

1. Panoramica

SCUM consente di creare e personalizzare le missioni tramite file JSON presenti in cartelle specifiche, a seconda che si utilizzi un server multigiocatore o si giochi in modalità sandbox. Tutte le modifiche ai file JSON delle missioni richiedono il riavvio del server (o del gioco in modalità sandbox) affinché le modifiche abbiano effetto.

Link a uno strumento che, attraverso una semplice interfaccia grafica, semplifica il processo di creazione, garantendo al contempo che l'output sia strutturato correttamente e convalidato rispetto ai dati di gioco noti:

[SCUMQuestTool](#)

Link al documento, tradotto e disponibile in cinese:

[SCUM自定义任务 - 中文说明文档](#)

Link al documento, tradotto e disponibile in tedesco:

[Benutzerdefinierte Quest-Modifikation](#)

Comandi rilevanti

- **#ExportQuests**
Crea la cartella Missioni (se non esiste già) ed esporta i file JSON relativi alle missioni, fornendo un'istantanea di tutte le missioni correnti.
- **#GetMeshInfo**
Utilizzato per le condizioni **di interazione** per recuperare informazioni sugli oggetti sulla mappa (ad esempio, per posizionare oggetti interattivi).

2. Struttura di cartelle e file

Cartella delle missioni

A seconda della configurazione, la cartella Missioni si trova nel seguente percorso:

- **Server multigiocatore:**
`<Server>\SCUM\Saved\Config\WindowsServer\Quests`
- **Sandbox :**
`%LocalAppData%\SCUM\Saved\Config\WindowsNoEditor\Quests`

Questa cartella viene creata automaticamente la prima volta che esegui il comando `#ExportQuests` . All'interno della cartella Quests troverai le seguenti sottocartelle:

1. Bloccato

- Contiene `BlockedQuests.json` , che consente di bloccare (disabilitare) determinate missioni in base al nome o di bloccare tutte le missioni predefinite.

2. Ignora

- Utilizzato per **aggiungere o aggiornare** missioni personalizzate. All'avvio del server (o del gioco in modalità sandbox), analizza i file JSON in questa cartella e applica eventuali missioni nuove o aggiornate.

3. Elenco missioni

- Contiene `CustomQuestList.json` (elenco di tutte le missioni personalizzate esportate) e `DefaultQuestList.json` (elenco di tutte le missioni predefinite).
- La modifica di questi file **non** influisce sul server o sul gameplay; sono solo a scopo di riferimento.

Blocco delle missioni predefinite

Se desideri impedire la visualizzazione di determinate missioni predefinite, puoi utilizzare il file `BlockedQuests.json` che si trova nella cartella `Blocked` . Aggiungi semplicemente i nomi delle missioni che desideri disattivare. Ad esempio:

```
{
  "BlockAllDefaultQuests": falso,
  "NomiblockQuest": [
    "T1_AR_Fetch_45ACPAmmobox",
    "T1_AR_Fetch_50AEAmmobox",
    "T1_AR_Fetch_9mmAmmobox"
  ]
}
```

Imposta `BlockAllDefaultQuests` su `true` se vuoi disabilitare *tutte* le missioni predefinite.

3. Creazione di missioni personalizzate

Tutte le missioni personalizzate devono essere inserite come **un oggetto JSON per file** nella cartella `Override` . Ogni file descrive una singola missione. Al riavvio del server (o del gioco), SCUM caricherà queste missioni.

Esempio di struttura del file

```
Missioni
├── Bloccato
│   └── BlockedQuests.json
├── Sostituisci
│   ├── MyFirstCustomQuest.json
│   └── MySecondCustomQuest.json
└── Elenco missioni
    ├── CustomQuestList.json
    └── DefaultQuestList.json
```

4. Anatomia di un file JSON di Quest

Ogni missione creata è definita da un **oggetto JSON "Quest"** con proprietà specifiche. Alcune proprietà sono **obbligatorie** (in grassetto), mentre altre sono *facoltative* (in corsivo) .

Di seguito la struttura, seguita da spiegazioni dettagliate:

```
{
  "AssociatedNPC": "Barista",
  "Livello": 1,
  "Titolo": "La mia prima missione personalizzata",
  "Descrizione": "Aiuta il barista con alcuni compiti."
  "OreLimite": 24,
  "RewardPool": [
    {
      // Qui possono essere inseriti uno o più oggetti Ricompensa.
    }
  ],
  "Condizioni": [
    {
      // Qui possono essere inseriti uno o più oggetti Condizione.
    }
  ]
}
```

Maggiori informazioni su `RewardPool` e `sulle relative condizioni` verranno fornite più avanti.

4.1 Proprietà obbligatorie

1. NPC associato

- Tipo: **stringa**
- Deve essere uno tra: "Armaiuolo" , "Banchiere" , "Barbiere" , "Barista" , "Dottore" , "Pescatore"/"Capitano di porto" , "Merceria" o "Meccanico" .
- Indica quale PNG offre la missione e da dove provengono le ricompense (incluse le ricompense con sconto nel negozio).

2. Livello

- Tipo: **intero** (intervallo valido: da 1 a 3)
- Specifica il livello di difficoltà o di importanza della missione.

3. Titolo

- Tipo: **stringa**
- Qualsiasi testo adatto per etichettare la tua ricerca.

4. Descrizione

- Tipo: **stringa**
- Una breve descrizione che illustra cosa deve fare il giocatore.

5. RewardPool

- Tipo: **array** di oggetti **Reward** .
- Ogni volta che viene generata una nuova istanza della missione, viene scelto casualmente **uno solo** di questi oggetti ricompensa.

6. Condizioni

- Tipo: **array** di oggetti **Condizione** .
- Definiscono gli obiettivi che devono essere raggiunti per completare la missione.

4.2 Proprietà opzionali

1. *Limite di tempo*Ore

- Tipo: **numero** (può essere intero o decimale)
- Imposta il limite di tempo in ore per il completamento della missione (ad esempio **24** , **48**, **5** , ecc.).
- Omettere se non c'è limite di tempo.

5. Oggetti premio

Ogni voce nell'array **RewardPool** è un oggetto **Reward** . SCUM inserisce un **massimo di cinque** "slot premio" in ogni oggetto, contati come segue:

1. **Gruppo Valuta (Normale, Oro, Fama):**

- Tutti e tre insieme contano come **1** slot premio, indipendentemente dal fatto che ne includa solo uno (ad esempio, solo CurrencyNormal) o tutti e tre (CurrencyNormal, CurrencyGold e Fame).

2. **Competenze:**

- Ogni **abilità** assegnata conta come **1** slot ricompensa per abilità.
- Esempio: se fornisci esperienza in Tiro con l'arco e Cucina, otterrai 2 slot ricompensa.

3. **Offerte commerciali:**

- Il **primo** TradeDeal all'interno di un singolo oggetto Ricompensa conta come **2** slot ricompensa.
- **Ogni TradeDeal aggiuntivo** nello stesso oggetto Ricompensa conta come **1** slot ricompensa.
- L'oggetto assegnato deve appartenere all'inventario del commerciante

Di seguito è riportato un esempio che mostra come funziona il conteggio delle ricompense:

```
{
  "ValutaNormale": 100,
  "ValutaOro": 1,
  "Fama": 10,
  "Competenze": [
    {
      "Abilità": "Cucinare",
      "Esperienza": 50
    }
  ],
  "Accordi commerciali": [
    {
      "Oggetto": "Arma_M9",
      "Prezzo": 50,
      "Importo": 2,
      "Consenti escluso": falso,
      "Fama": 10
    }
  ]
}
```

- **ValutaNormale (100) + ValutaOro (1) + Fama (10) → 1 slot ricompensa**
- **Abilità di cucina +50 EXP → 1 slot ricompensa**
- **Primo accordo commerciale → 2 slot premio**

Finora, il totale è di 4 slot ricompensa. Potresti aggiungere un'altra abilità o un altro TradeDeal (o qualsiasi altro elemento ricompensa a slot singolo) prima di raggiungere il limite di 5.

5.1 Proprietà degli oggetti ricompensa

- **CurrencyNormal** (**intero**)
L'importo di valuta normale che il giocatore riceve.
- **CurrencyGold** (**intero**)
La quantità di valuta d'oro che il giocatore riceve.
- **Fama** (**intero**)
Punti fama assegnati.
- **Abilità** (**serie** di oggetti **Abilità**)
Il giocatore ottiene esperienza abilità per ogni oggetto Abilità in questa serie.
- **TradeDeals** (**array** di oggetti **TradeDeal**)
Definisce i prezzi scontati o altrimenti modificati degli articoli nel negozio NPC corrispondente.

6. Oggetti abilità

Utilizzato all'interno dell'array **Abilità** all'interno di un oggetto Ricompensa.

```
{ "Abilità": "Cucinare",  
  "Esperienza": 50  
}
```

- **Abilità** (stringa)
Deve essere uno dei nomi di abilità predefiniti:
"Tiro con l'arco" , "Aviazione" , "Consapevolezza" , "Boxe " , "
Mimetizzazione " , "Cucina" , "Demolizione" , "Guida " , "Resistenza"
, "Ingegneria " , "Agricoltura " , "Pistola" , "Medicina" , "Armi da
mischia" , "Motocicletta" , "Fucili" , "Corsa " , "Cecchino " ,

"Furtività" , "Sopravvivenza" , "Tattiche" o "Furto" .

- **Esperienza** (intero)
Quantità di punti esperienza da concedere per quell'abilità.

7. Oggetti TradeDeal

Utilizzato all'interno dell'array `TradeDeals` all'interno di un oggetto Reward.

```
{  
  "Oggetto": "Arma_M9",  
  "Prezzo": 50,  
  "Importo": 2,  
  "Consenti escluso": falso,  
  "Fama": 10  
}
```

- **Oggetto** (stringa)
Il nome dell'oggetto nel gioco, lo stesso che useresti in `#SpawnItem` .
- **Prezzo** (intero, facoltativo)
Il prezzo scontato (o modificato). Omettere per mantenere il prezzo predefinito.
- **Importo** (intero ≥ 1 , facoltativo)
La quantità disponibile dell'articolo scontato.
- **AllowExcluded** (booleano, facoltativo):
consente di acquistare articoli normalmente non venduti dal commerciante se impostato su `true` .
- **Fama** (intero ≥ 0 , facoltativo)
Requisito di fama per acquistare l'oggetto. Omettere di utilizzare il requisito di fama predefinito.

8. Oggetti di condizione

L'array **Condizioni** in un oggetto JSON Quest è costituito da uno o più oggetti **Condizione** . Ogni condizione può essere di tre tipi: **Eliminazione** , **Recupero** o **Interazione** .

8.1 Proprietà delle condizioni comuni

```
{
  "Tipo": "Eliminazione",
  "CanBeAutoCompleted": falso,
  "TrackingCaption": "Elimina 3 pupazzi",
  "IndiceSequenza": 0,
  "Posizioni mostrate sulla mappa": [
    {
      "Posizione": { "X": 0.0, "Y": 0.0, "Z": 0.0 },
      "SizeFactor": 1.0
    }
  ]
}
```

- **Tipo** (stringa)
Deve essere **"Eliminazione"** , **"Recupero"** o **"Interazione"** .
- **CanBeAutoCompleted** (booleano, predefinito: **false**)
Se **false** , il giocatore deve tornare dal PNG che ha assegnato la missione per completare questo passaggio. Se **true** , la missione si completa automaticamente una volta soddisfatti i requisiti.
- **TrackingCaption** (stringa, facoltativa)
Ciò che appare nel diario del giocatore o nel registro delle missioni per descrivere questo passaggio.
- **SequenceIndex** (intero ≥ 0)
Definisce l'ordine in cui le condizioni diventano attive.
 - Almeno una condizione deve avere **SequenceIndex: 0** .
 - Più condizioni possono condividere lo stesso indice e tutte devono essere completate prima che la missione avanzi all'indice successivo.

- **LocationsShownOnMap** (array di oggetti **MapLocation** , facoltativo)
Disegna cerchi sulla mappa per le posizioni di questa condizione.

8.1.1 Oggetti MapLocation

Ogni oggetto in **LocationsShownOnMap** si presenta in questo modo:

```
{ "Posizione": { "X": 1000.0, "Y": 2000.0, "Z": 30.0 },
  "SizeFactor": 1.0 }
```

Anche Ctrl+C nel gioco per copiare la posizione funziona:

```
{ "Posizione": "{X=-157607,328 Y=-687586,562 Z=667,976|P=353,113800
Y=101,191971 R=0,000000}",
  "SizeFactor": 1.0 }
```

- **Posizione**
 - Un oggetto contenente le coordinate **X** , **Y** , **Z** nel mondo di gioco.
- **SizeFactor** (numero > 0,0)
 - La scala del cerchio disegnato (1,0 equivale a circa 300 m di diametro).

8.2 Condizioni di eliminazione

Utilizzato per i requisiti di uccisione:

```
{
  "Tipo": "Eliminazione",
  "Caratteri di destinazione": [
    "Fantoccio",
    "Prigioniero"],
  "Importo": 5,
  "Armi consentite": [
    "BP_Arma_M1911_C",
    "BP_Arma_M9_C"]}]}
```

- **TargetCharacters** (array di stringhe)
Elenca chi o cosa deve essere eliminato. Può includere nomi di spawn usati in **#SpawnZombie** (ad esempio, **Puppet**) o **#SpawnAnimal** . Segnaposto speciali:
 - **"Burattino"** (corrisponde a tutti i burattini)
 - **"Prigioniero"** (corrisponde a tutti i personaggi dei giocatori)
 - **"Rasoio"** , **"Sentinella"** , **"Vecchia sentinella"**
 - **"ArmedNPC"** (corrisponde a tutti i PNG armati)

- **Quantità** (numero ≥ 1)
Numero di uccisioni richieste.
- **AllowedWeapons** (array di stringhe, facoltativo)
Se presente, verranno consegnate **solo** le uccisioni effettuate con queste armi.
Omettere di consentire qualsiasi arma.

8.3 Condizioni di recupero

Utilizzato per ritirare o consegnare articoli:

```
{
  "Tipo": "Recupera",
  "Disabilita l'acquisto degli articoli richiesti": falso,
  "PlayerKeepsItems": falso,
  "ElementiObbligatori": [
    {
      "Articoli accettati": [ "Apple" ],
      "NumeroRichiesto": 3,
      "NumeroObbligatorioAggiuntivoCasuale": 2,
      "MinAcceptedItemUses": 1,
      "MinAcceptedCookLevel": "Crudo",
      "MaxAcceptedCookLevel": "Cotto",
      "MinAcceptedCookQuality": "Scarso",
      "Massa minima dell'articolo accettato": 100,0,
      "MinAcceptedItemHealth": 50.0,
      "Rapporto minimo risorse articolo accettate": 20,0,
      "Importo minimo risorse articolo accettato": 50,0
    }
  ]
}
```

- **DisablePurchaseOfRequiredItems** (booleano):
se **impostato su true** , nessuno può acquistare gli oggetti elencati in **RequiredItems** mentre la missione è attiva. **Rilevante solo** se il server ha **attivato** l'impostazione **BLOCCA ACQUISTO OGGETTI RICHIESTI** .
- **PlayerKeepsItems** (booleano)
Se è **vero** , gli oggetti non vengono rimossi dall'inventario del giocatore al

completamento della missione.

- **RequiredItems** (array di oggetti **Item**)
Ogni oggetto **Item** specifica quali e quanti elementi devono essere raccolti.

8.3.1 Oggetti elemento

Proprietà per ogni elemento richiesto:

- **AcceptedItems** (array di stringhe)
Elenco dei nomi degli elementi riconosciuti da `#SpawnItem` .
- **RequiredNum** (intero ≥ 1)
Quanti di questi elementi devono essere raccolti.
- **RandomAdditionalRequiredNum** (intero ≥ 1 , facoltativo)
Un importo aggiuntivo casuale (fino al numero specificato) che potrebbe essere richiesto.
- **MinAcceptedItemUses** (intero ≥ 0 , facoltativo)
: se applicabile, l'articolo deve avere almeno questo numero di utilizzi.
- **MinAcceptedCookLevel / MaxAcceptedCookLevel** (stringa, facoltativa)
Intervallo di livelli di cottura consentiti per gli alimenti. Livelli validi: `"Crudo"` , `"Poco cotto"` , `"Cotto"` , `"Troppe cotto"` , `"Bruciato"` .
- **MinAcceptedCookQuality** (stringa, facoltativa)
Qualità di cottura più bassa accettabile: `"Rovinato"` , `"Cattivo"` , `"Scarso"` , `"Buono"` , `"Eccellente"` , `"Perfetto"` .
- **MinAcceptedItemMass** (numero $\geq 0,0$, facoltativo)
In grammi, la massa di un articolo deve essere almeno questo valore.
- **MinAcceptedItemHealth** (numero da 0,0 a 100,0, facoltativo)
Percentuale minima richiesta di integrità dell'oggetto.
- **MinAcceptedItemResourceRatio** (numero da 0,0 a 100,0, facoltativo)
Utilizzato per articoli contenenti liquidi (ad esempio, bottiglie d'acqua). Deve contenere almeno questa percentuale di liquido.
- **MinAcceptedItemResourceAmount** (numero $\geq 0,0$, facoltativo)
Utilizzato per articoli contenenti liquidi. Deve avere almeno questo numero di grammi.

8.4 Condizioni di interazione

Utilizzato per interagire con gli oggetti posizionati sulla mappa (ad esempio, azionare interruttori, raccogliere note, ecc.).

Ottenere dati sulla posizione.

Utilizza `#GetMeshInfo` nel gioco mentre guardi un oggetto per catturarne il nome mesh, l'istanza, la trasformazione, ecc. Copia un frammento JSON negli appunti che puoi incollare direttamente nel file della tua missione.

```
{ "Type": "Interaction",  
  
  "Posizioni": [{  
    "AnchorMesh": "/Gioco/Mondo/QualcheMappa/BP_Switch.BP_Switch_C",  
    "Istanza": 4,  
    "FallbackTransform": "X=123,456 Y=234,567 Z=10,0 Beccheggio=0  
Imbardata=0, Rollio=0",  
    "VisibleMesh": "/Gioco/Mondo/UnaMappa/BP_SwitchModel.BP_SwitchModel_C"  
  },  
  {  
    "AnchorMesh": "/Gioco/Mondo/UnaMappa/BP_Door.BP_Door_C"  
  }],  
  "MinNeeded": 1,  
  "Massimo necessario": 2,  
  "SpawnOnlyNeeded": vero,  
  "WorldMarkerShowDistance": 50}
```

- **Posizioni** (serie di oggetti)
Posizione): ciascuna definisce il posizionamento di un oggetto o un punto di interazione nel mondo.
- **MinNeeded** / **MaxNeeded** (interi)
Determina casualmente con quanti oggetti della posizione specificata il giocatore deve interagire. Se `MinNeeded = 1` e `MaxNeeded = 2` , il gioco deciderà se sono necessarie 1 o 2 interazioni.
SpawnOnlyNeeded (booleano)
Se `true` , viene generato solo il numero di oggetti determinato casualmente. Se `false` , vengono generati tutti, ma il giocatore deve interagire solo con un numero sufficiente a soddisfare il numero richiesto.
- **WorldMarkerShowDistance** (intero ≥ 0)
La distanza (in metri) alla quale un indicatore è visibile nel mondo, guidando i giocatori alla posizione esatta dell'oggetto.

8.4.1 Oggetti di posizione

- **AnchorMesh** (stringa)
Identifica l'oggetto di gioco sulla mappa.
- **Istanza** (intero, facoltativo):
utilizzata se più copie della stessa mesh sono in stretta prossimità.
- **FallbackTransform** (stringa)
Dati di posizione/rotazione utilizzati in caso di errore di **AnchorMesh** (ad esempio, la modalità sandbox utilizza sempre **FallbackTransform**).
- **VisibleMesh** (stringa)
Il modello 3D da visualizzare.

Forme base che possono essere utilizzate e posizionate ovunque:

`/Game/ConZ_Files/Models/BasicShapes/Shape_Cube.Shape_Cube`

`/Gioco/File_ConZ/Modelli/Forme_di_base/Forma_cilindro.Forma_cilindro`

`/Gioco/File_ConZ/Modelli/Forme_di_base/Piano_Forma.Piano_Forma`

`/Gioco/File_ConZ/Modelli/Forme_di_base/Sfera_di_forma.Sfera_di_forma`

9. Mettere tutto insieme: esempio di ricerca

Di seguito è riportato un esempio semplice e dettagliato di un file JSON di una missione personalizzata. Salvate questo file come **MyFirstQuest.json** (o qualsiasi altro nome) nella cartella **Override** .


```
{
  "AssociatedNPC": "GeneralGoods",
  "Livello": 1,
  "Titolo": "Specialità del commerciante di beni generali",
  "Descrizione": "Raccogli le mele per il commerciante di beni di largo consumo e ricevi una piccola ricompensa.",
  "OreLimite": 24.0,
  "Pool di premi": [{
    "ValutaNormale": 100,
    "Fama": 5,
    "Competenze": [{
      "Abilità": "Cucinare",
      "Esperienza": 20}],
    "Accordi commerciali": [
      {
        "Oggetto": "Ananas",
        "Prezzo": 50,
        "Importo": 1,
        "Fama": 0 } ] } ],
  "Condizioni": [{
    "TrackingCaption": "Raccogli le mele",
    "IndiceSequenza": 0,
    "CanBeAutoCompleted": falso,
    "Tipo": "Recupera",
    "Disabilita l'acquisto degli articoli richiesti": falso,
    "PlayerKeepsItems": vero,
    "ElementiObbligatori": [{
      "Elementi accettati": [
        "Mela_2"],
      "NumeroRichiesto": 3,
      "MinAcceptedItemHealth": 50.0}],
    "Posizioni mostrate sulla mappa": [{
      "Posizione": {
        "X": 1000,0,
        "Y": 2000.0,
        "Z": 50.0},
      "Fattore di dimensione": 1.0} ]}]}
```

Spiegazione

1. PNG associato

Il barista offrirà la missione e le ricompense della serie **TradeDeals** si applicheranno al negozio del barista.

2. Livello

Il livello 1 indica una missione semplice o di basso livello.

3. RewardPool

Il giocatore riceve 100 monete normali, 5 punti fama, 20 punti esperienza di cucina e può acquistare un M9 a un prezzo scontato una volta completata la missione.

4. Condizioni

- Il tipo è **Fetch** . Il giocatore deve portare 3 mele (**AcceptedItems** : **Apple**) con almeno il 50% di salute.
- Devono tornare dal barista (**CanBeAutoCompleted** : false) per finalizzare la missione.
- **PlayerKeepsItems** è **true** , quindi non perderanno le mele una volta completato.

Dopo aver creato o modificato questo file, **riavvia il server o il gioco** affinché la missione diventi attiva.

10. Suggerimenti e buone pratiche

- **Conserva i backup** dei file JSON nel caso in cui sia necessario annullare le modifiche.
- **Convalida il tuo JSON** utilizzando validatori JSON online o locali per evitare errori di formattazione.
- **Utilizzare #GetMeshInfo** per dati di posizionamento accurati per le condizioni di tipo **Interazione** .
- **testa** prima la tua missione in modalità sandbox, prima di aggiungerla a un ambiente server live.
- **Ricorda** che non puoi modificare le missioni predefinite. Puoi solo bloccarle o crearne di completamente nuove.

Conclusione

Seguendo questa guida, dovresti essere in grado di creare, personalizzare e gestire le tue missioni in SCUM. Che tu voglia bloccare le missioni predefinite, aggiungerne di nuove con condizioni uniche o premiare i giocatori con valuta, fama, sconti o esperienza in abilità, la struttura JSON fornita sopra copre tutti gli aspetti essenziali.

Importante: riavvia sempre il server o il gioco dopo aver apportato modifiche ai file JSON, in modo che le missioni personalizzate vengano caricate correttamente.

Divertiti a creare esperienze SCUM uniche con le tue missioni personalizzate!