**Progetto di Basi di Dati**

**“Ingress for dummies”**

**Bellemo Lorenzo (Leggend)**

**Matricola 1073623**

**Solani Tommaso (tssp)**

**Matricola 544835**

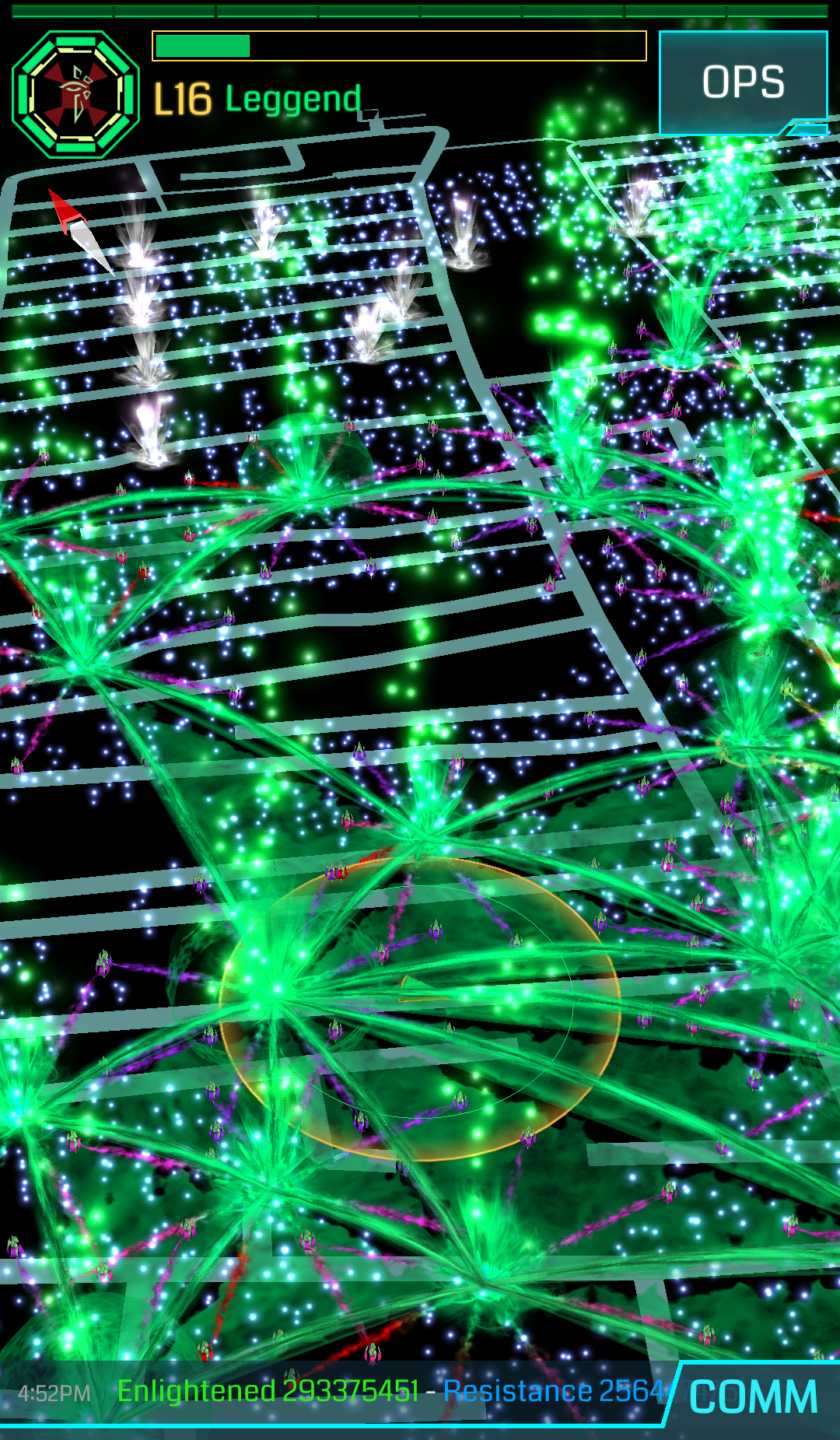
**23 giugno 2015**

**Abstract**

Ingress è un videogioco di realtà aumentata creato da NianticLabs, sviluppato per le piattaforme Android e iOS.

La dinamica del gioco vede due fazioni in continua competizione sulla base di ideali diversi: Resistance (Resistenza, rappresentati in blu) e Enlightened (Illuminati, rappresentati in verde). Lo scopo del gioco è aumentare il numero di Mind Unit (unità mentali) della propria fazione creando Link (collegamenti tra portali) e Control field (campi di controllo virtuali sulle aree geografiche), tramite le apposite Portal Key (chiavi dei portali), visibili attraverso l'interfaccia del gioco.

I giocatori sono "agenti" di una delle due fazioni e utilizzano uno scanner, sul proprio dispositivo mobile. La mappa dello scanner riproduce fedelmente le strade e gli edifici, senza però specificarne il nome, in modo da evidenziare gli elementi utili per il gioco: i portali (Portal), la materia esotica (Exotic Matter), i link (collegamenti tra due portali) e i control field (campi di controllo, che si generano collegando tra loro tre portali a formare un triangolo).

I portali sono punti visibili sulla mappa del gioco e sono associati a punti di riferimento reali come sculture, biblioteche, edifici pubblici e luoghi di interesse storico o artistico. I portali appaiono di colore blu quando vengono conquistati dalla Resistenza, verde se appartengono agli Illuminati e grigio se sono neutri, cioè non sono ancora stati conquistati da una delle due fazioni o neutralizzati. Ogni portale può essere equipaggiato con un massimo di otto risonatori (Resonator) e quattro Mod. Un portale neutro è privo risonatori e, per conquistarlo per la propria fazione, il giocatore deve piazzarvi un suo risonatore.

Un giocatore può creare un collegamento (link) con un altro portale, se il giocatore è in possesso della chiave del portale di destinazione. Per creare il link è necessaria solo la chiave del portale di destinazione: creando il link la chiave viene però consumata e scompare dall'inventario.

Quando tre portali sono collegati in un triangolo, si crea un Campo di Controllo (Control Field).

L'inventario di un giocatore può includere risonatori, XMP Burster, Virus, Ultra Strike, Power cubes, Mods e chiavi di portali. Questi si ottengono dai portali tramite “hacking”. Un giocatore può effettuare l'hack di un portale che appartiene a entrambe le fazioni.

I giocatori iniziano al livello 1 e passano di livello accumulando Action Point (AP); il livello più alto è L16. I giocatori ricevono AP effettuando hack di portali avversari, distruggendo risonatori, schierando risonatori (con punti aggiuntivi se si conquista un portale col primo risonatore), creando collegamenti e creando campi di controllo.

**Descrizione dei requisiti**

Si vuole realizzare una base di dati per la gestione di un videogioco di realtà aumentata (Ingress) che gestirà gli aspetti riguardanti le attività e le statistiche di gioco di ogni giocatore.

Per partecipare al gioco è necessario iscriversi utilizzando una e-mail e un nickname unico (contenente lettere e numeri), inoltre bisogna scegliere una fazione di appartenenza fra le due disponibili: Resistance ed Enlightened.

Ogni giocatore possiede un inventario che può contenere un massimo di 2000 oggetti, è possibile superare questa soglia ma fatto ciò, per poterne acquisire altri, sarà necessario scendere nuovamente sotto la quota.

Un giocatore ha un inventario nel quale può depositare i seguenti oggetti:

* weapon
* resonator
* portal mod
* powercube
* portal key

Weapon contiene Xmp, Ultrastrike, Ada Refactor, Jarvis Virus.

Portal Mod contiene, Common Shield, Rare shield, Very Rare Shield, AXA shield, Common Multi Hack, Rare Multi Hack, Very Rare Multi Hack, Common Heat Sink, Rare Heat Sink, Very Rare Heat Sink, Turret, Link Amp, Force Amp.

Un giocatore può usare oggetti dello suo stesso livello o inferiore per quelli che dispongono di un livello Gli oggetti che hanno un livello sono xmp e ultrastrike, resonator e power cube ed è compreso tra 1 e 8

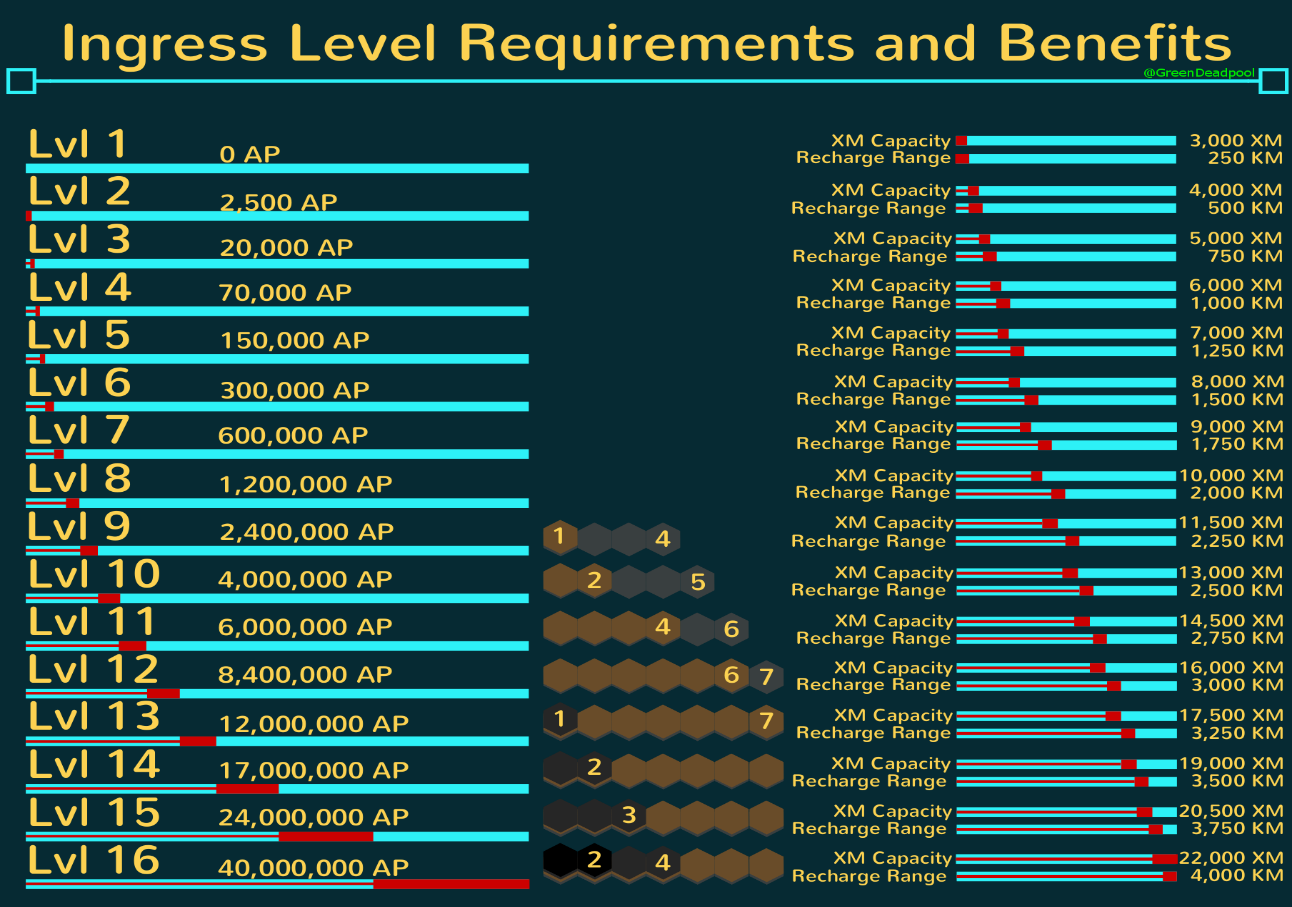
Mentre gli altri oggetti i quali non dispongono di livello e possono essere utilizzati indipendentemente dal livello del giocatore

Per acquisire oggetti è necessario effettuare hack.

C'è un limite a quanto frequentemente un giocatore può eseguire l'hack di un portale: si può fare l'hack una volta ogni 5 minuti per un massimo di 4 volte consecutive, dopodiché il portale va in surriscaldamento (*burn out*) e si deve aspettare 4 ore prima che si raffreddi.

Un nuovo giocatore avrà 0 AP, ovvero i punti esperienza, e livello 1. Per determinate azioni si viene ricompensati con AP che vanno ad aumentare il suo livello fino all’8, da qui in poi per aumentare di livello fino ad un massimo di 16 concorrono sia gli AP che medal. è possibile continuare ad accumulare AP e medal anche dopo aver raggiunto il livello massimo.

Compiendo azioni di gioco, un giocatore modifica le proprie statistiche che sono:

* Ap guadagnati
* Livello
* Giocatori reclutati
* Hack di un portale entro 24 ore
* Punti guadagnati tramite glifo
* Numero di hack effettuati
* Numero di missioni uniche completate
* Numero di giorni consecutivi di possesso di un portale
* Numero di risonatori nemici distrutti
* Numero di risonatori posizionati
* Numero di mod posizionate
* Numero di link creati
* Numero di field creati
* Numero di portali unici catturati
* Numero di portali unici visitati
* Quantità di XM ricaricata
* MU collezionati
* Km percorsi in cammino

Quando una statistica raggiunge un determinato valore sbloccherà della medaglia inerente di livello superiore.

Le medaglie hanno i seguenti livelli elencati da quello meno significativo al più significativo: bronzo, argento, oro, platino, onice.

Ogni portale può essere collegato ad altri tramite link e può supportarne un massimo di 8. I link devono avere molteplicità pari a 1.

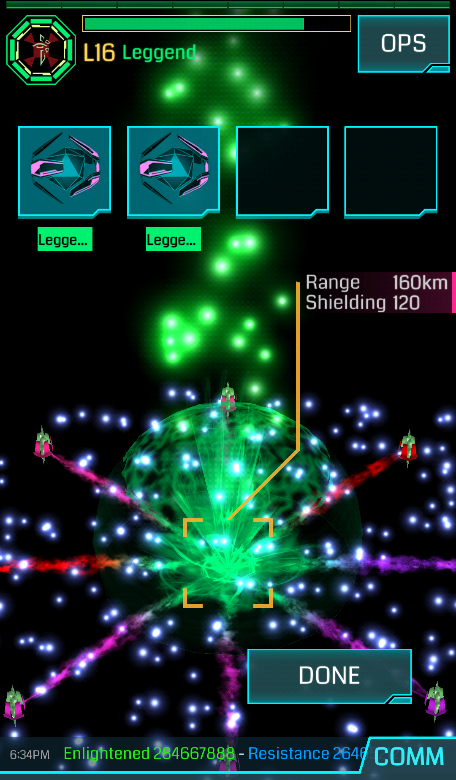
Per creare un link è necessario che che il portale abbia 8 risonatori inseriti, ovvero sia completo e sia della stessa fazione.

Un link appartiene al giocatore che lo ha creato e di conseguenza alla sua fazione, al momento della creazione vengono registrate data ora e cella di appartenenza. Un link può generare al massimo un field. Ogni cella ha un codice identificativo.

Un portale può essere abbattuto (portato allo stato neutrale) utilizzando delle armi su di esso.

Utilizzando Xmp e ultra strike è possibile diminuirne l’energia, quando questa raggiunge un valore pari o inferiore a 0 allora il portale perde tutti i suoi risonatori e le mod e diventa nuovamente catturabile.

Ogni portale dispone di una coppia di valori (coordinate Gps) che lo identifica univocamente, un nome, un possessore, una data di cattura, la fazione di appartenenza, numero di link totali,un valore energetico (calcolata a in base al livello del portale), una valore energetico dato dalle mod installate, e un livello compreso tra 1 e 8 e calcolato dalla media del livello dei suoi risonatori.

Un portale può contenere un massimo di 8 slot e in ognuno di essi può essere posizionato un risonatore, vi sarà inoltre associato il nome del giocatore che lo ha posizionato.

Inoltre sono presenti 4 slot sui quali è possibile

**Progettazione concettuale**

***Player***

La tabella player raccoglie tutte le informazioni basilari di un giocatore

* nickname (char): Indica lo pseudonimo utilizzato dal giocatore al momento dell’iscrizione e lo identifica univocamente nel mondo di gioco
* email(char): indica la email utilizzata da un giocatore per la registrazione, ad una email viene associato un solo nickname
* faction(char):indica la fazione scelta all’iscrizione, r=Resistance ed e=Enlighened
* level(tinyint): indica l’attuale livello del giocatore
* gpslat, gpslon (double): indicano le coordinate Gps (latitudine e longitudine) del giocatore nella mappa, viene utilizzato un tipo double per evitare errori di assegnazione nell’ultima cifra decimale

Primary key: nickname, email

ad ogni insert in questa tabella viene attivato il trigger newplayer

l’attributo level viene modificato dal trigger checklevel

***Inventory***

La tabella inventory associa la quantità totale di item posseduta da un giocatore

* nickname (char): Indica lo pseudonimo utilizzato dal giocatore al momento dell’iscrizione e lo identifica univocamente nel mondo di gioco
* itemnumber(integer): è la quantità totale di item

Primary key: nickname

itemnumber viene aggiornato dai trigger checkinventoryweapon, checkinventoryresonator, checkinventorypc, checkinventorypk2, checkinventorypk, checkinventorypm

***Weapon***

La tabella weapon raccoglie tutti gli oggetti di attacco che servono a danneggiare un portale nemico

* nickname (char): Indica lo pseudonimo utilizzato dal giocatore al momento dell’iscrizione e lo identifica univocamente nel mondo di gioco
* xmpl1, …, xmpl8(integer): attributi che contano il numero di xmp nell’inventario, è presente un attributo per livello
* usl1, …, usl8(integer): attributi che contano il numero di ultra strike nell’inventario, è presente un attributo per livello
* adarefractor(integer): indica quanti ada refractor possiede un giocatore
* jarvisvirus(integer): indica quanti jarvis virus possiede un giocatore

Primary key: nickname

ad ogni update di questa tabella viene attivato il trigger checkinventoryweapon

***Resonator***

La tabella resonator raccoglie tutti gli oggetti atti a catturare un portale neutro o potenziare il livello di un portale alleato

* nickname (char): Indica lo pseudonimo utilizzato dal giocatore al momento dell’iscrizione e lo identifica univocamente nel mondo di gioco
* rl1, …, rl8(integer): attributi che contano il numero di resonator nell’inventario, è presente un attributo per livello

Primary key: nickname

ad ogni update di questa tabella viene attivato il trigger checkinventoryresonator

***Portalmod***

La tabella resonator raccoglie tutti gli oggetti, che andranno inseriti nelle slot mod, utili a potenziare un portale alleato

* nickname (char): Indica lo pseudonimo utilizzato dal giocatore al momento dell’iscrizione e lo identifica univocamente nel mondo di gioco
* shieldcommon(integer): indica la quantità di shield common
* shieldrare(integer): indica la quantità di shield rare
* shieldveryrare(integer): indica la quantità di shield very rare
* axashield(integer): indica la quantità di shield axa
* multihackcommon(integer): indica la quantità di multi hack common
* multihackrare(integer): indica la quantità di multi hack rare
* multihackveryrare(integer): indica la quantità di multi hack very rare
* heatsinkcommon(integer): indica la quantità di heat sink common
* heatsinkrare(integer): indica la quantità di heat sink rare
* heatsinkveryrare(integer): indica la quantità di heat sink very rare
* turret(integer): indica la quantità di turret
* linkamp(integer): indica la quantità di link amp
* forceamp(integer): indica la quantità di force amp

Primary key: nickname

ad ogni update di questa tabella viene attivato il trigger checkinventorypm

***Powercube***

La tabella powercube raccoglie tutti gli oggetti atti a ripristinare l’energia di un giocatore

* nickname (char): Indica lo pseudonimo utilizzato dal giocatore al momento dell’iscrizione e lo identifica univocamente nel mondo di gioco
* pcl1, …, pcl8(integer): attributi che contano il numero di power cubenell’inventario, è presente un attributo per livello

Primary key: nickname

ad ogni update di questa tabella viene attivato il trigger checkinventorypc

***Portalkey***

La tabella powercube raccoglie tutte le chiavi dei portali in possesso a ogni giocatore

* nickname (char): Indica lo pseudonimo utilizzato dal giocatore al momento dell’iscrizione e lo identifica univocamente nel mondo di gioco
* gpslat, gpslon (double): indicano le coordinate Gps (latitudine e longitudine) del giocatore nella mappa, viene utilizzato un tipo double per evitare errori di assegnazione nell’ultima cifra decimale
* amount(integer): indica la quantità di chiavi in possesso riferita al singolo portale

Primary key: nickname

ad ogni update di questa tabella viene attivato i trigger checkinventorypk2 e ad ogni insert il trigger checkinventorypk

***Stats***

La tabella stats raccoglie tutte le informazioni inerenti alle statistiche di gioco dei giocatori

* nickname (char): Indica lo pseudonimo utilizzato dal giocatore al momento dell’iscrizione e lo identifica univocamente nel mondo di gioco
* ap(integer): Indica gli AP del giocatore
* explorer(integer): Indica quanti portali unici ha visitato il giocatore
* trekker(integer): Indica quanti Kilometri ha percorso il giocatore
* builder(integer): Indica quanti risonatori sono stati posizionati il giocatore
* connector(integer): Indica quanti link ha creato il giocatore
* mindcontroller(integer): Indica quanti field ha creato il giocatore
* illuminator(integer): Indica quante mind units ha catturato il giocatore
* recharger(integer): Indica quanta Xm ha caricato il giocatore
* liberator(integer): Indica quantportali ha catturato il giocatore
* pioner(integer): Indica quanti portali unici ha catturato il giocatore
* engineer(integer): Indica quante mod ha posizionato il giocatore
* purifier(integer): Indica quanti risonatori ha distrutto il giocatore
* guardian(integer): Indica il tempo massimo che il giocatore è riuscito a mantenere un portale
* specops(integer): Indica il numero di missioni completate dal giocatore
* hacker(integer): Indica gli hack effettuati da giocatore
* translator(integer): Indica il punteggio ottenuto dagli hack con glifo
* sojourner(integer): Indica il numero di hack consecutivi ogni 24 ore
* recruiter(integer): Indica il numero di giocatori invitati

Primary key: nickname

ad ogni update di questa tabella viene attivato i trigger checkmedal

***Medal***

La tabella stats raccoglie tutte le informazioni riguardo alle medaglie inerenti alle relative statistiche dei giocatori

* explorer(integer): indica la medaglia relativa a quanti portali unici ha visitato il giocatore
* trekker(integer): indica la medaglia relativa a quanti Kilometri ha percorso il giocatore
* builder(integer): indica la medaglia relativa a quanti risonatori sono stati posizionati il giocatore
* connector(integer): indica la medaglia relativa a quanti link ha creato il giocatore
* mindcontroller(integer): indica la medaglia relativa a quanti field ha creato il giocatore
* illuminator(integer): indica la medaglia relativa a quante mind units ha catturato il giocatore
* recharger(integer): indica la medaglia relativa a quanta Xm ha caricato il giocatore
* liberator(integer): indica la medaglia relativa a quantportali ha catturato il giocatore
* pioner(integer): indica la medaglia relativa a quanti portali unici ha catturato il giocatore
* engineer(integer): indica la medaglia relativa a quante mod ha posizionato il giocatore
* purifier(integer): indica la medaglia relativa a quanti risonatori ha distrutto il giocatore
* guardian(integer): indica la medaglia relativa al tempo massimo che il giocatore è riuscito a mantenere un portale
* specops(integer): indica la medaglia relativa al numero di missioni completate dal giocatore
* hacker(integer): indica la medaglia relativa agli hack effettuati da giocatore
* translator(integer): indica la medaglia relativa al punteggio ottenuto dagli hack con glifo
* sojourner(integer): indica la medaglia relativa al numero di hack consecutivi ogni 24 ore
* recruiter(integer): indica la medaglia relativa al numero di giocatori invitati

Primary key: nickname

i valori di questa tabella vengono regolati dal trigger checkmedal

***Portal***

La tabella portal raccoglie tutti i portali presenti nel mondo di gioco e le loro caratteristiche principali

* gpslat, gpslon (double): indicano le coordinate Gps (latitudine e longitudine) del portale nella mappa, viene utilizzato un tipo double per evitare errori di assegnazione nell’ultima cifra decimale
* name(char): indica il nome attribuito al portale
* owner(char): indica chi ha catturato il portale, se questo è neutro allora è settato a NULL
* cell(char): indica la cella di appartenenza del portale
* capturdate(timestamp): Indica la data di cattura o di distruzione del portale
* linknumber(integer): indica il numero di link in entrata o in uscita dal portale
* faction(char): indica la fazione che possiede il portale, r=Resistance ed e=Enlighened, se il portale è neutro allora il suo valore sarà n
* energy(integer): indica il livello di energia di un portale
* modenergy(integer): indica il livello di energia totale delle mod

Primary key: gpslat, gpslon

L’attributo level viene modificato dal trigger checkportallevel

L’attributo energy viene modificato dal trigger checkenergy

L’attributo linknumber viene modificato dal trigger linknumberinsert

Ad ogni insert viene attivato il trigger newportal

***Resonator level***

La tabella resonatorlevel raccoglie le informazioni sul livello dei risonatori di ogni portale

* gpslat, gpslon (double): indicano le coordinate Gps (latitudine e longitudine) del portale nella mappa, viene utilizzato un tipo double per evitare errori di assegnazione nell’ultima cifra decimale
* resonator1, …, resonator8(tinyint): viene salvato il livello del risonatore che ha posizionato in un determinato slot, se non è presente nessun risonatore il valore viene settato a 0

Primary key: gpslat, gpslon

ad ogni update di questa tabella viene attivato i trigger checkportallevel

***Resonator owner***

La tabella resonatorowner raccoglie il nickname di chi ha posizionato un risonatore su un certo slot

* gpslat, gpslon (double): indicano le coordinate Gps (latitudine e longitudine) del portale nella mappa, viene utilizzato un tipo double per evitare errori di assegnazione nell’ultima cifra decimale
* resonator1, …, resonator8(tinyint): viene salvato il nick del giocatore che ha posizionato un risonatore nel relativo slot, se non è presente un risonatore il valore viene settato a NULL

Primary key: gpslat, gpslon

***Mod status***

La tabella modstatus contiene quali mod sono posizionate negli slot di un portale

* gpslat, gpslon (double): indicano le coordinate Gps (latitudine e longitudine) del portale nella mappa, viene utilizzato un tipo double per evitare errori di assegnazione nell’ultima cifra decimale
* slot1, …, slot4(char): viene salvata la mod posizionata nello slot relativo

Primary key: gpslat, gpslon

***Link***

La tabella link raccoglie le informazioni dei link fra i portali

* gpslat1, gpslon1 (double): indicano le coordinate Gps (latitudine e longitudine) del portale da cui è stato effettuato il link (portale di origine)
* gpslat2, gpslon2 (double): indicano le coordinate Gps (latitudine e longitudine) del portale verso cui è stato effettuato il link (portale di destinazione)
* owner(char): indica chi ha catturato creato il link
* cell(char): indica la cella di appartenenza del link basata sulla posizione del portale di origine
* capturdate(timestamp): Indica la data di creazione del link
* faction(char): indica la fazione che possiede il link, r=Resistance ed e=Enlighened

Primary key: gpslat1, gpslon1, gpslat2, gpslon2

Ad ogni insert viene attivato il trigger linknumberinsert

***Field***

La tabella field raccoglie le informazioni dei field generati da link

* gpslat1, gpslon1, gpslat2, gpslon2, gpslat3, gpslon3(double): indicano le coordinate Gps (latitudine e longitudine) dei vertici del field
* cell(char): indica la cella di appartenenza del field basata sulla posizione del portale di origine del link che ha generato il field
* capturdate(timestamp): Indica la data di creazione del field
* faction(char): indica la fazione che possiede il field, r=Resistance ed e=Enlighened

Primary key: gpslat1, gpslon1, gpslat2, gpslon2, gpslat3

***Cell***

La tabella cell raccoglie tutte le celle presenti nel mondo di gioco e i loro punteggi

* name(char):indica il codice identificativo della cella
* totalmuresistance(integer): indica il punteggio della fazione resistente
* totalmuenlightened(integer):indica il punteggio della fazione illuminata

Primary key: name

***Hack***

La tabella hack raccoglie le informazioni per la gestione delle attività di hacking

* nickname (char): Indica il giocatore che ha effettuato hack
* gpslat, gpslon (double): indicano le coordinate Gps (latitudine e longitudine) del portale che è stato sottoposto ad hack
* firsthack(timestamp): indica l’ora in cui viene effettuato il primo hack
* previoushack(timestamp):indica l’ora in cui viene effettuato l’hack precedente
* hacknumber(int):indica il numero di hack effettuati sul portale

Primary key: nickname, gpslat, gpslon

**Descrizione delle relazioni:**

Le due entità fondamentali sono Player e Portal. Attorno ad esse si sviluppa gran parte del gioco.

***[Player] - <Capture> - [Portal]***

Un giocatore può catturare da 0 ad un numero indefinito di portali mentre un portale può essere neutro o di proprietà di al massimo un giocatore.

***[Player] - <Store> - [Inventory]***

Un giocatore possiede un solo inventario e un inventario appartiene ad un solo giocatore.

***[Player] - <gain> - [Stats]***

Un giocatore migliora un gruppo di statistiche e un gruppo di statistiche appartiene ad un solo giocatore.

***[Player] - <ulock> - [Medal]***

Un giocatore sblocca un gruppo di medaglie e un gruppo di medaglie appartiene ad un solo giocatore.

***[Player] - <unlock> - [Medal]***

Un giocatore può effettuare da 0 n hack mentre un hack può essere effettuato da almeno un giocatore fino ad un numero indefinito.

***[Portal] - <create> - [Link]***

Un portale può essere collegato da 0 a un massimo di 8 portali, quindi può generare al massimo 8 link (cardinalità da 1 a n). Un link viene generato da una e una sola coppia di portali.

***[Link] - <create> - [Field]***

Un link può avere da 0 a n occorrenze nella tabella field, mentre un field ha solo una serie di enuple nella tabella link.

***[Field] - <concur> - [Cell]***

Un field contribuisce al punteggio di una o più celle, mentre il punteggio di una cella viene modificato da 0 a più field

***[Portal] - <has> - [Resonatorlevel] [Portal] - <has> - [Resonatorowner]***

Un portale possiede un solo gruppo di 8 slot per i risonatori, e un gruppo di 8 slot per i risonatori appartiene ad un solo portale.

***[Portal] - <has> - [modstatus]***

Un portale possiede un solo gruppo di 4 slot per le mod, e un gruppo di 8 slot per le mod appartiene ad un solo portale.

**Descrizione generalizzazione**

Lo schema concettuale prevede un’unica generalizzazione dotata di un genitore chiamato Inventory e 5 figli: Weapon, Resonator, Portal Mod, Power Cube, Portal Key.

Weapon è a sua volta genitore e ha 2 figli XMP, Ultrastrike

Portal Mod è a sua volta genitore e ha 3 figli: Defense, Farm, Attak

Sia Weapon che Portal mod, nell’implementazione dello schema logico, inglobano i propri figli.

Tutti i figli ereditano l’attributo nickname dal genitore Inventory.

**Descrizione funzioni, view, procedure, trigger**

***function checkfullresonator()***

Controlla che il portale preso in considerazione abbia 8 risonatori posizionati.

Se è vero ritorna un valore true, altrimenti ritorna un valore false.

***function checklinknumber()***

Controlla che il portale preso in considerazione abbia meno di 8 link.

Se è vero ritorna un valore true, altrimenti ritorna un valore false.

***function checkexistinglink()***

Controlla che il link che si vuole creare non sia già presente in tabella.

Se è vero ritorna un valore true, altrimenti ritorna un valore false.

***function createlink()***

Controlla che il giocatore che cerca di creare il link sia della fazione corretta (in caso contrario restituisce false) e se è vero Inserisce il link nella tabella apposita restituendo un valore true.

***view startportal e view endportal***

Queste view vengono generate attraverso un comando da applicativo. Startportal riguarda il portale di origine del link che si è appena creato mentre endportal riguarda il portale di destinazione.

Entrambe generano una tabella che elenca tutti i portali al quale è collegato il portale interessato.

***funciton createfield()***

Utilizzando le view precedentemente create (attraverso un join) controlla quanti portali sono in comune fra portale di origine e portale di destinazione. Se viene trovata corrispondenza verrà selezionato il primo portale in comune nella lista per generare il field.

Il field sarà inserito nella tabella in questo ordine: portale di origine, portale di destinazione, portale comune.

Se viene generato un field viene ritornato un valore true, in caso contrario un valore false.

***function deploy()***

la funzione permette di effettuare un controllo sullo stato del portale, se è neutro verrà effettuata la cattura, mentre se è già catturato dalla fazione del giocatore allora esso potrà aggiungere ulteriori risonatori o potenziarli con alcuni di livello più alto ove possibile.

***function insertmod()***

la funzione permette di effettuare un controllo sullo stato del portale, se è neutro verrà restituito un valore false, mentre se è già catturato dalla fazione del giocatore allora esso potrà aggiungere delle mod.

***function checkhack()***

Permette di effettuare un hack e imposta un’attesa di 5 minuiti per il prossimo fino ad un massimo di 4 giri. Dopo il quarto l’attesa sale a 4 ore per poterne effettuare un’altro.

***procedure hackportal()***

Calcola un numero random di oggetti che poi verranno inseriti nell’inventario.

***procedure newplayer()***

Permette l’iscrizione di un giocatore.

***Procedure neutralizeportal()***

Dopo l’azione di attacco permette di settare tutti i valori del portale in modo che sia neutro.

***Procedure inventorycalc()***

Procedura utilizzata da diversi trigger per calcolare il numero totale di oggetti nell’inventario.

***trgger newplayer***

Dopo l’iscrizione di un nuovo giocatore, vengono inserite in tutte le tabelle inerenti all’inventario le enuple necessarie.

***trigger newportal***

Dopo l’inserimento di un nuovo portale nel database, vengono inserite in tutte le tabelle inerenti al suo stato le enuple necessarie.

***trigger checkmedal***

Trigger che permette lo sblocco delle varie tier delle medaglie in base al valore raggiunto dalle statistiche.

***trigger checkinventoryweapon, trigger checkinventoryresonator, trigger checkinventorypc, trigger checkinventorypk2, trigger checkinventorypk, trigger checkinventorypm***

Trigger che calcolano il numero totale di oggetti in inventario utilizzando la procedura inventorycalc().

***trigger checkportallevel***

Calcola il livello del portale e la sua energia in base al livello e alla quantità dei suoi risonatori.

***trigger checklevel***

Trigger che controlla e modifica il livello raggiunto da un giocatore in base agli AP guadagnati e alle medaglie sbloccate.

***trigger linknumberinsert***

Caclola il numero di link di un portale.

**Descrizione applicativo**

L’applicativo è formato dalle seguenti pagine php:

* db\_const.php
* function.php
* register.php
* login.php
* game.php
* portals.php
* portalstatus.php
* hack.php
* deploy.php
* mod.php
* link.php
* firexmp.php
* profile.php
* inventory.php
* logout.php

***db\_const.php:*** Contiene i parametri per la connessione alla base di dati e provvede a stabilire una connessione che poi verrà inclusa nelle altre pagine.

***function.php:*** Contiene la funzione che provvede a verificare se un utente ha effettuato il login e in caso positivo consente l’accesso mentre in caso negativo viene reindirizzato alla pagina di login.php. Tale funzione è stata inclusa in tutte le pagine dell’applicativo, eccetto nel modulo di login.php

***register.php:*** Modulo di registrazione dove vengono richiesti nome utente, fazione e e-mail i quali poi vengono inseriti nel database.

***login.php:*** Modulo di login che richiede di inserire nome utente ed email, effettua il controllo che vi siano entrambi in database e procede alla successiva pagine, oppure restituisce un messaggio di errore se uno dei due campi non è corretto.

***game.php:*** Pagina in cui è possibile inserire il codice della cella dove si intende interagire con i portali che vi saranno all’interno.

***portals.php:*** Pagina in cui vengono visualizzati tutti i portali, con relative informazioni sullo stato, appartenenti alla cella precedentemente selezionata e form di inserimento testo che permette di immettere le coordinate gps di un portale con bottone di submit per poter interagire con esso.

Vengono inoltre generate le tabelle per i link e per i field creati nella cella.

Vi è inoltre un pulsante in alto che permette di ritornare al file game.php per selezionare un’altra cella.

***portalstatus.php:*** Pagina in cui vengono stampate tutte le informazioni sullo stato del portale selezionato e pulsanti che chiamano le seguenti pagine:

* hack.php
* deploy.php
* mod.php
* link.php
* firexmp.php

Vi è in alto un pulsante per ritornare alla pagina portals.php

***hack.php:*** Pagina dove automaticamente calcola gli oggetti guadagnati se l’hack va a buon fine e li inserisce in inventario oppure stampa un messaggio che avverte di attendere per effettuare l’hack sucessivo.

***deploy.php:*** Pagina in cui viene visualizzato un select contenente il livello del risonatore che è possibile installare nel portale e un’altro select per scegliere lo slot.

Verranno poi stampate due tabelle, l’una contenente la lista crescente di slot con al di sotto il livello del risonatore inserito (deployato), l’altra con il nick del giocatore che l’ha inserito.

***mod.php:*** Pagina in cui viene visualizzato un select contenente il tipo di mod che è possibile installare nel portale e un’altro select per scegliere lo slot.

Verranno poi stampate due tabelle, l’una contenente la lista crescente di slot con al di sotto il tipo di mod inserito (installato), l’altra con il nick del giocatore che l’ha inserito.

***link.php:*** Pagina in cui vengono visualizzate le coordinate gps del primo portale dove partirà il link, la lista di portali per facilitare la scelta delle coordinate gps del secondo portale e un form di inserimento per tali coordinate con relativo bottone di submit.

Viene quindi chiamata la funzione che controlla se è possibile effettuare il link e restutisce un bool true se l’esito è positivo, in tal caso viene selezionata una seconda funzione che crea il link la quale nuovamente restituisce un bool true se l’esito è stato positivo.

Vengono quindi lanciate due query che distruggono le viste generate da richieste di link precedenti ne creano di nuove per quella corrente le quali servono per la successiva funzione che con la restituzione di un bool confermerà o negerà la creazione di un field.

***firexmp.php:*** Pagina in cui tramite form select viene visualizzato il livello del weapon xmp che si può utilizzare per diminuire il livello delle mod installate, del portale e un bottone di submit per compiere l’azione

Nelle pagine hack.php, deploy.php, mod.php, link.php, firexmp.php vi sono in alto due bottoni per ritornare alla pagina portals.php o portastatus.php

***profile.php:*** Pagina in cui venegono visualizzate tutte le statistiche del giocatore.

In alto vi sono i bottoni per ritornare alla pagine game.php se non è selezionata alcuna cella, altrimenti a portals.php o se è selezionato un portale a portalstatus.php.

***inventory.php:*** Pagina in cui venegono visualizzati tutti gli oggetti che il giocatore dispone in inventario.

In alto vi sono i bottoni per ritornare alla pagine game.php se non è selezionata alcuna cella, altrimenti a portals.php o se è selezionato un portale a portalstatus.php.

***logout.php:*** pagina che distrugge la sessione e reindirizza l’utente a login.php.

***La pagina principale per iniziare ad utilizzare l’applicativo è register.php***