Documentazione di CovidDiffusionSimulator

1 Introduzione 4

1.1 Informazioni sul progetto 4

1.2 Abstract 4

1.3 Scopo 4

2 Analisi 5

2.1 Analisi del dominio 5

2.2 Analisi e specifica dei requisiti 5

Use case 6

2.3 Pianificazione 7

2.4 Analisi dei mezzi 8

2.4.1 Software 8

2.4.2 Hardware 8

3 Progettazione 9

3.1 Design delle interfacce 9

3.2 Design Procedurale 11

4 Implementazione 12

4.1 Inizio 12

4.2 Creazione della classe Human 12

4.2.1 Costruttore 12

4.2.2 set\_infected 12

4.2.3 unsetInfected 12

4.2.4 setGravlyIll 13

4.2.5 unsetGravlyIll 13

4.2.6 setDead 13

4.2.7 Casual\_move 13

4.2.8 changeDirection 13

4.2.9 checkX e checkY 13

4.2.10 print 14

4.2.11 tryInfect 14

4.2.12 tryChangeInfectedState 14

4.3 File script.js 14

4.3.1 createHuman 14

4.3.2 startMovementInterval 15

4.3.3 pauseMovementInterval 15

4.3.4 stop 15

4.3.5 moveEveryOne 15

4.3.6 readInput 15

4.3.7 checkSliderValue 16

4.3.8 printSliderValue 16

4.3.9 implementCheckBoxes 16

4.3.10 infectionControl 16

4.3.11 menuIconOpen 16

4.3.12 menuIconClose 16

4.3.13 openMenu 17

4.3.14 menuSelectParametriSimulazione 17

4.3.15 menuSelectStatistiche 17

4.3.16 menuSelectHome 17

4.3.17 readState 17

4.3.18 makeStatistics 17

4.3.19 SetInfectionPercentageDefault 18

4.3.20 setVelocityMovement 18

4.3.21 setChangeStatePercentageDefault 18

4.4 Index.html 18

5 Test 18

5.1 Protocollo di test 18

5.2 Risultati test 20

5.3 Mancanze/limitazioni conosciute 20

6 Consuntivo 21

7 Conclusioni 22

7.1 Sviluppi futuri 22

7.2 Considerazioni personali 22

8 Bibliografia 22

8.1 Bibliografia per articoli di riviste: 22

8.2 Bibliografia per libri 22

8.3 Sitografia 23

9 Allegati 23

# Introduzione

## Informazioni sul progetto

Sono un allievo della SAMT della classe I3AA, Il progetto è stato supervisionato dal docente Geo Petrini e il mandante è il docente Luca Muggiasca.

Il progetto è iniziato il 9.9.2021 ed è finito il 23.12.2021.

## Abstract

* **Background/Situazione** **iniziale** Questo progetto è volto all’intrattenimento e vuole anche fare una simulazione pressoché verosimile dell’evoluzione dei contagi all’interno di una popolazione. Lo scopo di questo progetto è creare un simulatore di una popolazione la quale entra in contatto con il covid-19.  
  Bisognerà poter scegliere le quantità di persone nella popolazione e quanti di essi hanno che protezione o sono infetti.  
  Bisognerà poter cambiare diversi argomenti per modificare il corso dell’infezione generale.  
  L’utente potrà facilmente capire la simulazione grazie a delle statistiche in tempo reale comprendenti un grafico.
* **Approccio**: Non essendoci richieste particolari riguardo al linguaggio da utilizzare ho sviluppato un’applicazione web utilizzando html, java script e css.
* **Risultati**: Sono riuscito a rispettare tutti i requisiti e sono abbastanza soddisfatto del mio lavoro.

## Scopo

Questo progetto è volto all’intrattenimento e vuole anche fare una simulazione pressoché verosimile dell’evoluzione dei contagi all’interno di una popolazione. Lo scopo di questo progetto è creare un simulatore di una popolazione la quale entra in contatto con il covid-19.

Bisognerà poter scegliere le quantità di persone nella popolazione e quanti di essi hanno che protezione o sono infetti.

Bisognerà poter cambiare diversi parametri per modificare il corso dell’infezione generale.

L’utente potrà facilmente capire la simulazione grazie a delle statistiche in tempo reale comprendenti un grafico.

# Analisi

## Analisi del dominio

Questo progetto parte dalla situazione di pandemia globale attuale, serve a creare un’idea a chiunque del potenziale di esposizione al virus.

Questa applicazione sarà disponibile a tutti non appena sarà pubblicato, sarà accessibile senza bisogno di login ma non si potranno salvare i parametri per una prossima simulazione.

Il programma è quasi interamente parametrizzato in maniera da migliorare l’esperienza dell’utente.

## Analisi e specifica dei requisiti

|  |  |  |
| --- | --- | --- |
| ID | Requisiti | Priorità |
| 1 | Avere uno spazio in cui far girare le “persone” | Alta |
| 2 | Avere un numero scelto di persone rappresentate da dei cerchi | Media |
| 3 | Dare la possibilità all’utente di cambiare le percentuali di trasmissione con le varie protezioni | Media |
| 4 | Avere un determinato numero di persone infette all’inizio | Alta |
| 5 | Avere un determinato numero di persone con la mascherina all’inizio | Media |
| 6 | Avere un determinato numero di persone vaccinate all’inizio | Media |
| 7 | Le percentuali di trasmissione default devono essere reali | Bassa |
| 8 | Il numero di persone viene scelto dall’utente | Media |
| 9 | Il numero di persone inizialmente con la mascherina viene scelto dall’utente | Bassa |
| 10 | Il numero di persone inizialmente vaccinate viene scelto dall’utente | Bassa |
| 11 | Il numero di persone inizialmente infette viene scelto dall’utente | Media |
| 12 | Le persone si muovono in maniera casuale all’interno dello spazio | Alta |
| 13 | Il numero di persone iniziale viene scelto tramite un text box con limitazione(10-400) | Bassa |
| 14 | Le percentuali di vaccinati, con mascherine e positivi iniziali vengono scelte dall’utente tramite degli slider | Bassa |
| 15 | L’inserimento dei vaccinati e di coloro che indossano la mascherina di default è disattivata | Bassa |
| 16 | L’utente può attivare i vaccinati e coloro che indossano la mascherina tramite dei checkbox | Bassa |
| 17 | Quando una persona è infetta si visualizza colorata di rosso | Media |
| 18 | Deve essere presente un tasto per iniziare la simulazione | Alta |
| 19 | Deve essere presente un tasto per mettere in pausa la simulazione | Bassa |
| 20 | Deve essere presente un tasto per fermare la simulazione | Bassa |
| 21 | Quando una persona è vaccinata viene visualizzata in blu mentre se ha la mascherina in verde | Bassa |
| 22 | Non è possibile ridimensionare lo spazio di simulazione | Alta |
| 23 | Non è presente la possibilità di salvataggi | Bassa |
| 24 | Verranno visualizzate le statistiche dei contagi nel tempo | Media |

## Use case

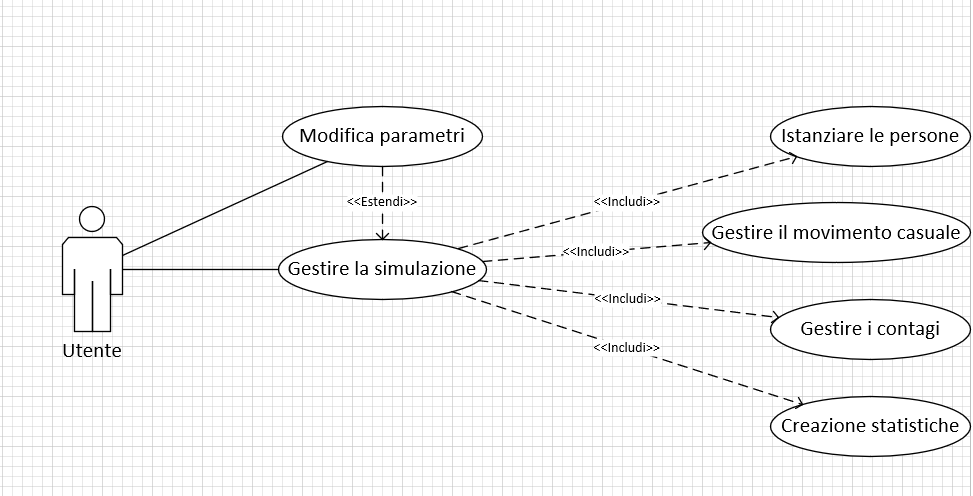


Figura Use Case

## Pianificazione

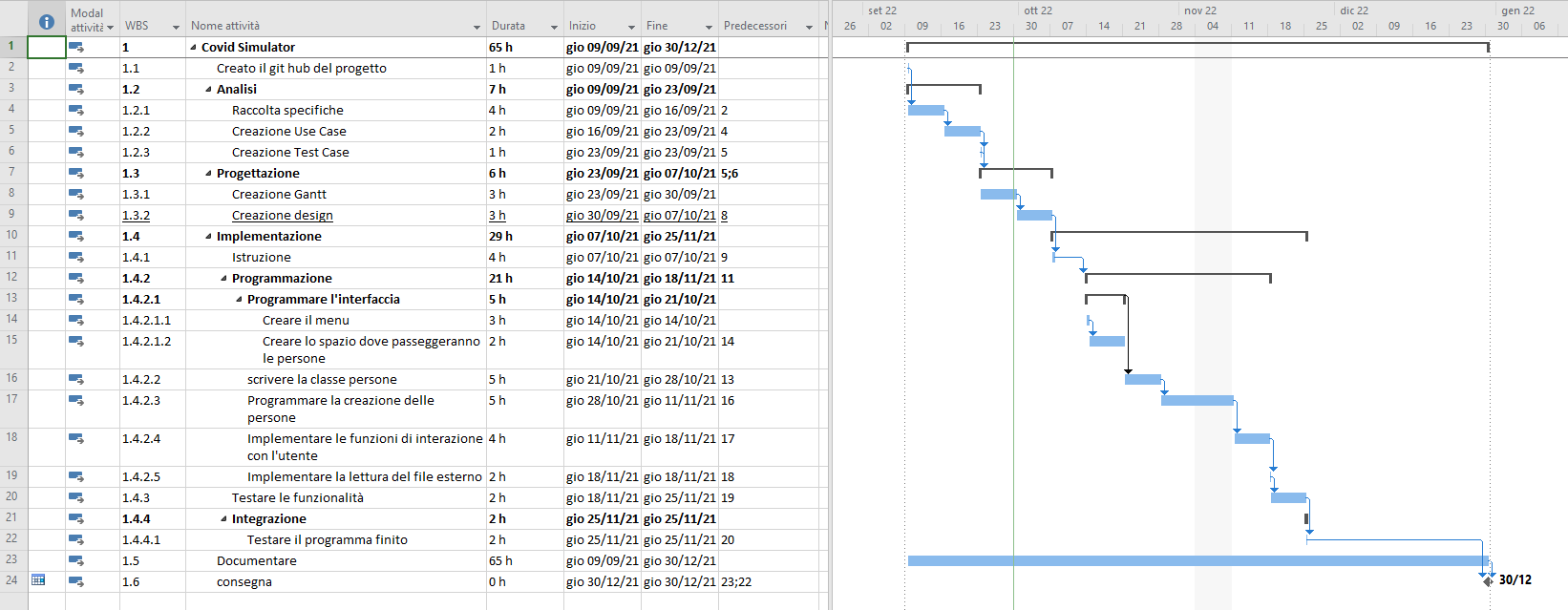
La pianificazione che ho eseguito è basata sul modello waterfall.

Figura Gannt iniziale

## Analisi dei mezzi

Per scrivere il codice necessario al funzionamento del software ho utilizzato Visual Studio Code versione 1.58.

### Software

Ho utilizzato le librerie boostrap per migliorare il css in alcuni casi ed ho usato la libreria plotly che mi permette di creare ed utilizzare il grafico necessario per raffigurare le statistiche.

### Hardware

Non è necessario nessun hardware particolare perché l’applicazione è un’applicazione web.

# Progettazione

## Design delle interfacce

Figura 4 Design Menu

Figura 3 Design Home

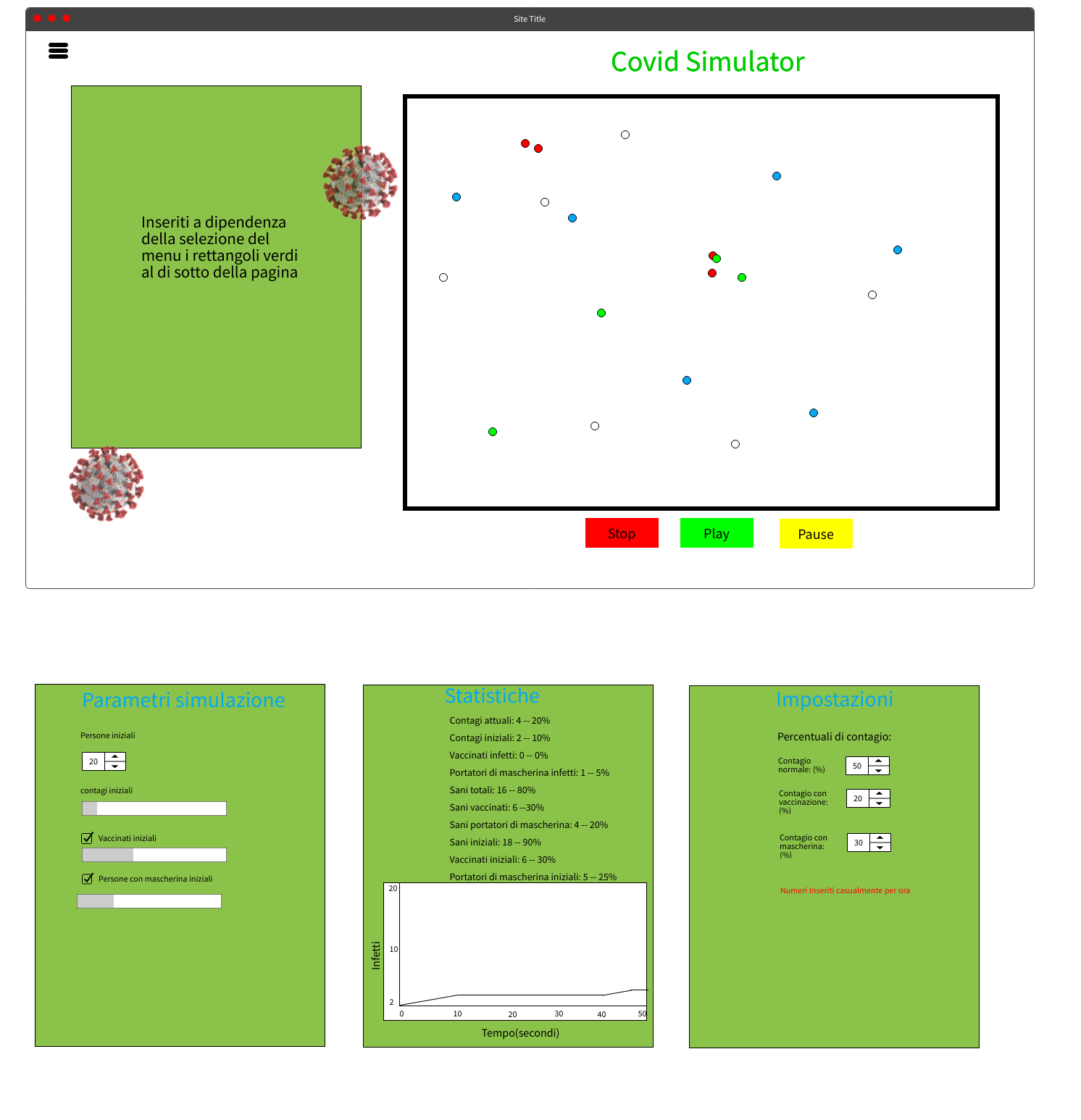


Figura 5 Design Sezioni

## Design Procedurale

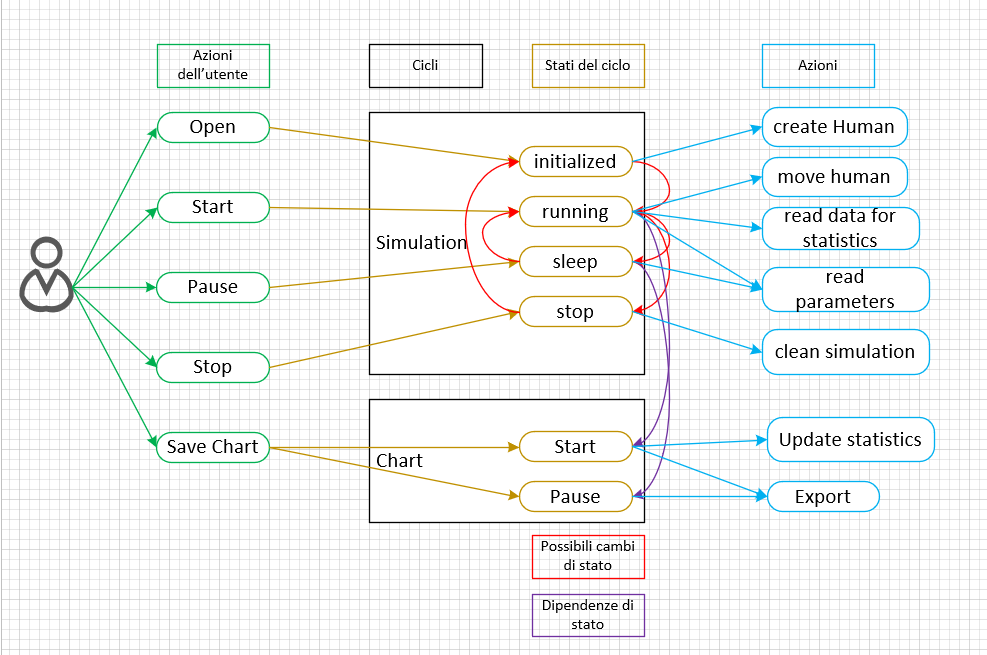


Figura 6 Design Procedurale

# Implementazione

## Inizio

Inizialmente ho creato 4 file:

1. index.html ovvero il file che conterrà il codice html.
2. style.css ovvero il file che conterrà tutto il codice css necessario alla pagina.
3. Human.js ovvero il file che conterrà la classe Human necessaria alla simulazione.
4. script.js ovvero il file che conterrà il codice javascript necessario al funzionamento della pagina.

In seguito ho cominciato a creare la base del file index.html inserendo le informazioni base.

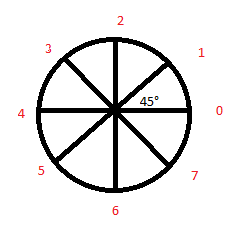
Il file css e script.js rimarranno vuoti mentre nel file della classe Human comincio a creare la classe vuota.

## Creazione della classe Human

### Costruttore

Il costruttore riceverà 3 valori booleani, ovvero se è infetto se ha la mascherina e se è vaccinato e un intero, ovvero il suo indice.

Ho inserito un controllo per vedere se si ricevono effettivamente dei valori booleani, se non è così viene creato un umano default ovvero con tutto falso.

Se i valori inseriti sono validi allora si prosegue con l’assegnare i valori passati, se l’umano è infetto sarà di colore rosso, se è vaccinato sarà di colore blu, se indossa la mascherina sarà di colore verde e se non sarà nulla di ciò verrà rappresentato in bianco.

Automaticamente verrà anche creata la posizione casuale dell’human con un attributo x e un valore y generati casualmente con massimo rispettivamente larghezza e altezza del canvas.

In seguito è stato inserito anche la generazione di una direzione iniziale, che servirà per generare uno spostamento simile alla realtà. Lo spostamento è descritto con un numero da 0 a 7 che raffigura uno spostamento in direzione 45° \* numero inserito.

Figura 7 direzione spostamento

Ulteriormente si settano gli attributi GravlyIll e Dead a false.

### set\_infected

Il metodo set\_infected è molto semplice, trasforma l’oggetto che lo invoca in infetto, ovvero mette a true l’attributo infected e assegna il colore rosso, inoltre scrive in console “infettatto” per eventuali debugging.

### unsetInfected

Questo metodo serve a far tornare gli infetti allo stadio iniziale ovvero quando erano sani, riprendendo il colore precedente e settando a false l’attributo infected.

### setGravlyIll

Questo metodo è molto semplice, serve a far diventare un infetto gravemente malato, lo fa diventare di colore viola e setta a true l’attributo GravlyIll.

### unsetGravlyIll

Questo metodo è il contrario del precedente, ovvero setta a false l’attributo GravlyIll e fa tornare di colore rosso l’Human.  
Questo perché è tornato un malato semplice e non è più a rischio di morire.

### setDead

Questo metodo richiama unsetInfected precedentemente spiegato, setta a true l’attributo dead e imposta il colore a nero, questo Human è morto, non è più infetto ma non si muove più.

### Casual\_move

Il metodo casual\_move serve a muovere in maniera casuale l’human all’interno del canvas.

Inizialmente generavo 4 numeri casuali e a dipendenza di quelli con degli xor decidevo in quale direzione muoversi di un pixel, ma ciò generava un movimento completamente irrealistico ovvero i pallini che raffigurano gli human si muovevano di veramente poco poi tornavano indietro e così via e restavano sul posto, più che delle persone che camminavano sembravano delle palline impazzite che tremolavano.

Per risolvere questo problema ho aggiunto l’attributo direction che serve per spostarlo di un pixel in quella direzione, con una certa frequenza bisogna cambiare la direzione, per fare ciò invoco il metodo changeDirection che verrà spiegato in seguito con frequenza casuale.

Dopo aver cambiato la posizione dell’human invoca il metodo repaint che sarà spiegato anche lui in seguito.

Ovviamente verranno mossi solamente gli human che non sono morti.

### changeDirection

Il metodo changeDirection se invocato cambia la direzione dell’human.

La cambia in maniera da creare una curva dolce ovvero cambia la direzione di 1 ovvero 45° in maniera casuale se aumentandola o diminuendola.

Nel metodo è anche presente un controllo per far rimanere nel range 0-7 la direzione ovvero se diventa 8 la riporta all’inizio ovvero a 0, mentre se diventa -1 la trasforma in 7.

### checkX e checkY

I metodi checkX e checkY sono pressoché uguali, cambia solo la proprietà che controllano, checkX controlla x mentre l’altro controlla y.

Controllano se la coordinata sia maggiore di 5 e minore dell’altezza o della larghezza entrambe meno 5.

5 è il raggio del cerchio che raffigura l’human e togliendolo dal massimo e aggiungendolo al minimo, gli human non potranno uscire dallo schermo nemmeno un po’.

Se l’human prova a uscire dallo schermo il metodo setta la direzione a opposta al lato dello schermo per evitare che ci riprovi subito, ad esempio se prova ad uscire a sinistra la direction verrà settata a 0 in maniera che riparta verso destra.

### print

Il metodo print serve a stampare nel campo di simulazione l’human, lo stampa sotto forma di cerchio con il proprio colore e il proprio indice.  
Prima di stampare controlla che le coordinate siano valide invocando i metodi checkX e checkY.

In seguito utilizzando i canvas disegna un arco passando coordinata x, coordinata y, raggio e l’ultimo argomento serve per chiudere l’arco e formare un cerchio.

### tryInfect

Il metodo tryInfect controlla se l’umano che invoca il metodo e l’umano che viene passato come argomento sono abbastanza vicini per provare a infettarsi (distanza minima tra centri raggio\*2).

Se sono abbastanza vicini e l’Human passato come argomento non è ancora contagiato estraggo un numero casuale, e a dipendenza della combinazione dei booleani dei due human viene dato un numero, se quello casuale è minore di quello calcolato l’human viene contagiato.

### tryChangeInfectedState

Questo metodo serve a rendere possibile agli Human di guarire come morire e diventare gravemente malati.  
Per fare ciò si estrae un numero casuale che serve per rendere appunto casuale il passaggio di stato e se l’Human è infetto si passa al cambio di stato: se l’Human è vaccinato ha diverse possibilità di guarire, se però non guarisce diventa gravemente malato. Se l’Human è già gravemente malato ha un basso rischio di morire, se non muore torna un infetto normale.  
La stessa cosa vale per tutti i non vaccinati, solo con le probabilità di aggravarsi e morire maggiori.

## File script.js

All’interno del file si gestiscono le interazioni dell’utente.

### createHuman

Il metodo createHuman crea tutti gli human che serviranno per la simulazione.

Al metodo vengono passati 4 interi, numero di persone totali, numero di infetti numero di persone che indossano la mascherina e numero di persone che sono vaccinate.

Viene controllato che i numeri vadano bene e per ogni categoria vengono creati tutti gli human con dei for con all’interno i costruttori e inserisce i nuovi human all’interno di un array che conterrà tutti gli human presenti nella simulazione.

### startMovementInterval

Questo metodo richiama al suo interno il metodo setInterval che richiama moveEveryOne ogni velocita(variabile che definisce i millisecondi di attesa tra un movimento e l’altro) millisecondi. Il setInterval viene salvato all’interno della variabile intervalMovement.

Una volta richiamato, il metodo disabilita il bottone start ovvero quello che ha invocato questo metodo e abilita il bottone pause utilizzabile per richiamare la funzione pauseMovementInterval. Inoltre viene settato nel sessionStorage l’Item stato a start.

### pauseMovementInterval

Questo metodo ferma momentaneamente la simulazione bloccando l’esecuzione del metodo moveEveryone rimuovendo l’intervallo contenuto in intervalMovement tramite il metodo clearInterval.

Una volta richiamato il metodo disabilita il bottone pause ovvero quello che ha invocato questo metodo e stop inoltre abilita il bottone start utilizzabile per richiamare la funzione startMovementInterval.

Inoltre viene settato nel sessionStorage l’Item stato a pause.

### stop

Il metodo stop quando invocato setta la variabile globale stopSimulation a true e stampa nella console stop.

### moveEveryOne

Il metodo moveEveryOne viene invocato da un setInterval ogni 100 millisecondi e svuota il canvas.

In seguito controlla se la variabile globale stopSimulation è true se è così blocca l’intervall come il metodo pauseMovementInterval, svuota l’array di Human e abilita il bottone start e disabilita i bottoni stop e pause, in seguito pulisce il canvas e setta nel sessionStorage l’Item stato a stop.

Se invece stopSimulation è false muove tutti invocando il metodo casual\_move spiegato precedentemente per ogni elemento dell’array di Human.

Dopo aver mosso tutti invoca il metodo infectionControl che verrà spiegato in seguito.

### readInput

Il metodo serve a leggere gli input che servono a creare gli human ovvero i number input e gli slider in cui inserire il numero o le percentuali di human che si vogliono per ogni tipo.

Calcola quanti human si vogliono per ogni tipo e infine chiama il metodo createHuman documentato in precedenza.

### checkSliderValue

Il metodo checkSliderValue controlla che gli slider delle varie percentuali non possano superare il 100% in totale.

Se uno slider supera la soglia consentita dal posizionamento degli altri viene spostato di 1 in dietro e ricontrolla per togliere tutta l’eccedenza. Infine invoca il metodo printSliderValue per stampare i nuovi valori.

### printSliderValue

Questo metodo passando l’id dello slider desiderato come argomento, stampa all’interno del paragrafo con id “p+id” dello slider il valore dello slider che gli corrisponde + %.

### implementCheckBoxes

Il metodo implementCheckBoxes permette di attivare e disattivare la possibilità di avere vaccinati e human che indossano la mascherina all’interno del campo di gioco, se il checkbox di cui viene passato l’id viene selezionato lo slider di cui viene passato l’id viene attivato.

Se il checkbox non è selezionato lo slider rimane disattivato.

Inoltre controlla per ogni check box se sono selezionati, se lo sono attivano le corrispondenti sezioni delle statistiche.

### infectionControl

Il metodo infectionControl controlla chi è infetto, una volta trovati per ognuno di loro invoca i metodi trychangeInfectedState e tryInfect della classe Human spiegati in precedenza.

### menuIconOpen

Questo metodo serve ad aprire l’icona del menu, ovvero da hamburger trasformarlo in una x.

Se il menu viene aperto viene invocato il metodo openMenu che verrà spiegato in seguito.

Se il menu è già aperto viene invocato il metodo menuIconClose che verrà spiegato in seguito.

### menuIconClose

Questo metodo serve a chiudere l’icona del menu ovvero da x torna un hamburger.

Quando viene invocato questo metodo cambia l’icona e apre la pagina da caricare, se la pagina non viene cambiata il metodo carica l’ultima aperta che è salvata in una variabile.

### openMenu

Questo metodo serve a rendere visibili i link presenti nel menu, per fare ciò setta il display della sezione html del menu a block, mentre le altre a none.

### menuSelectParametriSimulazione

Questo metodo viene invocato da una delle voci del menu, setta il display block della sezione parametriSimulazione per renderla visibile e la mette anche come last opened infine invoca il metodo per aprire il menu, che visto che è già aperto lo chiuderà.

### menuSelectStatistiche

Questo metodo viene invocato da una delle voci del menu, setta il display block della sezione statistics per renderla visibile e la mette anche come last opened infine invoca il metodo per aprire il menu, che visto che è già aperto lo chiuderà.

### menuSelectHome

Questo metodo viene invocato da una delle voci del menu, setta il display block della sezione home per renderla visibile e la mette anche come last opened infine invoca il metodo per aprire il menu, che visto che è già aperto lo chiuderà.

### readState

Questa funzione viene invocata ogni secondo da un intervallo, ciò che fa è passare l’array di Human e contare quanti infetti ci sono al momento, quanti di questi sono vaccinati e quanti indossano la mascherina e conta i morti.

Dopo aver contato invoca il metodo makeStatistics che spiegherò nel prossimo capito.

### makeStatistics

Questo metodo serve a creare diverse percentuali e a inserirle in degli slider per rendere più leggibili le statistiche.

Le statistiche che vengono calcolate sono:Persone in salute iniziali, persone in salute al momento, infetti iniziali, infetti al momento, vaccinati al momento, vaccinati infetti, vaccinati sani, persone che indossano la mascherina iniziali, persone con la mascherina infetti al momento, persone con la mascherina sani.

Infine setta l’item infect a quanti infetti ci sono al momento, l’item healt che contiene quante persone sono sane e l’item dead che contiene quanti sono i morti dentro al sessionStorage.

### SetInfectionPercentageDefault

Questo metodo setta gli slider che determinano con quanta possibilità gli human si infettano, la funzione viene anche invocata all’inizio per rendere verosimile la simulazione.

Questo metodo invoca a sua volta printSliderValue per i 3 slider che rappresentano i contagi.

### setVelocityMovement

Questo metodo serve a gestire ogni quanto muovere gli human, legge il value dello slider incaricato di settare la velocità e la inserisce nella variabile, e scrive il valore dello slider.

### setChangeStatePercentageDefault

Questo metodo Setta i valori default negli slider che fanno inserire dall’utente le percentuali per il quale si modificano i cambi di stato di contagio(guarire diventare gravi e morire).

## Index.html

Nell’index è presente tutto il codice html, che serve a generare il campo da gioco, la sezione home contiene le informazioni generali sul covid, la sezione statistics contiene le varie statistiche sulla simulazione e la sezione parametriSimulazione serve a creare una nuova simulazione e gestire le percentuali di Human da creare, Inoltre si possono gestire le percentuali di contagio e la velocità di movimento.

Le varie sezioni sono raggiungibili tramite un menu ad hamburger.

All’interno della sezione statistics è presente anche un grafico che tiene conto di quanti morti,infetti e sani sono presenti in quale secondo della simulazione, per realizzarlo nel file si apre un tag script e si inserisce valori sull’asse x e quelli sull’asse y e il grafigo plotly fa tutto da solo, per prendere i dati leggo gli item all’interno del session storage.

Tramite un interval di un secondo stampo il nuovo numero di infetti solo se l’item stato è settato a start, se è settato a pause non succede nulla e se invece è settato a stop i secondi e i 6 array di valori vengono svuotati.

# Test

## Protocollo di test

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-001  REQ-13  REQ-14  REQ-15  REQ-16  REQ-18  REQ-19  REQ-20 | **Nome:** | Controllare che gli elementi html interattivi funzionano |
| **Descrizione:** | Bisogna controllare che tutti gli elementi input funzionino correttamente. | | |
| **Prerequisiti:** |  | | |
| **Procedura:** | entrare nel sito web e fare:  per i checkbox: controllare che attivandoli si attivino i componenti che fanno parte dell’estensione appena sbloccata.  Per gli slider controllare tramite f12 che le percentuali sul totale siano calcolate correttamente.  Infine per i Number controllare che non possano essere inseriti caratteri speciali e numeri al di fuori delle limitazioni, attraverso f12. | | |
| **Risultati attesi:** | In f12 e visivamente deve essere tutto funzionante. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-002  REQ-4  REQ-5  REQ-6  REQ-8  REQ-9  REQ-10  REQ-11  REQ-13 | **Nome:** | Controllare che i numeri iniziali delle persone siano corrispondenti ai valori immessi dall’utente e che le persone si creino |
| **Descrizione:** | Bisogna controllare che le persone con le varie caratteristiche che vengono create corrispondano in quantità ai valori inseriti dall’utente. | | |
| **Prerequisiti:** | TC-001 | | |
| **Procedura:** | Andare sul sito web, cliccare F12 e controllare che i numeri corrispondano dall’inizio ai valori immessi nella sezione parametri simulazione e guardare che nel campo di simulazione gli human siano presenti. | | |
| **Risultati attesi:** | I numeri sono uguali. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-003  REQ-12 | **Nome:** | Controllare che le palline si muovano casualmente |
| **Descrizione:** | Bisogna controllare che le palline si muovano tutte in direzioni diverse in maniera casuale. | | |
| **Prerequisiti:** | TC-002 | | |
| **Procedura:** | Entrare nel sito web, e iniziare la simulazione. | | |
| **Risultati attesi:** | Le palline si muovono tutte in direzioni casuali diverse. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-004  REQ-24 | **Nome:** | Controllare che le statistiche vengano fatte correttamente |
| **Descrizione:** | Bisogna controllare che la pagina delle statistiche sia visibile e venga visualizzata in maniera corretta, inoltre bisogna controllare che i valori risultanti siano corretta. Il grafico deve essere visibile. | | |
| **Prerequisiti:** | TC-003 | | |
| **Procedura:** | entrare nel sito web e controllare che ci sia il menu in alto a sinistra, e al suo interno ci sia la voce Statistiche e cliccando si apra una sezione dove si vedano i dati in tempo reale, ovvero i contagiati e i non contagiati e le relative percentuali e confrontarli con ciò che si vede dentro f12. | | |
| **Risultati attesi:** | Vedere tutto in maniera corretta. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-005  REQ-13 | **Nome:** | Controllare che le limitazioni vengano imposte realmente |
| **Descrizione:** | Bisogna controllare che inserendo un numero maggiore o minore delle limitazioni (10-400) venga inserito il numero più vicino al limite (10-400). | | |
| **Prerequisiti:** | TC-001 | | |
| **Procedura:** | entrare nel sito web e inserire dei numeri non validi nei campi numerici. | | |
| **Risultati attesi:** | I valori inseriti vengono cambiati in base agli errori e se non escono dai limiti vengono tenuti. | | |

## Risultati test

|  |  |
| --- | --- |
| Test | Risultato |
| TC-004 | Provando un po’ tutti gli input html si vede che tutti producono il risultato sperato e descritto nel test case. Test passato. |
| TC-001 | Andando nella sezione dei parametri della simulazione si inseriscono dei numeri ad esempio 100 persone totali e il 10% di infetti, vaccinati e con la mascherina. Una volta cliccato start si creano le persone nel campo di simulazione e aprendo F12 si leggono quante persone e di che genere create. Il numero delle persone create corrisponde a 70 normali, 10 infetti, 10 vaccinati e 10 portatori di mascherina che è il risultato atteso. |
| TC-002 | Subito dopo aver fatto il TC-001 chiudo F12 e controllo che nella simulazione le palline si spostino in direzioni diverse, un po’ a destra un po’ a sinistra un po’ verso l’alto e un po’ verso il basso. Anche questo test è risultato positivo. |
| TC-003 | Dopo aver stoppato la simulazione reinserisco dei valori invalidi, come 2000 persone e per gli slider provo a superare il 100 ma vengo bloccato. All’avvio della simulazione anche il 2000 viene trasformato in 400 che è il limite, provando anche con 2 all’avvio il valore viene cambiato in 10. Anche questo test è passato. |
| TC-005 | Mentre la simulazione sta andando se si guarda nella sezione delle statistiche si possono notare i range che cambiano i valori in base allo spargimento dell’infezione e si vede anche il grafico che si aggiorna costantemente in automatico. Test risultato un successo. |

## Mancanze/limitazioni conosciute

Non dovrebbero essere presenti mancanze o limitazioni rispetto ai requisiti.

# Consuntivo

Figura 8 Gantt consuntivo

# Conclusioni

## Sviluppi futuri

Possono essere aggiunte molte cose essendo una simulazione, ad esempio implementare la possibilità di essere sia vaccinati che portatori di mascherina, aggiungere il degrado delle mascherine e della vaccinazione, inoltre potrebbe essere possibile l’aggiunta di alcuni punti di vendita delle mascherine oppure dei punti di vaccinazione.  
Potrebbe essere inserita la possibilità di rilevamento dei sintomi e di quarantene o funzionalità simili.  
Si potrebbe renderlo un videogioco complesso invece di lasciarlo una simulazione semplice.

## Considerazioni personali

In questo mio primo progetto ho imparato a gestire il tempo e a gestire un progetto da 0, ho ampliato le mie conoscenze riguardo allo sviluppo di applicazioni web, ho imparato a gestire una classe in JavaScript, ho imparato a usare i grafici plotly e ad usare le classi offerte dalla libreria bootstrap.  
Mi è piaciuto e mi ha divertito fare questo progetto, non era troppo difficile e non c’erano troppi elementi da gestire. Ovviamente c’è voluto dell’impegno e sono riuscito ad impegnarmi e a divertirmi anche quando spuntavano degli errori, grazie al fatto che essendo un programma con una parte grafica si vedevano le righe di codice che si sono scritte dare dei risultati.  
Ho avuto pochi problemi e quasi sempre facilmente risolvibili anche grazie all’aiuto dei professori, ciò ha tenuto alto il mio morale.  
Il mio progetto non rivoluzionerà il mondo ma sono fiero del mio lavoro e sono felice di aver scelto questo progetto.

# Bibliografia

## Bibliografia per articoli di riviste:

Non ho consultato riviste.

## Bibliografia per libri

Non ho consultato libri.

## Sitografia

* https://www.w3schools.com/, *W3School*, (consultato il 30.09.2021,10.10.2021,21.10.2021,28.10.2021,11.11.2021, 18.11.2021, 25.11.2021, 02.12.2021)
* <https://getbootstrap.com/>, *Bootstrap*, (consultato il 25.11.2021)
* <https://plotly.com/>, *Plotly*, (consultato il 28.10.2021, 11.11.2021, 02.12.2021)

# Allegati

Elenco degli allegati, esempio:

* Diari di lavoro
* Codici sorgente
* QdC