

ALMA MATER STUDIORUM · UNIVERSITÀ DI  
BOLOGNA

---

**Corso di Laurea Magistrale in  
Ingegneria Informatica**

# **2D Animazione & Gameplay**

## **LAB 2**

Fondamenti di Computer Graphics M

**Cristian Davide Conte 0001034932**

**Anno Accademico: 2022-2023**

# Contents

<b>1</b>	<b>Gameplay loop</b>	<b>2</b>
<b>2</b>	<b>Animazioni e collisioni</b>	<b>3</b>

# Chapter 1

## Gameplay loop

Per questa esercitazione si è creato un semplice gioco OpenGL 2D: Highway Surfer. Il gameplay si ispira al famoso gioco mobile "Subway Surfers" e consiste nel dover muovere una sfera schivando eventuali ostacoli presenti sul percorso. Al giocatore è concesso cambiare corsia e saltare, eventualmente anche combinando i movimenti (il gioco provvederà ad interpolare le animazioni necessarie a soddisfare l'input dell'utente). Dato che il gioco diventerà gradualmente più difficile (la velocità con cui gli ostacoli vengono generati aumenta col passare del tempo), maggiore sarà il tempo per cui si riuscirà a sopravvivere, maggiore sarà il punteggio finale: il gioco termina non appena si urta un ostacolo mostrando un pop-up contenente il punteggio raggiunto dell'utente.

Per spostarsi nella corsia a sinistra rispetto alla posizione corrente del giocatore, bisognerà premere il tasto "a".

Per spostarsi nella corsia a destra rispetto alla posizione corrente del giocatore, bisognerà premere il tasto "d".

Per saltare occorrerà premere la barra spaziatrice.

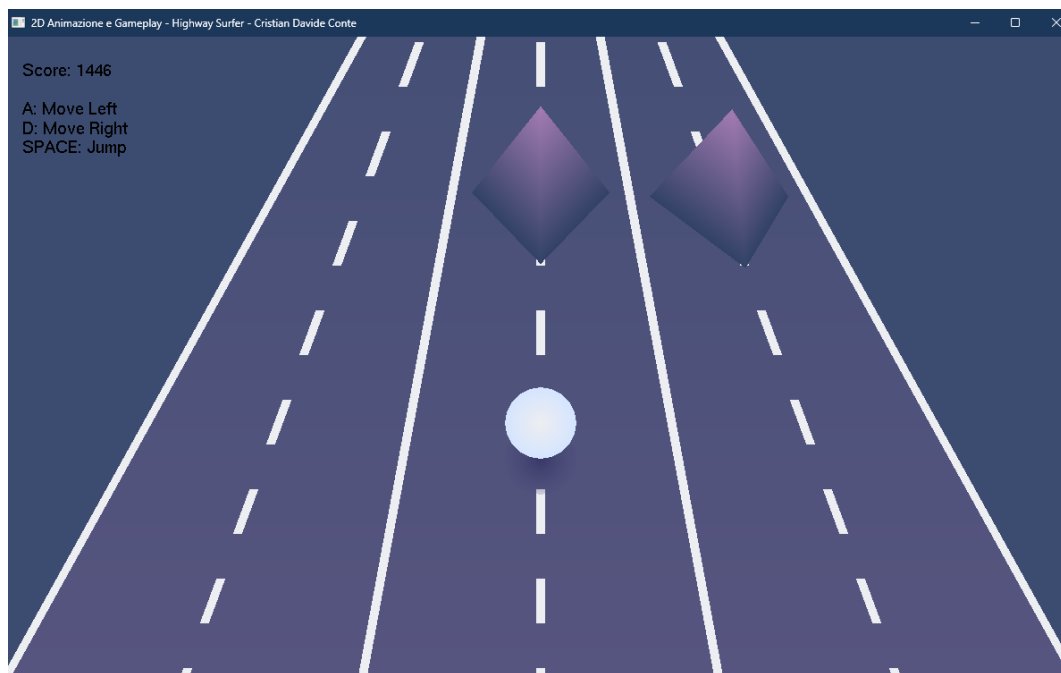


Figure 1.1: Highway Surfers - Gameplay

## Chapter 2

### Animazioni e collisioni

Al fine di rendere più piacevoli i comandi di gioco si è pensato di utilizzare delle funzioni di easing per realizzare le animazioni della sfera controllata dal giocatore. In particolare il movimento orizzontale del giocatore è controllato da un pattern ease-out-back mentre quello verticale (jump) da un pattern ease-out-bounce che permette alla sfera di dare l'illusione di rimbalzare contro l'asfalto dell'autostrada. Queste funzioni non sono altro che interpolazioni lineari tra la posizione corrente e quella finale della sfera utilizzando un peso che segue pattern non lineari (ad es. il pattern ease-out è dato da  $1 - (1 - t)^3$  dove  $t$  è il tempo trascorso dall'inizio dell'animazione rispetto alla durata totale dell'animazione). Per semplicità le easing function lavorano in  $[0..1]$  sia per il dominio che per il codominio.

Tutte le funzioni di easing utilizzate sono disponibili a questo link: <https://easings.net/>.

Per quel che riguarda le collisioni invece, queste sono realizzate analizzando ogni singolo punto sulla circonferenza della sfera (i vertici delle basi dei triangoli che la compongono) e confrontandoli con i triangoli che compongono gli ostacoli (i.e. gli ostacoli sono formati da 2 triangoli aventi un edge in comune). In particolare la funzione "sign" si occupa di controllare se il punto passato in input sta nel semipiano destro o sinistro di una retta passante per due punti. Dato che ogni triangolo individua 3 rette, basta osservare i segni dei risultati della funzione "sign" per capire se il giocatore ha colpito o meno un ostacolo. Per semplicità, quando il giocatore sta saltando gli viene garantito un breve periodo di invulnerabilità (nessuna collisione possibile) durante il quale però non gli sarà concesso di saltare nuovamente (una specie di cooldown per evitare che il gioco si rompa premendo ripetutamente la barra spaziatrice).

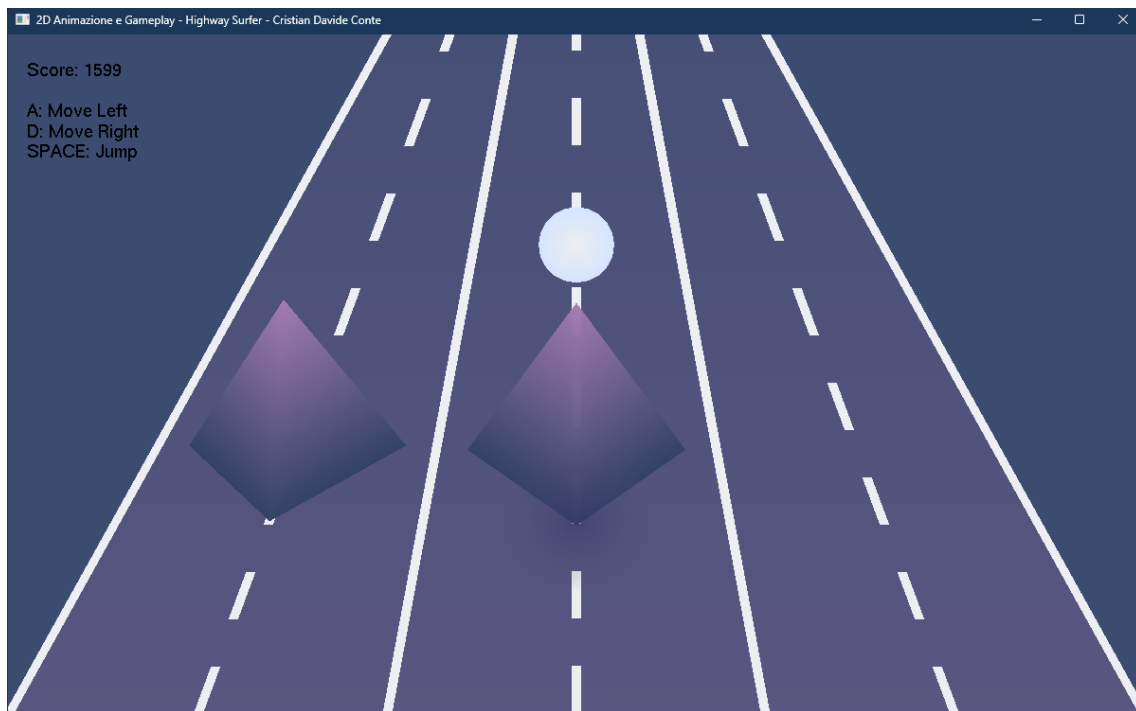


Figure 2.1: Highway Surfers - Jump, no collision

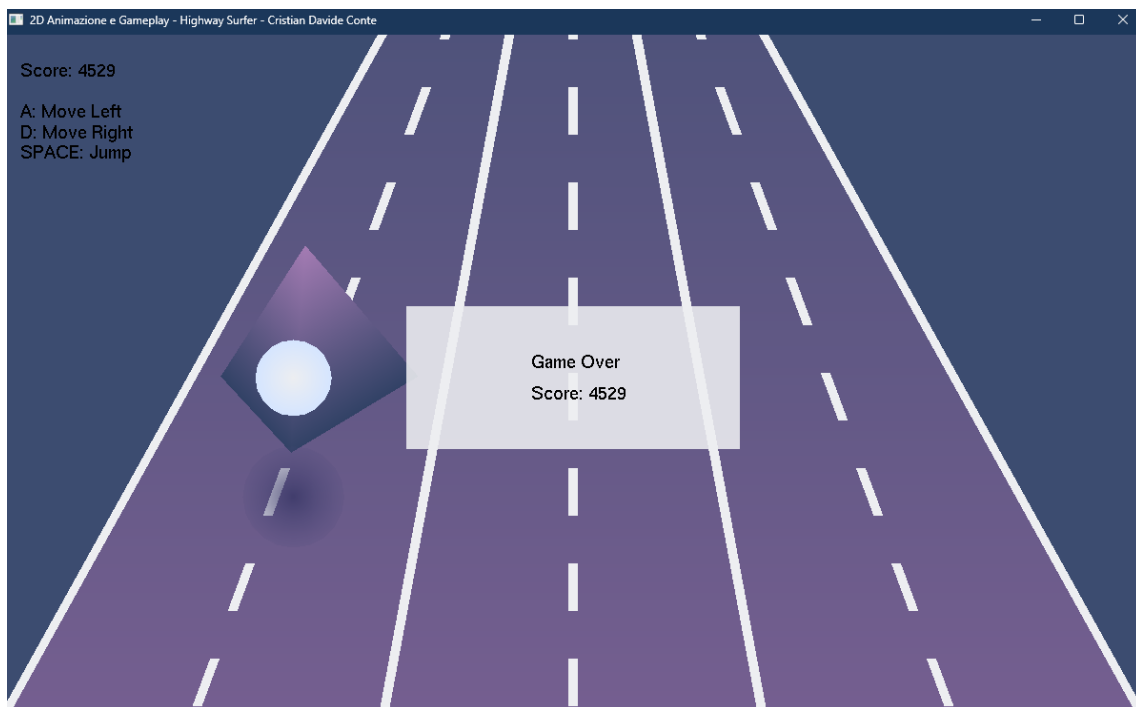


Figure 2.2: Highway Surfers - Collision, Game Over