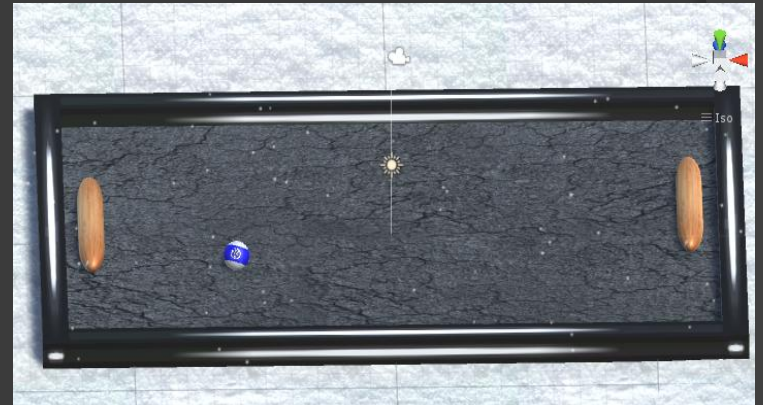
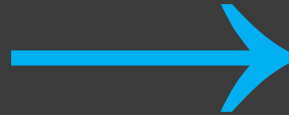
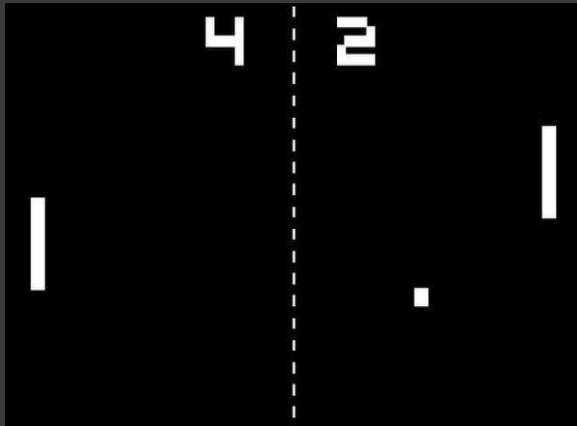


Anno accademico: 2016/2017

Francesco Gradi

LINGUAGGI DI REALTÀ VIRTUALE

3D PONG



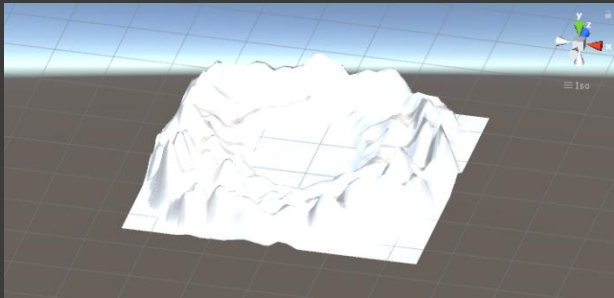
INDICE

● Software utilizzati	pag. 4
● Creazione del Terrain	pag. 5
● L'arena di gioco	pag. 6
● I giocatori	pag. 7
● Le sfere	pag. 9
● I Particle Systems	pag. 10
● Le textures	pag. 12
● La CHEAT MODE	pag. 13
● Il menu di pausa	pag. 16
● Oggetti di gioco secondari	pag. 18
● L'audio	pag. 20

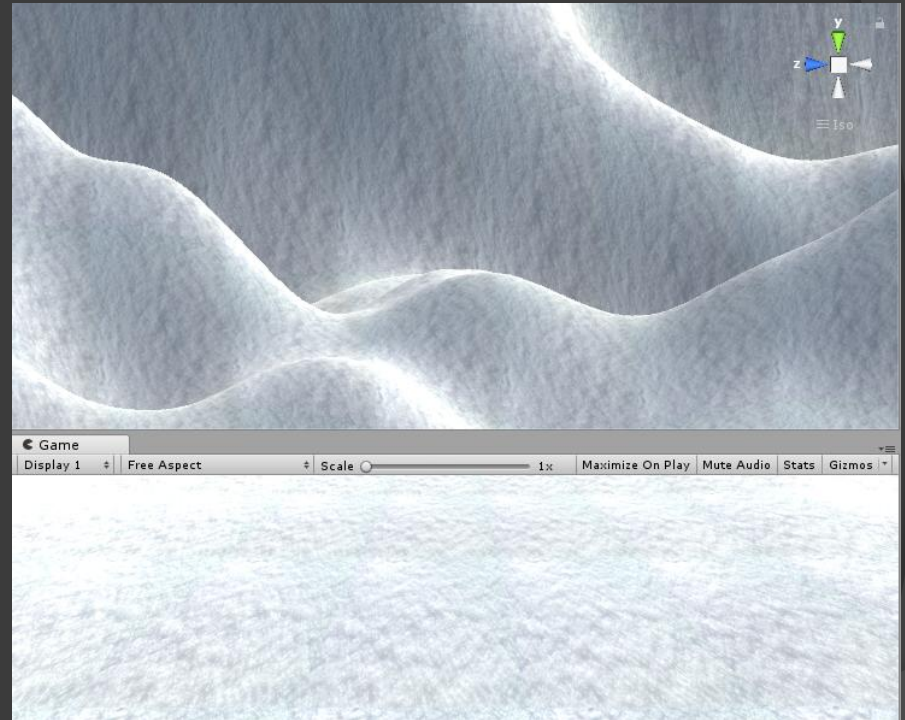
I SOFTWARE UTILIZZATI

- **Unity3D**: struttura e logica del progetto.
- **Blender**: oggetti più complessi.
- **GIMP**: textures.
- **Audacity / Sony Vegas**: audio.

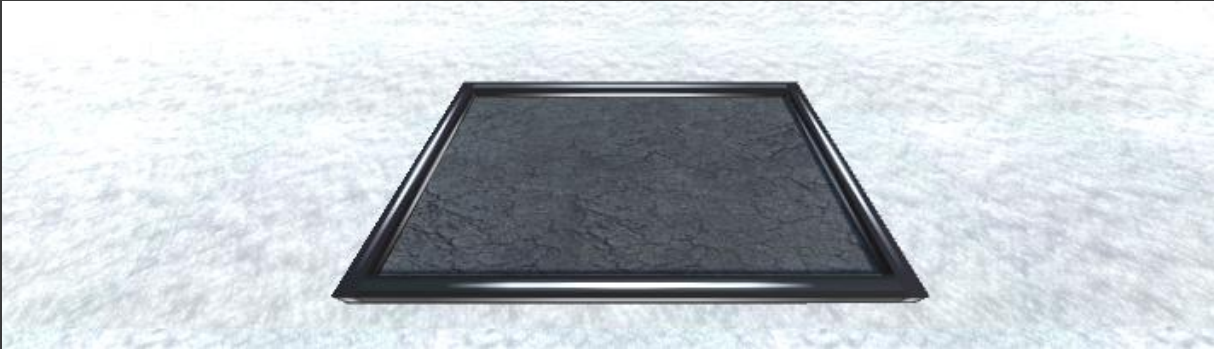
CREAZIONE DEL TERRAIN



Ho dapprima creato l'oggetto Terrain, modificato con alture e depressioni e infine applicata la texture precedentemente creata tramite GIMP. Tutto affinché il suo aspetto potesse essere quello di un paesaggio montano, dune di neve incluse.



L'ARENA DI GIOCO



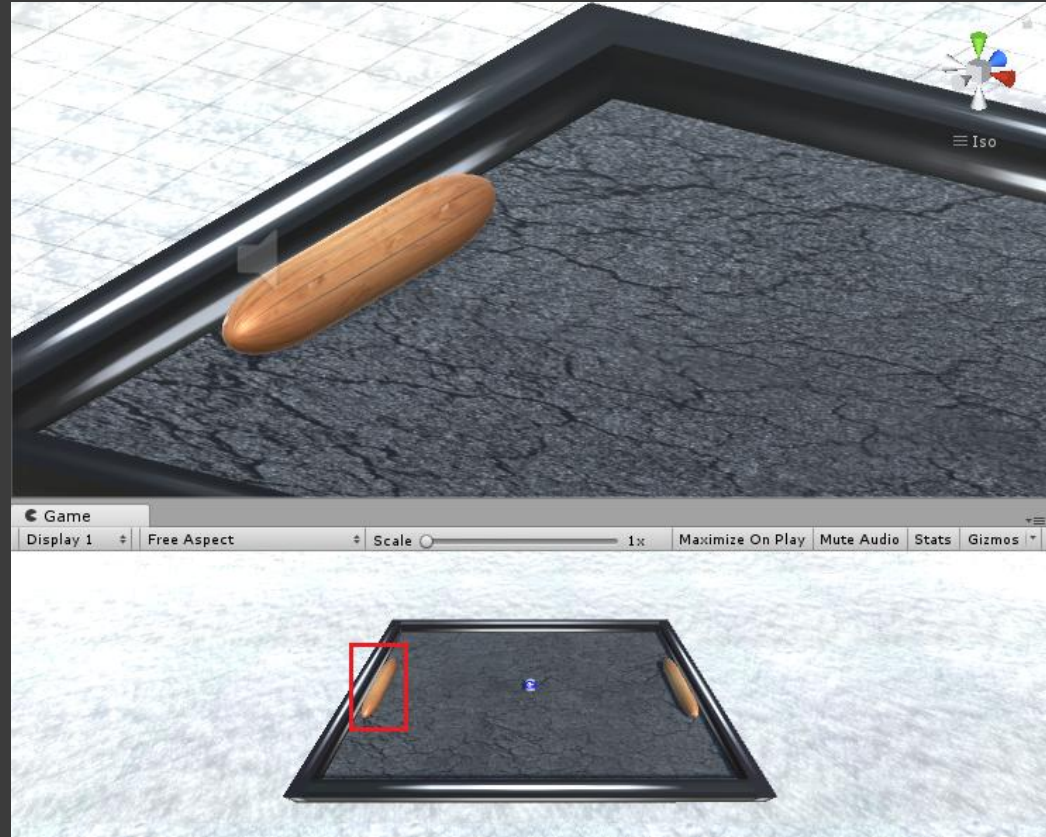
L'arena è composta di 4 parallelepipedi a formare i muri, più un piano di gioco. Tramite script, ho impostato che le collisioni con i muri destro e sinistro aggiornino il punteggio in base a quale dei due è stato colliso.



I GIOCATORI

● Il Player 1

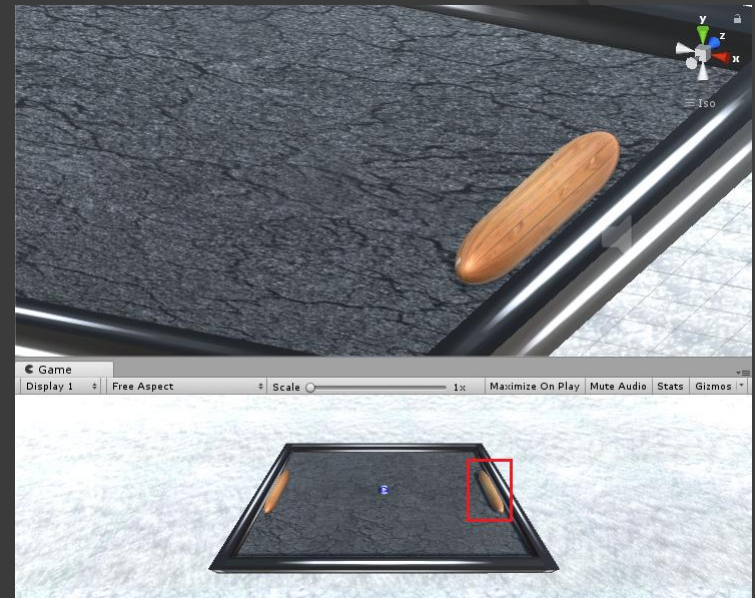
Il giocatore 1 è controllato dall'utente tramite un semplice script. Il suo corpo rigido è vincolato all'asse Z, in modo da impedirgli di muoversi in altre direzioni all'infuori di quella lungo l'asse Z stesso.



I GIOCATORI

● L'avversario

L'avversario invece è decisamente più complesso. Il gioco è pensato unicamente in modalità SinglePlayer, di conseguenza il giocatore 2 si deve muovere autonomamente per intercettare la palla. Il suo corpo rigido è anch'esso vincolato all'asse Z, in più come mostra lo script accanto segue il movimento della palla sempre sull'asse Z. La variabile "containerZ" gli impedisce di uscire fuori dall'arena di gioco e la funzione "MoveTowards" fa in modo che che non raggiunga immediatamente la sua destinazione, ma che ci arrivi con una piccola accelerazione. Il che rende il gioco più gradevole e l'avversario meno competitivo.



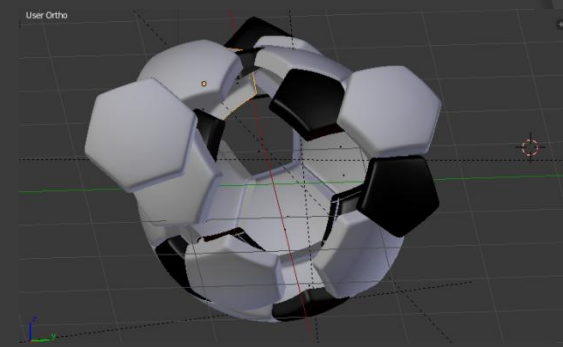
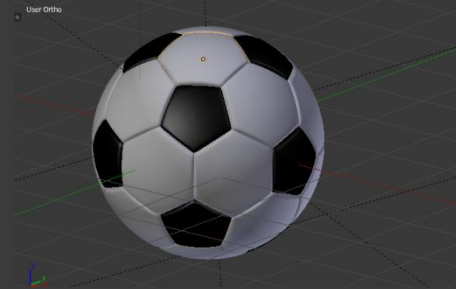
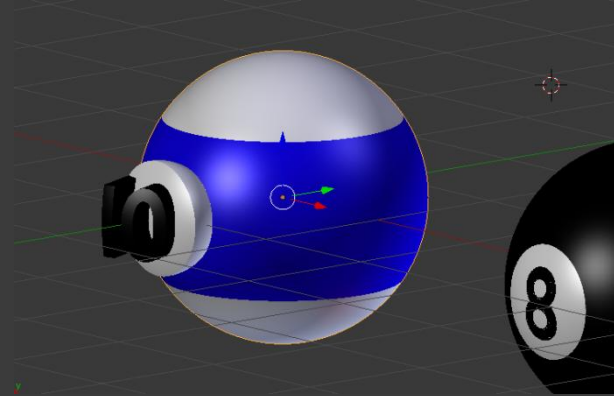
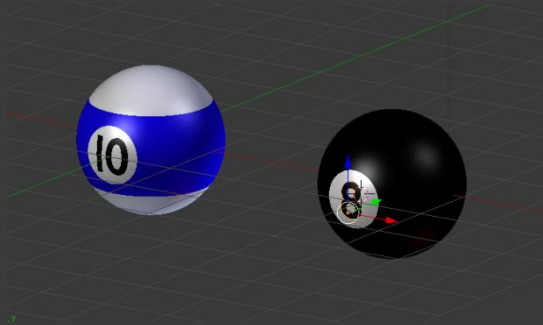
```
// Use this for initialization
void Start () {

    ball = GameObject.FindGameObjectWithTag ("Ball").transform;

}

// Update is called once per frame
void FixedUpdate () {
    containerZ = ball.position.z;
    float step = speed * Time.deltaTime;
    if(containerZ > 4)
        containerZ = 4;
    if(containerZ < -4)
        containerZ = -4;
    transVec = new Vector3 (11.9f, 0.75f, transform.position.z);
    ballVec = new Vector3 (0f, 0f, containerZ);
    transform.position = Vector3.MoveTowards(transVec, ballVec, step);
}
```


LE SFERE



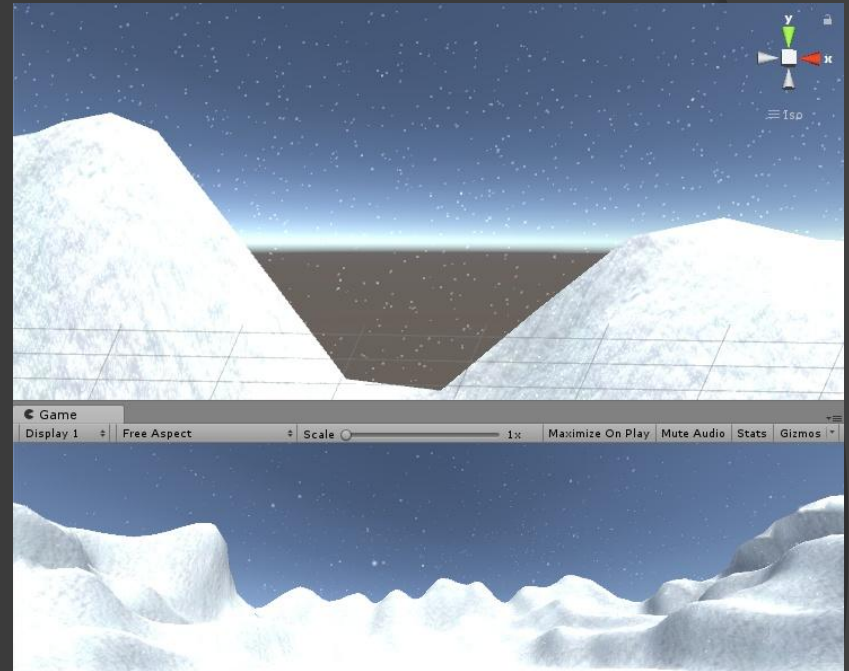
Le sfere sono un capitolo che mi ha richiesto parecchio tempo per essere realizzato. Ho prodotto diversi prototipi, tutti quanti realizzati in Blender. Ho composto le sfere da biliardo tramite intersezioni di altre sfere tra di loro e campi di testo “meshati” per renderli interagibili con dei solidi, la sfera da calcio invece è stata realizzata con un insieme di pentagoni a sé stanti ancorati però ad un oggetto comune.

Ho suddiviso ognuno degli oggetti che compone le sfere in sottofacce per dare l'impressione di una superficie liscia, infine ho applicato due diversi shader per dare lucentezza alle sfere da biliardo e semi-ruvidità a quella da calcio.

I PARTICLE SYSTEMS

● La neve

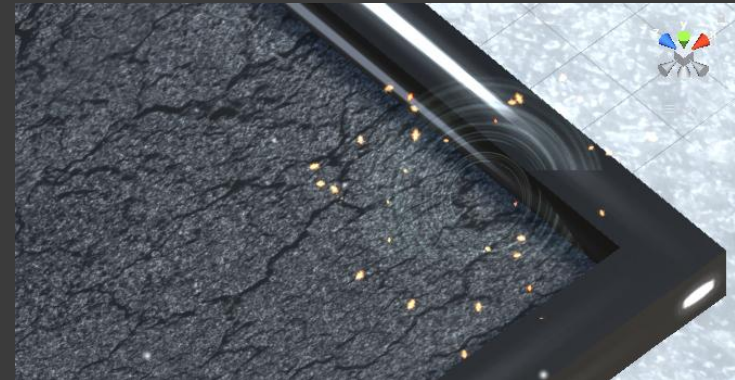
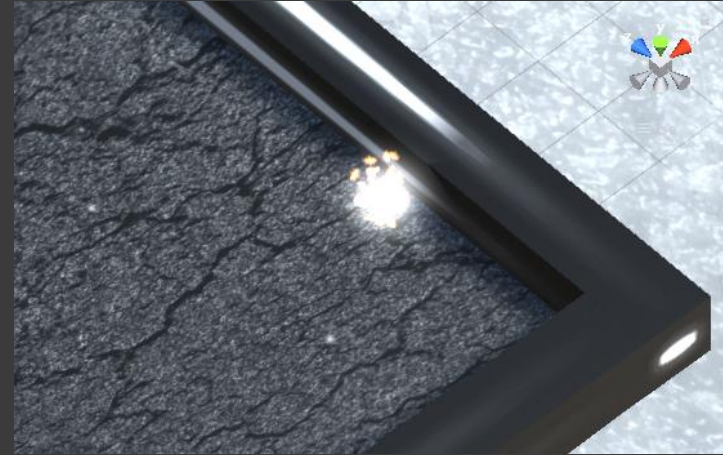
I Particle Systems sono un altro capitolo che ha richiesto non poco tempo di lavoro e studio approfondito. In particolare la neve, dovendo coprire un'area molto estesa, aveva bisogno di settaggi particolari. È stato necessario settare il numero di particelle generate al secondo a più di 100, dopodiché tramite altre impostazioni particolari ho abilitato la rilevazione delle collisioni di ogni fiocco di neve con il terreno e settato la "bounciness" a 0. In questo modo ogni fiocco si deposita a terra (e non trapassa il terrain) e lì rimane prima di scomparire in dissolvenza pochi attimi dopo.



I PARTICLE SYSTEMS

⦿ Le esplosioni

Per dare un tocco di realismo, ad ogni collisione con i muri destro e sinistro dell'arena istanzio tramite script il secondo sistema particellare del progetto: le esplosioni. Il che mi sembrava una buona idea dato che una volta colliso la palla torna al centro del campo per continuare la partita. Ho dovuto successivamente implementare un altro script che dopo un paio di secondi eliminasse tutti i cloni dell'esplosione che venivano istanziati (figura in basso a destra).



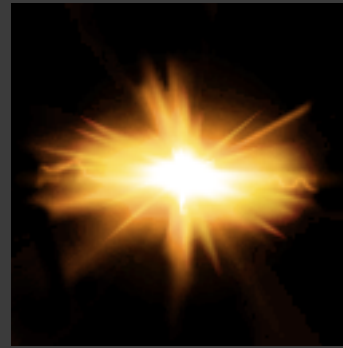
```
► explosion(Clone)  
► explosion(Clone)  
► explosion(Clone)  
► explosion(Clone)  
► explosion(Clone)  
► explosion(Clone)  
► explosion(Clone)
```


LE TEXTURES



Terrain

Ho realizzato le textures con GIMP facendo ben attenzione a rendere i pixel alle estremità simili tra loro, in modo che una volta applicate dessero l'idea di continuità.



Esplosioni



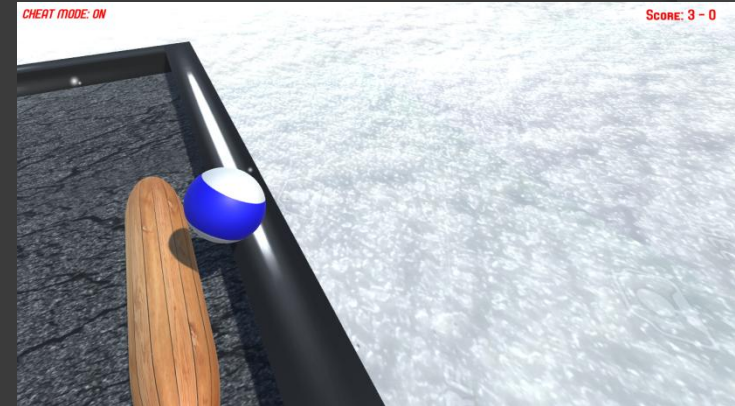
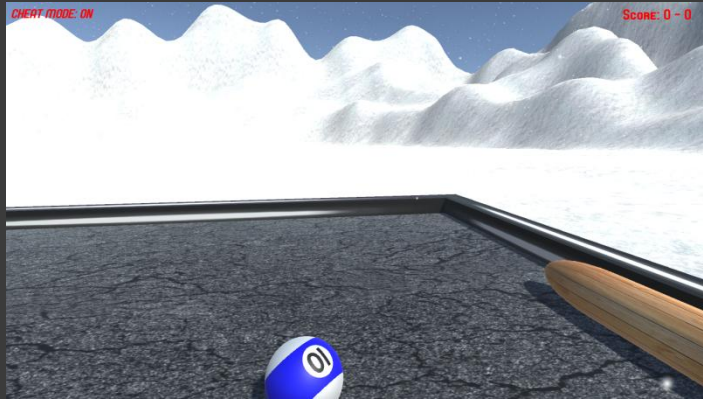
Giocatori



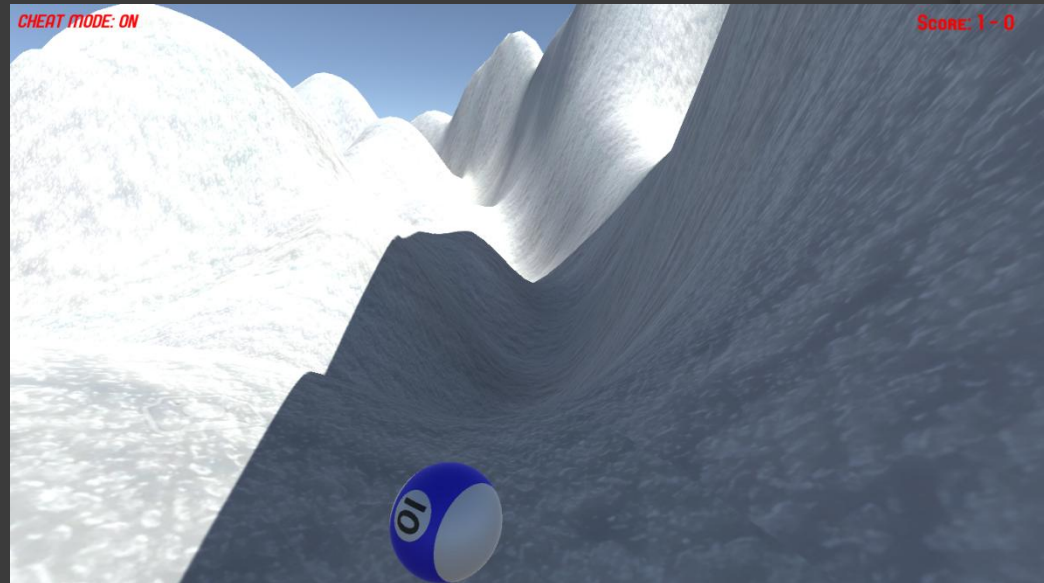
Arena



LA CHEAT MODE



La Cheat Mode è il secondo grande pilastro del progetto intero. La si può accedere premendo il tasto sinistro del mouse e permette di controllare la palla in terza persona: per segnare all'avversario saltandolo, per esplorare il mondo circostante o per percorrere le “piste” di neve che ho messo appositamente ai lati della mappa.



LA CHEAT MODE

La foto a destra in alto illustra il codice che permettere di muovere la palla.

Sul piano tecnico la pressione del tasto sinistro agisce sulla telecamera (che inizia a seguire la palla in terza persona e viene ruotata con il mouse), sul materiale della palla (che da 100% “bouncy” diviene vincolato all’attrito) e sul testo in alto a sinistra che informa sullo stato stesso della Cheat Mode.

Perché la telecamera si muova di conseguenza è stato necessario creare 2 variabili statiche che informassero se l’utente sta controllando la palla o meno. Di conseguenza ho dichiarato statici anche i metodi qui a fianco, affinché potessero essere acceduti all’infuori dello script corrente.

```
if (ballControllable) {  
    //ON CHEAT MODE:  
    transform.Rotate(0f, 0f, 0f);  
    float moveHorizontal = Input.GetAxis ("Horizontal");  
    float moveVertical = Input.GetAxis ("Vertical");  
    if(Input.GetKeyDown (KeyCode.Space))  
        rb.AddForce(new Vector3(0, 6, 0), ForceMode.Impulse);  
  
    Vector3 movement = new Vector3 (moveHorizontal, 0.0f, moveVertical);  
  
    rb.AddForce (movement * speed);  
    if (Input.GetMouseButtonDown (1)) {  
        |  
        DisableCheatMode ();  
    }  
}
```

```
public static void EnableCheatMode(){  
    //CHEAT MODE ENABLED!  
    playerControllable = false;  
    BallBehaviour.ballControllable = true;  
    BallBehaviour.coll.material.dynamicFriction = 0.9f;  
    BallBehaviour.coll.material.staticFriction = 0.9f;  
    BallBehaviour.coll.material.bounciness = 0.2f;|  
    chmStatus.text = "CHEAT MODE: ON";  
}
```

```
public static void DisableCheatMode (){|  
    //CHEAT MODE DISABLED!  
    ballControllable = false;  
    Player1Controller.playerControllable = true;  
    coll.material.dynamicFriction = 0;  
    coll.material.bounciness = 1;  
    coll.material.staticFriction = 0;|  
    Player1Controller.chmStatus.text = "";  
}
```


LA CHEAT MODE

La realizzazione di questa modalità è ciò che ha richiesto più tempo e studio in assoluto. La sua disattivazione ripristina il materiale della palla e la posizione iniziale della telecamera e può essere eseguita premendo il tasto destro del mouse o direttamente dal menu di pausa.



IL MENU DI PAUSA

Il menu di pausa è accessibile premendo Esc e contiene 3 pulsanti: “Riprendi”, “Cheat Mode”, “Esci”.

La comparsa del menu comporta l’attivazione del relativo canvas, il fermarsi del tempo e la comparsa del cursore.

Con il bottone “Riprendi” è possibile ripristinare lo scorrere del tempo e disattivare il canvas sopra citato comportando la scomparsa del menu stesso (come da figura accanto).

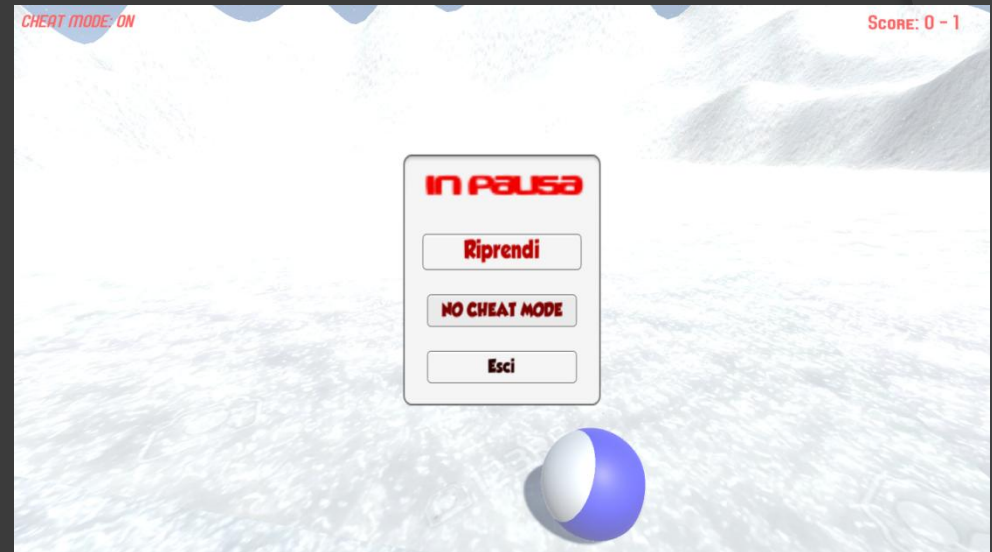


```
public void Pause(){  
    if (canvas.gameObject.activeInHierarchy == false) {  
        canvas.gameObject.SetActive (true);  
        Time.timeScale = 0;  
        Cursor.visible = true;  
    } else {  
        canvas.gameObject.SetActive (false);  
        Time.timeScale = 1;  
        Cursor.visible = false;  
    }  
}
```

IL MENU DI PAUSA

Il pulsante relativo alla Cheat Mode attiva/disattiva la stessa (come da tasto sinistro/destro del mouse) e il testo varia a seconda che sia attiva o meno.

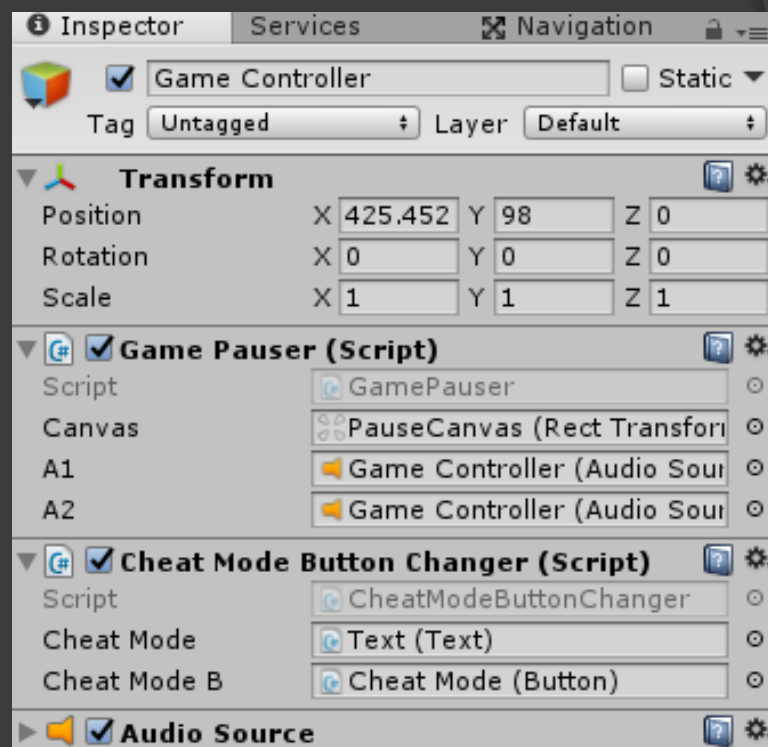
Il pulsante “Esci” chiude l’applicazione.



OGGETTI DI GIOCO SECONDARI

● Il Game Controller

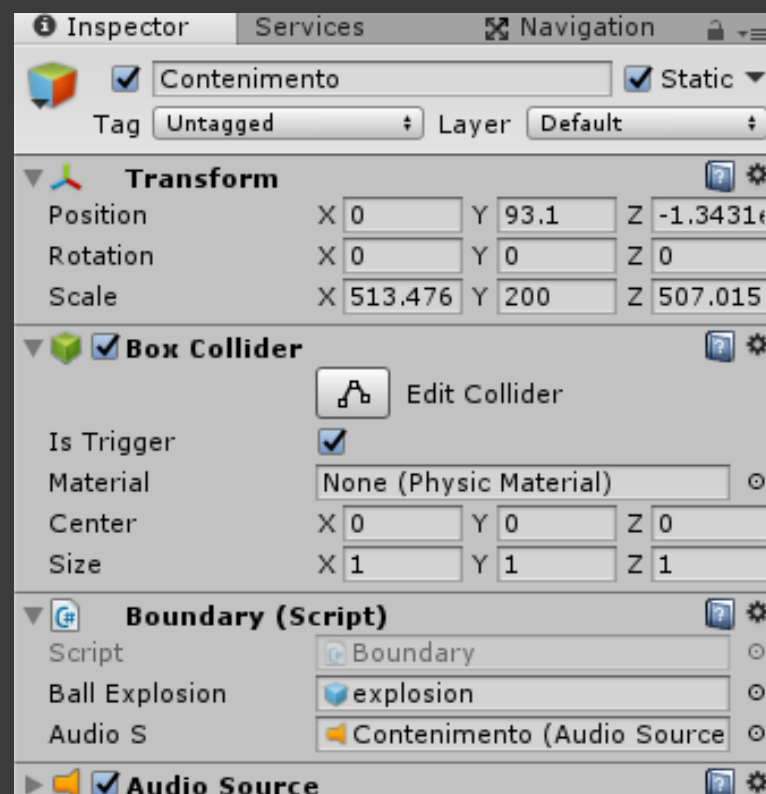
È un oggetto che non ha presenza fisica nel gioco, ma ha più compiti fondamentali. Le sue funzioni principali sono attivare/disattivare il canvas alla pressione del tasto Esc, modificare il testo dei bottoni, nascondere il cursore durante il gioco e riprodurre la musica di sottofondo.



OGGETTI DI GIOCO SECONDARI

● Il Collider di contenimento

Ho creato un oggetto vuoto con solo un collider e uno script annessi che, nel malaugurato caso in cui la palla trapassi il terrain o schizzi via oltre le dune, la riporti al centro dell'arena di gioco. Avendo un collider annesso, basterà che la palla lo oltrepassi per istanziare un'esplosione e riportarla nel suo punto d'origine.



L'AUDIO



Gli audio sono stati realizzati con Audacity e Sony Vegas, ma principalmente con quest'ultimo. La musica di sottofondo è stata editata per essere riprodotta all'inizio e rimanere in loop per tutta la durata del gioco, l'audio relativo alle esplosioni viene riprodotto ad ogni collisione con i muri dietro ai due giocatori e all'attraversamento del collider di contenimento. L'audio "Boing" invece viene riprodotto ad ogni rimbalzo della palla su qualsiasi oggetto.