UNIVERSITÀ DEGLI STUDI DI PERUGIA

Dipartimento di Matematica e Informatica

CORSO DI LAUREA MAGISTRALE IN INFORMATICA

RELAZIONE INTELLIGENZA ARTIFICIALE

Progetto "Crowd Seating Simulation"

Studente

Enrico Maria Falcone

Professore

Alfredo Milani

Indice

Elenco delle figure

1 Introduzione generale

L'obiettivo di questo progetto è quello di effettuare una Crowd Seating Simulation, ovvero una simulazione che vede degli agenti camminare in uno spazio seguendo delle regole precise in modo da riprodurre il più fedelmente possibile il comportamento di una folla di esseri umani che entrando da un'entrata, si vuole sedere su delle sedie posizionate e orientate verso un palcoscenico. Questi individui verranno difatti attratti/respinti dalle sedie più vicine al palco, seguendo comunque i criteri della liberta della sedia stessa.

Figura 1: Icona del programma

Per svolgere questo progetto è stato utilizzato NetLogo, un software sviluppato da Uri Wilen-sky nel 1999 presso il Center for Connected Learning and Computer-Based Modeling della Northwestern University che offre un ambiente di sviluppo ideale per la realizzazione di modelli di simulazioni ad agenti.

All'interno di NetLogo è possibile studiare l'evoluzione nel tempo di un sistema e visualizzarla in tempo reale all'interno di un display virtuale, in due o tre dimensioni.

NetLogo è stato scritto in Java ma è completamente programmabile per mezzo di un meta linguaggio ad oggetti basato sulla sintassi del Logo, un linguaggio di programmazione risalente agli anni 60.

Dal momento che l'ambiente di sviluppo di NetLogo interpreta direttamente il codice senza compilarlo, al suo interno è possibile in tempo reale interagire con il sistema attraverso buttons e sliders per modificarne i parametri di controllo.

Nella prossima sezione verrà spiegata l'interfaccia realizzata per lo scopo precedentemente spiegato.

1.1 Testo del progetto dato

Crowd Seating Simulation (1 persona)

Un simulatore del posizionamento di un insieme di individui su un insieme di sedie orientate verso un palcoscenico.

Regole: attrazione verso il palco, distanza minima dal palco, massimizzazione della distanza tra vicini; trovare sedia.

La disposizione, il numero di sedie e di individui sono definibili da utente.

L'interfaccia deve mostrare graficamente il comportamento nel tempo e consentire di modificare i parametri in tempo reale.

Strumenti: NetLogo

2 Descrizione del simulatore

Nella seguente sezione verranno spiegate le varie funzionalità degli elementi grafici presenti nel simulatore stesso, inoltre verrà fornita una guida sull'utilizzo del simulatore per effettuare una simulazione impostando i parametri desiderati.

2.1 Presentazione Interfaccia

L'interfaccia presentata nel seguente simulatore è stata divisa in tre sezioni per facilitare il setting e la visualizzazione della simulazione:

- La sezione a sinistra (come visibile in Figura 2) presenta i vari bottoni e gli slider per la specifica dei parametri essenziali nella simulazione; specifici bottoni per lo spawn di diverse simulazioni e anche la scelta del parametro di repulsione, .

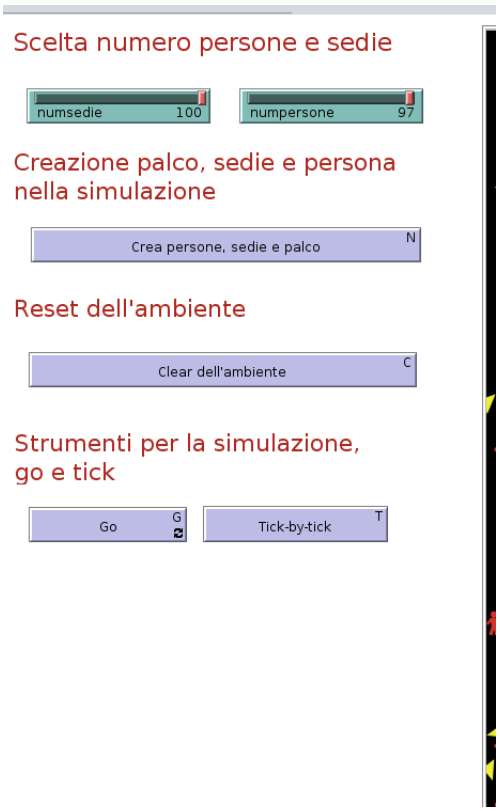

Figura 2: Sezione sinistra.

- Nella sezione centrale è presente un riquadro con il quale si può avere

una visione in tempo reale della simulazione specificata dai parametri settati nella sezione sinistra, come visibile in Figura 3.

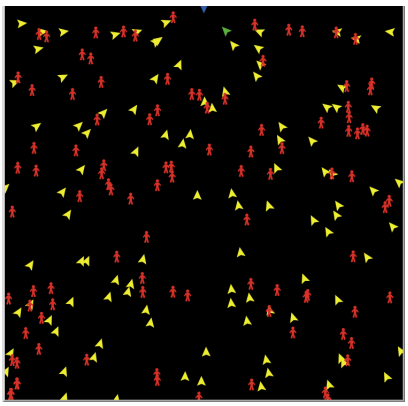

Figura 3: Sezione centrale.

- La sezione di destra contiene bottoni dedicati al salvataggio, al caricamento, alla creazione e alla modifica di un modello precedentemente personalizzato, come è possibile notare in Figura 4.

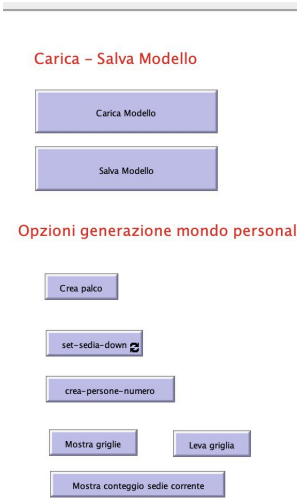

Figura 4: Sezione destra.

2.1.1 Sezione sinistra di specifica

Nella sezione di sinistra, più nello specifico, sono presenti slider e bottoni, come presentati di seguito:

- In alto è presente uno slider che permette di scegliere quante persone inserire nelle configurazioni predefinite, ovvero quella a rettangolo, e quella a triangolo.
- A seguire sono presenti due slider che permettono di scegliere il numero di sedie delle rispettive configurazioni predefinite. Lo slider di sinistra riguarda la disposizione a rettangolo, anche definita "disposizione 1", mentre quello a destra, riguarda la disposizione a triangolo "disposizione 2".
- Uno slider riguardante la repulsione che ogni spettatore presenta nel-

la scelta del posto, ovvero ogni spettatore sceglierà un posto libero, mostrando una determinata repulsione dagli altri posti occupati.

- In seguito, sono presenti 4 bottoni per resettare il simulatore, far spawnare le persone in una predeterminata posizione scelta con il mouse, e due bottoni per far spawnare le disposizioni del teatro.

- Alla fine, sono presenti due ulteriori bottoni per l'avvio della simulazione.

  Il bottone "*go*" che una volta premuto, resta selezionato fino alla fine della simulazione, e il bottone "*tick-by-tick*" che una volta premuto, permette di osservare l'esito della simulazione al tick successivo. Questo risulta essere veramente utile quando si vuole osservare nel dettaglio la scelta del posto e le varie interazioni degli spettatori.

2.1.2 Sezione centrale

Come anche descritto precedentemente, la sezione centrale presenta un riquadro dove viene mostrata la simulazione, nella quale gli elementi sono i seguenti:

- Le sedie sono rappresentate dalle frecce gialle orientate verso un palco.

- Il palco viene rappresentato da una freccia blu con una predeterminata orientazione, rivolto verso il basso per rappresentare il fatto che la scena sia orientata verso gli spettatori, e quindi le sedie del teatro.

- Le persone sono rappresentate da icone a forma di "persona" colorate con il colore rosso.

2.2 Sezione destra

Come anche già visibile nella Figura 4, in sezione di destra, accanto al riquadro centrale, sono presenti i comandi per caricare e salvare il modello. Una volta premuti, l'utente viene invitato a specificare il nome del file e la directory desiderata.

Oltre a questi, sono presenti ulteriori bottoni che permettono la creazione di una disposizione di sedie personalizzata:

- Il bottone "*Crea palco*", crea un palco nella stessa posizione delle predisposizioni presenti.
- Il bottone "*set-sedia-down*", permette di scegliere la disposizione della singola sedia nel mondo.
- Il bottone "*crea-persone-numero*", permette di far spawnare un numero di persone pari a quelle specificate nello slider a sinistra.
- I bottoni *Mostra griglie*, e *Leva griglia*, permettono di mostrare una griglia in modo da specificare con il tocco del mouse, la disposizione della sedia nel mondo. Questo è visibile nella Figura 5.

  Ovviamente, una volta terminata la disposizione delle sedie, è possibile eliminare la griglia tramite il bottone "*Leva griglia*".

  Va sottolineato che nel caso in cui si effettua un salvataggio della disposizione con la griglia in visione, allora, quando questa viene ricaricata, verrà mostrata nuovamente.

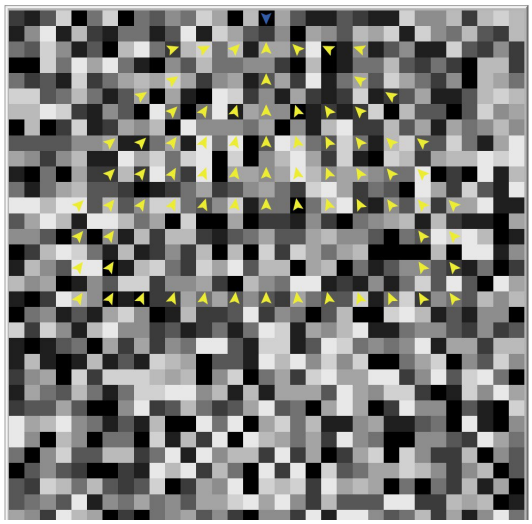

Figura 5: Sezione centrale con griglia in evidenza.

- Il bottone finale "*Mostra conteggio sedie corrente*", restituisce sul terminale il numero attualmente presente di sedie utile per la configurazione della disposizione personalizzata.

2.3 Guida ad un primo utilizzo e specifica simulazione

All'avvio del simulatore, una procedura specifica si occupa automaticamente dell'inizializzazione del contatore dei tick di clock, che assume il valore 0.

All'avvio della simulazione, gli elementi presenti sullo schermo interagiscono tra di loro, mostrando queste interazioni nel riquadro centrale.

È possibile caricare una disposizione precedentemente creata, tramite il bottone presente nella sezione di destra, oppure è possibile premere i pulsanti dedicati allo spawn personalizzato. Una volta che sono presenti tutti gli elementi della simulazione, è possibile premere il bottone "go" per far partire

la simulazione e osservare il comportamento degli agenti nel teatro.

In ogni caso, è sempre possibile stoppare la simulazione, ripremendo il bottone "go", ed è anche possibile prendere ed apportare delle modifiche alla simulazione, aggiungendo o diminuendo il numero delle sedie, o degli spettatori, nonchè modificare la loro posizione.

Nelle prossime pagine, verranno spiegate nel dettaglio le funzioni che vengono fatte partire alla pressione di ogni bottone, inoltre verrà anche mostrato l'algoritmo di scelta delle sedie in base alla repulsione e alla visibilità della scena.

3 Codice sorgente

Nella seguente sezione vengono spiegate le variabili, gli oggetti e le funzioni definite nel codice proposto come soluzione al problema d'esame.

3.1 Introduzione

I primi oggetti che vengono definiti sono le turtles, ovvero oggetti in grado di muoversi, cambiare colore, forma e altre proprietà.

Sono inoltre anch'esse in grado di immagazzinare informazioni sotto forma di variabili proprietarie.

Ogni agente generico che viene generato appartiene alla classe turtle, ma è possibile definire delle "sottoclassi" dette breed che permettono di differenziare gli agenti, facendogli possedere variabili e comportamenti che dipendono dalla breed di appartenenza.

All'interno del codice vengono definite sia il nome della breed (al plurale), sia quello del generico agente che ne fa parte (al singolare).

```
breed
[
    sedie sedia
]
breed
[
    persone persona
]
breed
[
    palchi palco
]
```

E' importante specificare come, anche se un agente faccia parte di una breed, esso venga comunque considerato anche una turtle.

3.2 Definizioni variabili

In NetLogo sono presenti tre tipologie di variabili:

- **Variabili globali**: esistono in esemplari unici e sono visibili da qualunque punto del codice. Vengono dichiarate fuori da ogni procedura e aggiornate con il comando "set nome variabile valore" oppure direttamente dall'interfaccia attraverso slider, switch o chooser.
- **Variabili proprietarie**: possono genericamente appartenere a tutte le turtles (turtles-own), patches (patches-own) oppure ad una breed

specifica (breed-own). Queste variabili sono replicate per ogni agente alle quali appartengono.

- **Variabili locali**: esistono in esemplari unici e sono visibili solo all'interno della routine dove sono state dichiarate con "let nome variabile valore".

Tornando al codice, prima della definizione delle procedure vengono definite le variabili, sia quelle globali che quelle proprietarie delle breed presenti.

```
globals
[
    primo
    patch-data
]
persone-own
[
    seated? ;true se la persona e'' seduta, falsa altrimenti
]
sedie-own
[
    libero? ; proprieta' della sedia, true se la sedia e' libera, falsa
    altrimenti
]


```

3.3 Definizione procedure

All'interno del linguaggio di programmazione NetLogo ci sono due tipologie di procedure:

- **Subroutine**: sono richiamate da altre subroutine o direttamente dall'interfaccia per mezzo della pressione di un button. Possono ricevere valori in ingresso, ma non restituiscono alcun valore in uscita. Iniziano sempre con la parola chiave *to* e terminano con *end*.
- **Reporter**: sono richiamati da altri reporter o subroutine. Possono ricevere valori in ingresso e restituiscono sempre un valore in uscita, attraverso il comando *report*. Iniziano sempre con la parola chiave *to-report* e terminano con *end*.

Tornando al codice, la prima procedura si chiama clear-everything.

3.3.1 Clear-everything

Richiamata dalla pressione del button omonimo all'interno dell'interfaccia, si occupa di riportare in qualunque momento la simulazione allo stato iniziale. Con questa subroutine infatti vengono eliminati tutti gli agenti presenti all'interno del display e resettato il contatore dei tick di clock.

```
1 to clear-everything
2   clear-all
3   reset-ticks
4   print "reset dell'ambiente completato"
5 end
```

3.3.2 Crea-palco

La procedura *Crea-palco* prevede la creazione di un palcoscenico rappresentato con una freccia blu, posto sulle cordinate come specificato dal codice seguente.

```
to crea-palco
  set primo true
  let conto-palchi (count palchi)
  if conto-palchi >= 0 [
    ask palchi [die]
    let palco-id 0
    create-palchi 1 [
      set color blue
      set palco-id who
      set xcor 0
      set ycor 16
      set heading 180
    ]
  ]
end
```

3.3.3 Prova-config1

La procedura *prova-config1* prevede di andare a creare un palco, e un numero di sedie specificato dallo slider visto nelle sezioni precedenti. Tale procedura dispone le sedie in modo rettangolare, orientandole verso il palco, e colorandole di giallo, per distinguerle dal palco e dalle sedie già occupate.

```
to prova-config1 ;spawn delle sedie configurazione 1
  clear-everything
```

```
3   if numsedie < numpersone [
4     ;print "non ci sono abbastanza sedie"
5     user-message ("non ci sono abbastanza sedie disponibili per tali persone
      ")
6     clear-everything
7     stop
8   ]
9 crea-palco
10 crea-persone
11
12 foreach (range -14 16 2) [ x ->
13   foreach (range 14 -16 -1)[ y ->
14     create-sedie 1 [
15         set libero? true
16         ;set shape "arrow"
17         set color yellow
18         set xcor (- y)
19         set ycor (- x)
20         face one-of palchi
21
22       ]
23
24       if count sedie = numsedie [
25         stop; se il numero di sedie e quello, allora esco
26       ]
27   ]
28 ]
29 ;per mettere in mezzo semplicemente ogni volta conto se la riga e arrivata a
       30, verifico se ce la possibilita di metterna altri 30, se non ce
     allora li metto a partire dal mezzo
30
```

```
31 end
```

3.3.4 Prova-config2

La procedura *prova-config2* prevede di andare a creare un palco, e un numero di sedie specificato dallo slider visto nelle sezioni precedenti. Tale procedura dispone le sedie in modo triangolare, orientandole verso il palco, e colorandole di giallo, per distinguerle dal palco e dalle sedie già occupate.

```
1  to prova-config2;sedia a parabola, spawn delle sedie
2    clear-everything
3    if numsedie2 < numpersone [
4      ;print "non ci sono abbastanza sedie"
5      user-message ("non ci sono abbastanza sedie disponibili per tali persone
       ")
6      clear-everything
7      stop
8    ]
9  crea-palco
10 crea-persone
11
12 foreach (range 12 -2 -4) [ z ->
13   let j 1
14     if count sedie >= numsedie2 [
15         stop; se il numero di sedie e quello, allora esco
16       ]
17    create-sedie 1 [
18         set libero? true
19         ;set shape "arrow"
20         set color yellow
```

```
21        set xcor (0)
22        set ycor (z)
23        face one-of palchi
24
25      ]
26
27    foreach (range z 15 1) [
28     i ->
29      if count sedie >= numsedie2 [
30        stop; se il numero di sedie e quello, allora esco
31      ]
32      create-sedie 1 [
33        set libero? true
34        ;set shape "arrow"
35        set color yellow
36        set xcor (j)
37        set ycor (i + 1)
38        face one-of palchi
39
40      ]
41      if count sedie >= numsedie2 [
42        stop; se il numero di sedie e quello, allora esco
43      ]
44      create-sedie 1 [
45        set libero? true
46        ;set shape "arrow"
47        set color yellow
48        set xcor ( - j)
49        set ycor (i + 1)
50        face one-of palchi
51
```

```
52      ]
53
54      set j (j + 1)
55      if count sedie >= numsedie2 [
56        stop; se il numero di sedie e quello, allora esco
57      ]
58    ]
59  ]
60 end
```

3.3.5 Crea-persone

La procedura *crea-persone* crea e fa spawnare un numero di persone nelle coordinate specificate dal codice. Il numero di persone viene specificato dallo slider presentato in sezione sinistra e precedentemente spiegato.

```
1  to crea-persone ;dato che le persone sono le ultime ad essere create, ogni
       persona individua la sedia piu' vicina al palco e la colora di verde
2
3    create-persone numpersone[; quindi creo le persone
4      set color red
5      set shape "person"
6      set seated? false
7      set xcor -16
8      set ycor -16
9      face one-of palchi
10   ]
11
12 end
```

Ovviamente, tale procedura va a predisporre anche le variabili delle persone, andando ad indicare che ogni persona è in piedi e non seduta.

3.3.6 Colora-sedia-vicina

La procedura *colora-sedia-vicina* si occupa di andare a decidere se colorare o meno altre sedie candidate ad essere occupate da una persona in piedi.

```
1  to colora-sedia-vicina;la variante del programma e che ogni persona cerca la
       sedia immediatamente vicina e libera subito dopo la prima vicina gia
      occupata
2
3    ;questo funziona
4    if count sedie with [color = green and libero? = true] = 0 [
5      let i 0
6      while [count sedie with [color = green and libero? = true] = 0][;sto
      ciclo cicla finche non vengono colorate sedie
7        ifelse i < 100 [;se per 100 volte non trova una sedia, prova a
      diminuire la repulsione alla linea 175
8        print "wow"
9         set i (i + 1)
10       prova;funzione che determina il prossimo da colorare di verde
11       ][
12         ifelse repulsione > 0 [set repulsione (repulsione - 1)
13        set i 0
14         colora-sedia-vicina][
15         set repulsione (repulsione + 2)
16        set i 0
17         colora-sedia-vicina
18         ]
19
```

```
      ]
      ]

  ]

end
```

3.3.7 Prova

La procedura *prova* viene richiamata dalla procedura precedente, provando ad a effettuare un processo di query, selezionando prima le sedie candidate, andando a predisporre come unica sedia candidata quella più vicina al palco.

```
to prova

  let nearest-neighbor one-of sedie with [libero? = true and color != green
    ]; inusato ora

 let candidati sedie with [libero? = true and color != green and not any?
    other sedie with [color = green] in-radius repulsione ];seleziono tutti
     i possibili candidati a distanza 3, da metterci anche se non voglio
    davanti nessuno
 let nearest-neighbora (min-one-of (other candidati) [distance one-of palchi
    ]);seleziono il minimo e lo coloro

  carefully [
    ask nearest-neighbora [;qua va messo un carefully per diminuire la
    repulsione
    set color green
    ]
  ][
```

```
13    set repulsione (repulsione - 1 )
14    ]
15
16    ;ask nearest-neighbor [
17    ;  set color green
18    ;]
19  end
```

3.3.8 Go

La procedura **go** va a selezionare una persona in modo randomico, per simulare una corsa verso le sedie, e permette a tale persona di avvicinarsi alla sedia candidata e di occuparla in sua vicinanza.

```
1  to go ;rimane solo un bug per la quale se vengono create due sedie troppo
       vicine, allora se si avvicina uno, le prende tutte e due
2
3
4    if not any? persone with [seated? = false];questo if parte se tutte le
       persone si sono sedute, rispettando il vincolo che ci siano i posti
5      [stop]
6
7    if count persone with [seated? = false] > 0 [;se il conteggio delle
       persone che non si sono ancora sedute e' > 0 allora coloro ancora sedie
8      colora-sedia-vicina
9    ]
10
11   ask one-of persone with [seated? = false][
12     face one-of sedie with [color = green and libero? = true]
13     fd 1
14
```

```
15    let current-persona self
16    ask sedie with [color = green and libero? = true and distance current-
      persona < 0.5 ][
17      set libero? false ;occupo la sedia
18      set hidden? true
19
20      ask current-persona [
21        set seated? true; mi siedo
22        face one-of palchi
23      ]
24    ]
25
26  ]
27
28 end
```

3.3.9 Set-persona-down

La procedura **set-persona-down** permette di andare a creare una persona disponendola sullo schermo alle rispettive coordinare specificate dalla posizione del puntatore del mouse. La persona viene creata al click del tasto sinistro.

```
1
2 to set-persona-down;procedura da capire che permette di far spawnare le
      persone alla pressione del mouse
3   if mouse-down?
4   [
5     if count persone = count sedie [
6       ;print "non puoi creare persone se non ci sono abbastanza sedie"
```

```
7       user-message ("non puoi creare persone se non ci sono abbastanza sedie
        , continua con la simulazione")
8       stop
9     ]
10    if timer > 0.5[
11      ask patch (round mouse-xcor) (round mouse-ycor)
12      [
13        sprout-persone 1
14        [
15          setxy (round xcor) (round ycor)
16
17          ;disegna sul display l'area coperta dal raggio (in trasparenza)
18          set color red
19          set shape "person"
20          set seated? false
21          if any? palchi [
22            face one-of palchi
23          ]
24          ;face one-of palchi
25        ]
26      ]
27
28    reset-timer
29    ]
30      ;stop
31
32  ]
33 end
```

3.3.10 Set-sedia-down

Omonima procedura uguale alla precedente, solo che invece di creare persone, crea delle sedie libere. Serve in particolar modo per la creazione di mondi personalizzati.

```
1  to set-sedia-down;procedura da capire che permette di far spawnare le
       persone alla pressione del mouse
2    if mouse-down?
3    [
4      if timer > 0.5[
5        ask patch (round mouse-xcor) (round mouse-ycor)
6        [
7          sprout-sedie 1
8          [
9            setxy (round xcor) (round ycor)
10
11           ;disegna sul display l'area coperta dal raggio (in trasparenza)
12           set libero? true
13         ;set shape "arrow"
14         set color yellow
15           if any? palchi [
16             face one-of palchi
17           ]
18           ;face one-of palchi
19         ]
20       ]
21
22     reset-timer
23     ]
24       ;stop
```

```
25
26   ]
27 end
```

3.3.11 Salva-modello

Procedura che permette il salvataggio di un mondo personalizzato in formato txt. Richiamata dall'opposito bottone spiegato in precedenza, richiede la specifica dall'utente di un nome e di un'opportuna estensione.

```
1  to salva-modello
2    ;funzione salva mondo con export-world "filename.txt"
3
4    let file user-new-file
5
6    if ( file != false )
7    [
8       export-world file
9    ]
10 end
```

3.3.12 Carica-modello

Procedura che permette il caricamento di un mondo personalizzato in formato txt. Richiamata dall'opposito bottone spiegato in precedenza, richiede la specifica dall'utente di un nome e di un'opportuna estensione.

```
1  to carica-modello
2    ;funzione carica mondo con import-world "filename.txt"
3    let file user-file
4
```

```
  if ( file != false )
  [
    import-world file
  ]
end
```

3.3.13 Crea-persone-personale

La procedura *crea-persone-personale* permette di creare un numero di persone specificato dall'utente. A differenza della precedente funzione spiegata nelle sottosezioni precedenti, tale funzione crea solamente le persone nelle configurazioni personalizzate, ovvero quelle destinate ad un salvataggio o ad una prova su una disposizione appena creata.

```
to crea-persone-personale
  ifelse (count persone + numpersone) <= count sedie [
  crea-persone
  ][
  user-message ("non puoi creare persone se non ci sono abbastanza sedie,
    continua con la simulazione")
      stop
  ]
end
```

3.3.14 set-spawn-point

La procedura *set-spawn-point* permette di selezionare un punto nel mondo dalla quale verranno create un numero di persone pari al numero selezionato dall'apposito slider "numpersone".

```
to set-spawn-point
  if mouse-down?
  [
    if (count persone) + numpersone >= count sedie [
      ;print "non puoi creare persone se non ci sono abbastanza sedie"
      user-message ("non puoi creare persone se non ci sono abbastanza sedie
    , continua con la simulazione")
      stop
    ]
    if timer > 0.5[
      ask patch (round mouse-xcor) (round mouse-ycor)
      [
        sprout-persone numpersone
        [
          setxy (round xcor) (round ycor)
          ;disegna sul display l'area coperta dal raggio (in trasparenza)
          set color red
          set shape "person"
          set seated? false
          if any? palchi [
            face one-of palchi
          ]
          ;face one-of palchi
        ]
      ]
    reset-timer
    ]
   ;stop
  ]
end
```

3.4 Commento sul codice

Ovviamente tutte le funzioni interagiscono tra di loro durante l'esecuzione della simulazione, inoltre, si evidenzia come alcune di loro vengano richiamate a cascata, come anche visibile nei codici elencati in precedenza.

Le funzioni cercano di emulare in modo fedele il comportamento della folla in cerca di un posto accordandosi ad alcuni parametri specificati dall'utente.

Ovviamente, il comportamento delle persone è analogo a quello delle persone reali, cercando sempre di muoversi coerentemente con un movimento veritiero, prestando pertanto attenzione alle altre persone e/o ad oggetti quali sedie presenti nella disposizione.

4 Conclusioni

Viene mostrata come l'interazione degli individui nella scelta del posto possa influenzare la disposizione degli stessi all'interno del programma. Si evidenzia come tutte le persone corrano verso un posto definito migliore, accontentandosi alla fine di una visione migliore in mancanza di posti con una distanza adeguata tra loro, anche studiandolo nelle diverse disposizioni create precedentemente con la sezione delle disposizioni personalizzate, muovendosi in accordo allo spazio di movimento disponibile predefinito dal creatore della disposizione.

Inoltre, si nota anche come, una volta che le sedie a distanza maggiore siano state già occupate, l'agente si rassegna, preferendo quindi una visibilità

maggiore, rispetto ad una repulsione dai vicini. Questo emula perfettamente quello che è il comportamento reale di una folla in cerca di un posto.