

- Zombie
- Armi
- Diversi scenari?
- Armi da mischia? (consumabili?)
- Armi speciali? (mosse speciali?)
- Cuori o healthbar?
- Abilità di movimento? (Sprint, Dash, Double jump, Rampino, Rotolamento)
- Animazioni? ()
- Solo tastiera?
- Tasti personalizzabili?
- Ceste?
- Macchina della fortuna? (a gettoni, monete?)
- Inventario? (minimale? (u see everything u got on screen, no inventory) , full like minecraft?)

NOTE

Ciò che è **EVIDENZIATO** di questo colore, non è stato implementato

GAMEPLAY

L'obiettivo è superare i livelli sfruttando l'ambiente, si possono raccogliere risorse e oggetti, uccidere i nemici è opzionale (ma possono dropare risorse e/o item).

Alla fine di ogni livello si trova un portale, e puoi scegliere un livello da giocare (tra quelli sbloccati) oppure tornare alla base dove puoi lasciare gli oggetti trovati nei container e potenziare/comprare armi, risorse e oggetti

Movimento

Normal movement

Ci si muove con:

- A → sx,
- D → dx,
- Space → salto (non puoi saltare se sei già in aria)
- S? → crouch,

Shooting

Guns

Interazione arma con:

- K? → shoot,
- O? → reload,
- I? → unequip/equip,

Throwables

Interazione lanciabili con:

- P? → throw,

Fighting

Fight

Combatti con:

- K? → colpo a mani nude, o con arma da mischia
- O? → parata
- P? → colpo speciale, calcio rotante?

Interazione oggetti

Pick

Raccolta oggetti con:

- E → raccolta collezionabili,
- E''' → loot,

Suggerimento raccolta quando player vicino oggetto (outline shader)

Interaction

Interazione hotbar:

- J → equipaggia oggetto selezionato
- U → rilascia oggetto selezionato (deve essere equipaggiato)

Enemies

I nemici ti seguono e ti cercheranno una volta visto, anche se ti perdono di vista raggiungeranno l'ultimo punto in cui sei stato visto e cercheranno avanti e indietro per un breve periodo di tempo, per poi entrare in idle se non sono più dentro al loro percorso

Movement

Ogni nemico ha un proprio campo visivo, e [caratteristiche speciali](#)

Attack pattern

Ogni attacco presenta un delay, potrebbero esserci delle combinazioni di colpi da parte dei nemici

Damage

Ogni nemico ha due tipi di attacco:

- Attack damage → danno da attacco
- Body damage → [danno se si tocca il nemico](#)

[Un danno esclude l'altro](#)

Inventory

Reference: [YouTube Series Overview and Demo | Unity Inventory System Tutorial - Part 0](#)

Hai a disposizione la hotbar, [inventario](#) e dei container nella base per contenere i tuoi item.

Store your items

Ogni container nel mondo salva il suo contenuto ad ogni interazione ([all'interno dei livelli no \(da fare\)](#))

Manage your items

Clicca uno slot per muovere gli item:

- Left Mouse btn → Muovi tutti gli item da Mouse slot allo slot cliccato (se stesso item, allora sposta la massima qt possibile, se Mouse slot vuoto prendi da slot cliccato, se item diversi scambia)
- Left-Shift + Left Mouse btn → Muovi un solo item da Mouse slot allo slot cliccato (se item diverso scambia)

Note/Considerazioni

Inventory

Il player può raccogliere e utilizzare oggetti, presenta un suo inventario di base, e può sfruttare contenitori all'interno del mondo per depositare gli oggetti trovati.

Ciò che si trova nella hotbar, nell'inventario e nei container viene salvato.

Problema: Mouse_slot impediva al click di raggiungere i pulsanti degli Slot_UI, risolto tramite il componente CanvasGroup attaccato a Mouse_slot grazie a BlocksRaycast = false, che permette al click di oltrepassare l'object e raggiungerne altri al di sotto.

Problema: All'apertura di una chest inventory_manager si resettava, il problema era lo start() di Slot_UI che andava a fare Clear_slot() a prescindere, risultando in un annullamento del contenuto.

Problema: Creazione e distruzione degli Slot visibili, essendo che i container condividono lo stesso inventory display, devo ogni volta ricreare gli slot perché le ceste potrebbero avere una dimensione diversa, e items diversi al loro interno, il problema consisteva nella distruzione degli slot precedenti che però non venivano distrutti in tempo e GetComponentInChildren continuava a considerarli, ciò risultava in una Null Reference in quanto i vecchi slot non esistevano più risolto con una coroutine che aspetta il prossimo frame, dando tempo agli slot di essere distrutti.

TODO:

Fare in modo che gli oggetti droppati nel mondo siano figli di un empty parent che li gestisce tramite una lista di gameobjects, quando un oggetto viene pickato, viene rimosso dalla lista e dalla hierarchy, salvare direttamente i gameobject non è possibile

PROVARE salvando la posizione e l'item data, così posso recuperare il prefab dall'item data, instanziarlo e posizionarlo nella posizione salvata