-width list; narration on.
Hard of hearing	Captions on. Every sound cue is written on screen, including what each room sounds like when you LOOK.
Motor assistance	Every control at least 56 pixels; reduced typing; the story is finishable by clicking alone; no timed input anywhere in the game.
Cognitive support	One column, narrower measure, and the panels that only add pressure — score, statistics, achievements, conversation history — removed.

Table 11.1 — the six profiles, from the ACCESSIBILITY panel.

Text size is separately adjustable from 90 to 150 per cent, and every individual setting can be toggled on its own, which turns the presets into starting points rather than cages.

The cues

Four room ambiences and four one-shot cues, 8 in all, synthesised with the Python standard library: nothing sampled, nothing recorded, nothing licensed. Each is 22 kHz mono WAV. The ambiences are cross-faded so they loop without a click, and follow the act rather than the room, so walking about inside one act does not restart them.

Cue	The caption that goes with it
harbour	<i>Rigging, gulls, and a crowd on the jetty.</i>
stone	<i>Water on stone, a long way underground.</i>
sea	<i>Open water and canvas.</i>
salon	<i>A clock, and carriages on gravel outside.</i>
score	<i>A point of reckoning.</i>
unlock	<i>A way opens.</i>
death	<i>A bell, once.</i>
click	— (a button click; nothing to say)

Table 11.2 — no information in this game is carried by sound alone.

Chapter twelve

Verification

Nothing in this manual is claimed on the strength of having written it. Each of the following was measured against the built game running on the engine as uploaded.

- **2,101 assertions** in the harness, all passing. The engine services and the game's own patch layer are loaded into a sandbox in the same order the entry file loads them, with the small amount of browser furniture they touch, so anything the harness proves it proves about the shipped game on the shipped engine.
- **Both routes played end to end** through the real parser, one command at a time, every response checked for the phrases the engine emits when it has not understood.
- **All six fatal endings** reached along the route, confirmed fatal, confirmed warned in the room text beforehand, and confirmed to refuse ordinary commands afterwards.
- **Every rule checked against the engine's actual vocabulary** — trigger, conditions and effects — so a rule cannot silently reference something Core 2.3.0 does not implement.
- **Every word in the game checked for collisions**, in each room and across the whole cast, because this engine resolves an object against the room and the inventory and a person against everybody in the game.
- **A dead-end sweep**: the route resumed from every fourth move of itself, each time finishing on 150.
- **Staged rules** confirmed to fire exactly one stage per turn.
- **Save, restore and restart**, including restoring into a game that had ended, the RESTART crash reproduced on an unpatched engine and then shown fixed, and the patched save format checked to carry what the engine's did not.
- **A browser playthrough in Chromium** through this package's own shell, typing all 115 commands into the command form and finishing on 150, with layout and overflow checks in all six accessibility profiles, a turn taken by mouse alone, a caption confirmed for a sound cue, and ten screenshots from that run.

Chapter thirteen

Reference tables

Everything shipped, counted

	Count
Acts	5
Rooms	20
Ways between rooms	31
shut at first	9
Objects	66
portable	25
People	8
Things they say	37
Errands	9
Story rules	110
Achievements	14

	Count
Hint ladders	49
Sound cues	8
Keepsakes	4
Words of prose	10,082
Maximum score	150
Par	115 moves

Table 13.1 — from the built game.

Every room

Room	Act	Ways out
The Deck of the Pharaon	I	down
The Quai des Belges	I	east, north, up, west
Pere Pamphile's Tavern	I	west
The Arbour at La Reserve	I	down, south
The Village of the Catalans	II	east
The Landing Stair, Chateau d'If	II	down
The Dungeon of Edmond Dantes	II	west
The Deputy Procureur's Cabinet	II	down
The Tunnel	III	east, west
The Cell of the Abbe Faria	III	east, up
The Sea Below the Rampart	III	down
The Isle of Tiboulén	III	east
The Tartan Jeune Amelie	IV	east
The Grotto of Monte Cristo	IV	north
No. 30, Champs-Elysees	IV	east, north, northwest, south, southwest, west
The Hotel de Morcerf	V	south
The Chamber of Peers	V	southeast
The Banker's Study	V	west
The House at Auteuil	IV	east
Villefort's Garden	V	northeast

Table 13.2 — all 20 rooms.

Colophon

The Count of Monte Cristo version 2.0.0, a game package for the Tronsoft Games Portal built to Package Specification v1.4 and running on TIFE Core 2.3.0. Published by Tronsoft Games, 2026-07-30.

Adapted from Alexandre Dumas, *Le Comte de Monte-Cristo*, first serialised 1844–1846 and in the public domain worldwide. Tronsoft Games claims no ownership of the novel. The adaptation, the game text, the puzzle structure, the artwork and the sound are original work.

This manual was generated from `core23/build/facts.json` and from the `built game.js`, so every table and every number in it is a statement about the game as shipped rather than the game as intended. Set in Lora and Caladea, with DejaVu Sans Mono. Artwork drawn with the Python Imaging Library, sound synthesised with the Python standard library, this document produced with ReportLab.

Produced with the assistance of a large language model under human direction. Full credits, including every tool and typeface licence, are in `docs/credits.txt`.