

DOCUMENTO DI GAME DESIGN GDD: CARS HELL

LOGO STUDIO DI SVILUPPO



LOGO GIOCO



Versione	Autori	Data	Note
1.0	D.C	17/03/2022	Prima versione del template per GDD, stesura della trama
1.1	D.C.	20/05/2022	Aggiunte bozze relative a meccaniche di gioco ed interfacce utente
1.2	D.C, M.L	15/07/2022	
1.3	D.C, M.L.	03/08/2022	
2.0	D.C, M.L , F.F., M.F.	10/09/2022	Revisione finale

Descrizione del progetto (elevator pitch)

Questo documento di game design descrive i dettagli di un gioco mobile multiplayer, arcade, con meccaniche battle royale, il cui obiettivo è quello di giungere all'ultimo livello. Pertanto, vi troverete a dover controllare un veicolo in un'arena a forma sferica simil mondo, dove dovrete cercare di distruggere tutti gli avversari in un tempo limitato prima che lo facciano loro.

Ogni arena sarà caratterizzata da delle caratteristiche ben precise e andando avanti potrete sbloccare nuovi veicoli che con le loro peculiarità vi aiuteranno a raggiungere il vostro obiettivo. Non sempre però scegliere il veicolo migliore sarà la scelta giusta.

Team

Il Team si compone di 4 membri:

- **Daniele Cecca**
- **Marco Lillo**
- **Francesco Ferrulli**
- **Michele Fraccalvieri**

Tutti i membri del team possiedono spiccate abilità di programmazione, in particolare nella programmazione ad oggetti. Sebbene sia la prima volta che i membri del team si apprestano ad utilizzare un motore grafico come Unity e sebbene alcuni di loro non abbiano mai programmato in C# (linguaggio usato in Unity) la forte conoscenza delle basi di OOP, dovuta alla conoscenza di Java, e grazie alla loro curiosità e volontà di apprendere, sono riusciti a destreggiarsi in maniera egregia all'interno di Unity, occupandosi non solo del coding ma anche degli aspetti grafici: modelli 3D e GUI.

Indice

Personaggi	5
Storia	5
Tema	6
Trama	6
Gameplay	7
Obiettivi	7
Abilità del giocatore	7
Meccaniche di gioco	7
Arene	8
Oggetti e power-ups	10
Auto disponibili	11
Progressione e sfida	11
Sconfitta	12
Demo	13
Bozza e prototipi	15
Art Style	16
Musica e Suono	16
Dettagli Tecnici	17
Mercato	19
Target	19
Piattaforma e monetizzazione	19
Localizzazione	19
Idee	20

Personaggi

Il gioco prevede un solo protagonista che dovrà gareggiare contro altri giocatori . Si prevede un solo protagonista poiché il nostro obiettivo era quello di far entrare in simbiosi l'utente giocante con il personaggio; difatti la storia si articola in modo tale da far sentire l'utente giocante il protagonista stesso.

Oltre al protagonista ci saranno dei personaggi secondari che non svolgono un ruolo realmente attivo ma permettono di far interagire il personaggio con alcune meccaniche del gioco; difatti i seguenti personaggi non sono realmente sviluppati ma permettono di giustificare alcune meccaniche:

1. **il traghettatore**, a cui pagare il pedaggio per passare da una arena all'altra
2. **l'aiutante** del giocatore che funge sia da consigliere (guida del gioco e tutorial) che da mercante
3. **Entità** indefinita che premia il giocatore alla fine del suo viaggio permettendogli di tornare in vita

Storia

La storia del gioco riguarda il tragico incidente del protagonista, che mentre stava tornando a casa da una escursione in una selva viene coinvolto in un incidente d'auto che lo porterà alla morte.

Inaspettatamente, il protagonista si risveglia in un mondo a lui sconosciuto, e scopre di non essere solo.

Dopo essersi ristabilito il protagonista ricorda di essere morto nell'incidente e giunge alla conclusione che tutti coloro che si trovavano con lui erano morti: infatti, il mondo in cui si risveglia è popolato da persone anch'esse morte in un incidente d'auto, accaduto nello stesso minuto, in tutto il mondo.

I "giocatori" scopriranno che è un mondo che li riserva un'ultima possibilità per evitare di morire definitivamente: ci sarà un solo vincitore e quest'ultimo avrà due possibili scelte:

- 1) Ritornare in vita e dimenticare di tutto l'accaduto
- 2) Riportare in vita un "giocatore", ma rimanere per sempre in questo mondo

Il mondo si compone di 9 arene in cui sopravvivere e sbaragliare gli avversari.

Tema

Il gioco presenta un tema centrale, quello della redenzione.

La metafora della **Divina Commedia** di Dante, è evidente in alcuni aspetti del gioco:

1. il viaggio attraverso dei gironi (le arene)
2. la presenza di 9 arene come i 9 gironi dell'inferno
3. la figura del traghettatore, che sblocca il passaggio da un'arena all'altra
4. la figura del consigliere/commerciante, simile alla figura di Virgilio
5. La morte del protagonista avvenuta mentre stava tornando da una escursione in una selva

Inoltre, è stata presa ispirazione anche dal fumetto **Gantz**, in particolare:

- la sfera nera del fumetto, nota come Gantz, equivale al pianeta
- il trasferimento della coscienza/copia di tutti i personaggi all'interno dell'arena prima di un incidente che avrebbe portato alla loro morte

Trama

Il gioco si apre con la prima arena che funge da tutorial e che permette al nostro personaggio di muovere i primi passi, imparando così le basi, inizialmente a bordo della stessa auto con la quale è deceduto. In seguito dovrà affrontare arene via via più complesse a cui vi potrà accedere solamente vincendo più partite possibili e, per farlo, dovrà anche cercare di migliorare il proprio veicolo.

Gameplay

Obiettivi

L'obiettivo del gioco è raggiungere l'ultima arena superando tutte le arene precedenti e per farlo bisognerà vincere più partite possibili distruggendo gli avversari e raccogliendo più soulcoin possibili. Man mano che si progredirà nel gioco si potranno sbloccare nuove auto più performanti.

La valuta (i Soulcoin) verrà utilizzata per sbloccare nuove auto, e sarà la stessa utilizzata per raggiungere nuove arene, pagando il traghettatore; pertanto, il giocatore dovrà compiere una scelta:

- progredire sempre con la stessa auto
- sbloccare una nuova auto e tornare ad una arena precedente

Abilità del giocatore

Il protagonista mediante l'uso dei veicoli ha delle abilità che potranno essere potenziate e modificate durante il gioco raccogliendo i soulcoin e progredendo fra le arene. In particolare, le abilità del protagonista sono:

- **guidare** qualsiasi auto
- **raccogliere** sia monete sia oggetti in game(power-up) attivabili in game
- **danneggiare** gli avversari
 - un attacco dalla media-lunga distanza(un' arma che toglierà parte di un punto vita al giocatore colpito)
 - un attacco a distanza ravvicinata(una sorta di sprint che toglierà un intero punto vita al giocatore colpito)

Meccaniche di gioco

Le meccaniche principali sono:

- Il **looting** degli oggetti e dei soulcoin
- Il poter **guidare** un'auto e **danneggiare** gli avversari (combattimento ranged/mounted)

Come meccaniche secondarie:

- Il poter acquistare dal mercato in-game veicoli più performanti, ognuno con delle caratteristiche peculiari;
- un "garage" dal quale scegliere il miglior veicolo per una determinata arena

- la presenza di un timer in ogni arena: al suo scadere il giocatore con più soulcoin vince
- Il looting di una minima parte dei soulcoin degli altri giocatori, che verranno droppati quando colpiti.
- Il poter acquistare tramite valuta reale delle gemme che potranno essere utilizzate per acquistare soulcoin
- La presenza di eventi in game che modificheranno l'esperienza di gioco
- Un sistema di vita basato su punti vita che vanno ad identificare un valore continuo di base di 100.
- Un sistema che mostra durante una partita i giocatori rimanenti e il numero di soul coin raccolti

Looting

Il giocatore potrà raccogliere nelle arene:

- I **soulcoin**, monete utili per progredire nel gioco, da utilizzare per:
 - sbloccare il passaggio per ogni arena
 - acquistare nuovi veicoli dal mercato-in game
- **Power-up temporanei**, disponibili nelle arene in-game:
 - saranno attivabili da un apposito pulsante a schermo, la cui icona cambierà in relazione al power-up raccolto in quel momento
 - se si è in possesso di un power up, non sarà possibile raccoglierne un altro (power-up non cumulativi)

Guida e danneggiamento nemici

Il giocatore potrà guidare ogni veicolo cliccando sulla sezione **destra** e **sinistra** dello schermo per muoversi rispettivamente a dx e sx.

Il veicolo invece, si muoverà **automaticamente** verso **avanti**, senza il bisogno di premere alcun tasto.

Arene

Le arene corrispondono ai vari livelli del gioco, di difficoltà crescente, ognuna con delle caratteristiche distintive:

Arena caratteristica	Numero giocatori	Power-up	Eventi in-game	Malus
Arena 1	1	/	/	/
Arena 2	3	1. Attivare lo scudo	/	/
Arena 3	4	1. Attivare lo scudo 2. Aumentare il danno dello shooting	/	/
Arena 4	6	1. Attivare lo scudo 2. Aumentare il danno dello shooting 3. Ricaricare la vita	Compaiono periodicament e delle zone dell'arena che rallentano il giocatore	/
Arena 5	6	1. Attivare lo scudo 2. Aumentare il danno dello shooting 3. Ricaricare vita 4. Rallentare il tempo	Delle meteore che infliggono danno al giocatore se colpito	/
Arena 6	6	1. Attivare lo scudo 2. Ricaricare munizioni 3. Rallentare tempo 4. Aumentare il danno dello shooting	Un tornado che infligge danno al giocatore e fa perdere tutte le moente	1. Perdere munizioni
Arena 7	8	1. Attivare lo scudo 2. Aumentare il danno dello shooting 3. Ricaricare vita 4. Rallentare il		1. Perdere munizioni 2. Avvelenare giocatore

		tempo 5. Aumentare la durata del dash		
Arena 8	8	1. Attivare lo scudo 2. Aumentare la durata del dash 3. Aumentare danno shooting 4. Ricaricare vita	Compare una nebbia che oscura la visuale ai giocatori	1. Avvelenare giocatore 2. Rallentare giocatore 3. Invertire comandi
Arena 9	8	1. Attivare lo scudo 2. Aumentare la durata del dash 3. Aumentare danno shooting 4. Ricaricare munizioni 5. Rallentare il tempo	Compaiono periodicament e delle zone dell'arena che avvelenano il giocatore	1. Perdere munizioni 2. Avvelenare giocatore 3. Rallentare giocatore

Oggetti , power-ups e malus

Il giocatore potrà raccogliere durante l'esperienza nelle arene, come visto precedentemente, vari power-up temporanei:

POWER-UP caratteristiche	
Rallentare il tempo	La velocità degli altri giocatori è dimezzata
Ricaricare la vita	La vita del giocatore si ricarica di 1 punto vita
Aumentare il danno dello shooting	Il danno del proiettile è raddoppiato
Ricaricare munizioni	Il numero di proiettili disponibili torna ad essere quello iniziale
Attivare lo scudo	Attorno al giocatore si attiva uno scudo che lo protegge da qualsiasi attacco di un giocatore nemico: dash o shooting
Aumentare la durata del dash	La durata del dash è aumentata

MALUS	
Perdere munizioni	Il giocatore perde tutte le munizioni
Avvelenare giocatore	Il giocatore viene avvelenato e pertanto perderà per un periodo di tempo punti vita
Rallentare giocatore	La velocità del giocatore viene dimezzata
Invertire comandi	I comandi per guidare l'auto: sx e dx sono invertiti

Auto disponibili:

Auto \ caratteristiche	Danno Dash	Danno Shooting	Difesa	Vita	Velocità
Paddy	30	10	30	100	20
Ambulance	30	10	40	200	10
Roger	50	20	30	80	30
God	60	30	40	150	30
Reaper	30	30	20	100	30
Fireman	40	10	40	180	10

Progressione e sfida

Il giocatore potrà progredire nel gioco sbloccando di volta in volta delle arene, ma esse avranno una difficoltà crescente considerevole. Infatti, i fattori che incrementeranno la difficoltà di ogni arena sono:

- **eventi in-game** che ostacoleranno il giocatore
- **numero crescente di avversari**
- **malus** raccogliibili come finti power-up

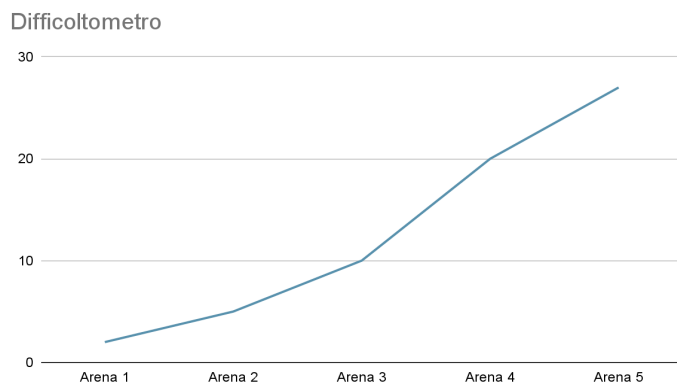
Contestualmente, la difficoltà sarà maggiore nelle arene più avanzate perché popolate da più giocatori e da giocatori con maggiore esperienza di gioco e con veicoli più performanti e potenti.

Man mano che si progredisce nelle arene, non sarà solo la difficoltà e il livello di sfida richiesto al giocatore per superare le arene a crescere, ma aumenteranno anche i power-up disponibili, come sorta di bilanciamento tra bonus e malus.

Nella pratica, è possibile risultare il “vincitore” di ogni arena:

- risultare il **Last Player Standing** dopo lo scontro con gli avversari
- allo scadere del Timer, risultare il **giocatore con più Soulcoin**.

Grafico ipotetico della progressione della difficoltà:



i coefficienti sono stati ipotizzati considerando tutti i vari fattori che contribuiscono alla difficoltà di ogni arena.

N.B. all’inizio la difficoltà cresce non troppo rapidamente, per far ambientare gradualmente il giocatore.

Sconfitta

Il giocatore può essere sconfitto in due modi:

- se scade il limite di tempo settato a due minuti e non ha il numero maggiore di soulcoin;
- se viene colpito troppe volte dagli avversari, perdendo tutta la vita disponibile

In entrambi i casi, il giocatore ripartirà dal menù iniziale senza le monete raccolte nell’arena nella partita precedente, in cui è stato sconfitto.

Demo

All'interno di questa demo sono state realizzate 4 scene:

- **HOME**
- **GARAGE**
- **SHOP**
- **BATTLE**(scena dove avviene effettivamente il gameplay)

All'interno della **Home** è stato implementato:

- il sistema che permette di scegliere una tra le quattro arene implementate selezionando l'immagine che la rappresenta, tramite uno slide sullo schermo.
- il pulsante "Battle" che se piggiato permette di iniziare la partita.

All'interno del **Garage** è stato implementato:

- Il sistema che permette di selezionare il veicolo desiderato(per via di alcune problematiche dovute all'istanziazione a run-time dei prefabs è possibile solamente vedere i veicoli disponibili e non effettivamente selezionarli)

All'interno del **Shop** è stato implementato:

- L'ipotetica interfaccia che mostra la possibilità di acquistare gemme e soulcoin nei vari pacchetti disponibili(la funzione dei pulsanti non è effettivamente implementata)

In tutte e 3 le scene è stato poi implementato un **sistema di navigazione** che permette all'utente di navigare attraverso queste 3 scene e un pulsante a forma di rotella che se piggiato apre la sezione dedicata ai settings.

Nella scena "**Battle**" è stato implementato:

- Il sistema di movimento, attacco e looting sopra descritto.
- Diversi aspetti grafici per indicare soulcoin, giocatori, tempo e proiettili rimanenti.
- L'interfaccia che mostra se al termine della partita il giocatore ha vinto o perso con il numero rispettivo di monete

Come detto precedentemente sono state realizzate **4 arene**, le quali rispettano in parte le caratteristiche delle arene sopra elencate; infatti non sono stati implementati gli eventi in game, i malus ed eventuali ostacoli.

In particolare nella prima arena, che funge da tutorial è stata implementata una schermata che viene mostrata prima della partita che mostra e spiega i comandi disponibili.

Per fare in modo che vengano mostrate all'utente tutte le macchine disponibili nell'arena in ogni arena il giocatore si troverà a usare un veicolo differente.

Sebbene questo gioco si presenta come gioco multiplayer, quest'ultimo non è stato implementato ma i nemici sono gestiti da una ai.

Art Style

E' un gioco 3D con grafica low-poly e sprite 2D, dallo stile cartoon.
In particolare, gli sprite 2D riguardano l'HUD, il menù;
mentre gli asset 3D sono stati usati per i veicoli, le arene, i proiettili.

Tutti gli asset 3D e 2D sono stati selezionati dallo Unity Asset Store e modificati in base alle esigenze.

Sono presenti elementi VFX in seguito alla fase di shooting, dash ed in seguito alla morte di un giocatore.

Per creare un UI più coerente con la storia si è cercato di usare nelle scene di navigazione: Home , Garage e Shop una gamma cromatica che ricordasse l'inferno mentre nelle varie arene si è deciso di usare dei colori per il background che fossero coerenti con il colore del pianeta presente in quella scena.

Per quanto riguarda la fase di gameplay si è deciso di utilizzare una gamma cromatica sul blu per identificare i proiettili e l'effetto del dash del giocatore mentre si è deciso di utilizzare una gamma cromatica sul rosso per identificare quelli del nemico/i.

Musica e Suono

Il gioco presenta una selezione accurata di suoni per ogni evento presente nel gioco, tutti copyright-free.

Nelle schermate di navigazione HOME SHOP E GARAGE è stata utilizzata la seguente traccia : "Purple Planet Music- Awakening" ed è stato scelto di usare la stessa traccia per dare un senso di continuità al giocatore che sta navigando nell'applicazione.

Nella fase di gameplay effettivo ogni arena avrà una traccia audio differente che sarà via via con bpm e un ritmo crescente in maniera proporzionale all'aumento di difficoltà, in modo tale da coinvolgere maggiormente emotivamente l'utente.

In particolare avremo nella demo le seguenti tracce:

1. Arena 0 :Purple Planet Music- Time Perception
2. Arena 1 :Purple Planet Music- Hybrid Tech Theme
3. Arena 2 :Purple Planet Music- Deep In Tran
4. Arena 3 :Purple Planet Music- Digital Motion

Sono poi stati aggiunti degli effetti sonori per lo shooting dei proiettili e per il dash, in particolare per questo è stato aumentato il pitch del suono del motore dell'auto. Suono del motore dell'auto che sarà differente da auto ad auto. Sarà inoltre implementato il suono 3d per il suono dei motori nemici.

Il livello del volume della musica e degli effetti sonori sarà regolabile dal menu generale del gioco in qualsiasi momento

Dettagli Tecnici

Il gioco è stato sviluppato interamente con Unity Engine, ed è disponibile al seguente link Github: <https://github.com/Ferru2000/Progetto-SVG>

Il progetto è attualmente open source e chiunque può contribuire al suo miglioramento/modifica.

E' stata realizzata sia una demo per dispositivi mobili sia per desktop.

Qui sotto spieghiamo uno dei sistemi chiave per il funzionamento del gioco:
il sistema di gravità.

Funzionamento della gravità

Il sistema di gravità funziona mediante 2 script:

- **PlanetScript**
- **Gravity**

Il primo di questi due script è assegnato all'arena che sarà il centro della gravità e pertanto fungerà da attrattore per gli altri oggetti.

E' stata realizzata una funzione **Attract()** che prende come parametro **l'oggetto da attrarre**, a cui sarà applicata la forza gravitazionale; Per individuare la direzione della forza gravitazionale viene calcolato il vettore **gravityUp**

La forza gravitazionale è poi aggiunta al **RigidBody** dell'oggetto attratto tramite il metodo **AddForce**, che prende come parametri **gravityUp** e un float number **gravity** che indica effettivamente quanto sarà forte la gravità.

Per fare in modo che l'oggetto ruoti verso il lato giusto è stata modificata la **componente rotation** dell'oggetto attratto

```

public class PlanetScript : MonoBehaviour
{
    [SerializeField]
    private float gravity = -12;

    1 riferimento
    public void Attract(Transform playerTransform)
    {
        Vector3 gravityUp = (playerTransform.position - transform.position).normalized;
        Vector3 localUp = playerTransform.up;

        playerTransform.GetComponent<Rigidbody>().AddForce(gravityUp * gravity);

        //parte che serve per ruotare il corpo dell'oggetto verso il lato giusto, in modo tale che sia piano rispetto alla
        Quaternion targetRotation = Quaternion.FromToRotation(localUp, gravityUp) * playerTransform.rotation;
        playerTransform.rotation = Quaternion.Slerp(playerTransform.rotation, targetRotation, 50f * Time.deltaTime);
    }
}

```

Il secondo script invece è assegnato all'oggetto attratto.

Questo metodo avrà come attributo lo script **PlanetScript** che permette di chiamare nella funzione **FixedUpdate** il metodo **Attract()**, passando come parametro l'oggetto a cui si sta assegnando questo script.

```

// Update is called once per frame
Messaggio Unity | 0 riferimenti
void FixedUpdate()
{
    if (attractorPlanet)
    {
        attractorPlanet.Attract(objectTransform);
    }
}

```

Mercato

Abbiamo deciso di realizzare un mobile game essendo l'industria del mobile nel settore gaming la più florida, **rappresentando il 45% dell'intero settore** e avendo registrato nel secondo trimestre del 2022 un guadagno di 22 miliardi con una crescita superiore al **30% rispetto l'anno precedente**.

Target

Essendo un mobile game, è destinato ad un pubblico casual, ma soprattutto giovane, **dai 7 anni in su**, poiché è possibile giocare solamente possedendo un semplice smartphone. Non sono previste opzioni di accessibilità per pubblico disabile.

Piattaforma e monetizzazione

Mobile game disponibile sui sistemi operativi seguenti:

- iOS
- Android
- HarmonyOS - Huawei

Il gioco è free-to-play ma presenta la possibilità di fare acquisti in-game, con cui è possibile:

- comprare skin per i veicoli (es. come skin ci potrebbe essere la delorean)
- comprare le Gemme, scambiabili al mercato per i soulcoin

Localizzazione

Il gioco supporterà completamente le seguenti lingue, essendo destinato ad un mercato non solo italiano ma internazionale:

- Italiano
- Inglese

Idee e ispirazioni

Nel corso dei meeting con il team sono sorte le seguenti idee che non si è ancora sicuri di voler sviluppare nel gioco finale.

- Un sistema che permette al giocatore di guadagnare esperienza in modo tale da salire di livello e sbloccare ricompense varie (skin, moltiplicatori di monete,ecc)
- Riempire le arene con degli ostacoli-edifici (cimiteri, chiese, tombe, palazzi,ecc)
- Un sistema di ranking che mostra il giocatore con il maggior numero di vittorie
- Una modalità storia parallela al gioco in se ,in cui si affrontano dei nemici controllati dall'intelligenza artificiale in squadra
- Una modalità tutti contro tutti (FFA) a squadre, dove le squadre sono composte da 2 player
- Un sistema che permette di potenziare i veicoli tramite un'altra valuta che si trova in delle sorte di chest che vengono date al termine della partita in caso di vittoria
- Un sistema che limiti il numero di partite che l'utente può svolgere giornalmente(una sorta di sistema di energia giornaliera)
- Aggiungere movimento con giroscopio

Per la realizzazione di questo gioco abbiamo preso come punto di riferimento :

- **mario kart balloon battle** per le meccaniche base del gioco
- un **qualsiasi battle royale** per le meccaniche battle royale
- **Silent Hill, Doom, Fallout,GTA** per la realizzazione del gdd
- **La Divina Commedia**, per la storia
- il fumetto **Gantz**, per la storia