Documentazione di CovidSimulator

1 Introduzione 3

1.1 Informazioni sul progetto 3

1.2 Abstract 3

1.3 Scopo 3

1.4 Analisi del dominio 4

1.5 Analisi e specifica dei requisiti 4

1.6 Use case 7

1.7 Pianificazione 7

1.8 Analisi dei mezzi 7

1.8.1 Software 7

1.8.2 Hardware 7

2 Progettazione 8

2.1 Design delle interfacce 9

3 Implementazione 11

3.1 Inizio 11

3.2 Creazione della classe Human 11

3.2.1 Costruttore 11

3.2.2 Set\_infected 11

3.2.3 Casual\_move 12

3.2.4 changeDirection 12

3.2.5 checkX e checkY 12

3.2.6 print 13

3.2.7 tryInfect 13

3.3 File script.js 13

3.3.1 createHuman 13

3.3.2 startMovementInterval 13

3.3.3 pauseMovementInterval 14

3.3.4 moveEveryOne 14

§3.3.5 readInput 14

3.3.6 checkSliderValue 14

3.3.7 implementCheckBoxes 14

3.3.8 infectionControl 14

3.4 Index.html 15

4 Test 15

4.1 Protocollo di test 15

4.2 Risultati test 16

4.3 Mancanze/limitazioni conosciute 17

5 Consuntivo 17

6 Conclusioni 17

6.1 Sviluppi futuri 17

6.2 Considerazioni personali 17

7 Bibliografia 17

7.1 Bibliografia per articoli di riviste: 17

7.2 Bibliografia per libri 17

7.3 Sitografia 18

8 Allegati 18

# 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

È una breve e accurata rappresentazione dei contenuti di un documento, senza notazioni critiche o valutazioni. Lo scopo di un abstract efficace dovrebbe essere quello di far conoscere all’utente il contenuto di base di un documento e metterlo nella condizione di decidere se risponde ai suoi interessi e se è opportuno il ricorso al documento originale.

Può contenere alcuni o tutti gli elementi seguenti:

* **Background/Situazione iniziale**
* **Descrizione del problema e motivazione**: Che problema ho cercato di risolvere? Questa sezione dovrebbe includere l'importanza del vostro lavoro, la difficoltà dell'area e l'effetto che potrebbe avere se portato a termine con successo.
* **Approccio/Metodi**: Come ho ottenuto dei progressi? Come ho risolto il problema (tecniche…)? Quale è stata l’entità del mio lavoro? Che fattori importanti controllo, ignoro o misuro?
* **Risultati**: Quale è la risposta? Quali sono i risultati? Quanto è più veloce, più sicuro, più economico o in qualche altro aspetto migliore di altri prodotti/soluzioni?

Esempio di abstract:

*As the size and complexity of today’s most modern computer chips increase, new techniques must be developed to effectively design and create Very Large-Scale Integration chips quickly. For this project, a new type of hardware compiler is created. This hardware compiler will read a C++ program, and physically design a suitable microprocessor intended for running that specific program. With this new and powerful compiler, it is possible to design anything from a small adder, to a microprocessor with millions of transistors. Designing new computer chips, such as the Pentium 4, can require dozens of engineers and months of time. With the help of this compiler, a single person could design such a large-scale microprocessor in just weeks.*

## Scopo

Questo progetto è volto all’intrattenimento e vuole anche fare una simulazione pressoché verosimile dell’evoluzione dei contagi all’interno di una popolazione.Analisi

## Analisi del dominio

Questo capitolo dovrebbe descrivere il contesto in cui il prodotto verrà utilizzato, da questa analisi dovrebbero scaturire le risposte a quesiti quali ad esempio:

* Background/Situazione iniziale
* Quale è e come è organizzato il contesto in cui il prodotto dovrà funzionare?
* Come viene risolto attualmente il problema? Esiste già un prodotto simile?
* Chi sono gli utenti? Che bisogni hanno? Come e dove lavorano?
* Che competenze/conoscenze/cultura posseggono gli utenti in relazione con il problema?
* Esistono convenzioni/standard applicati nel dominio?
* Che conoscenze teoriche bisogna avere/acquisire per poter operare efficacemente nel dominio?
* …

## Analisi e specifica dei requisiti

Il progettista, dopo aver ricevuto il mandato, in collaborazione con il committente redige una lista di requisiti. Durante questi incontri, tramite interviste (da inserire nei diari), il progettista deve cercare di rispondere alle seguenti domande:

* Quali sono i bisogni del committente?
* Quali funzioni deve svolgere il prodotto?
* Come devono essere implementate?
* L’utente, come vorrebbe/dovrebbe interagire con il prodotto?
* Come verrà utilizzato il prodotto?
* Che tipo di interfaccia si immagina?
* Che prestazioni minime deve fornire il prodotto?
* Che grado di sicurezza deve avere il prodotto?
* …

In base alla lista dei requisiti e all’analisi degli stessi, il progettista redige una *specifica dei requisiti* in cui elenca e descrive in modo dettagliato quali sono le funzionalità che il prodotto fornirà. La specifica dovrebbe essere abbastanza dettagliata da poter essere utilizzata come base per lo sviluppo, ma non troppo; ad esempio non dovrebbe contenere dettagli di implementazione, o definizioni dettagliate dell’interfaccia grafica a meno che questi non siano considerati cruciali. Non si deve scordare che i requisiti non rappresentano delle attività bensì delle caratteristiche che il prodotto dovrà possedere.

|  |  |  |
| --- | --- | --- |
| ID | Requisiti | Importanza |
| 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(5-100) | 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 |

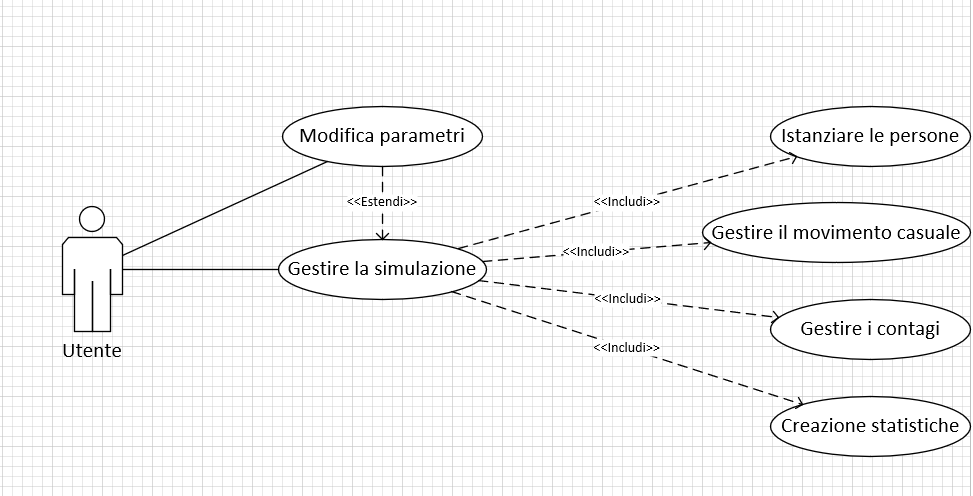
**Spiegazione elementi tabella dei requisiti:**

**ID**: identificativo univoco del requisito

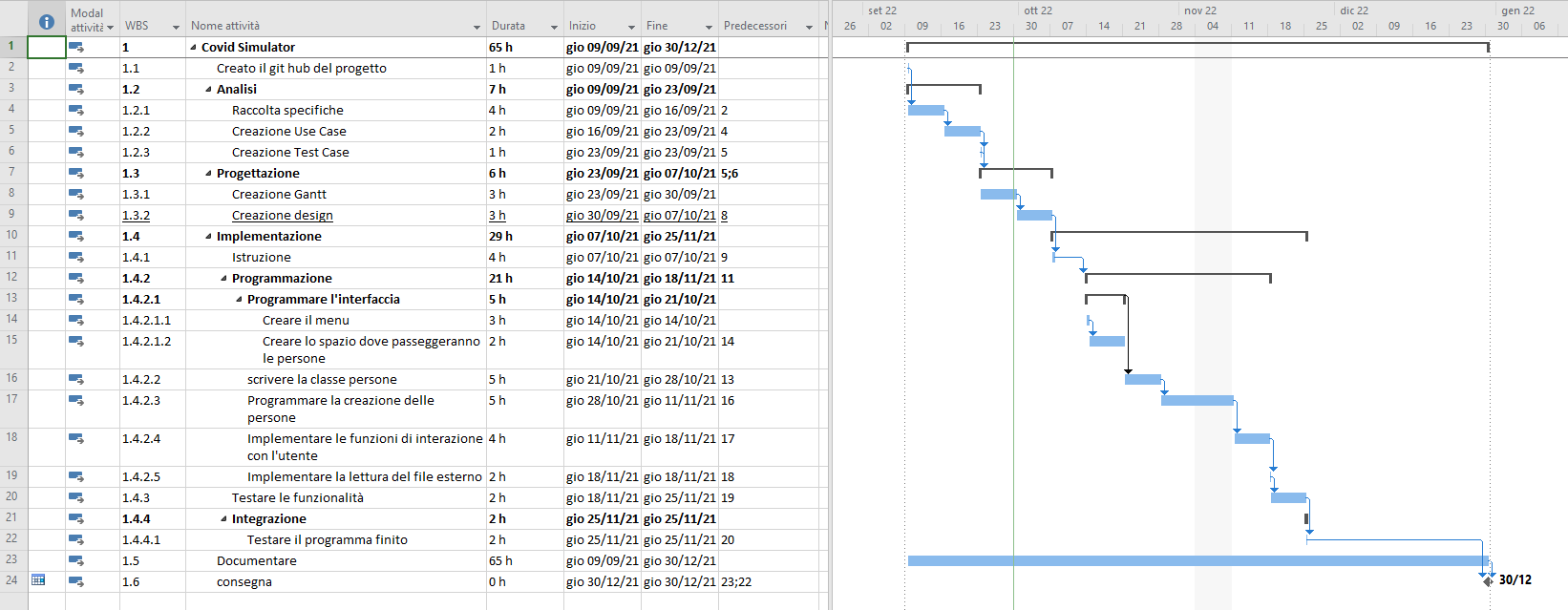
**Nome**: breve descrizione del requisito

**Priorità**: indica l’importanza di un requisito nell’insieme del progetto, definita assieme al committente. Ad esempio, poter disporre di report con colonne di colori diversi ha priorità minore rispetto al fatto di avere un database con gli elementi al suo interno. Solitamente si definiscono al massimo di 2-3 livelli di priorità.

## Use case



## Pianificazione



## Analisi dei mezzi

Elencare e descrivere i mezzi disponibili per la realizzazione del progetto. Ricordarsi di sempre descrivere nel dettaglio le versioni e il modello di riferimento.

### Software

SDK, librerie, tools utilizzati per la realizzazione del progetto e eventuali dipendenze.

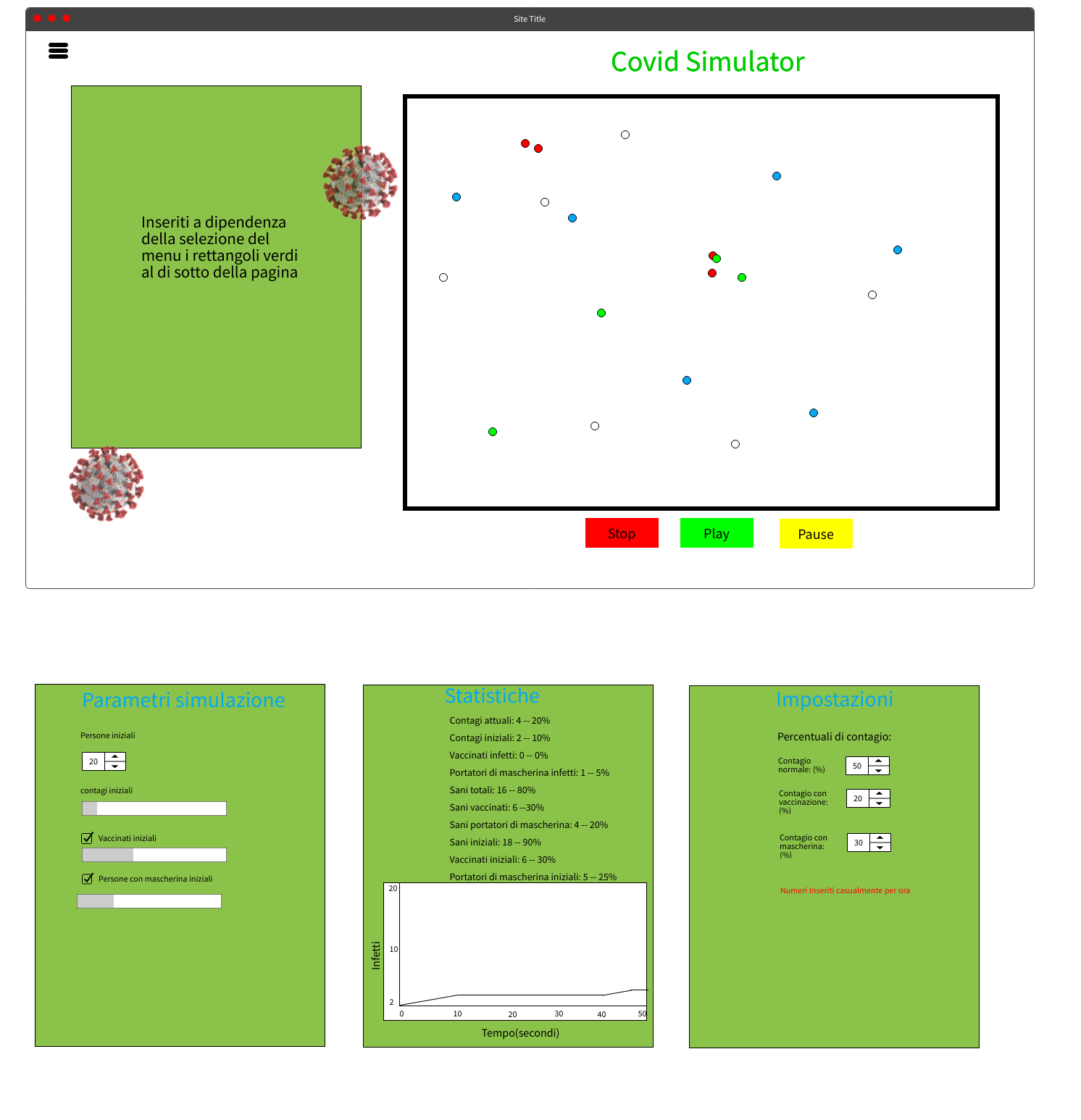
### Hardware

Su quale piattaforma dovrà essere eseguito il prodotto? Che hardware particolare è coinvolto nel progetto? Che particolarità e limitazioni presenta? Che HW sarà disponibile durante lo sviluppo?

# Progettazione

Questo capitolo descrive esaustivamente come deve essere realizzato il prodotto fin nei suoi dettagli. Una buona progettazione permette all’esecutore di evitare fraintendimenti e imprecisioni nell’implementazione del prodotto.

## Design delle interfacce



# 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.

Figura File index.html dopo la creazione

## Creazione della classe Human

### Costruttore

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

Figura prototipo costruttore

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

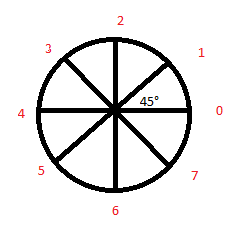
Se i valori inseriti sono validi allora si prosegue con l’assegnare i valori passati, se l’umano è infetto sarà di colore rosso, se è vaccinato sarà di colore blu, se indossa la mascherina sarà di colore verde e se non sarà nulla di ciò verrà rappresentato in bianco.

Figura direzione spostamento

Automaticamente verrà anche creata la posizione casuale dell’human con un attributo x e uno 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.

### 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.

### 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 si 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.

Per risolvere questo problema ho aggiunto l’attributo direction che serve per spostarlo di un pixel in quella direzione, e con l’estrazione di 3 numeri casuali e mettendoli tutti in un if con degli and invoco il metodo changeDirection che verrà spiegato in seguito.

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

### changeDirection

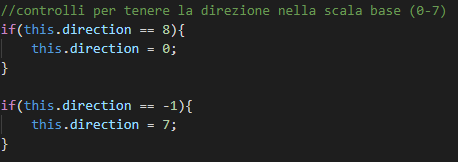
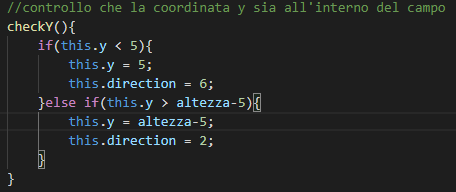
Il metodo changeDirection se invocato cambia la direzione.

Figura controllo direzione range 0-7

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.

Figura codice del metodo(checkY)

### 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.

Figura creazione del 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).

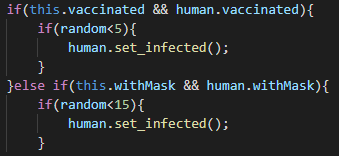
Se sono abbastanza vicini e l’Human passato come argomento non è ancora contagiato estraggo un numero casuale, e a dipendenza della combinazione dei booleani dei due human viene dato un numero, se quello casuale è minore di quello calcolato l’human viene contagiato.

Figura esempio di prova di contagio

Esempio della prova del contagio nella foto affianco.

## 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.

Figura prototipo createHuman

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 100 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 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 i bottoni start pause e createHuman, 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.

Per richiamare una funzione ogni 100 millisecondi bisogna scrivere il codice scritto nella foto.

Figura setInterval

### 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.

Una volta invocato disabilita il bottone che lo ha invocato ovvero il bottone createHuman.

### 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 checkato lo slider di cui viene passato l’id viene attivato.

Se il checkbox non è checkato lo slider rimane disattivato.

### infectionControl

Il metodo infectionControl controlla chi è infetto, una volta trovati cerca chi è entro una zona di raggio\*2 una volta trovati invoca il metodo tryInfect della classe Human spiegato in precendenza.

### 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.

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 calcolate sono tutte in base al totale, queste statistiche sono: Persone in salute iniziali, persone in salute al momento, infetti iniziali, infetti al momento, vaccinati al momento, vaccinati infetti, 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 dentro al sessionStorage.

## 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.

Le varie sezioni sono raggiungibili tramite un menu ad hamburger.

All’interno della sezione statistics è presente anche un grafico, 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 secondi e i due array di valori vengono svuotati.

# Test

## Protocollo di test

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-001  REQ-3 | **Nome:** | Controllare che il file di configurazione sia raggiungibile |
| **Descrizione:** | Bisogna controllare che il file di configurazione contenente le configurazioni iniziali sia raggiungibile e il programma lo legga correttamente. | | |
| **Prerequisiti:** |  | | |
| **Procedura:** | Andare sul sito web, aprire lo strumento console del tasto f12 e leggere il messaggio. | | |
| **Risultati attesi:** | File conf trovato.  File conf non trovato, creazione del file di conf con dei valori predefiniti. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |
| **Descrizione:** | Bisogna controllare che le persone con le varie particolarità che vengono create corrispondano in quantità ai valori inseriti dall’utente. | | |
| **Prerequisiti:** |  | | |
| **Procedura:** | Andare sul sito web, aprire la sezione delle statistiche e controllare che i numeri corrispondano dall’inizio. | | |
| **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:** |  | | |
| **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-13 | **Nome:** | Controllare che le limitazioni vengano imposte realmente |
| **Descrizione:** | Bisogna controllare che inserendo un numero maggiore o minore delle limitazioni (5-100) venga inserito il numero di default ovvero 50. | | |
| **Prerequisiti:** |  | | |
| **Procedura:** | entrare nel sito web e inserire dei numeri non validi nei campi numerici. | | |
| **Risultati attesi:** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-005  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-006  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. | | |
| **Prerequisiti:** |  | | |
| **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. | | |

## Risultati test

Tabella riassuntiva in cui si inseriscono i test riusciti e non del prodotto finale. Se un test non riesce e viene corretto l’errore, questo dovrà risultare nel documento finale come riuscito (la procedura della correzione apparirà nel diario), altrimenti dovrà essere descritto l’errore con eventuali ipotesi di correzione.

## Mancanze/limitazioni conosciute

Descrizione con motivazione di eventuali elementi mancanti o non completamente implementati, al di fuori dei test case. Non devono essere riportati gli errori e i problemi riscontrati e poi risolti durante il progetto.

# Consuntivo

Consuntivo del tempo di lavoro effettivo e considerazioni riguardo le differenze rispetto alla pianificazione (cap. 1.7) (ad esempio Gantt consuntivo).

# Conclusioni

Quali sono le implicazioni della mia soluzione? Che impatto avrà? Cambierà il mondo? È un successo importante? È solo un’aggiunta marginale o è semplicemente servita per scoprire che questo percorso è stato una perdita di tempo? I risultati ottenuti sono generali, facilmente generalizzabili o sono specifici di un caso particolare? ecc.

## Sviluppi futuri

Migliorie o estensioni che possono essere sviluppate sul prodotto.

## Considerazioni personali

Cosa ho imparato in questo progetto? ecc.

# Bibliografia

## Bibliografia per articoli di riviste:

1. Cognome e nome (o iniziali) dell’autore o degli autori, o nome dell’organizzazione,
2. Titolo dell’articolo (tra virgolette),
3. Titolo della rivista (in italico),
4. Anno e numero
5. Pagina iniziale dell’articolo.

## Bibliografia per libri

1. Cognome e nome (o iniziali) dell’autore o degli autori, o nome dell’organizzazione,
2. Titolo del libro (in italico),
3. ev. Numero di edizione,
4. Nome dell’editore,
5. Anno di pubblicazione,
6. ISBN.

## Sitografia

1. URL del sito (se troppo lungo solo dominio, evt completo nel diario),
2. Eventuale titolo della pagina (in italico),
3. Data di consultazione (GG-MM-AAAA).

**Esempio:**

* http://standards.ieee.org/guides/style/section7.html, *IEEE Standards Style Manual*, 07-06-2008.

# Allegati

Elenco degli allegati, esempio:

* Diari di lavoro
* Codici sorgente/documentazione macchine virtuali
* Istruzioni di installazione del prodotto (con credenziali di accesso) e/o di eventuali prodotti terzi
* Documentazione di prodotti di terzi
* Eventuali guide utente / Manuali di utilizzo
* Mandato e/o QdC
* Prodotto
* …