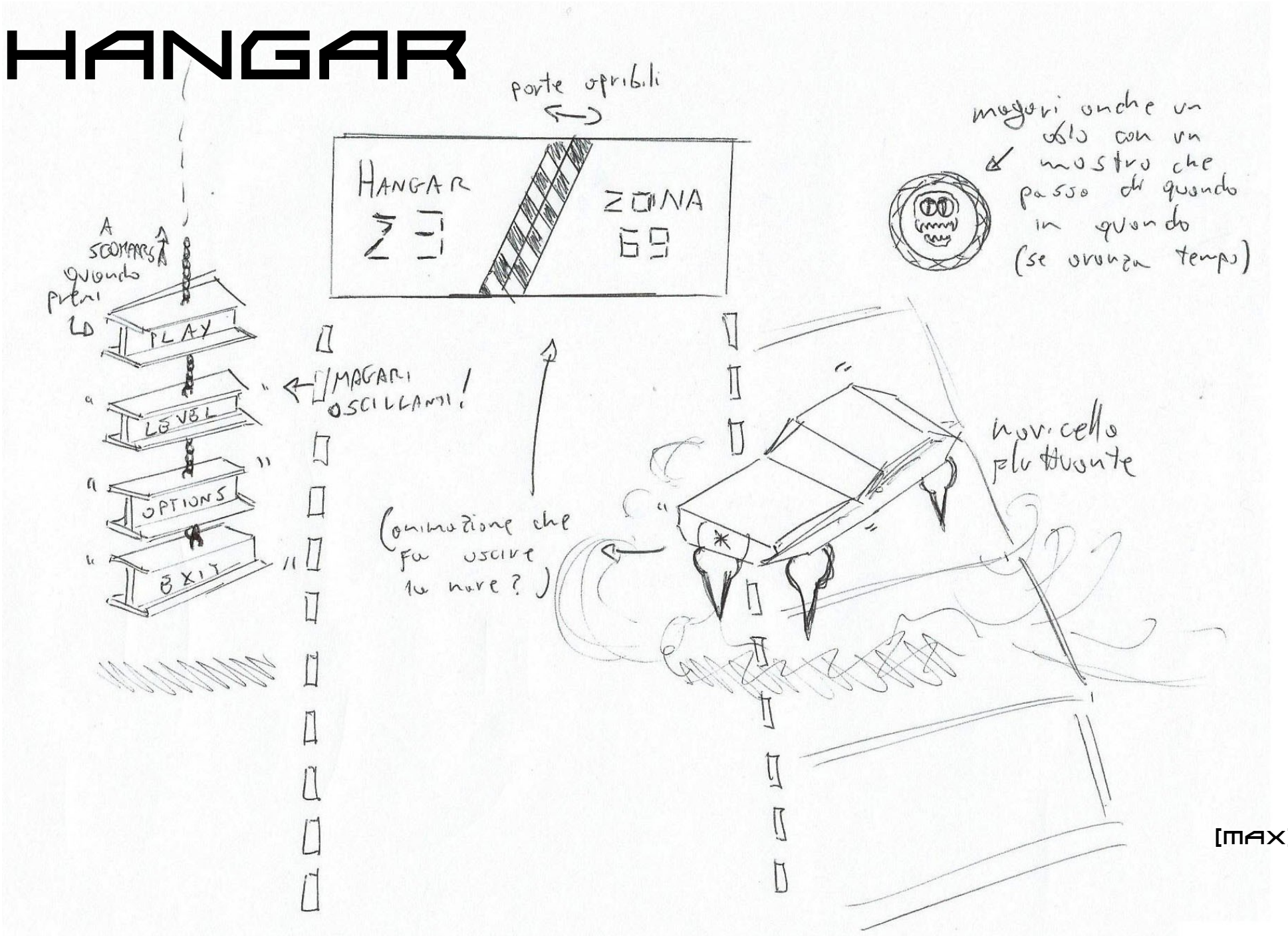
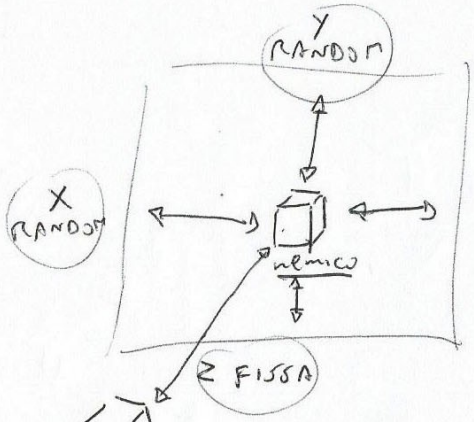
HANGAR

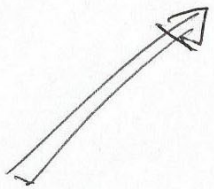

porte apribili
HANGAR 23
ZONA 69
magari anche un oblò con un mostro che passa di quando in quando (se avanza tempo)
A SCOMPARSA quando premi LO
PLAY
LEVEL
OPTIONS
EXIT
MAGARI OSCILLANTI!
(animazione che fa uscire la nave?)
navicella pluttuante

[MAX]

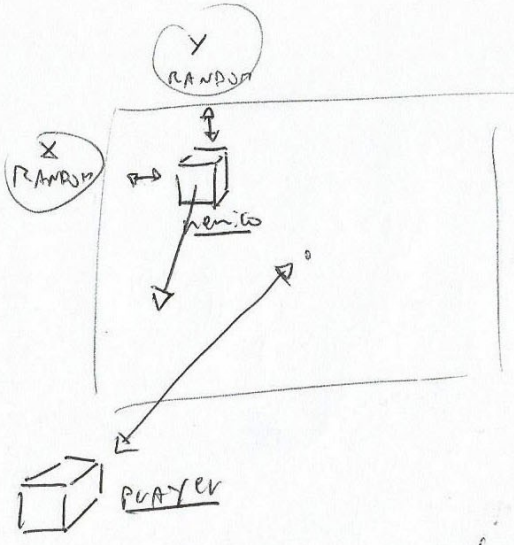
Per realizzare il menù radice di gioco fatto ad hangar abbiamo pensato di mantenere un certo dinamismo come nell'ambiente di gioco, inserendo delle animazioni; per questo sono stati inseriti i tasti fatti a travi oscillanti e la nave di gioco che parte, ed esce dall'hangar appena il giocatore e' pronto a giocare.
C'era anche l'idea di inserire un piccolo mostro in gabbia, ma non c'è stato il tempo, quindi si può vedere la gabbia rimasta vuota in scena, ma che fa comunque la sua figura.
E' stato reso possibile selezionare i singoli livelli dove ora c'è il tasto "score", tutti con un punteggio massimo, un record, attraverso il quale il giocatore potrà provare a sfidarsi ogni volta che vuole, cercando di far fuori più nemici possibili senza morire.
L'hangar è stato poi ottimizzato graficamente di pari passo al gioco facendo un bake delle luci statiche, inserendo qualche piccolo particellare e ottimizzando la nave del player.

NEMICO 1





La mare si deve muovere in una direzione e avere un nemico che gli stia sempre fisso davanti cercando di scansare i colpi del player, ma riuscendo a centrare la mare con un determinato scarto.



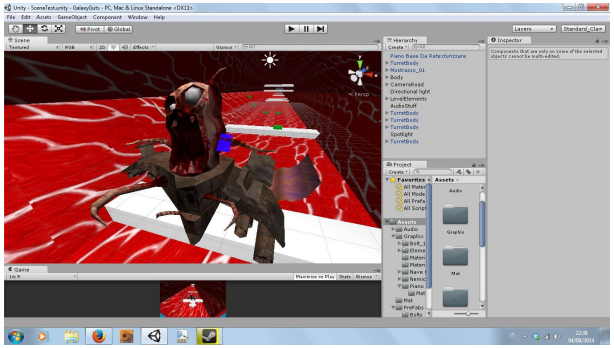
Il nemico dovrà guardare il player per poter sparare nella sua direzione.

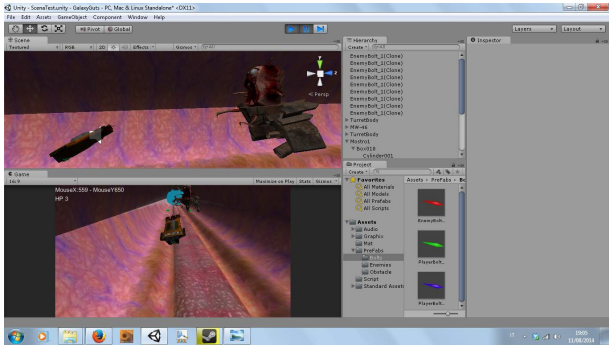
Il player potrà rispondere al fuoco con ~~[illegible]~~ dei cannoni rivolenti montati sulla mare.

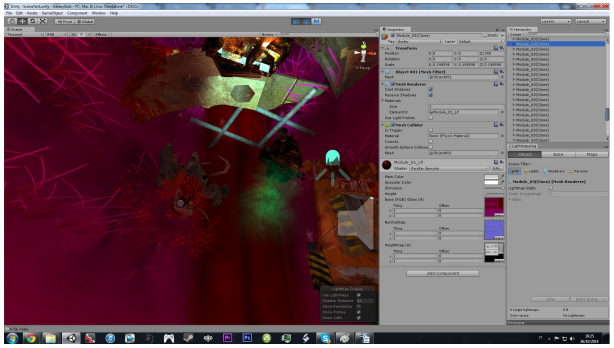
Per schivare i colpi basta ritardare il fuoco del nemico dopo che si è posizionato per dare la possibilità al player di muoversi.

[MAX]

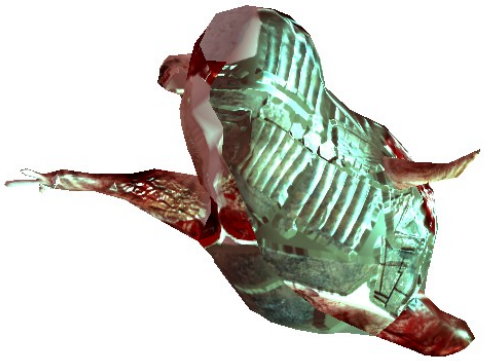
Quando abbiamo deciso di realizzare il primo nemico avevamo pensato a qualcosa che stesse nel mezzo del gioco ad "irritare" il giocatore, quindi abbiamo pensato ad inserire un oggetto con una distanza fissa, ma un movimento verticale ed orizzontale variabile per schivare i colpi del giocatore.
C'era però la problematica di farlo sparare nella direzione giusta, quella del giocatore, così abbiamo pensato di far semplicemente ruotare il nemico nella direzione del giocatore, e sparare mezzo secondo dopo, giusto il tempo di permettere al giocatore di schivare il colpo nemico.
Come per la nave del giocatore si può notare un miglioramento grafico aggiungendo cose come la normal map, anche se non c'è stato tempo per un'animazione su questo nemico.
Quando il nemico viene colpito mortalmente dopo un tot di colpi alla fine viene istanziato al suo posto un altro prefab, sempre del nemico, ma senza script, e con aggiunto un rigidbody con gravità e un particellare del sangue, come se fosse effettivamente colpito a morte.

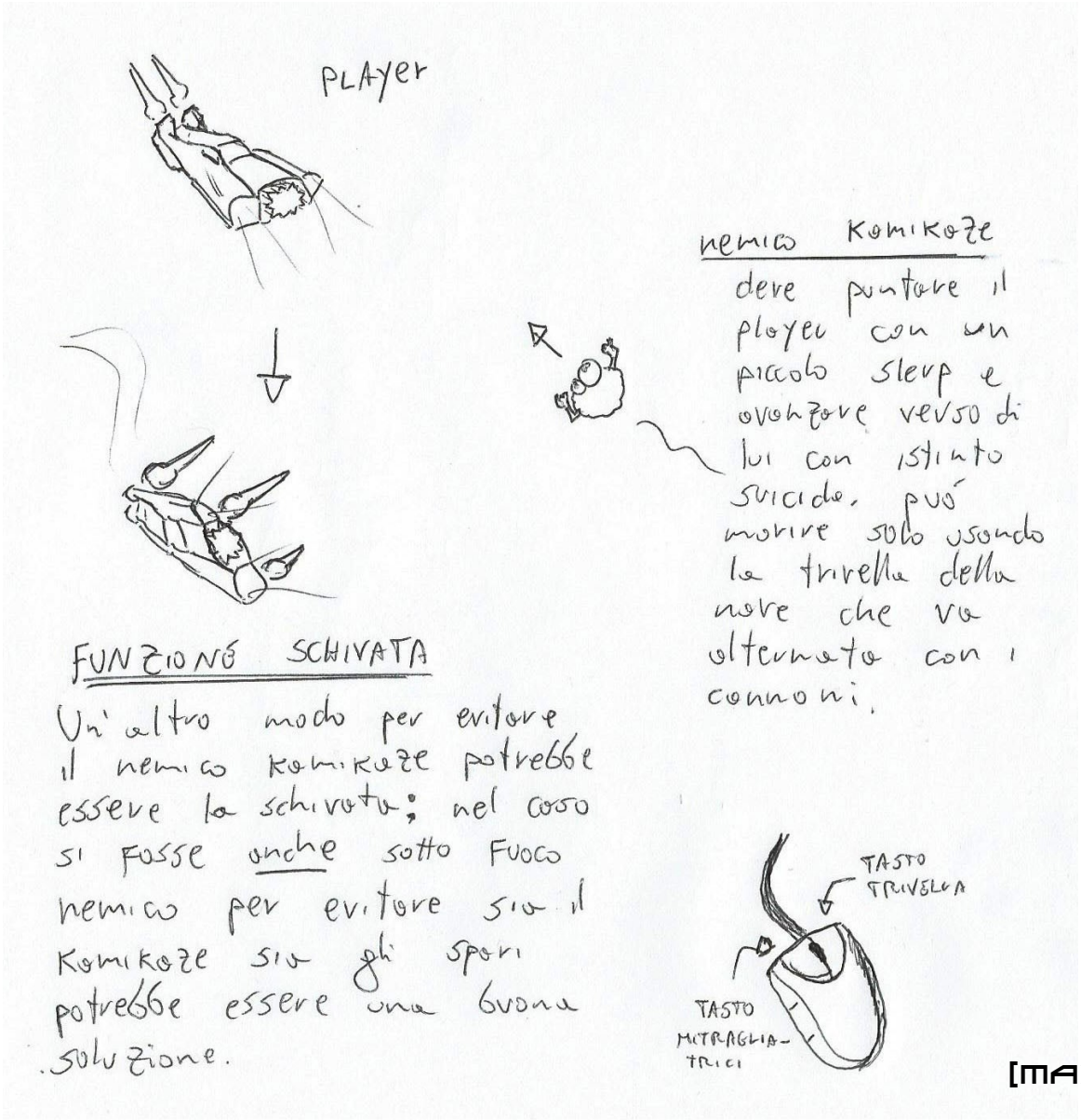






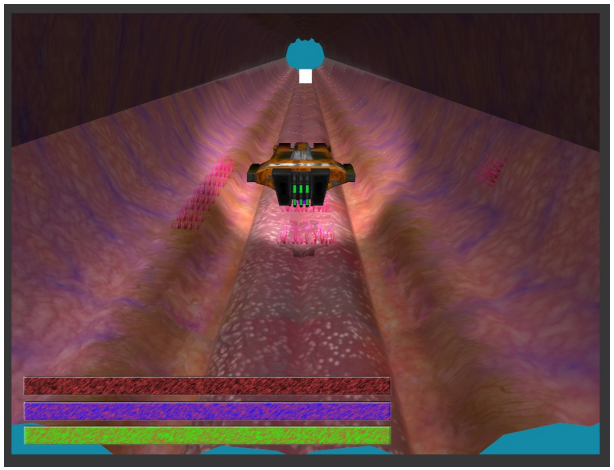
NEMICO KAMIKAZE

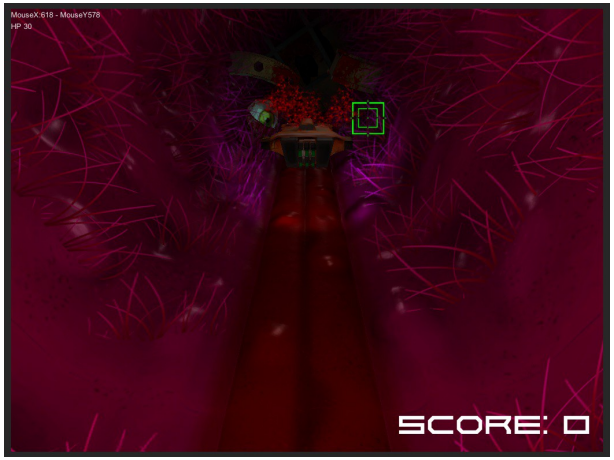


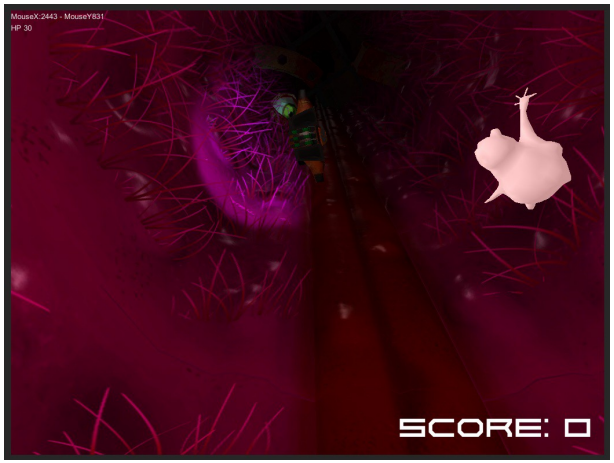


[MAX]

Per il secondo nemico eravamo partiti con l'idea di dover usare la trivella per farsi strada, così abbiamo pensato ad un nemico che si sarebbe schiantato contro il giocatore, causando dei danni a meno che non avesse usato la trivella.
Solo dopo, giocandoci, ci siamo resi conto che era troppo facile per il giocatore usare costantemente la trivella, senza così subire danni, così abbiamo pensato ad un paio di soluzioni: dare alla trivella un tempo di ricarica, come una barra di surriscaldamento oppure alternarla alle mitragliatrici. Abbiamo preso la seconda strada perché non piaceva molto l'idea di una barra a schermo quando già la visibilità era ridotta dalla nave stessa essendo il gioco in terza persona.
Successivamente abbiamo inserito la possibilità di poter schivare il nemico con una rapida sterzata, permettendo così al giocatore di poter continuare a sparare nel caso ci fossero anche dei nemici al quale dover rispondere al fuoco.







TORRETTA

TORRETTA

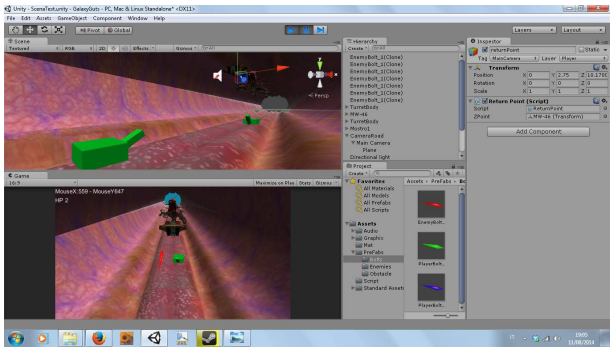
La torretta è pensata come un nemico con posizione fissa che si muove girando su se stesso nella direzione del giocatore appena lo vede in lontananza. I cannoni, figli della BASE si dovranno muovere solo sull'asse X, mentre la base solo sull'asse Y.

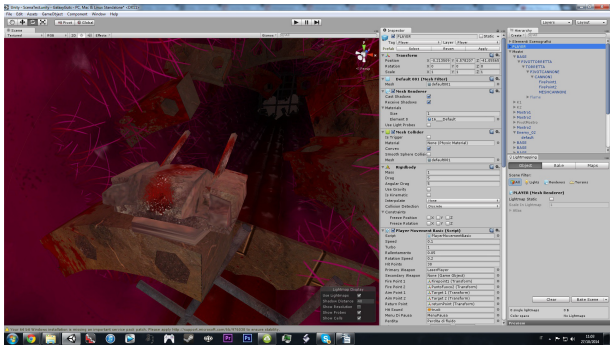
Come per i nemici che stanno davanti al giocatore ad una posizione fissa ci dovrà essere uno SLEEP nel movimento per permettere al giocatore di poter schivare i colpi.

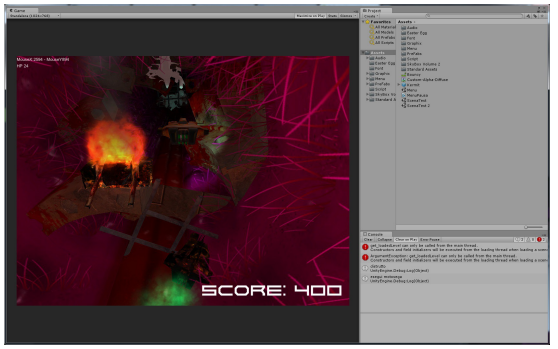
Una volta che la torretta è stata distrutta basta dare una direzione o ai cannoni per farli abbassare, con maggiori distanziamento di un particellare, come fuoco o scheggie.

[MAX]

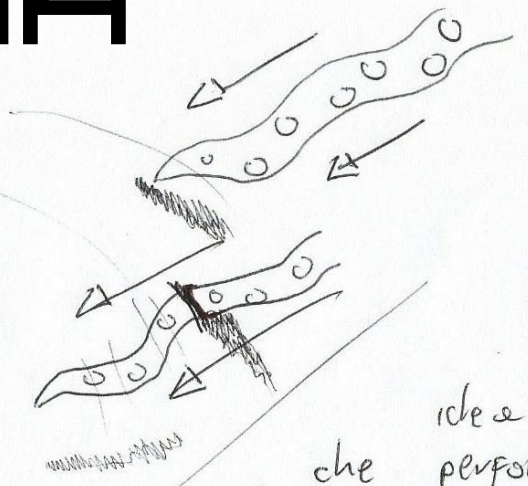
La torretta, è stata inserita dopo aver visto che serviva almeno una tipologia di nemico statico, presente sempre negli stessi punti del livello di gioco, per dare al giocatore qualche riferimento.
E' stata pensata divisa in due parti: la base e i cannoni. L'idea è che la base si deve muovere su se stessa orizzontalmente, mentre i cannoni rimangono figli della base (muovendosi quindi insieme a lei), e nel mentre si alzano indipendentemente su un asse verticale, tutto seguendo il giocatore con uno slerp, sia per dare alla torretta una dinamica più realistica, sia per permettere al giocatore di schivare i colpi con più facilità.
Dopo che la torretta viene distrutta ricevendo un tot di colpi abbiamo pensato di non sostituirla con un altro prefab come il primo tipo di nemico, ma semplicemente di cambiargli la texture con una arrostita, inserire un particellare di fuoco, disattivare il rateo di fuoco e dare una nuova direzione fissa da guardare, che è la stessa, ma con i cannoni che devono guardare verso il basso, come se fossero danneggiati e di conseguenza non ci sia più niente ad alimentarli.





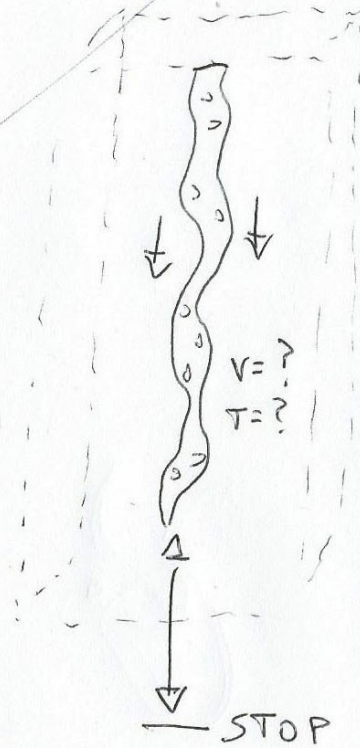


COLONNA



COLONNE / TENTACOLI A SCATTO

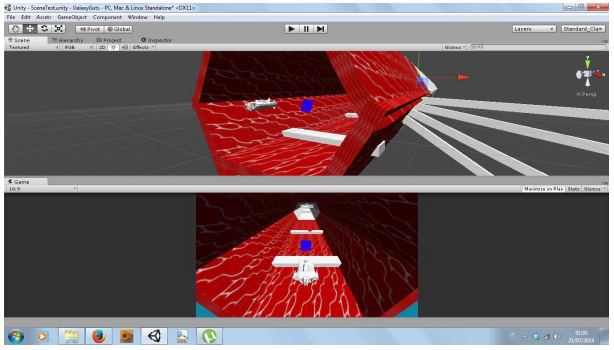
Per dare noia al player e per aggiungere una maggior fluidità al gameplay potrebbe essere una buona idea aggiungere degli ostacoli che perforano l'area di gioco con l'intento di ferire il giocatore.

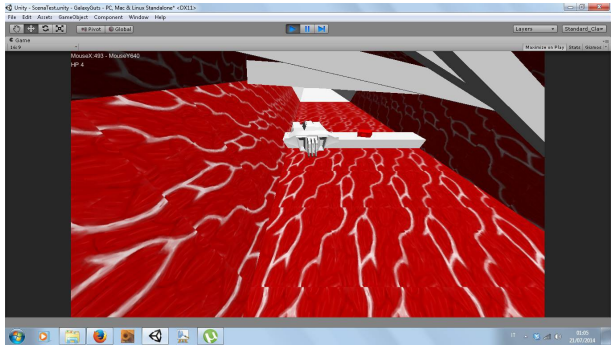


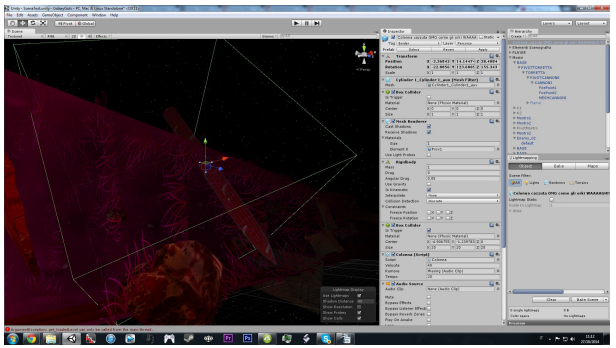
Per farle muovere potrei aggiungere un TRIGGER COLLIDER, che a contatto con il giocatore avanza ~~per~~ AD una velocità stabilita per un tratto definito da editor.
Il codice dovrà quindi usare (1) una variabile tempo per la velocità di movimento e una variabile tempo per dare una fine allo script (T) prima che l'oggetto del tentacolo finisca dall'altra parte del gioco.

[MAX E MARCO]

La colonna è nata come l'idea di una trappola da inserire nel gioco, che aggiungesse quell'elemento "terrificante" che serve in un gioco con un minimo di tensione. Infatti appena il giocatore invade l'area a rischio, definita con un collider trigger, la colonna farà uno scatto in avanti emettendo un boato, che se il giocatore ha le casse accese un gridolino lo può anche fare.
Per rendere le colonne diverse fra loro ho introdotto due variabili, una relativa alla velocità con la quale si sposta la colonna, una relativa al tempo per la quale si deve spostare. Modificando queste due variabili è possibile far muovere qualsiasi colonna per un determinato tratto di spazio in una qualsiasi velocità.
Se il giocatore non riesce a scansare una di queste colonne viene ovviamente danneggiato, ma non ha modo, neanche con la trivella di poterci passare attraverso, anche se inizialmente era stata pensata una cosa del genere.



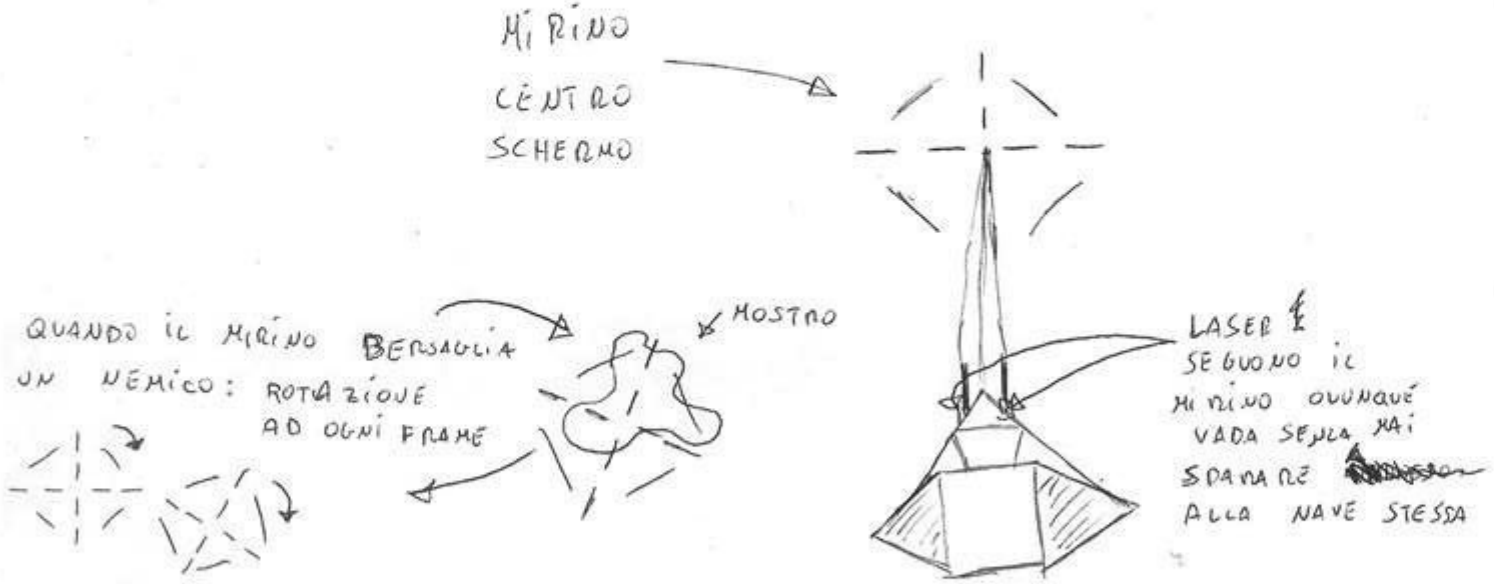




MOVIMENTO MIRINO

IL MIRINO SI SPOSTA MEDIANTE IL MOVIMENTO DEL MOUSE (POSSIBILE CAMBIO COLORE, E POSIZIONE)

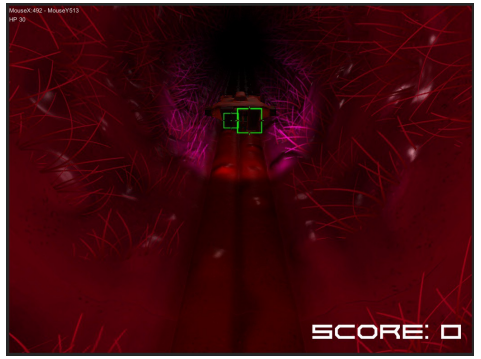
COMANDO CAM. SCREENTO WORLDDOINT PER LA TRASFORMAZIONE DELLE CORDINATE DEL MOUSE IN CORDINATE 3D WORLD



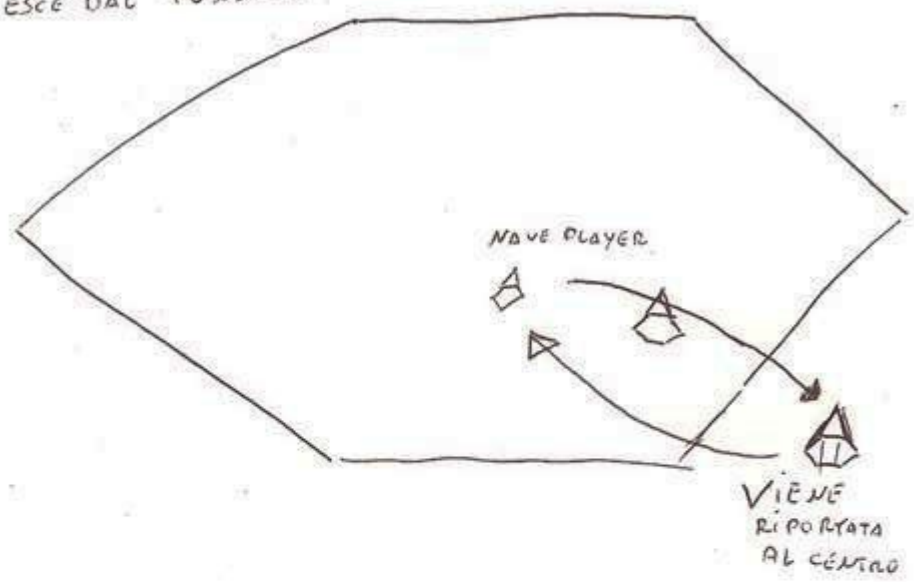
IL CAMBIO COLORE, E ROTAZIONE SI APPLICA TRAMITE UN RAYCAST

[MARCO]

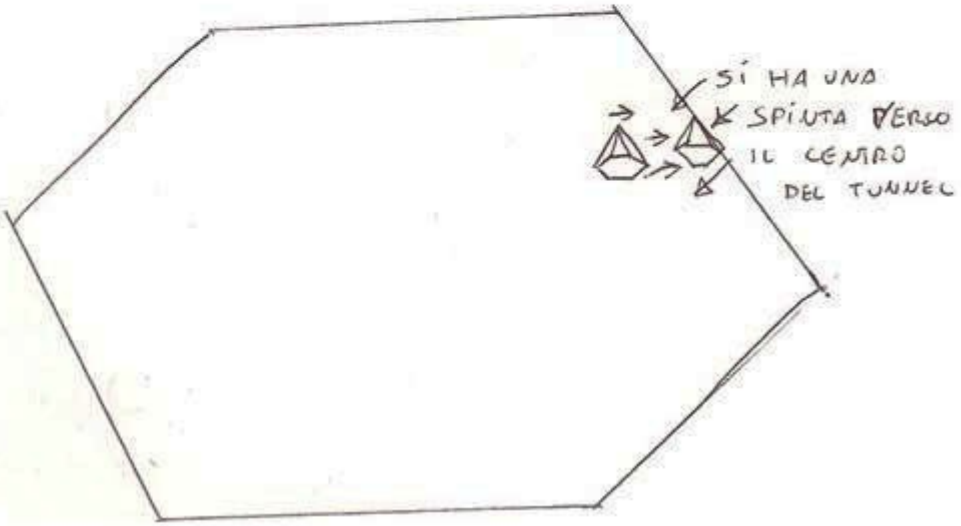
Per realizzare il mirino siamo partiti con due idee... La prima era che la nave avesse dei cannoni fissi che sparassero solo davanti a se e che per direzionare lo sparo si dovesse direzionare la nave, ma sembrava troppo un approccio da gioco 2D; per cui abbiamo pensato di creare un vero e proprio mirino, gestibile tramite il movimento del mouse e che i cannoni della nostra nave puntassero sempre verso quel mirino.
Abbiamo poi riscontrato alcuni problemi di giocabilità, quindi abbiamo inserito una funzione sul mirino che ti permette di riconoscere nemici, alleati o punti critici del livello cambiando colore: rispettivamente, Rosso per un bersaglio nemico, Giallo nella possibilità di punti critici o strategici del gioco (come pareti da abbattere) e verde se non inquadra niente.
Per realizzare ciò abbiamo utilizzato un Raycast che tramite la lettura della tag sull'oggetto colliso, seleziona il colore del mirino.
Abbiamo anche aggiunto piccole animazioni, alcune sul cambio di colore per aggiungere quel tocco dinamico al gioco che non guasta mai (come rotazioni del mirino o ridimensionamenti), una aggiungendo un figlio al mirino, che seguisse il padre con un piccolo slerp, in modo che dia l'effetto dell'effettivo ritardo nel posizionamento dei cannoni.



1- ESCE DAL TUNNEL



2- RIMBALZA SUL BORDO



SISTEMA DI PREVENZIONE DEI BUG

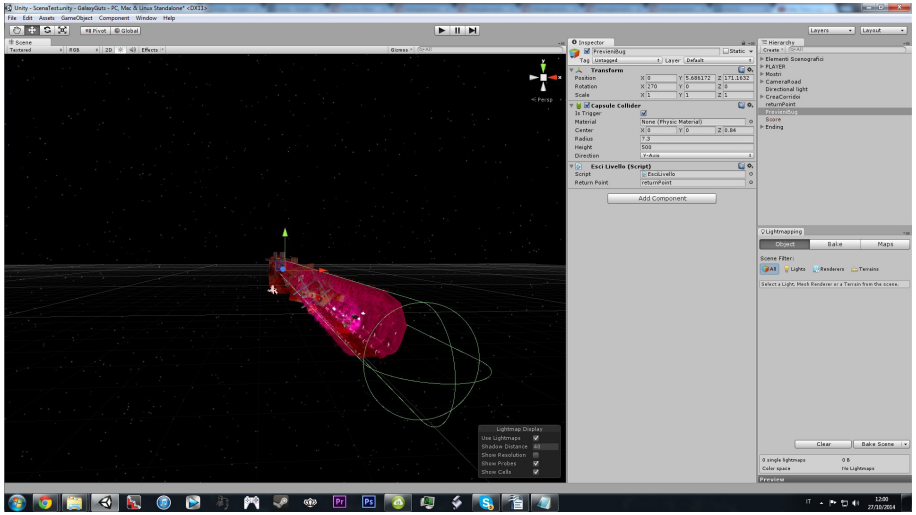
1 - SE LA NAVE, ACCIDENTALMENTE, FINISCE FUORI TUNNEL, AUTOMATICAMENTE VIENE RIPORTATA AL CENTRO

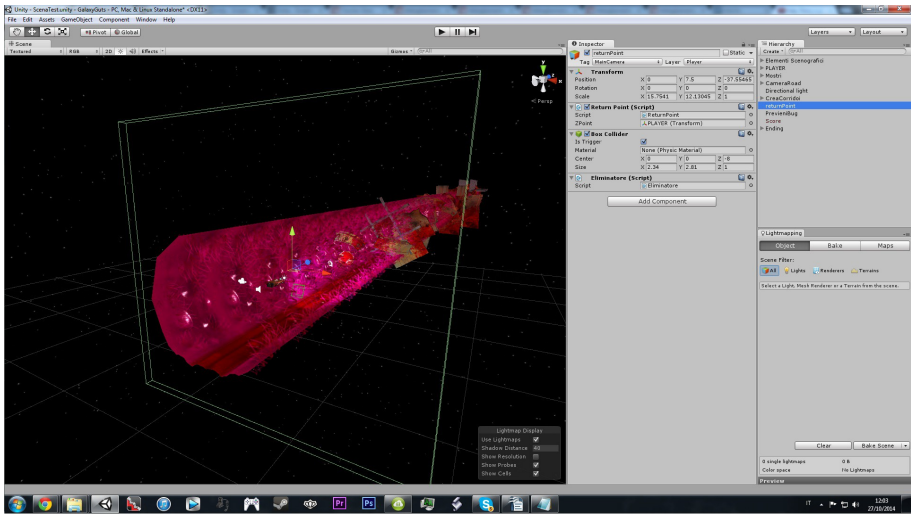
2 - SE LA NAVE SI SCONTRA CON UN BORDO VIENE RESPINTA VERSO IL CENTRO TUNNEL

GESTIONE AREA DI GIOCO E BUG

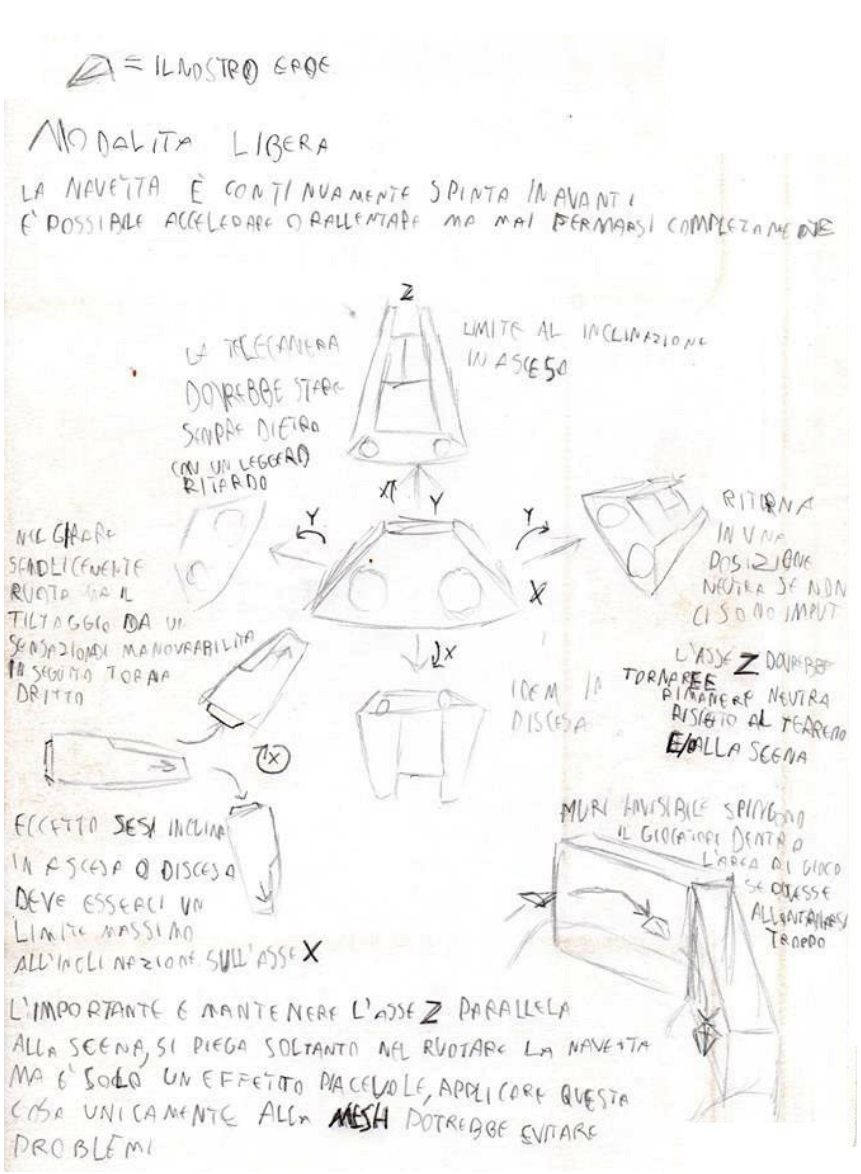
[MAX, MARCO E CLAUDIO]

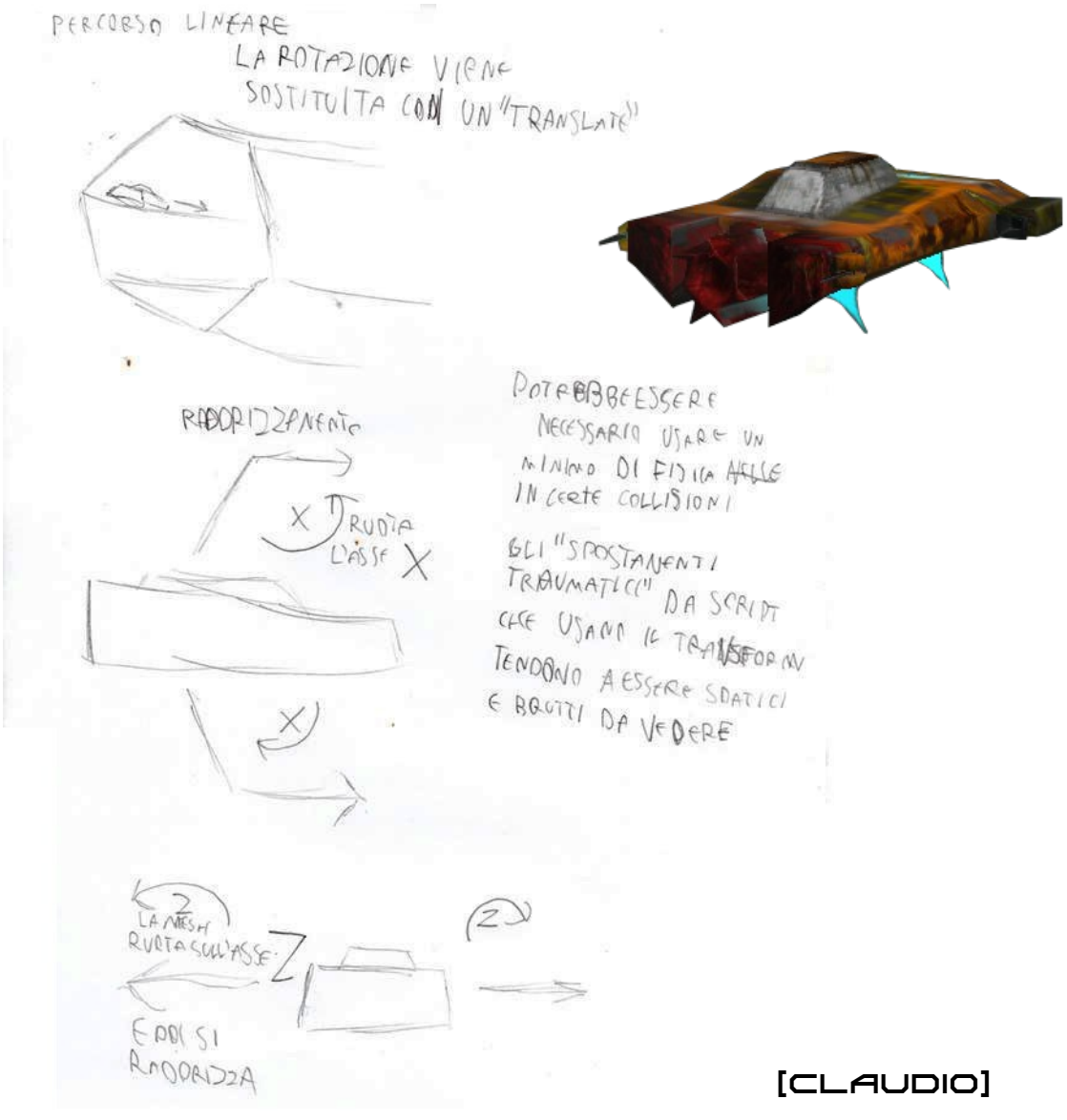
Per evitare che il giocatore potesse uscire fuori dall'area di gioco abbiamo pensato di inserire uno script sulla nave del player che la facesse rimbalzare ogni volta che essa collideva su una parete verso un "return point", un oggetto vuoto al centro del tunnel, che segue il giocatore solo sull'asse z.
Nel caso il giocatore riuscisse per qualche bug ad uscire dall'area di gioco c'è comunque un ulteriore script associato ad un gameobject con un collider trigger che comprende tutto il livello giocabile, che riporta il giocatore nuovamente al return point sulle sue coordinate x e y quando esce da questo trigger, mantenendo la distanza raggiunta nel livello dal player (z).
Per evitare che i nemici continuino a far fuoco dopo che sono stati superati, o che semplicemente tornino all'attacco, insieme al return point è stato inserito uno script che elimina qualsiasi nemico venga colliso dal suo trigger, che si trova subito dietro il giocatore.





MOVIMENTO PLAYER





[CLAUDIO]

Durante l'ideazione del progetto, pensammo a due modalità di movimento, entrambe ispirate a giochi spaziali come StarFox o Star Wars: Rogue Squadron, una libera, in stile open world e una su un percorso che gli facesse da vera e propria rotaia. Scegliemmo la seconda, anche dopo una prova con la prima, che ci sembrava troppo dispersiva.
Decidemmo di non usare la fisica come mezzo principale di movimento, in quanto tendeva a essere difficile da controllare e non adatta alla stile tendente all'arcade che avevamo in mente.
Una volta che decidemmo la il movimento su rotaia, il passaggio al movimento libero a quello a rotaia fu molto semplice, in quanto sostanzialmente si trattava di cambiare un'unica linea di codice, il movimento libero presente sul prototipo usa lo stesso codice, seppure molto semplificato. In seguito, nell'aggiungere il tunnel che funge da gola del mostro e i vari ostacoli il codice è andato complessandosi diventando quello che è adesso.