PROGETTO DI SISTEMI

AZZOLIN EDOARDO

ANALISI DEL PROGETTO

FORMULAZIONE DELL'IDEA INIZIALE:

- L'obiettivo era ricreare, in alcune delle sue parti, il videogioco «Super Mario»

COME MI SONO INFORMATO:

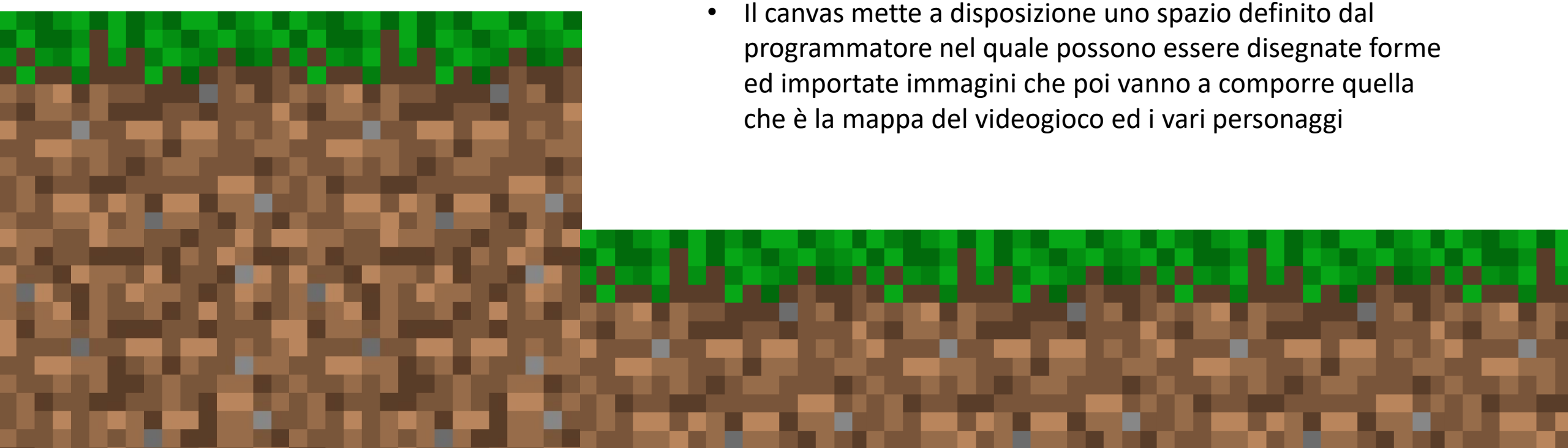
- Seguendo vari video tutorial mi sono reso conto che per realizzare un progetto di tale complessità era essenziale approcciare la programmazione ad oggetti
- Senza perdere di vista l'obiettivo finale, ho tratto le varie informazioni da diverse fonti per poi unirle nel mio elaborato

OSSERVAZIONI A PROGETTO ULTIMATO:

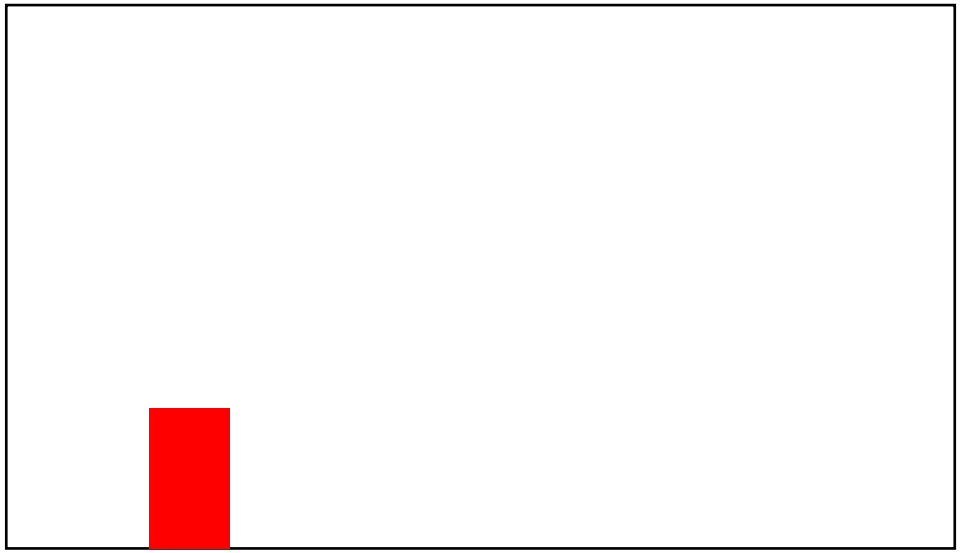
- Sono riuscito a sviluppare la mia idea iniziale con qualche piccola variazione che analizzeremo in seguito

LA **PIANIFICAZIONE**

- Il lavoro è quasi totalmente concentrato sulla parte di javascript
- Per sfruttare al meglio le funzionalità del linguaggio dedicate allo sviluppo di un videogioco, mi sono servito di una libreria (e corrispondente tag HTML) di nome canvas
- Il canvas mette a disposizione uno spazio definito dal programmatore nel quale possono essere disegnate forme ed importate immagini che poi vanno a comporre quella che è la mappa del videogioco ed i vari personaggi

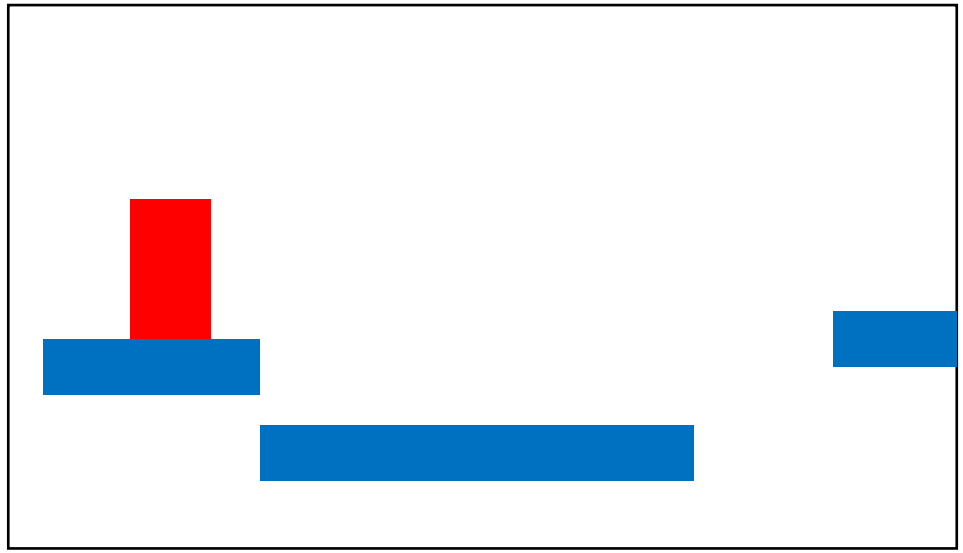


CREAZIONE DEL **PERSONAGGIO**



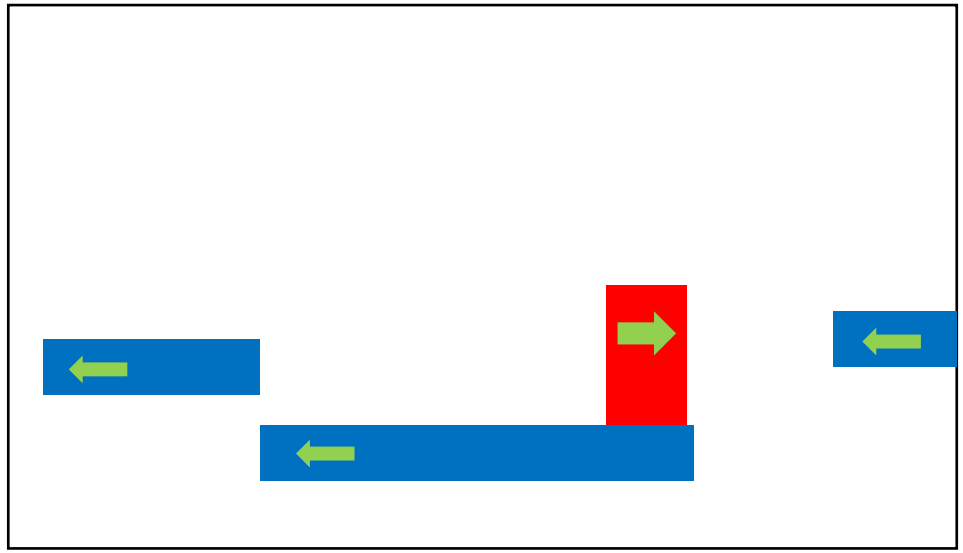
- La prima cosa che ho creato è stato il personaggio di classe Player che inizialmente è un semplice rettangolo che resta fermo a mezzaria
- Successivamente ho simulato la gravità che porta il personaggio ad accelerare la caduta fino a quando non si ferma toccando il suolo
- A questo punto ho implementato delle funzioni che mi permettono di rilevare gli input da tastiera programmandole in modo da interagire con il personaggio

CREAZIONE DELLE **PIATTAFORME**



- Proseguendo mi sono posto il problema di creare delle piattaforme
- Inizialmente ho considerato la singola piattaforma e ho fatto in modo tale che se il personaggio, cadendo, capitasse sopra di essa, si fermasse
- Successivamente, per avere a disposizione più piattaforme, ho convertito la singola variabile in un array che ne contenesse un certo numero e mi sono ingegnato per creare una pseudo-mappa, aggiungendo la condizione di reset in caso il personaggio toccasse il fondo del canvas
- A questo punto mi sono adoperato per ricercare immagini da sostituire ai rettangoli importandone la sorgente che, modificando a dovere il codice, permette di riprodurle sul canvas

ESTENSIONE DELLA **MAPPA**

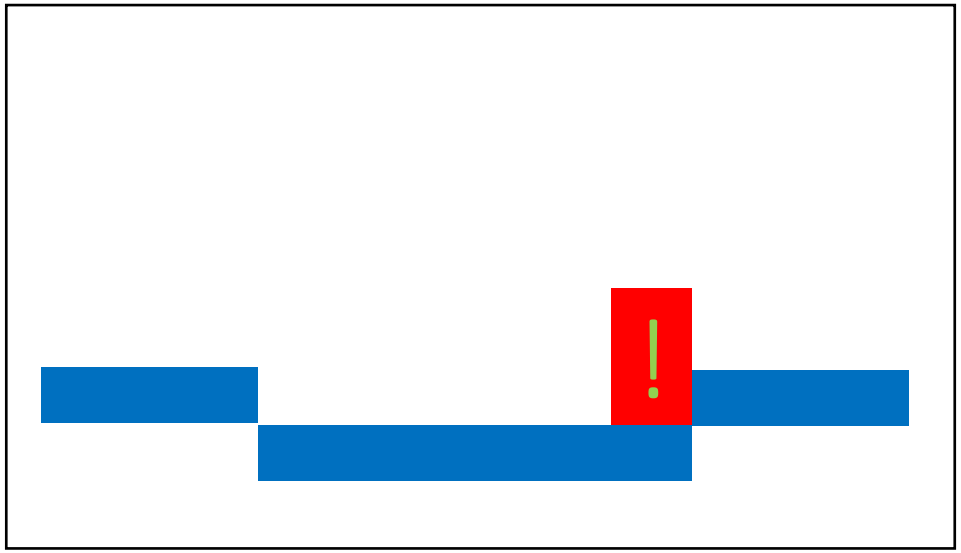


- Per fare in modo che la mappa non sia costretta nei limiti imposti dalla larghezza del canvas, ho dato origine ad un sistema che assegna al personaggio una limitata libertà di movimento in quanto, raggiunto uno dei 2 estremi impostati, il personaggio cessa di muoversi e cominciano a spostarsi tutte le piattaforme attorno ad esso

LO **SFONDO**

- Per aggiungere lo sfondo ho creato una classe apposita ed ho editato un’immagine scaricata da internet in maniera tale da darle un effetto di continuità lungo l’asse orizzontale
- Successivamente ho trovato un modo per rendere lo scorrimento delle immagini più realistico agendo sulla velocità di movimento
- In sostanza, le piattaforme si muovo alla stessa velocità del personaggio, mentre lo sfondo si sposta più lentamente simulando un effetto di parallasse

COLLISION DETECTION COMPLETO

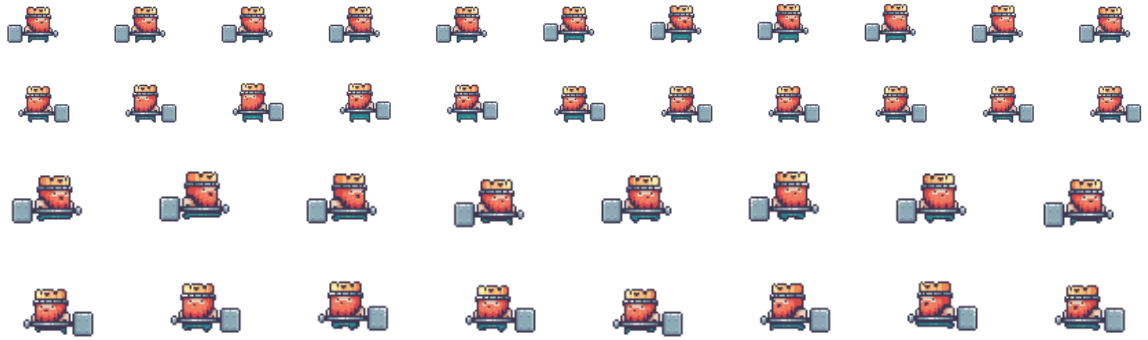


- A questo punto dello sviluppo, il personaggio era in grado di rilevare solamente le collisioni provenienti dal basso
- Di conseguenza ho creato delle funzioni che permettono di verificare se il personaggio entra in contatto con una piattaforma, ed in caso affermativo su che lato è avvenuta la collisione, e di verificare se ha impattato con un ostacolo (che vedremo nella slide successiva). Anche il caso in cui un ostacolo entra in contatto con una piattaforma è stato implementato

GLI **OSTACOLI**

- Per rendere il gioco più complicato e far sì che assomigli di più all'idea iniziale di Super Mario, ho inserito qualche ostacolo lungo il percorso
- Gli ostacoli inseriti sono quelli rappresentati in figura
- La mazza chiodata si muove in una direzione fino a che non entra in contatto con una piattaforma. Dopodiché inverte il proprio senso di marcia.
- Il pugnale, invece, cade dall'alto ed una volte che raggiunge la base del canvas si riporta alla sua posizione iniziale e ricomincia la caduta
- Anche per gli ostacoli è stato necessario adattare la posizione al cambiamento di quella delle piattaforme a seconda che il personaggio prosegua lungo la mappa o faccia marcia indietro

ANIMAZIONI DEL PERSONAGGIO



- Per poter aggiungere animazioni al personaggio ho utilizzato diverse immagini (ognuna che corrisponde ad una azione) che a loro volta sono composte da svariate figure. Questo per far sì che con un'unica immagine fisica si possa già ottenere un'animazione.
- L'animazione viene creata sovrapponendo di volta in volta il ritaglio di immagine successivo, fino all'ultimo ritaglio d'immagine per poi ripartire dalla prima, creando così un'illusione di movimento
- Per passare da un'animazione ad un'altra (es. dallo stare fermo sul posto al correre verso destra) ho agito sul rilevatore di input da tastiera per far sì che quando viene premuto un tasto e quando lo si rilascia il personaggio si comporti di conseguenza

CONCLUSIONI

- Il progetto è stato abbastanza complesso da realizzare, soprattutto nella parte iniziale mi sono trovato in difficoltà nell'apprendere il funzionamento degli elementi della programmazione ad oggetti che non ci era ancora stata introdotta
- Il risultato ottenuto può essere considerato un esempio di layout in quanto il codice è scritto in maniera tale che, modificando qualche componente o aggiungendo piattaforme per espandere la mappa, non venga stravolto tutto il programma ma, invece, le nuove modifiche verranno integrate senza dover rivalutare il codice daccapo