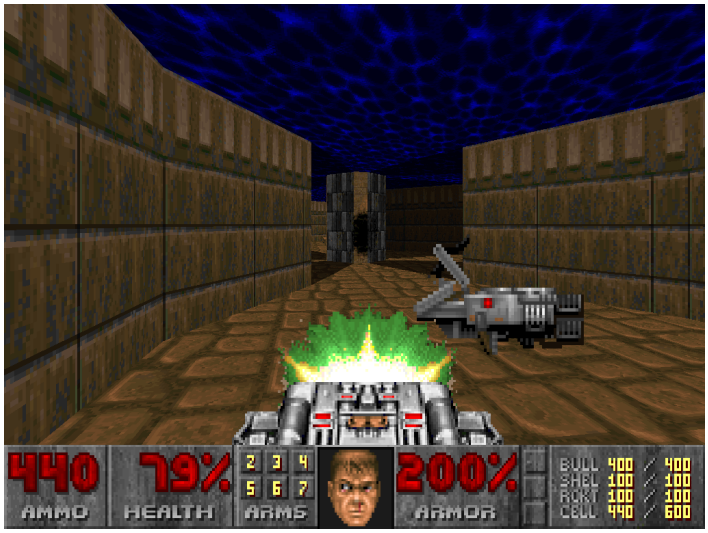
Diana Christian VB

Come creare un arma nuova per DOOM



DOOM, lo sparatutto classico del 1993 non ha soltanto reso popolare il genere degli FPS, ma ha anche cresciuto una generazione di appassionati di programmazione che hanno speso, giorni, mesi o anni per trasformare il loro gioco preferito in qualcosa di completamente nuovo.
Se hai mai avuto la curiosità o la voglia di creare qualcosa di tutto tuo con l'engine di doom sappi che dovrai anche creare delle armi uniche per far sembrare unica la tua mod. Con questa guida passo passo, potrai creare con un po di pazienza e di voglia un sacco di armi diverse!

Prima di iniziare

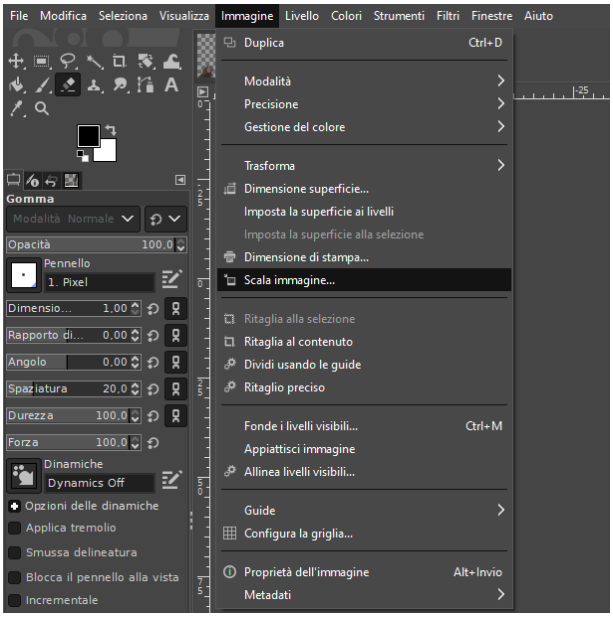
Ti serviranno:

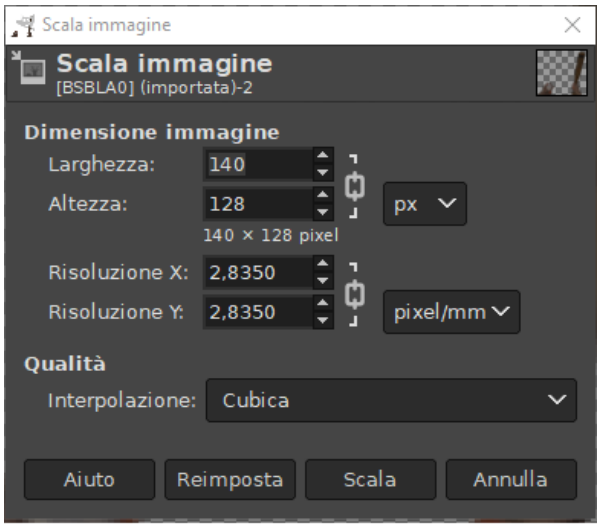
- Un computer (DOOM gira praticamente su ogni macchina)
- Il file .wad della versione di doom che stiamo usando
- Slade 3, GZDoom, GIMP

Passo 1 : Creare gli sprite

Per iniziare, una volta che si è scelto il concept della propria arma dovremo preparare le immagini PNG delle nostre armi, almeno 5 per creare un'animazione fluida ma non troppo complessa da sviluppare.

Per creare gli sprite qua sopra ho usato un programma gratuito chiamato GIMP che mi ha permesso di cancellare lo sfondo dell'immagine originale e di ridimensionare l'immagine perché possa essere utilizzata su Doom (la risoluzione ideale sarebbe 100-200 px di altezza, GIMP scalerà automaticamente la larghezza).

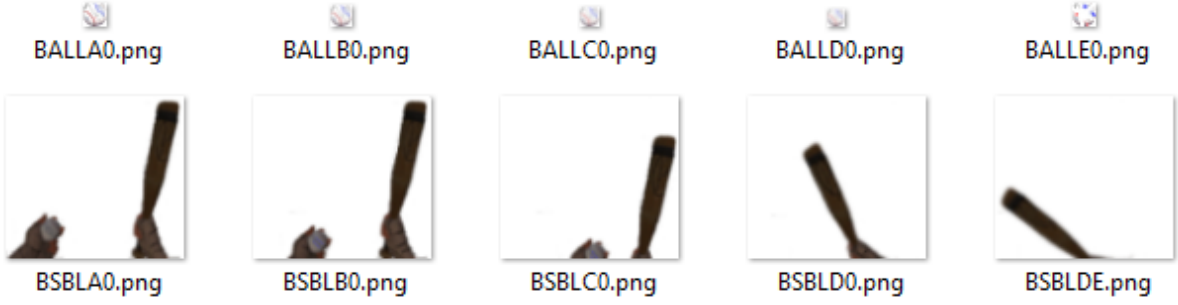




se la nostra arma dovesse avere dei proiettili personalizzati allora bisognerà produrre anche 5 sprite per il nostro proiettile.

I file vanno poi salvati usando una nomenclatura specifica detta "NNNNFA" dove le quattro N rappresentano il nome della nostra texture, la F rappresenta il frame dell'animazione e la A l'angolo.

Per esempio:



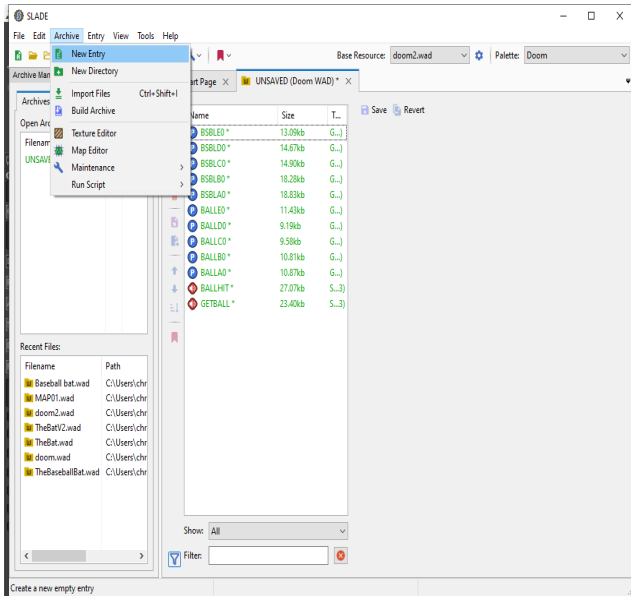
dove BSBL è il nome dello sprite, A,B,C,D,E sono i frame, e l'angolazione per comodità è sempre 0

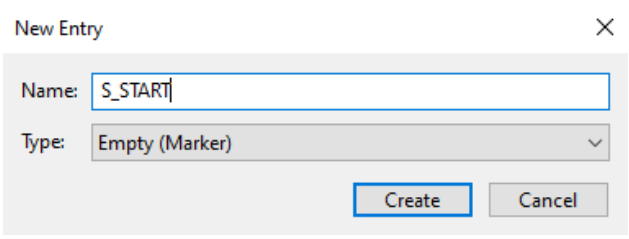
Una volta fatto ciò, prima di iniziare a creare il codice per la nostra arma bisognerà anche trovare un sound effect di massimo 1 secondo per lo sparo e per la raccolta della nostra mod.

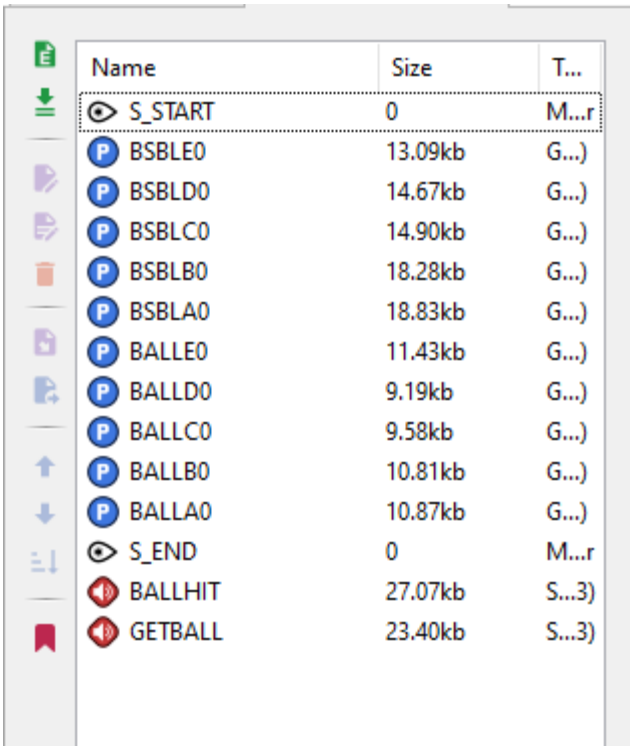
Passo 2 : Creare il codice dell'arma

Ora che le nostre texture sono pronte possiamo iniziare ad utilizzare Slade 3, un programma gratuito per creare e modificare file .wad

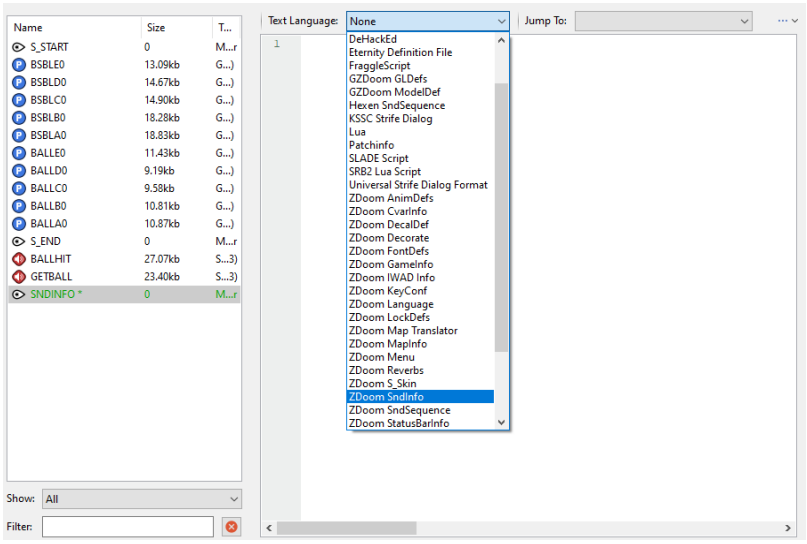
Per iniziare, una volta aperto il programma, creato un nuovo archivio e inserito i nostri file all'interno di esso bisognerà creare due marker chiamati **S_START** e **S_END** e inserire le nostre immagini tra di essi ponendo S_START in cima e S_END in fondo così che DOOM sappia quali sono le texture.

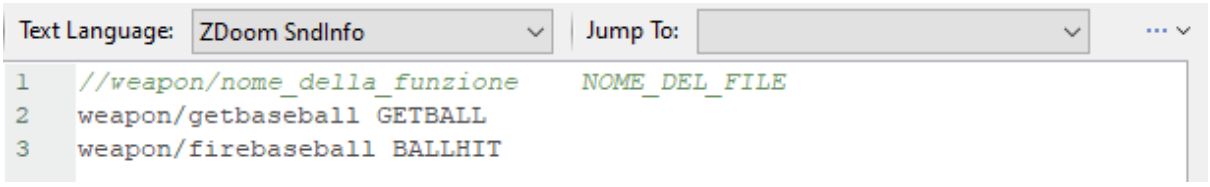




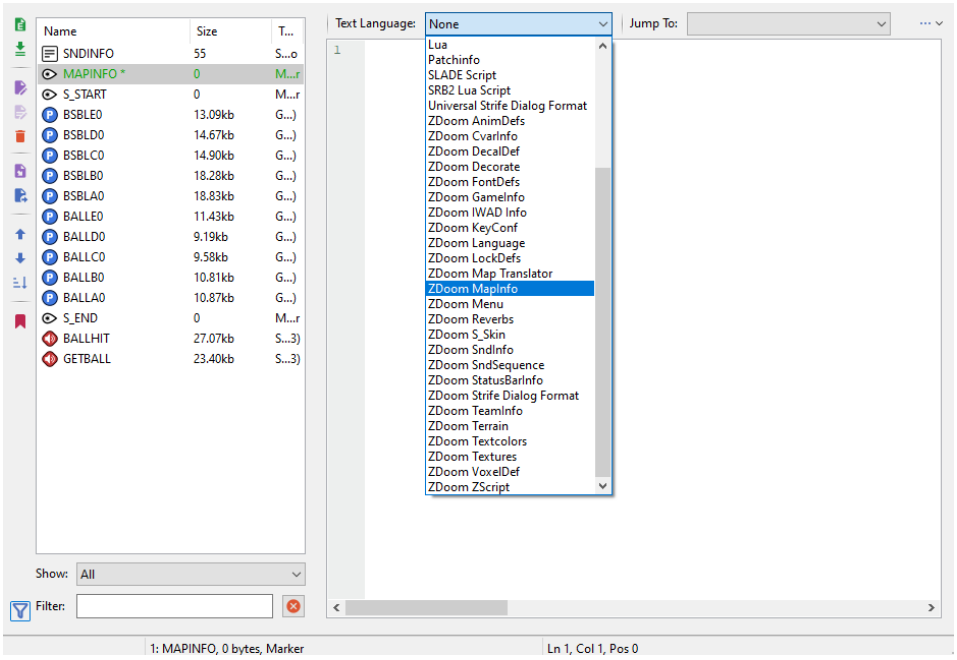


Fatto ciò si inizia a creare un po di codice per la nostra arma, iniziando dalla entry **SNDINFO**. Una volta creata la entry bisognerà cambiare il linguaggio di scrittura in Zdoom SndInfo per poi scrivere il seguente codice:





In seguito, si dovrà ripetere un procedimento simile con la entry **MAPINFO**, il codice però viene utilizzato per creare una nuova classe su DOOM che in seguito sarà equipaggiata con la nostra nuova arma.




```
1  //PlayerClasses = "NOME_DEL_PERSONAGGIO"
2  GameInfo {
3      PlayerClasses = "RAPTOR"
4  }
```

Passo 2 - 2 : Decorate

Ora si passa alla parte più consistente della programmazione e dello sviluppo dell'arma. La creazione della entry DECORATE che conterrà il codice della nostra classe, della nostra arma e del suo proiettile.
Per cominciare basterà creare una nuova entry come fatto in precedenza e usare come linguaggio di scrittura ZDoom Decorate.
Inizieremo dal codice della classe che è:

```
//ACTOR nome_scelto_su_MAPINFO: DoomPlayer
//Player.StartItem serve a decidere l'arma di partenza
//Player.WeaponSlot decide la posizione di un arma nell'inventario

ACTOR RAPTOR: DoomPlayer {
    Player.StartItem "Fist"
    Player.WeaponSlot 1, Fist, Chainsaw
    Player.WeaponSlot 2, //metti la tua arma in uno di questi slot
    Player.WeaponSlot 3, Shotgun, SuperShotgun
    Player.WeaponSlot 4, Chaingun
    Player.WeaponSlot 5, RocketLauncher
    Player.WeaponSlot 6, PlasmaRifle
    Player.WeaponSlot 7, BFG9000

}
```

poi si definisce la propria arma con il nome che abbiamo inserito nello slot (per esempio se hai messo la tua arma nello slot 4 sotto il nome ArmaX , dovrai scrivere *ACTOR ArmaX)* e si codifica usando il metodo che segue per personalizzarla.

```
//ACTOR nome_arma: Weapon Replaces arma_che_si_vuole_sostituire + N°
//del inventario di ZDoom libero

//Weapon.AmmoUse numero di munizioni usate da uno sparo
//Weapon.AmmoGive numero di munizioni ottenute da un caricatore
//Weapon.AmmoType tipo di munizioni utilizzate
//Inventory.PickupSound "nome_inserito_su_SNDINFO"
//Inventory.PickupMessage "Il messaggio che esce a schermo quando trovi l'arma"

ACTOR BaseballBat: Weapon Replaces Pistol 20000 {
   Weapon.SelectionOrder 50
   Weapon.AmmoUse 1
   Weapon.AmmoGive 5
   Weapon.AmmoType "Clip"
   Inventory.PickupSound "weapon/getbaseball"
   Inventory.PickupMessage "You found something to throw"
```

prima di chiudere le parentesi graffe del nostro ACTOR dell'arma bisogna creare le animazioni dei nostri sprite. Per farlo il metodo è molto semplice e diretto grazie alla nomenclatura che abbiamo prima utilizzato (NNNNFA).

```
States {
      Spawn:
      BALL A -1
      Stop

      Select:
      BSBL A 1 A_Raise
      Loop

      Deselect:
      BSBL A 1 A_Lower
      Loop

      Ready:
      BSBL A 1 A_WeaponReady
      Loop

      Fire:
      BSBL BCD 4
      Goto strike

      strike:
      BSBL E 0 A_PlaySound("weapon/firebaseball",CHAN_WEAPON, 0.8)
      BSBL E 0 A_Recoil(10)
      BSBL E 3 A_FireCustomMissile("Ball",0,1,0,-8)
      BSBL C 20
      BSBL D 0 A_Refire("Ball")
      Goto Ready
      }
}
```

per spiegare meglio cosa sta succedendo in States:

- Spawn : è la funzione con cui l'arma appare in giro per la mappa
- Select : è la funzione con cui la nostra arma appare quando la selezioniamo dall'inventario, solitamente si usa il primo sprite della nostra arma con la funzione A_Raise che serve a far apparire l'arma dal basso in maniera fluida
- Deselect : è la funzione opposta a select e usa la funzione opposta A_Lower che fa sparire l'arma dal centro dello schermo verso il basso
- Ready : è lo stato in cui entra l'arma quando è uscita dall'inventario ed è pronta a fare fuoco.
- Fire : è lo stato che si avvia nel momento in cui il giocatore fa fuoco con l'arma pronta, per l'esempio ho usato NNNN FFF A, dove le tre lettere di fila servono a dire che l'animazione usa il frame B, C e D alla velocità di 4 frame tra uno e l'altro.

strike è invece invece il nome che ho dato alla mia funzione, serve a far apparire il proiettile dell'arma che in questo caso ho chiamato "Ball" e serve a modificare alcune statistiche dell'arma come il rinculo deciso dalla funzione A_Recoil(x).
Se cambiamo il numero del frame prima di A_Refire("nome_proiettile") possiamo anche regolare il tempo trascorso tra un proiettile e l'altro.

Infine dobbiamo dichiarare il nome e la funzione del nostro proiettile personalizzato.

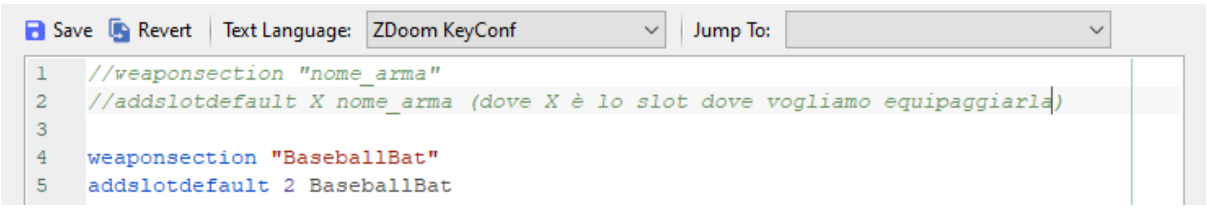
```
ACTOR Ball : Actor 20001 {
    +RANDOMIZE

    Decal "Scorch"
    Projectile

    Radius 7
    Height 7
    Speed 25
    Damage 20

    DeathSound "weapon/plasmax"

    States {
        Spawn:
        BALL AB 4 Bright
        Loop

        Death:
        BALL CDE 4 Bright
        Stop
        }
}
```

In breve, la funzione Radius e Height ci permette di dare una forma e dimensione al nostro proiettile. Speed serve a regolare la sua velocità e Damage il suo danno.
Con DeathSound si utilizza una serie predefinita di suoni o dei suoni aggiuntivi personalizzati che servono a decidere il suono che il proiettile emetterà una volta che colpisce un muro o un nemico.

Gli States sottostanti invece servono a far apparire il proiettile e a dargli un animazione personalizzata.

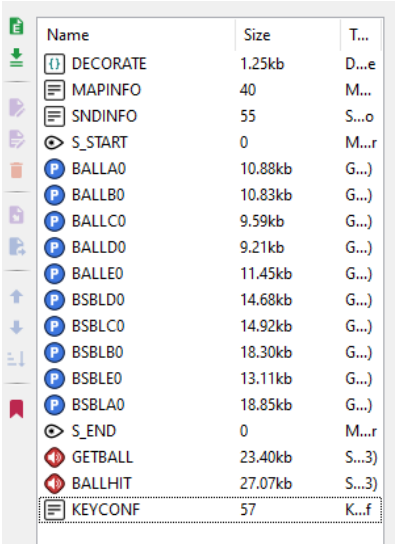
Passo extra : Keyconf

Arrivati a questo punto rimane un solo piccolo passaggio aggiuntivo che renderà leggermente meglio la nostra mod, creando un'altra entry chiamata KEYCONF possiamo settare la nostra arma ad uno slot permettendoci, premendo il tasto desiderato di farla tornare nelle nostre mani in ogni momento.

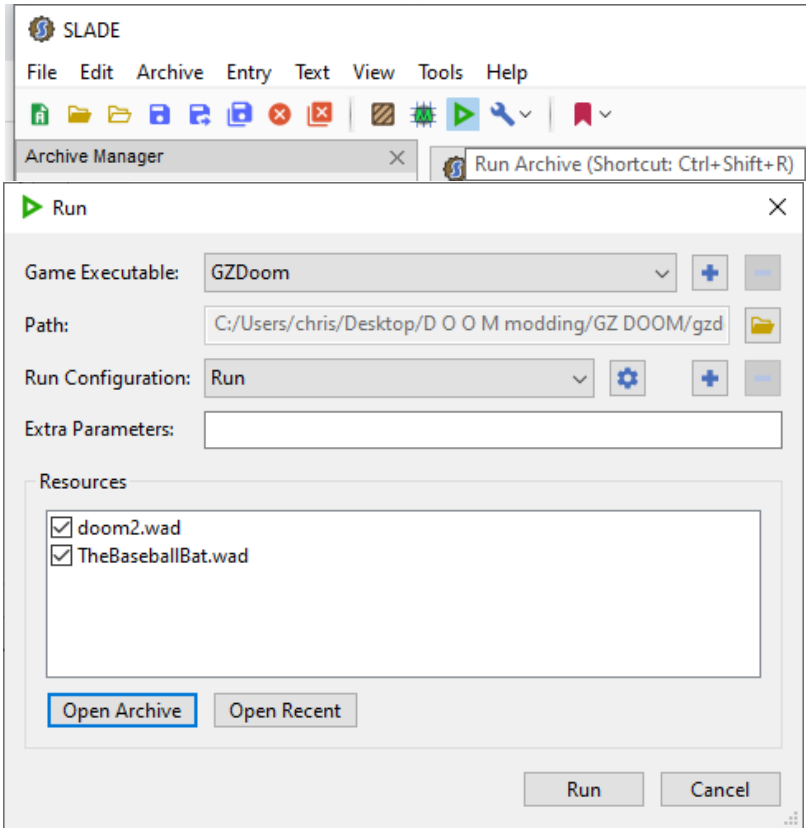


Passo 3 : Testare la mod

Ora il nostro archivio dovrebbe avere questo aspetto:



Quindi quello che ci rimane da fare è trasformare tutto in un file wad e testare la nostra mod su una versione completa del gioco. Una volta salvato il file seguite i passi qui sotto per verificare che tutto sia andato a buon fine:



aprite l'archivio delle risorse ed inserite una versione di DOOM e la vostra mod dopo di che premete RUN e GZDoom farà il resto.

Per testare la propria arma senza mappe personalizzate basta usare la console dei comandi per scrivere “summon nome_arma” e vedersela apparire davanti. Nel caso in cui il comando summon non funzioni o dica unknown actor “nome_arma” allora è probabile che ci sia un errore nella scrittura del file decorate.

Se la tua mod ha invece funzionato, congratulazioni, hai appena creato la tua prima arma su Doom!