

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

Scuola di Scienze
Dipartimento di Fisica e Astronomia
Corso di Laurea in Fisica

**ANALISI COMPARATIVA DI
SIMULAZIONI PHET DI SISTEMI
MECCANICI CLASSICI E SIMULAZIONI
NETLOGO DI SISTEMI COMPLESSI**

Relatrice:
Prof.ssa Olivia Levrini

Presentata da:
Roberta Pia Castelli

Correlatrice:
Dott.ssa Eleonora Barelli

Anno Accademico 2020/2021

Desidero ringraziare la Prof.ssa Olivia Levrini, mia relatrice, e la dott.ssa Eleonora Barelli, sempre pronte a rispondere ad ogni mio dubbio.

Un particolare ringraziamento va a tutta la mia famiglia per il sostegno.

Abstract

L'introduzione di tecnologie digitali nella didattica è un tema che, da sempre, divide la comunità educante tra quanti rimangono ancorati a una didattica più tradizionale, spesso frontale e nozionistica, a quanti invece sono disposti a sperimentare nuovi approcci. Nell'ambito delle scienze in generale e della fisica in particolare, tra gli strumenti che stanno progressivamente entrando a far parte dei metodi d'insegnamento vi sono le simulazioni, di cui questa tesi tratta. La tesi parte affrontando la nozione di simulazione per poi distinguere tra simulazioni utilizzate nella pratica scientifica e simulazioni utilizzate invece in didattica come laboratori virtuali che consentono di osservare su schermo fenomeni in condizioni controllate e di eseguire esperimenti computazionali basandosi su modelli. Successivamente si tratta della distinzione tra simulazioni di sistemi classici e simulazioni di sistemi complessi, analizzando il diverso ruolo che il laboratorio virtuale assume nei due tipi di sistemi. Basandosi sulla letteratura di ricerca in didattica, si individuano i vantaggi e le criticità che si presentano, negli studenti, quando entrano in contatto con i due tipi di simulazione. Il focus della tesi consiste nell'analisi di alcune simulazioni di sistemi meccanici classici e di sistemi complessi utilizzando rispettivamente le piattaforme PhET e NetLogo. Per esplicitare le profonde differenze che esistono tra i sistemi meccanici classici, alla base delle simulazioni PhET, ed i sistemi complessi, alla base delle simulazioni NetLogo, viene costruito uno strumento di analisi in forma di tabella. Ogni simulazione è stata analizzata nel dettaglio, dagli aspetti di visualizzazione fino al codice sottostante. La tabella ha permesso di evidenziare che, mentre le simulazioni PhET adottano un approccio deterministico, con un chiaro rapporto di causa-effetto, quelle NetLogo adottano un approccio probabilistico. La conoscenza profonda di come ogni simulazione connetta l'input con l'output e l'eventuale presenza di elementi probabilistici, consentirebbe ai docenti di progettare guide strategiche per ridurre le criticità nell'uso delle simulazioni in didattica.

Indice

Introduzione

Capitolo 1: Le simulazioni nella didattica della fisica e delle scienze

- 1.1 Verso una caratterizzazione di simulazione in ambito scientifico
- 1.2 L'utilizzo delle simulazioni in ambito educativo
- 1.3 Risultati di ricerca sulle potenzialità delle simulazioni nella didattica e sulle difficoltà incontrate dagli studenti
- 1.4 Il progetto PhET

Capitolo 2: Le simulazioni per affrontare i sistemi complessi

- 2.1 Verso una caratterizzazione di sistema complesso
- 2.2 Simulare i sistemi complessi: approcci equation-based e agent-based
- 2.3 Risultati di ricerca sull'utilizzo delle simulazioni di sistemi complessi in ambito educativo e sulle difficoltà incontrate dagli studenti
- 2.4 Il progetto NetLogo

Capitolo 3: Analisi comparativa di simulazioni Phet di meccanica classica e simulazioni NetLogo di sistemi complessi

- 3.1 Analisi di simulazioni PhET di meccanica classica
 - 3.1.1 Masse e molle
 - 3.1.2 Conservazione dell'energia meccanica
 - 3.1.3 Forze sul piano inclinato
 - 3.1.4 Leve
- 3.2 Analisi di simulazioni NetLogo di sistemi complessi
 - 3.2.1 Predazione
 - 3.2.1 Segregazione
 - 3.2.3 Cambiamento climatico
 - 3.2.4 Traffico
- 3.3 Analisi comparativa

Conclusioni

Introduzione

Dal Marzo 2020, a causa dell'emergenza epidemiologica da Covid-19, la didattica nelle scuole e nelle università ha subito una rivoluzione. Infatti, i Paesi di tutto il mondo hanno investito rapidamente in soluzioni di didattica e apprendimento a distanza erogate attraverso diversi canali, tra cui piattaforme online. In questo senso, la pandemia ha contribuito a rendere visibile un tema che da diversi anni si sta definendo, relativamente alle nuove tecnologie nella pratica scolastica. Quindi il loro utilizzo, che prima era spesso considerato importante per alcuni istituti ma non indispensabile per altri, è stato riconosciuto come fondamentale e necessario a tutti i livelli di studio. Tra gli strumenti che possono accompagnare questa rivoluzione nella didattica vi sono le simulazioni, di cui questa tesi tratta.

Tuttavia, le simulazioni non sono solo strumenti didattici ma nella pratica scientifica vengono considerate il "terzo pilastro" della ricerca. Infatti, con la diffusione di computer potenti e veloci, le simulazioni sono sempre più utilizzate nell'ambito della ricerca a partire dalle scienze naturali, come la fisica, finanche alle scienze sociali, passando per moltissimi ambiti del sapere. Tra gli scopi delle simulazioni vi è il fatto che possono essere considerate "laboratori virtuali" in cui lo scienziato osserva fenomeni in condizioni controllate, agisce su variabili e osserva gli effetti delle sue manipolazioni.

Sebbene vi siano innegabili differenze tra le simulazioni utilizzate come strumento di ricerca nella pratica scientifica e le simulazioni comunemente usate a scopo didattico, anche per le simulazioni didattiche è molto importante il concetto di laboratorio virtuale. Infatti, le simulazioni possono essere usate didatticamente usando un approccio simile ai laboratori sperimentali, con numerosi vantaggi. La letteratura di ricerca evidenzia che le simulazioni possono risultare accattivanti e coinvolgenti per molti studenti, facilitando l'approccio alla fisica e ad altre discipline e la comprensione concettuale di alcuni temi. Il ruolo attivo dello studente nel processo di apprendimento è facilitato dall'utilizzo di questo strumento che è basato sul concetto di interazione dello studente con il mezzo tecnologico. Inoltre, il loro utilizzo risulta poco costoso e alla portata di tutti gli istituti scolastici. Come vedremo, infatti, esistono piattaforme gratuite che mettono a disposizione di studenti e insegnanti molte simulazioni.

In questa tesi affronteremo l'utilizzo di simulazioni di sistemi classici (PhET) e di sistemi complessi (NetLogo) nella didattica della fisica, cercando di individuare le competenze che sono richieste agli studenti per affrontare i due tipi di simulazioni, le criticità che possono sorgere quando gli studenti entrano in contatto con esse e quali sono i vantaggi nel loro utilizzo. Cercheremo, inoltre, di individuare dei criteri di confronto non solo tra i su citati linguaggi di programmazione dei due tipi di simulazioni ma, a partire dalle simulazioni, cercheremo di evidenziare categorie più ampie di confronto tra i sistemi meccanici classici, che sono alla base delle simulazioni PhET, e i sistemi complessi, su cui si basano le simulazioni NetLogo.

La tesi è strutturata in tre capitoli. Nel capitolo 1 esponiamo la definizione di simulazione in ambito scientifico e alcune sue applicazioni in ambito didattico, presentando i risultati di ricerca in didattica delle scienze relativi ai loro punti di forza e alle criticità evidenziate. Infine, viene presentato il progetto PhET che verrà approfondito nel terzo capitolo.

Nel capitolo 2 introduciamo il tema dei sistemi complessi e il ruolo che le simulazioni hanno nella loro modellizzazione e risoluzione. Anche in questo caso, vengono riportati alcuni esempi della letteratura di ricerca in didattica in cui si è fatto uso delle simulazioni di sistemi complessi individuandone potenzialità e difficoltà. Il capitolo termina presentando il progetto NetLogo che verrà approfondito nel terzo capitolo.

Il capitolo 3 si concentra sull'analisi di quattro simulazioni PhET di meccanica classica (masse e molle; conservazione dell'energia meccanica; Forze sul piano inclinato; Leve) e di altrettante simulazioni NetLogo di sistemi complessi (Predazione; Segregazione; Cambiamento climatico; Traffico). Oltre a descriverle, presentiamo una griglia di analisi comparativa costruita per poter identificare le principali differenze che emergono tra sistemi classici e simulazioni di sistemi complessi e come queste possano essere riconosciute interagendo con le rispettive simulazioni.

Capitolo 1

Le simulazioni nella didattica della fisica e delle scienze

In questo capitolo, commentiamo il ruolo che le simulazioni ricoprono nella didattica delle scienze e della fisica. Per farlo abbiamo raccolto e studiato vari articoli della letteratura di ricerca in didattica. Nel primo paragrafo definiamo la simulazione e l'importanza che essa ricopre in ambito scientifico, procedendo, nel secondo paragrafo, con l'utilizzo che si fa di essa in ambito educativo e la presentazione di alcuni laboratori virtuali. Nel terzo paragrafo, esponiamo le potenzialità e le difficoltà emerse nel loro utilizzo nella didattica. In conclusione, viene presentato il progetto PhET e i principi didattici su cui si basa.

1.1 Verso una caratterizzazione di simulazione in ambito scientifico

Nel linguaggio comune il termine “simulare” viene percepito come un termine negativo, infatti, nella definizione comune si intende l'atto di fingere o imitare un comportamento. Tuttavia, in ambito tecnico e scientifico, e in particolare nella teoria dei sistemi, “simulare” significa riprodurre il comportamento di un determinato sistema, processo o fenomeno tramite un modello che permette di variare i parametri del sistema o dell'ambiente in maniera più semplice rispetto al sistema che si vuole analizzare. A tal fine, nel corso dei secoli, sono stati realizzati dispositivi in grado di fare delle previsioni su problemi specifici: i simulatori. Esempi di simulatori sono la macchina di ANTIKITHERA (ὁ μηχανισμός τῶν Ἀντικυθήρων, I sec. a. C.), un meccanismo ad ingranaggi costruito per prevedere il moto dei pianeti, oppure la macchina FERMIAC (1946), che studia le reazioni di decadimento nucleare, o ancora la galleria del vento, in grado di calcolare come l'aria si distribuisce intorno ad una automobile o un aeroplano. Con l'avvento dei calcolatori elettronici, essi sono diventati i nuovi simulatori e le simulazioni con essi elaborate -dette simulazioni computazionali- sono diventate uno strumento di analisi molto potente, largamente utilizzato in ambito scientifico. Quello per la definizione di che cosa sia una simulazione scientifica per darne una caratterizzazione rigorosa è un dibattito che va avanti da alcuni decenni nella comunità

dei filosofi della scienza (Grüne-Yanoff & Paul Weirich, 2010).

Una delle prime definizioni di simulazione fu data da Humphreys nel 1991: “Una simulazione è un metodo implementato al computer per esplorare le proprietà di un modello matematico laddove non siano disponibili metodi analitici” (Humphreys, 1991). Secondo questa definizione una simulazione è un metodo computazionale che ha il ruolo di “esplorare” un modello quando non è possibile utilizzare un metodo analitico per analizzarlo.

Nel 1996 Hartmann affermò che: “Una simulazione imita un processo tramite un altro processo” (Hartmann, 1996). Questa definizione, diversamente dalla precedente fornita da Humphreys, concerne l'evoluzione temporale del sistema, escludendo i fenomeni stazionari che invece erano inclusi nella caratterizzazione di Humphreys. Secondo tale definizione, la simulazione fornisce quindi l'evoluzione del sistema in tempo reale, risolvendo le equazioni che si riferiscono al modello.

Una definizione più completa fu data da Humphreys nel 2004, a completamento di quella formulata 15 anni prima: “Un sistema S offre una core simulation di un oggetto o di un processo B se e solo se S è un dispositivo computazionale concreto che produce, tramite un processo temporale, soluzioni per un modello computazionale che rappresenta correttamente B, o dinamicamente o staticamente. Se, in più, il modello computazionale usato da S rappresenta correttamente la struttura di un sistema reale R, allora S fornisce una core simulation di un sistema R rispetto a B” (Humphreys, 2004). Questa definizione risulta particolarmente efficace in quanto ci permette di separare tre aspetti: il sistema simulante (S), un modello di processo (B) e un sistema reale (R). Secondo questa definizione la simulazione è sempre un processo temporale, tuttavia non è detto che il sistema simulato sia sempre dinamico: infatti essa potrebbe anche rappresentarne la struttura in modo statico. A differenza delle definizioni presentate, quest'ultima individua il principale obiettivo della simulazione nella produzione di soluzioni per il modello computazionale (S) di un sistema, basandosi su un modello del sistema stesso (R). Possibili soluzioni del sistema possono essere, ad esempio, le sequenze di valori numerici caratteristici del sistema nel tempo, oppure lo stato finale in cui un sistema evolve fino a raggiungere una situazione di equilibrio. Nel capitolo 2 di questa tesi commenteremo, utilizzando esempi specifici, come lo scopo risolutivo della simulazione diventa cruciale quando si vanno ad affrontare i sistemi complessi.

Sulla base delle definizioni riportate in questo paragrafo, possiamo affermare che in ambito scientifico le simulazioni sono strumenti che, basandosi sulla costruzione di modelli computazionali a partire da modelli teorici di tipo fisico o matematico, mirano a una migliore comprensione di fenomeni e processi del mondo reale.

1.2 L'utilizzo delle simulazioni in ambito educativo

A differenza delle simulazioni utilizzate nella pratica scientifica, le simulazioni educative possono essere definite come ambienti di apprendimento interattivi in cui un modello simula le caratteristiche di un sistema a seconda delle azioni compiute dallo studente (Kirchner & Huisman, 1998). Esse mirano a sviluppare una comprensione del modello base e dei principi teorici che ne derivano. È possibile dividere in due categorie le simulazioni utilizzate in ambito educativo (Greca et al., 2014):

- 1) Laboratori virtuali: utilizzati per osservare su schermo fenomeni in condizioni controllate, per manipolare variabili e osservare gli effetti della manipolazione. Questi laboratori virtuali possono essere utilizzati sia a livello scolastico che per la ricerca scientifica. Inoltre, i laboratori virtuali presentano vari vantaggi pratici come la sicurezza, l'economicità e l'indipendenza dallo spazio e dal tempo (un esempio è nel Planetario Virtuale 1.2.1), la possibilità di fare pratica senza vincoli di spazio e tempo consente di provare e riprovare, fare errori e verificare ipotesi alternative. In questa fase il modello è già costituito e il laboratorio virtuale permette allo studente di esplorarlo.
- 2) Simulazioni di fenomeni: usate per modellare un sistema o un processo. In questa modalità lo studente ha la possibilità di modellare fenomeni, attraverso l'uso di equazioni, e di combinare la rappresentazione di oggetti con rappresentazioni analitiche, analogiche e grafiche. Questo è possibile tramite programmi come Modellus e Easy Java Simulator. In alternativa, un sistema di modellizzazione più semplice è STELLA15 che consente agli studenti di modellare sistemi e genera automaticamente le relazioni matematiche.

In questa tesi ci dedicheremo principalmente alle simulazioni del primo tipo, ovvero i "laboratori virtuali" mostrando alcuni esempi molto studiati nella letteratura di ricerca in didattica della fisica (López & Pintó, 2017; Smetana & Bell, 2012).

A seguire, mostriamo due esempi di simulazioni in ambito fisico che ci saranno

particolarmente utili nella prossima sezione per descrivere difficoltà incontrate dagli studenti con esse.

- **Simulazione sull'attrito "Friction"**

Alcune tra le più famose simulazioni progettate ad hoc per la didattica delle scienze sono disponibili al sito <https://phet.colorado.edu/it/>. Tra esse vi è, ad esempio, "Friction" (Figura 3).

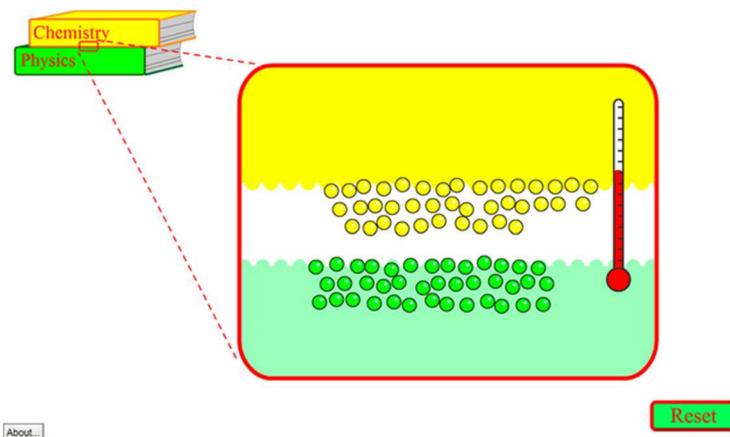


Figura 1 Applet "Friction", rappresenta la relazione tra attrito e calore

La simulazione intende rappresentare la relazione tra attrito e aumento di temperatura di un corpo. Si può notare come la simulazione mostri una rappresentazione sia macroscopica, data dai libri uno sull'altro (in alto a sinistra), e una rappresentazione a livello molecolare rappresentata dai due gruppi di particelle al centro dell'immagine accompagnate da un termometro sulla destra. Muovendo i libri l'uno contro l'altro, a causa delle scabrosità, si può osservare come vibrino e si scontrino le particelle facendo aumentare l'energia cinetica del moto molecolare e di conseguenza la temperatura del corpo.

- **Simulazione sulla legge di Faraday "Faraday's law"**

La simulazione "Faraday's law" (Figura 4) intende mostrare l'interazione elettromagnetica tra una bobina e un magnete. Muovendo il magnete all'interno o intorno alla bobina, viene prodotta corrente elettrica, rappresentata dalla lampadina che si accende e contemporaneamente dall'ago del voltmetro che si sposta.

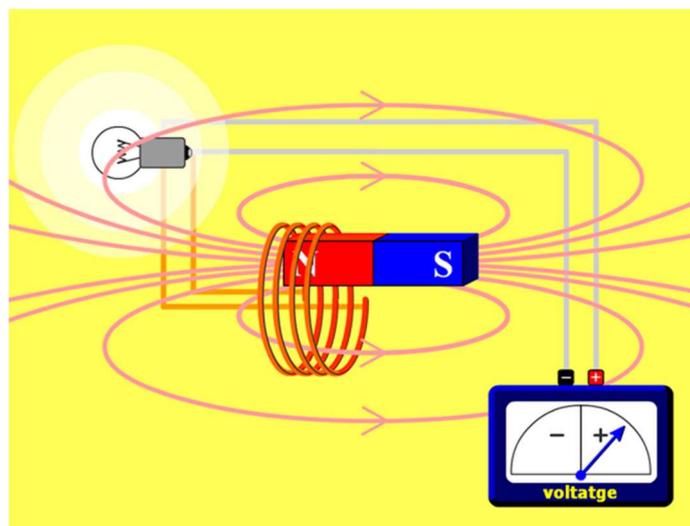


Figura 2 Applet "Faraday's law", rappresenta l'induzione elettromagnetica

1.3 Risultati di ricerca sulle potenzialità delle simulazioni nella didattica e sulle difficoltà incontrate dagli studenti

Le simulazioni al computer sono modelli e processi del mondo reale realizzati al computer. Queste sono oggetto di ricerca per dimostrare come e in quali condizioni il loro utilizzo nella didattica delle scienze faciliti l'interattività dello studente rinforzando la pratica, il problem-solving e l'higher-order thinking.

La maggior parte delle simulazioni utilizzate nelle scuole sono state progettate da insegnanti specializzati o ricercatori che hanno cercato, tramite esse, di rendere i concetti scientifici più comprensibili.

Nonostante tale intento, la letteratura di ricerca in didattica ha individuato diversi problemi che gli studenti incontrano nel "leggere" le simulazioni. Molte di queste difficoltà riguardano la delicata relazione tra il significante (immagine acustica o visiva: parola scritta o pronunciata) e il significato (contenuto concettuale del significante). Nel caso delle simulazioni, per significante intendiamo la "superficie materiale" della simulazione, quella che un utente può osservare esplicitamente, mentre con significato intendiamo il livello sottostante a tale rappresentazione esplicita, che potremmo definire il "contenuto concettuale" della simulazione stessa. Dallo studio (Amettler & Pintó, 2002) è emerso che spesso gli studenti hanno difficoltà a capire la struttura compositiva delle immagini raffigurate, cioè la disposizione visiva dei diversi elementi della composizione,

o la disposizione sintattica e la relazione tra i diversi elementi visivi, come la disposizione sinistra-destra o su-giù.

Nell'articolo di ricerca, "Identifying secondary-school students' difficulties when reading visual representations displayed in physics simulations" (López & Pintó, 2017), si affronta la classificazione delle difficoltà che incontrano gli studenti di scuola secondaria di secondo grado per poterle prevenire in un contesto di insegnamento.

Le difficoltà incontrate dagli studenti sono state categorizzate in:

- Difficoltà nella lettura della struttura compositiva della simulazione (CS)
- Difficoltà nel dare adeguata rilevanza ad ogni elemento visivo (RE)
- Difficoltà nel dare l'adeguato significato semantico per ogni elemento visivo (SEM)
- Difficoltà nel trattare rappresentazioni multiple (MR)
- Difficoltà di percepire e interpretare informazioni dinamiche (DY)

Ognuna di queste categorie è stata divisa in sottocategorie.

Nella categoria Struttura Compositiva (CS) si possono individuare due sottocategorie: la prima (CS1) racchiude il fraintendimento del rapporto tra elementi visivi diversi della composizione che sono posti vicini nel piano bidimensionale dello schermo. Un esempio può essere visto nella simulazione "Friction" (figura 3) in cui alcuni studenti hanno interpretato le etichette dei libri come struttura compositiva della simulazione indicando "atomi di chimica" e "atomi di fisica". Questo accade perché gli studenti non danno il giusto significato alla struttura compositiva della rappresentazione confondendo il rapporto che c'è tra il lato sinistro (rappresentazione macroscopica dei libri) e il lato destro (rappresentazione molecolare delle superfici di contatto tra i libri). La seconda (CS2) racchiude il fraintendimento della relazione tra i diversi elementi che sono sovrapposti nei diversi strati della simulazione. Ad esempio, nell'utilizzo della simulazione "Faraday's law" (figura 4) alcuni studenti hanno immaginato che le due bobine fossero la parte più alta e più bassa di un toroide affermando che il magnete dovesse essere spostato con movimenti circolari.

L'articolo mostra che alcuni studenti tendono a dare una rilevanza errata ad alcuni elementi: spesso accade che si focalizzino eccessivamente su elementi, in realtà,

irrilevanti oppure trascurino elementi fondamentali. Questo tipo di difficoltà va a costituire le sottocategorie delle difficoltà collegate alla Rilevanza degli Elementi visivi (RE), corrispondenti ciascuna a uno degli elementi irrilevanti ma erroneamente considerati importanti: RE1 colore degli elementi visivi, RE2 forma, RE3 posizione e RE4 etichette verbali incorporate. Ad esempio, nella simulazione “Faraday’s law” (figura 4) alcuni studenti hanno enfatizzato il colore rosso e blu del magnete come se fosse un’informazione rilevante rientrando nella categoria RE1; inoltre è stato individuato che la bobina, nella concezione degli studenti, è un cilindro cavo e non un filo a spirale: questo tipo di affermazione va a collocarsi nella sottocategoria RE2. Alcuni esempi di RE3 e RE4 sono stati individuati nella simulazione “Friction” (figura 3) in cui alcuni studenti si sono concentrati principalmente su quello che accadeva al centro dello schermo senza notare i libri raffigurati nell’angolo; altri, invece, sono stati forviati dalle etichette dei libri fino al punto di pensare che la simulazione trattasse di una reazione chimica poiché appariva il termine “chimica” su uno dei libri.

Le difficoltà relative al Significato Semantico di ogni elemento visivo (SEM) sono state divise in due categorie. Le prime, SEM1, sono state individuate quando il significato che gli studenti danno ad un elemento della rappresentazione corrisponde al significato di un’altra rappresentazione. Un esempio di ciò è dato dalla risposta di uno studente sulla spiegazione del voltmetro nella simulazione “Faraday’s law” (figura 4): qui, egli afferma che il voltmetro è una macchina che rileva la forza del magnete facendo capire che in realtà sta pensando alla forma di una bussola. La seconda categoria, SEM2, è individuata quando gli studenti mostrano mancanza di conoscenze che non permettono loro di dare il giusto significato agli elementi. Ad esempio, alcuni studenti hanno dichiarato di non conoscere il significato del voltmetro dando soluzioni alternative come “una macchina che regola la corrente per evitare l’esplosione della lampadina”.

Per quanto concerne le difficoltà nel trattare Rappresentazioni Multiple (MR), sono state individuate tre sottocategorie. Le MR1 sono difficoltà riguardanti l’integrazione di due pezzi di informazioni sorte quando, ad esempio nella simulazione “Friction” (figura 3), gli studenti hanno dovuto combinare la rappresentazione particellare e continua della materia. La difficoltà MR2 consiste in una combinazione errata di due informazioni complementari; ad esempio, uno studente ha affermato che la lampadina della simulazione “Faraday’s law” (figura 4) si accende solo quando la tensione è positiva. La

terza sottocategoria, MR3, consiste in una mancanza di discriminazione di due informazioni diverse. Ad esempio, alcuni studenti hanno collegato direttamente la lampadina alle linee del campo magnetico.

Anche la categoria relativa all'Interpretazione di Informazioni Dinamiche (DY) viene divisa in quattro sottocategorie. La DY1 riporta la confusione degli studenti tra il comportamento dipendente dalla posizione e quello dipendente dal movimento; a questa categoria appartengono le difficoltà mostrate nella simulazione "Friction" (figura 3), secondo cui si crede che le due superfici si scaldino a causa della posizione di contatto e non a causa del movimento. Nella sottocategoria DY2 si riporta la confusione tra i processi simultanei e rappresentati in successione; ad esempio, per la simulazione "Faraday's law" (figura 4) sono state fatte affermazioni del tipo "la corrente passa prima nel voltmetro e poi nella lampadina". La DY3 racchiude le difficoltà degli studenti nell'imporre linearità a cambiamenti e situazioni non lineari; ciò accade per la temperatura nella simulazione "Friction" (figura 3) la quale, secondo uno studente, diminuisce a velocità costante. L'ultima sottocategoria DY4 si riferisce alla mancanza di percezione dell'intensità di una rappresentazione dinamica; ad esempio, nella simulazione "Faraday's law" (figura 4) alcuni studenti non hanno fatto alcun riferimento ai livelli di voltaggio o alla diversa intensità luminosa della lampadina facendo emergere che non hanno notato come più fili nella bobina o più velocità nel muovere il magnete comportino una maggiore o minore corrente elettrica indotta.

Nonostante le difficoltà sopraelencate, alcune recenti ricerche mostrano dei vantaggi nell'utilizzo di laboratori virtuali. Infatti, gli studenti possono lavorare con i dati in un ambiente controllato avendo maggior controllo sulle variabili, cosa che generalmente non è possibile nei laboratori standard. Inoltre, l'utilizzo di laboratori virtuali non solo permette un risparmio di tempo sia nella configurazione dell'esperimento sia nell'organizzazione dei risultati di output ma è anche in grado di rendere visibili fenomeni non osservabili. Nello studio (Chinn & Malhotra, 2002) vengono categorizzati i vantaggi delle simulazioni in ambito educativo:

- Esperimenti a livello teorico del meccanismo: lo studio di entità teoriche che altrimenti non potrebbero essere visualizzate;
- Diversi tipi di esperimenti su un singolo caso;

- Progetti scientifici relativamente complessi.

Nella review “Computer Simulations to Support Science Instruction and Learning: A critical review of the literature”(Smetana & Bell, 2012), come si evince dal titolo, sono stati raccolti e catalogati articoli di ricerca dal 1972 al 2010 per poter presentare una revisione attuale sull’impatto delle simulazioni nell’insegnamento e nell’apprendimento delle scienze. Gli articoli sono stati divisi in quattro categorie:

- “Promoting Content knowledge” (impatto sull’acquisizione di contenuti)
- “Promoting Science Process Skills” (impatto sulle abilità degli studenti di applicare il ragionamento scientifico)
- “Promoting Conceptual Change” (impatto sui cambiamenti concettuali)
- “Pedagogical issues in using computer simulation” (problemi pedagogici nell’uso delle simulazioni).

In questa analisi della lettura è emerso che la combinazione tra metodi didattici classici e simulazioni al computer dà ottimi risultati. Un esempio si può trovare nella sperimentazione di didattica della biologia (Akpan & Andre, 2000) dove sono stati comparati quattro diversi approcci nella dissezione di una rana:

- SBD (Simulation before dissection)
- DBS (Dissection before simulation)
- SO (Simulation only)
- DO (Dissection only).

I risultati dei test somministrati hanno evidenziato che gli studenti che hanno utilizzato le simulazioni prima della dissezione (SBD) e le sole simulazioni (SO) hanno ottenuto punteggi migliori rispetto agli studenti che hanno utilizzato la simulazione dopo la dissezione (DBS) e la sola dissezione (DO). I risultati positivi ottenuti da SBD e SO hanno permesso agli autori della ricerca di concludere che la simulazione utilizzata come introduzione all’esperimento ha permesso di creare un ambiente di apprendimento semplificato in cui gli studenti hanno potuto farsi un’esperienza di base su cui edificare una conoscenza più complessa. Mentre i risultati negativi di DBS hanno suggerito che gli studenti sono stati meno partecipi alla simulazione e l’hanno usato solo come conferma. Gli autori hanno concluso che l’utilizzo delle simulazioni prima dell’esperienza di laboratorio può contribuire alla comprensione teorica dell’esperimento.

Lo studio di (Chang et al., 2008) ha aggiunto che le simulazioni devono essere affiancate da alcuni supporti come informazioni generali sull'indagine, assistenza nella formulazione e nella revisione delle ipotesi e da una guida step-by-step durante l'indagine. Lo studio di (Liu et al., 2008) ha mostrato quanto gli studenti usino le simulazioni in base alle loro precedenti conoscenze: gli studenti con un livello di competenze più alto utilizzano le simulazioni come conferma di quanto già conoscono, mentre quelli con un livello di competenze più basso le usano come risorsa primaria per risolvere problemi attraverso tentativi ed errori.

In conclusione, la letteratura di ricerca mostra che le simulazioni hanno effetti positivi quando:

- 1) Vengono utilizzate come supporto ad altri metodi didattici: in tal caso, sono un ottimo punto di riferimento per visualizzare, rappresentare e semplificare problemi complessi grazie alla idealizzazione dei modelli.
- 2) Gli studenti dispongono di un buon supporto e guida degli insegnanti: questi tipi di supporto dipendono dagli obiettivi di apprendimento e dalle capacità ed esigenze degli studenti.

Promuovono processi di cambiamento concettuale: le simulazioni permettono agli studenti di confrontare dati ed eventi virtuali che sfidano le loro conoscenze e convinzioni, favorendo così, in alcuni casi, l'innescarsi di una modifica della loro conoscenza.

1.4 Il progetto PhET

PhET Interactive Simulations è un progetto, dell'University of Colorado Boulder, OER (Open Educational Resource) fondato dal premio Nobel Carl Wieman nel 2002 con l'obiettivo di migliorare il mondo dell'insegnamento e dell'apprendimento scientifico. PhET offre simulazioni interattive di scienza e matematica testate e valutate per assicurarne l'efficacia didattica. Le simulazioni sono scritte in Java, Flash o HTML5, quindi, possono essere eseguite online o scaricate sul computer. Tutte le simulazioni vengono realizzate in modo tale che gli studenti possano cimentarsi nel campo delle scienze attraverso l'indagine utilizzando i seguenti principi di progettazione: incoraggiare il metodo dell'indagine scientifica; fornire interattività; rendere visibile l'invisibile; visualizzare modelli concettuali; includere rappresentazioni multiple (per esempio moto dell'oggetto, grafici, valori, ecc.); collegarsi a esperienze e fenomeni del mondo reale;

fornire agli utenti percorsi guidati (per esempio controllo dei fattori limitanti) in una proficua esplorazione; creare una simulazione che possa essere usata con flessibilità in diversi contesti educativi (Wieman et al., 2010).

Le simulazioni PhET sono largamente utilizzate nella ricerca sulla didattica delle scienze ed è possibile trovarne vari utilizzi nella letteratura scientifica. Un esempio è l'articolo "What levels of guidance promote engaged exploration with interactive simulation?" (Adams et al., 2008) in cui vengono analizzate quattro procedure per l'utilizzo delle simulazioni in ambito didattico:

- Simulazione senza nessuna istruzione: in questa modalità l'apprendimento è fortemente legato al tipo di simulazione. Infatti, se essa è ben progettata questo tipo di approccio può essere incredibilmente produttivo. Al contrario, se la simulazione è mal progettata o troppo complessa questo approccio è fallimentare.
- Simulazione accompagnata da domande di guida: anche in questa modalità l'apprendimento è legato alla simulazione in sé. In questa modalità gli studenti esplorano la simulazione concentrandosi soprattutto sulle domande fornite, quindi non esplorano liberamente ma restano legati alle domande. Se la simulazione non è "intimidatoria" o troppo complessa si osservano ottimi risultati nell'apprendimento.
- Simulazione con guida "soft": questa modalità include una serie di domande volte ad aiutare gli studenti a identificare gli elementi presenti sullo schermo. In questa modalità gli studenti difficilmente tendono ad andare oltre le domande a meno che la simulazione non sia eccessivamente coinvolgente. Quindi l'apprendimento è fortemente legato al tipo di domanda.
- Simulazione fortemente guidata: questa modalità fornisce una guida step-by-step all'analisi della simulazione. Essa sembra creare una sorta di barriera tra lo studente e la simulazione. Infatti, per rispondere alle domande, lo studente si basa principalmente sulle conoscenze teoriche acquisite precedentemente. Inoltre, lo studente non esplora quasi per niente la simulazione di sua iniziativa ma tende a rimanere legato alle domande.

Un altro esempio di letteratura che analizza l'utilizzo delle simulazioni PhET è il già citato articolo "Identifying secondary-school students' difficulties when reading visual

representations displayed in physics simulations” (López & Pintó, 2017), in cui si procede, attraverso delle interviste, nell’individuazione delle difficoltà che possono avere gli studenti nell’utilizzo delle simulazioni. Tuttavia, le difficoltà emerse possono essere superate fornendo agli studenti materiale didattico specifico per costruire un ponte tra le simulazioni e i concetti alla base di esse. Ad esempio, risulta utile fornire una guida passo-passo che chieda agli studenti di individuare la struttura e il significato della rappresentazione con domande del tipo:

- Cosa fa la simulazione?
- Cosa vedi in questa immagine?
- Che cosa pensi che significhi questo elemento specifico?

Le domande, inoltre, dovrebbero affrontare anche il livello pragmatico della rappresentazione:

- Perché pensi che gli autori della simulazione abbiano deciso di includere questo elemento visivo in questa rappresentazione?

In conclusione, la letteratura di ricerca mostra che il significato che gli studenti tendono a dare alle informazioni rappresentate nelle simulazioni può distaccarsi completamente dal significato originale atteso dai progettisti della stessa. Quindi, l’utilizzo della simulazione non implica capirne il significato. Infatti, i risultati migliori a livello didattico tendono ad essere raggiunti quando le simulazioni sono accompagnate all’inizio da una guida step-by-step, che può aiutare nella distinzione tra gli elementi rilevanti e decorativi della simulazione o a chiarire informazioni ambigue, e poi, man mano che lo studente acquisisce dimestichezza e impara a dare il giusto peso agli elementi visivi, tale guida diventa più leggera fino ad arrivare all’utilizzo libero della sola simulazione.

Capitolo 2

Le simulazioni per affrontare i sistemi complessi

In questo capitolo, commentiamo il ruolo che le simulazioni ricoprono nell'ambito dei sistemi complessi e nella loro didattica. Per farlo abbiamo raccolto e studiato vari articoli della letteratura di ricerca in didattica. Nel primo paragrafo definiamo il sistema complesso e le sue proprietà. Nel secondo paragrafo illustriamo i due tipi di approcci per simulare i sistemi complessi: equation-based e agent-based. Nel terzo paragrafo, esponiamo i risultati relativi all'utilizzo delle simulazioni di sistemi complessi in ambito educativo e alle difficoltà incontrate dagli studenti. In conclusione, viene presentato il progetto NetLogo.

2.1 Verso una categorizzazione di sistema complesso

Molti dei sistemi che ci circondano sono complessi. Infatti, lo studio dei sistemi complessi è fortemente legato ai progressi che sono stati fatti in diversi settori che vanno dalla fisica all'antropologia. È possibile osservare campi ben separati unificarsi in questa unica disciplina favorendo l'interdisciplinarietà. Un modo iconico di rappresentare questo rapporto del settore dei sistemi complessi con gli altri campi scientifici è indicato in Fig. 5, come suggerito in (Bar-Yam, 1997).

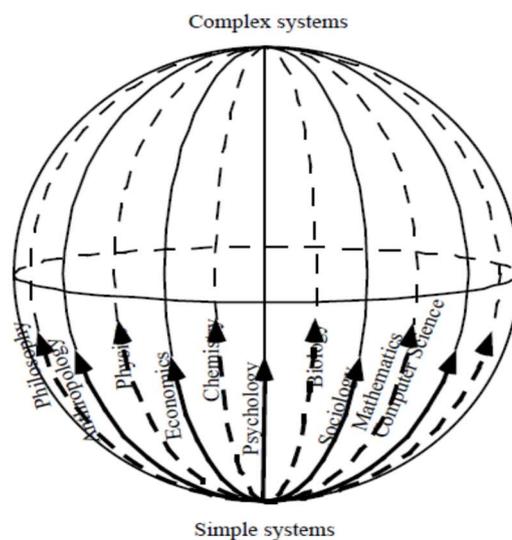


Figura 3 Rapporto tra sistemi semplici dei diversi campi scientifici e i sistemi complessi. Figura tratta da (Bar-Yam, 1997)

Questa visualizzazione suggerisce che tutti i sistemi complessi hanno proprietà universali perché ogni campo sviluppa strumenti per affrontare la complessità dei sistemi nel loro dominio. Questi strumenti possono essere adattati per un uso più generale, riconoscendo così applicabilità universale.

Passando da un'immagine di complessità che riguarda il "sapere complesso" e la sua relazione con le discipline tradizionali, ci spostiamo ora verso una definizione di complessità più adatta ad affrontare la scienza (e la fisica) dei sistemi complessi. Un sistema complesso può essere definito come un sistema per cui è difficile restringere la sua descrizione in un numero di parametri o variabili caratterizzanti senza perderne le proprietà globali. Secondo la definizione data da Cilliers nel 2007: "Un sistema complesso è un sistema composto da diversi elementi individuali (agenti) i quali, interagendo tra loro secondo relazioni non lineari, danno al sistema complesso risultante alcune proprietà che i sistemi classici non hanno" (Cilliers, 2007). In tale definizione si sottolinea che le relazioni che descrivono questo tipo di sistemi sono non lineari.

I sistemi complessi sono, quindi, sistemi deterministici che evolvono nel tempo in modo caotico (caos deterministico) e ciò comporta che piccole variazioni dello stato iniziale, anche piccole incertezze sulle misure iniziali, possono portare ad uno stato finale del sistema completamente diverso, cioè una piccola incertezza sulle condizioni iniziali crea degli attrattori, che sono delle forme nello spazio delle fasi, dove i punti possono collocarsi in maniera probabilistica. Un esempio di caos deterministico si ha nel caso delle previsioni atmosferiche. Infatti, anche ricorrendo a misurazioni da satellite dello stato iniziale, a un modello matematico e ad algoritmi molto accurati non si potrà andare oltre un tempo dell'ordine $\frac{c}{\lambda}$, dove λ dipende dalle proprietà della dinamica dell'atmosfera ($\frac{1}{\lambda} \sim 10$ *giorni*). Ciò non avviene per i sistemi semplici poiché essendo governati da equazioni lineari, tanto piccola è l'incertezza sulle condizioni iniziali, tanto piccola risulta quella sulle condizioni finali.

Un'altra caratteristica dei sistemi complessi è la causalità reciproca, cioè se un elemento influenza l'altro questo andrà a retroagire attraverso dei meccanismi di feedback (causalità circolare) cosa che non avviene nei sistemi semplici. Un esempio di questo meccanismo è dato dallo scioglimento dei ghiacciai in climatologia: infatti un aumento della

temperatura terrestre causa lo scioglimento dei ghiacciai, questo, a sua volta, causa una diminuzione del potere riflettente, detto albedo, per cui di fatto causa più radiazione assorbita dalla terra aumentando la temperatura e così via.

Altra causa della non linearità è che il ruolo dei singoli elementi nel determinare il comportamento del sistema non è ben identificabile come si evince dalle equazioni alla base del modello Lotka-Volterra (paragrafo 2.2). Le proprietà globali che si verificano alla scala macroscopica del sistema vengono dette proprietà emergenti nel senso che non vengono derivate dai singoli elementi, come accade per i sistemi semplici, ma emergono dalle interazioni reciproche dei singoli elementi. Per questo motivo, per capire il comportamento di un sistema complesso dobbiamo capire non solo il comportamento delle parti, ma anche come esse interagiscono tra loro per far emergere il comportamento globale di tutto il sistema.

Riassumendo, i sistemi complessi possono essere caratterizzati dalle seguenti proprietà:

- Non decomponibilità, ovvero non possono essere decomposti o ridotti in parti più piccole a causa della struttura caotica dei componenti.
- Non linearità, ovvero le equazioni che li governano sono di tipo non lineare.
- Effetto Farfalla, ovvero un'elevata sensibilità alle condizioni iniziali.
- Retroazione, ovvero casualità circolare.
- Emergenza, ovvero facendo evolvere nel tempo i sistemi complessi emergono comportamenti o proprietà della collettività non predicibili.
- Auto-organizzazione, ovvero la capacità dei componenti di auto organizzarsi all'interno del sistema.

2.2 Simulare i sistemi complessi: approcci equation-based e agent-based

Le simulazioni sono largamente usate nella ricerca scientifica, come accennato nel capitolo 1, soprattutto quando si ha a che fare con la risoluzione dei sistemi complessi. Infatti, mentre per i sistemi semplici è possibile riprodurre il fenomeno in laboratorio per poterne verificare le ipotesi e trovare soluzioni analitiche al set di equazioni che costituiscono il modello, per i sistemi complessi ciò è spesso impossibile. In generale, per i sistemi complessi fortemente non lineari non esistono soluzioni analitiche. Pertanto, si

utilizzano le simulazioni che definiscono l'evoluzione spazio-temporale del sistema costruendo uno stato dopo l'altro tramite un metodo numerico.

Si possono distinguere due categorie di simulazioni: quelle che descrivono il sistema attraverso delle equazioni (equation-based) e quelle che generano la dinamica del sistema attraverso delle regole imposte sugli agenti del sistema (agent-based). Le simulazioni equation-based usano un insieme di equazioni differenziali e risolvendo queste derivano lo stato futuro del sistema a partire dallo stato presente; nelle equazioni compaiono variabili che si riferiscono a quantità macroscopiche poiché la modellizzazione avviene a livello aggregato. Le simulazioni agent-based, invece, non sono basate sulla descrizione macroscopica del sistema ma generano la dinamica del sistema lasciando evolvere i singoli agenti che compongono il sistema tramite delle regole, facendo così emergere le proprietà macroscopiche del sistema.

Per rendere più esplicite le differenze tra equation-based e agent-based si prenderà in considerazione uno stesso modello, quello di Lotka-Volterra, e si mostreranno i due tipi possibili di simulazioni.

Il modello di Lotka-Volterra, molto usato in ecologia, si occupa della coesistenza di prede e predatori. Le ipotesi su cui si basa il modello sono che le prede dispongano di cibo illimitato, l'unica fonte di sostentamento dei predatori siano le prede, i predatori muoiano solo di morte naturale, non siano in atto meccanismi evolutivi e l'ambiente esterno non si modifichi a favore di nessuna delle specie. Tradotte analiticamente, queste condizioni danno luogo al seguente sistema di equazioni:

$$\frac{dx}{dt} = Ax(t) - By(t)x(t) \qquad \frac{dy}{dt} = Cx(t)y(t) - Dy(t)$$

dove:

- $x(t)$ = numero di prede al tempo t
- $y(t)$ = numero di predatori al tempo t
- dx/dt = tasso di variazione della popolazione di prede nel tempo
- dy/dt = tasso di variazione della popolazione dei predatori nel tempo
- A = coefficiente di nascita delle prede
- B = coefficiente di predazione
- C = coefficiente di incontro tra prede e predatori

- D = coefficiente di morte naturale dei predatori.

Come si può notare nelle equazioni, il tasso di variazione della popolazione delle prede diminuisce con l'aumentare degli incontri con i predatori ($B y(t)x(t)$), mentre il tasso di variazione della popolazione dei predatori aumenta con l'aumentare degli incontri con le prede ($C x(t)y(t)$).

Il grafico in figura 6 rappresenta il risultato di una simulazione equation-based del modello Lotka-Volterra, ottenuta inserendo le equazioni, che descrivono il modello, in un compilatore ottenendo così una risoluzione numerica step by step.

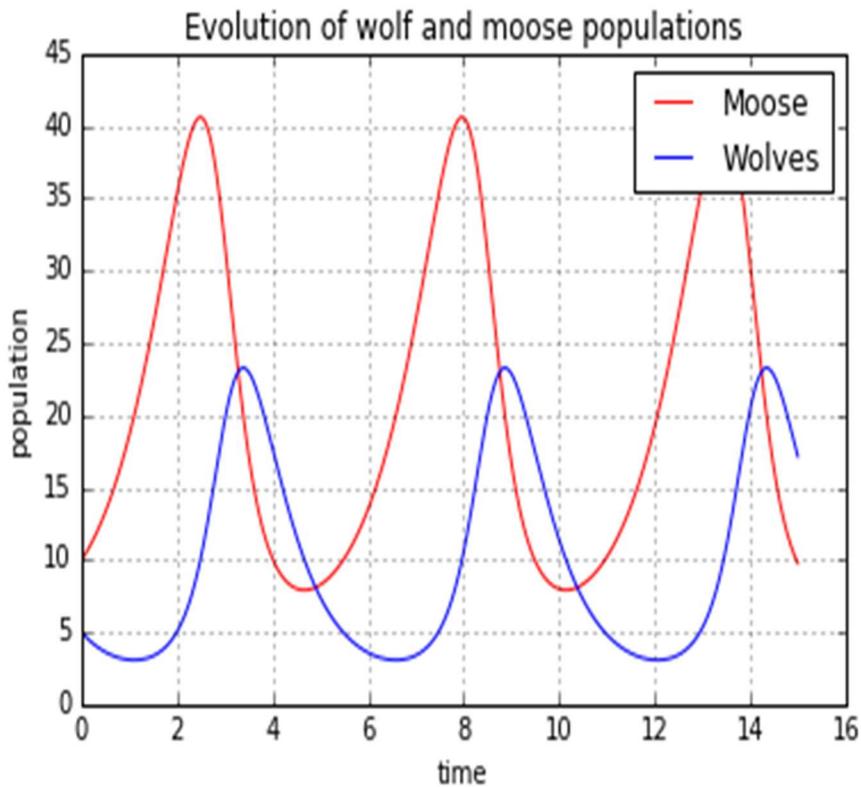


Figura 4 Grafico prodotto da una simulazione del modello Lotka-Volterra equation-based

Di seguito si mostrerà lo stesso modello in una simulazione agent-based. In tale modello, due tipi di agenti (linci e lepri) si muovono in una griglia in modo random. Esse, in un

determinato istante, possono trovarsi in prossimità di altri agenti: in questo caso, interagiranno tra di loro.

Nello specifico, le regole che i singoli agenti devono seguire nella simulazione preda-predatore agent-based sono:

- se un predatore non ha almeno tre prede come sui primi vicini muore di fame;
- se un predatore prende (si trova vicino ad) almeno tre prede, sopravvive e si riproduce;
- se una preda riesce a scampare al predatore sopravvive e si riproduce.

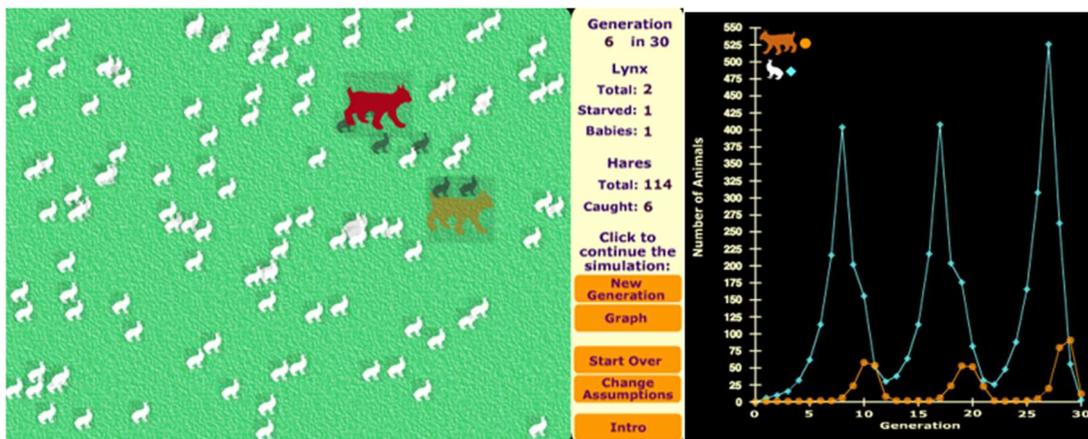


Figura 5 Visualizzazione del modello di Lotka-Volterra in simulazione agent-based

In questo caso ogni singolo elemento evolve creando la dinamica del sistema. Si può notare, anche in questo caso, la causalità reciproca: infatti, all'aumentare dei predatori corrisponde una diminuzione di prede che a sua volta causa la diminuzione dei predatori e così via.

Un altro esempio di simulazione agent-based molto noto è quello basato sul modello di Schelling. Questo modello fu concepito alla fine degli anni Sessanta per spiegare la comparsa di aree di segregazione urbana, i cosiddetti ghetti. Si suppone che un ambiente sia costituito da due tipologie di individui diversi tra loro. Ciascun individuo è autonomo nelle proprie scelte, cioè decide il suo comportamento in relazione alla situazione in cui si trova, non ci sono indicazioni generali che gli impongano dove e come spostarsi. L'unica regola alla base di questo modello è che ogni individuo decide di trasferirsi solo se meno di un terzo dei vicini sono del suo stesso tipo, cioè se il numero degli individui

diversi da lui è maggiore al 67%. Gli agenti agiscono secondo una funzione non lineare del tipo:

$$\begin{aligned}
 & \textit{insoddisfatto} (\textit{agente}, \textit{tipo}, [\textit{vic}_1, \textit{vic}_2, \dots, \textit{vic}_n]) \\
 & = \begin{cases} \textit{vera se} & \frac{n \textit{ vicini stesso tipo}}{n \textit{ vicini non vuoti}} \leq 0.3 \\ \textit{falsa se} & \frac{n \textit{ vicini stesso tipo}}{n \textit{ vicini non vuoti}} > 0.3 \end{cases}
 \end{aligned}$$

Per mostrare il funzionamento di questa regola, prendiamo il caso della figura 8. Abbiamo una griglia in cui gli agenti sono posizionati randomicamente.

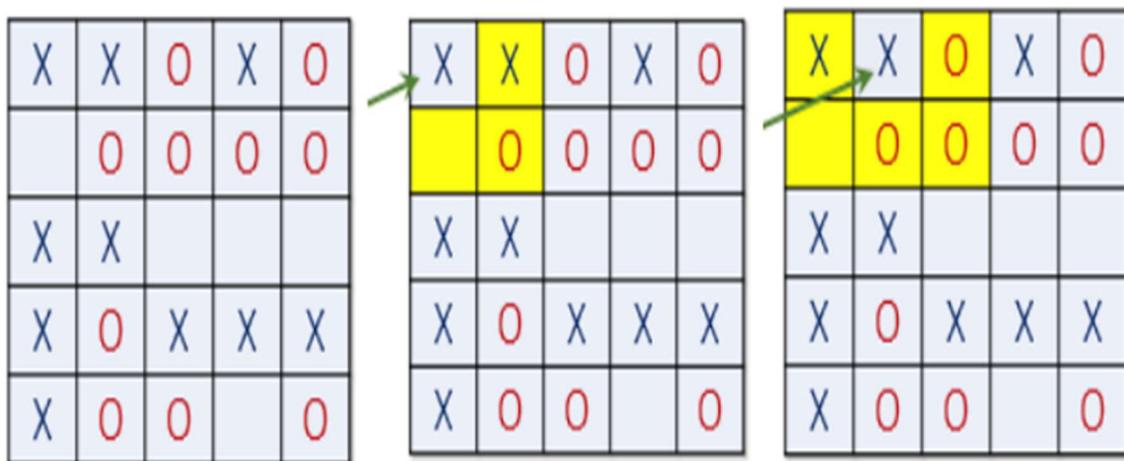


Figura 6 Rappresentazione del modello Schelling

Analizzando il primo elemento in alto a sinistra e valutando il suo livello di soddisfazione, esso è vicino ad un individuo simile e uno diverso quindi la percentuale è 50% che è minore del 67% quindi è soddisfatto e non si sposta.

Prendendo in considerazione il secondo elemento, quest'ultimo è insoddisfatto perché il 75% (3 su 4) dei suoi vicini sono diversi; allora, l'elemento si sposta finché non è soddisfatto. La simulazione continua così finché tutti gli individui sono soddisfatti. Il risultato che si ottiene è un ambiente segregato e la segregazione risulta come una proprietà emergente poiché compare in modo inatteso a partire dalle sole regole che governano la dinamica del sistema, senza alcuna consapevolezza o pre-determinazione negli individui che la determinano né misure imposte dall'esterno. Più in generale si

evidenza che il sistema ha la capacità di auto organizzarsi e che ciò è causato dalla sole regole interne del sistema.

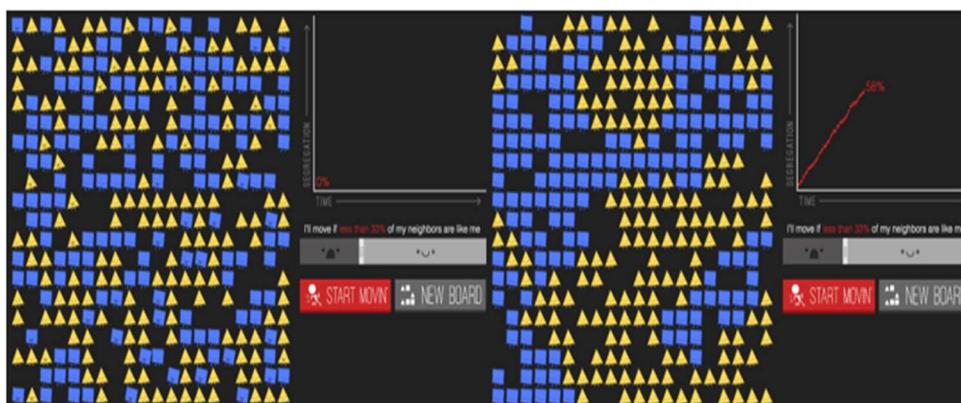


Figura 7 Visualizzazione di una simulazione del modello Schelling

Nella simulazione mostrata in Fig. 9 si nota proprio questo effetto di segregazione, infatti, i triangoli e i quadrati si dispongono come se fossero dei quartieri.

In entrambi gli esempi si può notare la codipendenza tra gli elementi del sistema che è in effetti una proprietà tipica delle simulazioni ad agente (Grüne-Yanoff & Weirich, 2010).

2.3 Risultati di ricerca sull'utilizzo delle simulazioni di sistemi complessi in ambito educativo e sulle difficoltà emerse

Le simulazioni di sistemi complessi sono spesso utilizzate in didattica per favorire cambiamenti concettuali e progressi cognitivi su temi che solitamente presentano per gli studenti notevoli difficoltà. Diversamente da quello che accade nella pratica e nella ricerca scientifica, quando le simulazioni vengono utilizzate per scopo didattico, la loro interpretazione come strumenti scientifici può risultare un punto particolarmente problematico.

Tuttavia, dalla letteratura (Barelli et al., 2019; M. Jacobson, 2000; M. J. Jacobson & Wilensky, 2006; Nerdel & Precht, s.d.; Smetana & Bell, 2012) si evince che le simulazioni come nuovo strumento di indagine offrono numerosi vantaggi. Oltre ai vantaggi tipici delle simulazioni in senso lato (già discussi nel capitolo precedente), quando si affrontano i sistemi complessi le simulazioni permettono di modellizzare fenomeni altamente non lineari e risolvere equazioni a moltissime variabili che non sarebbe possibile risolvere in modo analitico (approcci ad equazione) oppure di simulare

moltissime entità, ciascuna con le proprie caratteristiche, di stabilire le regole di interazione locali tra esse e di osservare l'emergere degli effetti globali delle interazioni (approcci ad agente).

Nonostante ciò, dalla recente ricerca (Barelli et al., 2019) si evince che quando gli studenti entrano in contatto con questo tipo di simulazioni sorgono diverse difficoltà concettuali. Una delle principali consiste nel rinunciare al cosiddetto “deterministic-centralized mindest” (M. J. Jacobson & Wilensky, 2006) ovvero al senso di controllo centralizzato sui sistemi e alla causalità deterministica a favore di descrizioni che coinvolgono auto-organizzazione, modelli stocastici (che descrivono situazioni che variano in base a leggi probabilistiche e non deterministiche) e processi decentrati. Collegato a ciò vi è anche una difficoltà a rinunciare alla concezione di una relazione lineare tra la dimensione dell'azione e l'effetto corrispondente (una piccola azione ha un effetto piccolo e viceversa), accettando effetti non classici come il cosiddetto “effetto farfalla”.

Le recenti teorie del cambiamento concettuale propongono come causa primaria di queste difficoltà strutture cognitive e credenze ontologiche ed epistemologiche. Questo suggerisce che nell'apprendimento dei sistemi complessi il focus non deve essere posto solo sugli aspetti concettuali dei sistemi complessi ma soprattutto sull'arricchimento della rete cognitiva di credenze e intuizioni che gli studenti hanno sul mondo. Infatti, è ampiamente accettato nella didattica della scienza che gli studenti debbano fare esperienze dirette per costruire credenze su come le cose si comportano nel mondo. Quindi, il modo migliore per la comprensione ontologica e concettuale dei sistemi complessi è quello di sperimentare fenomeni complessi nella vita quotidiana in modo da poter attingere all'esperienza e incanalarla per costruire sotto classifiche di sistemi complessi. Tuttavia, per molti sistemi complessi la pratica sperimentale quotidiana è resa impossibile da difficoltà tecniche, pratiche ed etiche (basti pensare allo studio delle città o dei sistemi sociali, i sistemi complessi per eccellenza). In questo senso, le simulazioni costituiscono l'ambiente laboratoriale privilegiato per i sistemi complessi, consentendo di investigare virtualmente, grazie a una procedura di modellizzazione, sistemi che non consentirebbero una manipolazione “manuale”.

Un'ulteriore difficoltà evidenziata da recenti ricerche (Barelli et al., 2019) consiste nella spiegazione dei fenomeni che la simulazione mostra e le cause che producono quel tale fenomeno. Infatti, chiamati ad analizzare simulazioni di sistemi complessi (come, ad

esempio, simulazioni ad agente del modello predatore-preda o del modello di Schelling), la maggior parte degli studenti interpreta le simulazioni come uno strumento che mostra un determinato comportamento senza fornirne le cause. Nella maggior parte dei casi, gli studenti riescono a individuare i fenomeni mostrati dalla simulazione, le entità di base, le condizioni iniziali e le regole assegnate alle entità dando una esplicazione superficiale del fenomeno senza indagarne le cause. La difficoltà nella spiegazione dei fenomeni complessi, inoltre, è collegata a una scarsa fiducia nell'utilizzo delle simulazioni per la pratica scientifica, e tali atteggiamenti si mostrano tanto più marcati quanto più scarsa è la consapevolezza su questioni epistemologiche quali il ruolo di un modello e la sua funzione.

2.4 Il progetto NetLogo

NetLogo è un linguaggio di programmazione multi-agente che fornisce un ambiente di modellizzazione per simulare fenomeni naturali e sociali. In particolare, viene utilizzato per la modellizzazione e la simulazione dei sistemi complessi e la loro evoluzione nel tempo. Esso consente agli utenti sia di esplorare e modificare simulazioni già esistenti, sia di creare i propri modelli anche se non si è programmatori professionisti. NetLogo è un'applicazione stand-alone scritta in Java in modo che possa essere eseguita su tutte le principali piattaforme di calcolo. Chi programma può creare e dare istruzioni a centinaia o migliaia di agenti indipendenti, che agiscono simultaneamente e ciò rende possibile esplorare la connessione tra i comportamenti degli individui ad un micro-livello e i comportamenti globali ad un macro-livello. Gli agenti mobili, detti turtles, si muovono su una griglia di "patch". Tutti gli agenti possono interagire tra loro ed eseguire più compiti contemporaneamente.

NetLogo nasce da una miscela di StarLisp e Logo. Da Logo ha ereditato il concetto di "turtle" e la filosofia di facilità di utilizzo, mentre da StarLisp ha ereditato i molteplici agenti e la contemporaneità delle loro azioni, cioè ogni agente può svolgere più azioni contemporaneamente. NetLogo è in fase di sviluppo dal 1999 ed è in ampia crescita nella comunità educativa.

Nella ricerca "Software review: NetLogo, a multi-agent simulation environment"(Sklar, 2007) NetLogo ha dimostrato di essere un valido strumento sia per la ricerca che per l'insegnamento a diversi livelli. Infatti, è facile da installare, da usare ed è molto utile per

via delle simulazioni sviluppate per applicazioni di ricerca nel campo della modellizzazione del mondo reale. È stato utilizzato con successo per insegnare nozioni di programmazione, infatti, nello studio “Learn-Lego robot and NetLogo” (Zaharija et al., 2014) la simulazione NetLogo viene utilizzata con un robot Lego Mindstorms, che connessi tra loro, comunicano tramite adattatori Bluetooth e attraverso un’applicazione .NET framework e C#. Da questo studio risulta che gli studenti hanno migliorato la comprensione dei concetti di base del campo della intelligenza artificiale e della programmazione.

Le simulazioni NetLogo vengono anche utilizzate nella ricerca per promuovere un cambiamento significativo nella didattica e in particolare nella didattica dei sistemi complessi (Barelli et al., 2019).

Capitolo 3

Analisi comparativa di simulazioni PhET di meccanica classica e simulazioni NetLogo di sistemi complessi

In questo capitolo, analizziamo le simulazioni PhET di meccanica classica e NetLogo di sistemi complessi. Per farlo abbiamo scelto dai rispettivi siti web alcune simulazioni che riteniamo rappresentative delle due tipologie di sistemi rappresentati e delle rispettive metodologie di simulazione. Nel primo paragrafo abbiamo analizzato simulazioni PhET riguardanti argomenti di meccanica classica (Masse e molle, Conservazione dell'energia meccanica, Forze sul piano inclinato e Leve). Nel secondo paragrafo abbiamo studiato simulazioni NetLogo di sistemi complessi (Predazione, Segregazione, Cambiamento climatico e Traffico). Nel terzo paragrafo presentiamo con una tabella lo strumento analisi comparativa tra le simulazioni PhET e NetLogo. In conclusione, presentiamo i risultati ottenuti dall'applicazione della griglia.

3.1 Analisi di simulazioni PhET di meccanica classica

Vediamo ora alcune simulazioni disponibili al sito https://phet.colorado.edu/_m/it/.

3.1.1 Masse e Molle

La simulazione dal titolo “Masse e molle” del PhET (<https://phet.colorado.edu/it/simulation/masses-and-springs-basics>) ha come obiettivi di apprendimento la comprensione delle grandezze fondamentali che caratterizzano i sistemi formati da masse e molle e della loro relazione. Più in dettaglio, gli obiettivi espliciti suggeriti dalla guida all'utilizzo della simulazione sono: descrivere lunghezza a riposo e posizione di equilibrio della molla; trovare la relazione tra massa, forza della molla (costante elastica) e allungamento (spostamento); determinare i fattori che influenzano il periodo di oscillazione; descrivere la relazione tra vettori velocità e accelerazione e la loro relazione con il moto. Inoltre, è anche possibile confrontare i dati ottenuti cambiando

la gravità del pianeta. Aprendo la simulazione si viene posti di fronte a tre possibili situazioni: Stretch, Bounce e Lab.



Figura 8 Menù iniziale della simulazione Masse e Molle: introduzione

Scegliendo la prima opzione si ha la possibilità di “giocare” con una o due sistemi a molla a cui si possono appendere le masse a disposizione per scoprire la relazione lineare tra la massa e l’allungamento della molla, tramite la relazione lineare con la costante elastica.

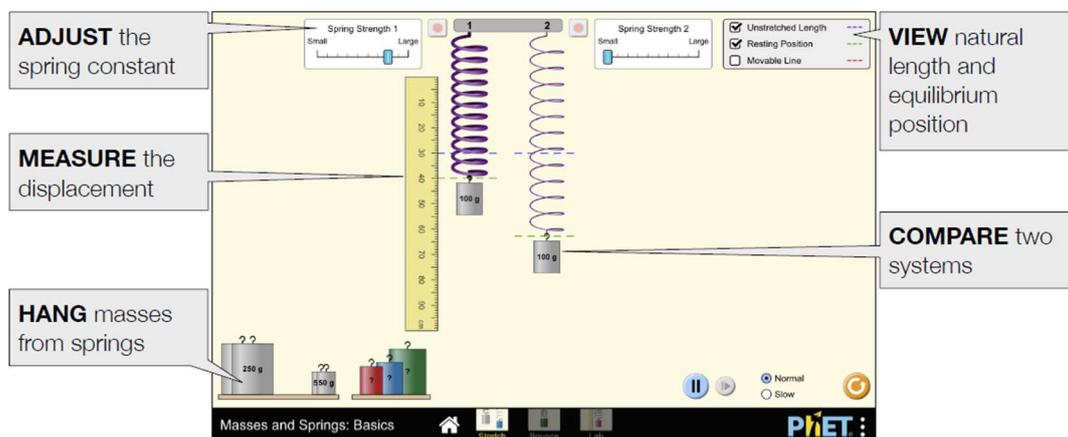


Figura 9 Stretch

Come si nota nell’immagine 11, la simulazione mette a disposizione diverse masse, un righello e la possibilità di regolare a piacere la forza delle molle. In alto a destra, invece, la simulazione permette di visualizzare a schermo la lunghezza della molla e la posizione di equilibrio. È importante far notare che solo in questa sezione le molle sono state formalmente smorzate per ridurre al minimo le oscillazioni e favorire, così, l’obiettivo di apprendimento della legge di Hooke.

Scegliendo l'opzione Bounce si può osservare un esperimento con molle oscillanti in cui è possibile indagare quali sono le variabili (come massa, costante elastica della molla o spostamento) che influiscono sul periodo di oscillazione.

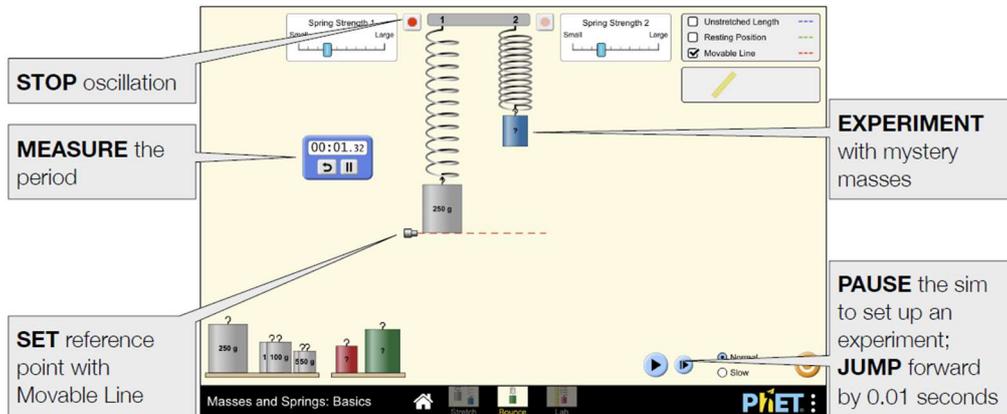


Figura 10 Bounce

In questo caso si hanno a disposizione anche un cronometro ed un riferimento mobile. È possibile, inoltre, rallentare o fermare il sistema in modo da agevolare le misure.

Dopo aver visto le sezioni Stretch e Bounce e dopo aver appreso una certa familiarità con la simulazione stessa e con i concetti base sulle molle ci si può cimentare in una vera e propria esperienza di laboratorio nella sezione Lab.

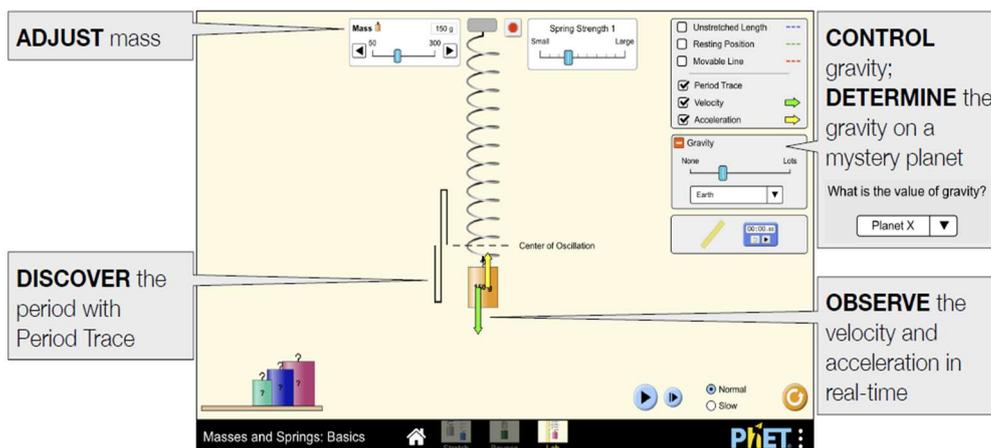


Figura 11 Lab

In questa sezione si possono raccogliere dati in modo da poter determinare o una massa incognita o la gravità di un pianeta. La simulazione permette di visualizzare a schermo i vettori velocità e accelerazione e questo permetterà di individuare il centro di oscillazione

che rappresenta il centro di massa all'equilibrio (lo studente potrà così scoprire che esso è un riferimento più appropriato rispetto alla posizione a riposo) e la traccia del periodo (che disegna il percorso di un'oscillazione completa attorno alla posizione di equilibrio del centro di massa).

Questo tipo di simulazione può essere proposta agli studenti per:

- Progettare un esperimento controllato per determinare come una variabile (come massa, gravità, costante della molla, o spostamento) influenza il periodo;
- Determinare la massa delle masse incognite o il valore dell'accelerazione di gravità g (qualitativamente o quantitativamente) spiegando il metodo usato;
- Determinare il rapporto tra la forza applicata e lo spostamento;
- Spiegare cosa rappresenta il periodo e determinare un metodo per misurarlo.

3.1.2 Conservazione dell'energia meccanica

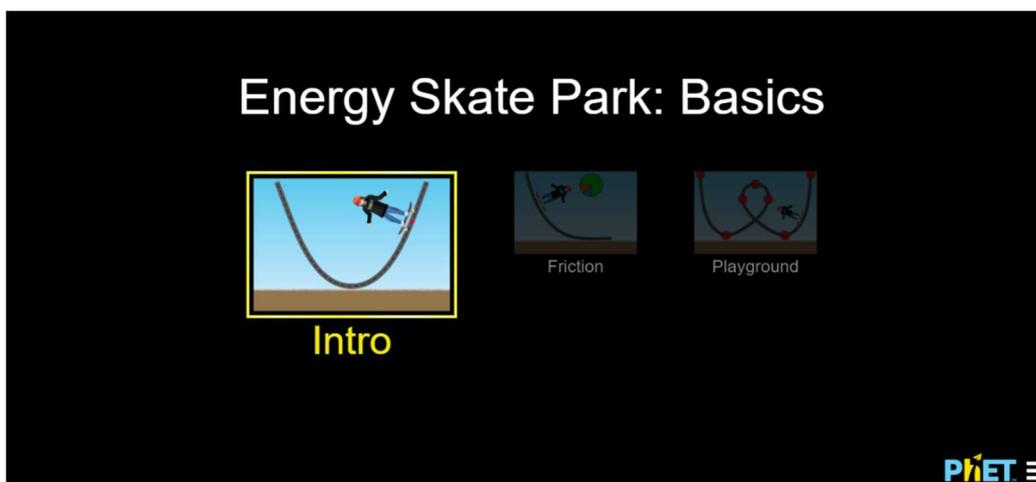


Figura 12 Energia con lo Skateboard: concetti base

L'obiettivo principale della simulazione "Energy Skate Park" del PhET (https://phet.colorado.edu/sims/html/energy-skate-park-basics/latest/energy-skate-park-basics_it.html) è quello di spiegare il principio di conservazione dell'energia meccanica usando l'energia cinetica e l'energia potenziale costruendo percorsi, rampe e salti per lo skater e osservando l'energia cinetica, l'energia potenziale e l'attrito mentre si muove. La schermata iniziale della simulazione presenta tre diverse scelte: Intro, Friction e

Playground. Nella sezione introduzione è possibile familiarizzare con la simulazione.

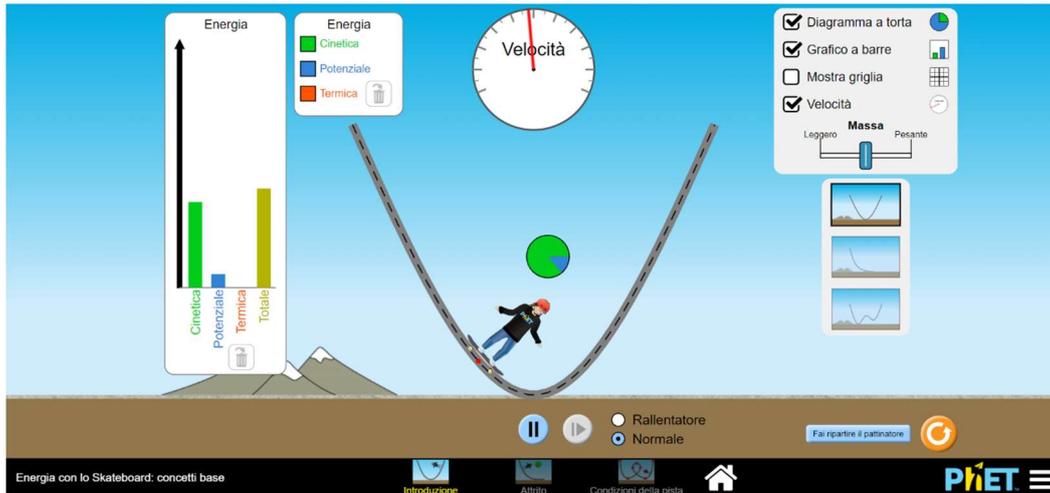


Figura 13 Intro

In questa sezione gli studenti possono utilizzare diversi percorsi e analizzare il rapporto tra l'energia cinetica e l'energia potenziale del pattinatore. È importante far notare che in questa sezione la pista è realizzata senza attrito.

Nella sezione Attrito, come si può intuire dal nome, viene aggiunto l'attrito alla pista, che è possibile modificare a proprio piacimento. Inoltre, la simulazione mostra, tramite grafico a torta o istogramma, come cambia l'energia al variare dell'attrito.

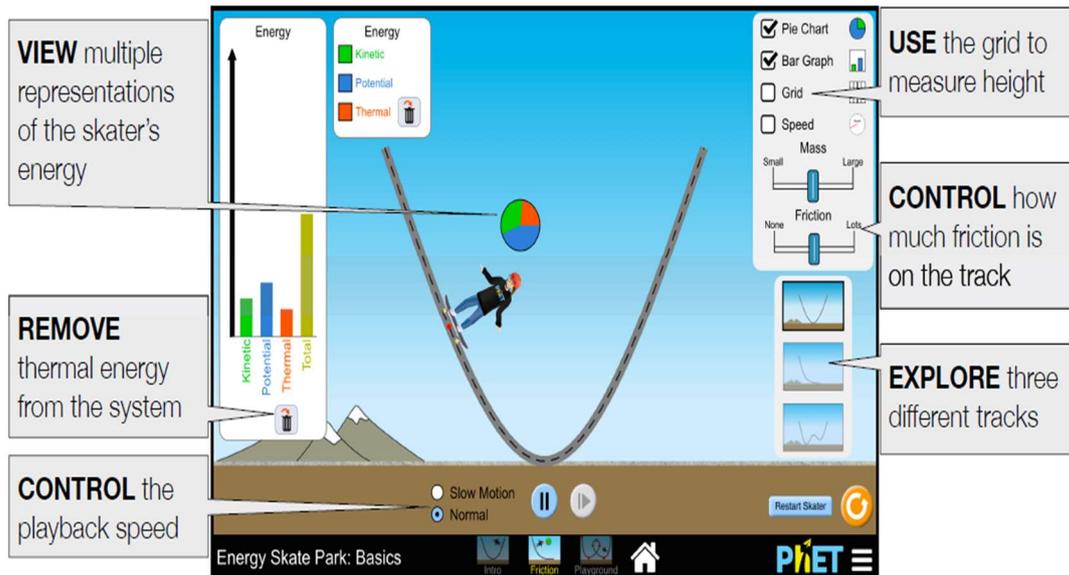


Figura 14 Friction

L'ultima sezione a disposizione è Playground.

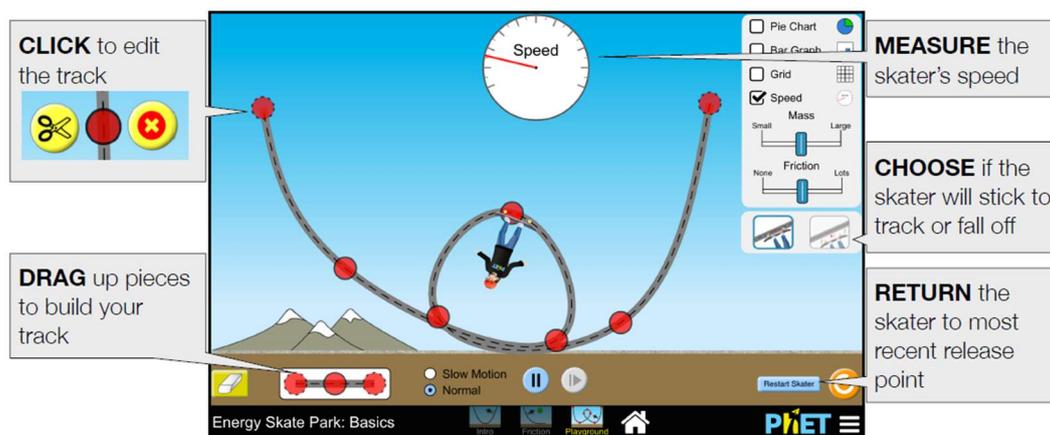


Figura 15 Playground

In questa sezione gli studenti possono costruire la pista inserendo anche dei salti. Quando lo skater atterra sulla pista, la componente verticale dell'energia cinetica viene convertita in energia termica. È possibile, eliminando l'attrito e assicurandosi che lo skater non lasci la pista, effettuare esperimenti in cui non vi è perdita di energia meccanica in energia termica. Questa simulazione può essere proposta per determinare la relazione tra energia cinetica e velocità e a quale punto della pista la maggior parte dell'energia meccanica dello skater viene trasferita all'ambiente in forma di energia termica. Esempi di domande che si possono proporre agli studenti sono:

- Dato il grafico della barra energetica, determinare la velocità del pattinatore.
- Il grafico energetico a torta corrisponde con la posizione in pista.
- Se l'energia cinetica dello skater è sempre più grande, determinare la direzione del suo movimento.
- Stabilire se lo skater può superare una collina data la sua posizione di partenza.

3.1.3 Forze sul piano inclinato

Questa simulazione consente di esplorare le forze e il movimento spingendo oggetti domestici su e giù per una rampa. Questa simulazione permette di abbassare e sollevare la rampa per vedere come l'angolo di inclinazione influenza le forze parallele mostrando i grafici di forza, energia e lavoro.

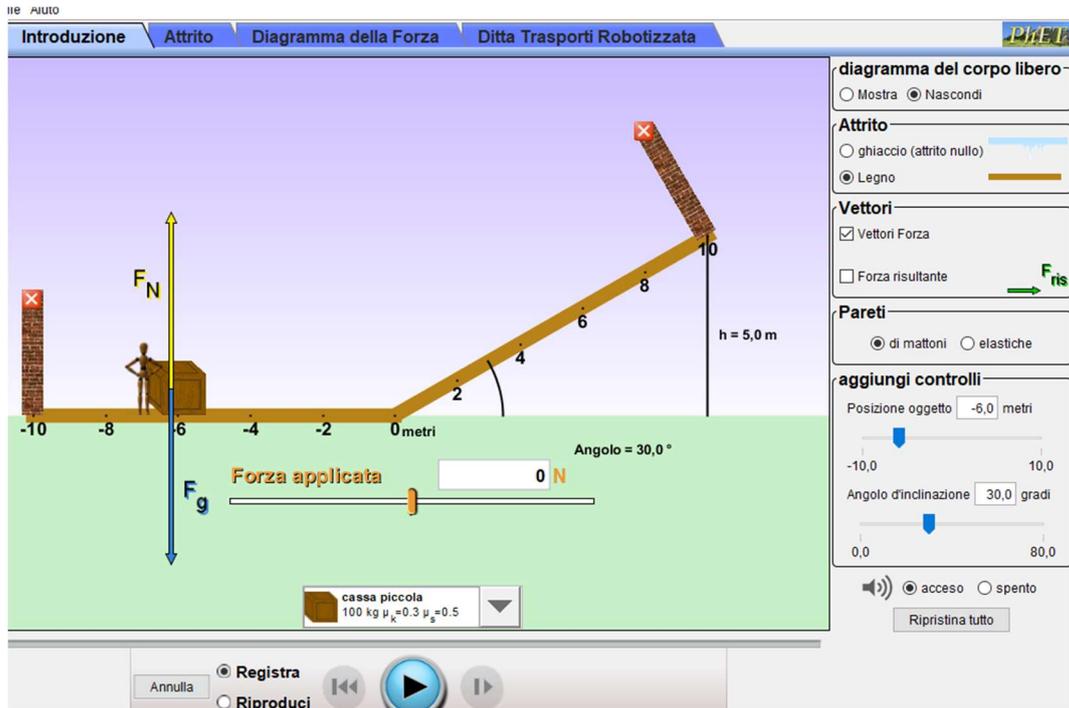


Figura 16 "Piano Inclinato: Forza e Moto" Introduzione

Aprendo la simulazione, c'è la pagina di introduzione in cui è possibile spingere una cassa su una rampa, di cui possiamo scegliere l'inclinazione, applicando una determinata forza. Questa sezione della simulazione serve a familiarizzare con la simulazione stessa.

La seconda sezione a disposizione permette di familiarizzare con il concetto di attrito.

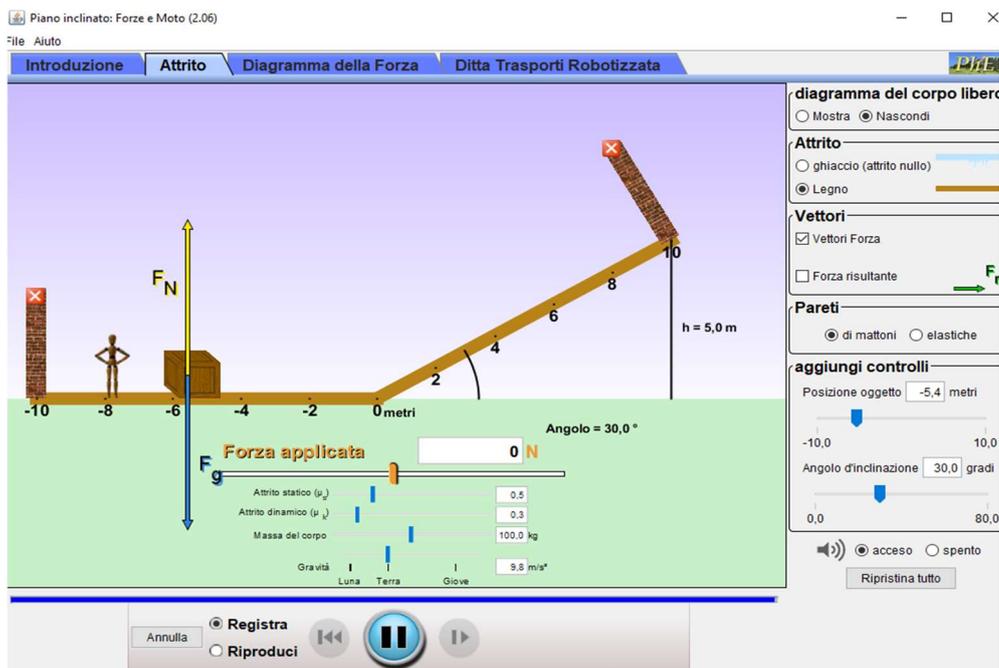


Figura 17 Attrito

Infatti, in questa sezione, oltre a variare la forza applicata, l'inclinazione della rampa e la massa dell'oggetto è possibile variare l'attrito statico, l'attrito dinamico e l'accelerazione di gravità.

Nella terza sezione "Diagramma delle Forze" è possibile visualizzare i grafici delle forze in funzione del tempo.

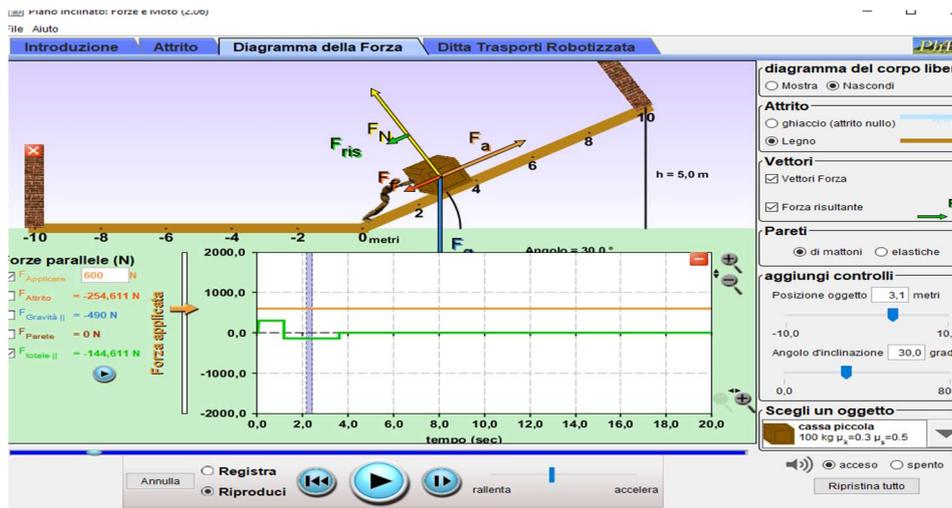


Figura 18 Diagrammi delle Forze

Nell'ultima sezione c'è un minigioco con cui gli studenti possono cimentarsi. Lo scopo di questo gioco è quello di utilizzare l'esperienza acquisita nelle sezioni precedenti per far consegnare al robot gli oggetti col minimo dispendio di energia.

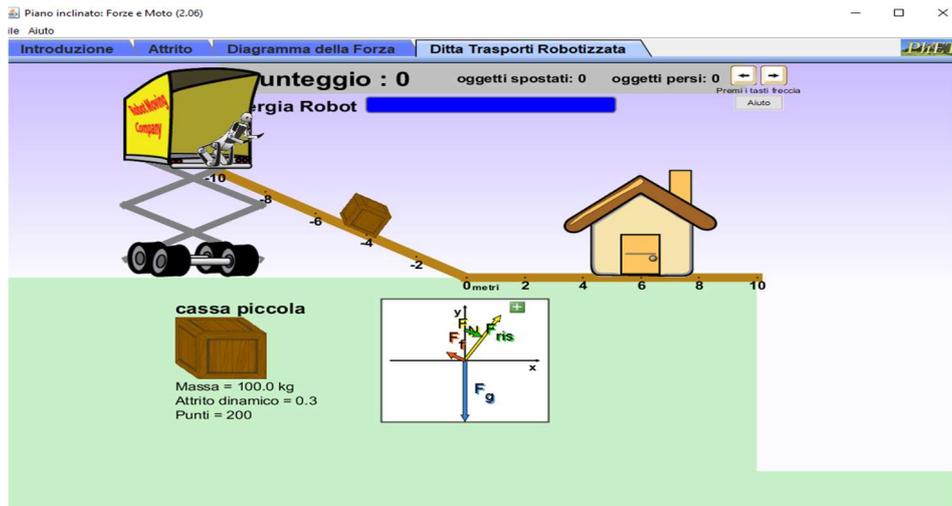


Figura 19 Ditta di Trasporti Robotizzata

Questa simulazione può essere utilizzata per svolgere compiti a casa, attività in classe o attività di laboratorio. Può, inoltre, essere utile per l'introduzione o l'approfondimento di concetti.

3.1.4 Leve

Aperto questa simulazione (<https://phet.colorado.edu/it/simulation/balancing-act>) c'è la schermata del menù in cui si possono scegliere tre diverse tipologie: Intro, Balance Lab e Game.

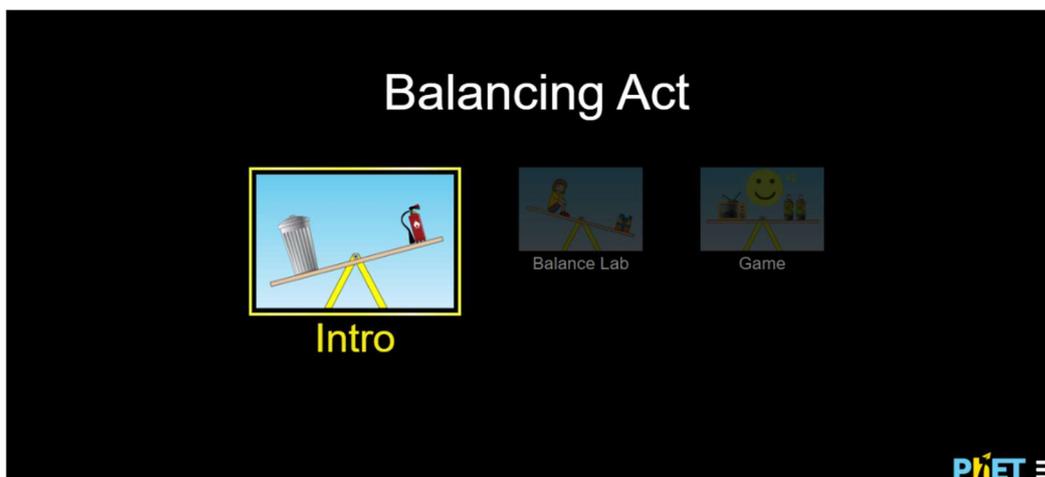


Figura 20 Menù Iniziale

Nella sezione Intro è possibile “giocare” con degli oggetti su un’altalena per imparare l’equilibrio.

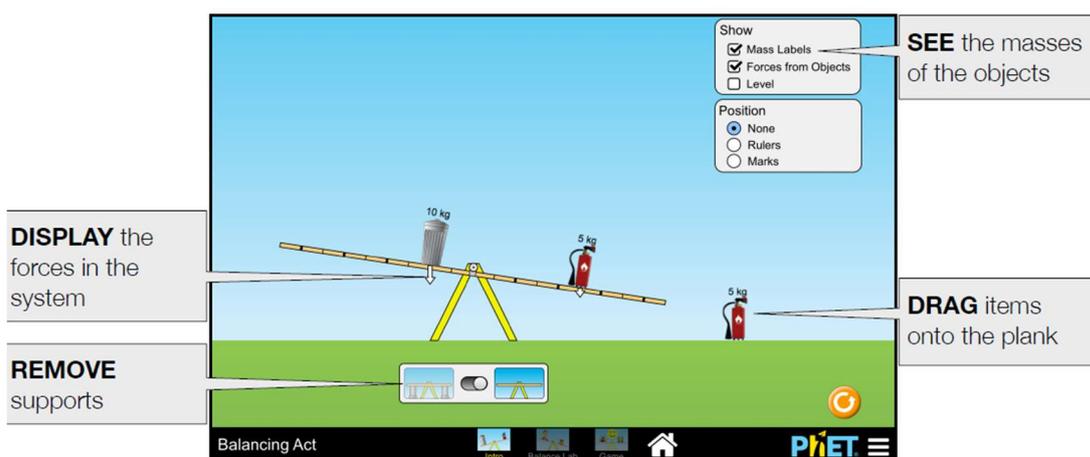


Figura 21 Intro

Questa specifica sezione serve a familiarizzare con la simulazione e con i diversi comandi a disposizione. Infatti, nelle finestre in alto a destra è possibile scegliere di visualizzare la massa degli oggetti, il vettore forza peso, la posizione di equilibrio dell'altalena e strumenti come il righello.

Nella seconda sezione "Balance Lab" vengono messi a disposizione vari oggetti di cui alcuni con massa incognita. Lo scopo di questa sezione è quello di bilanciare l'altalena utilizzando gli oggetti con massa nota ed in seguito quello di determinare le masse incognite.

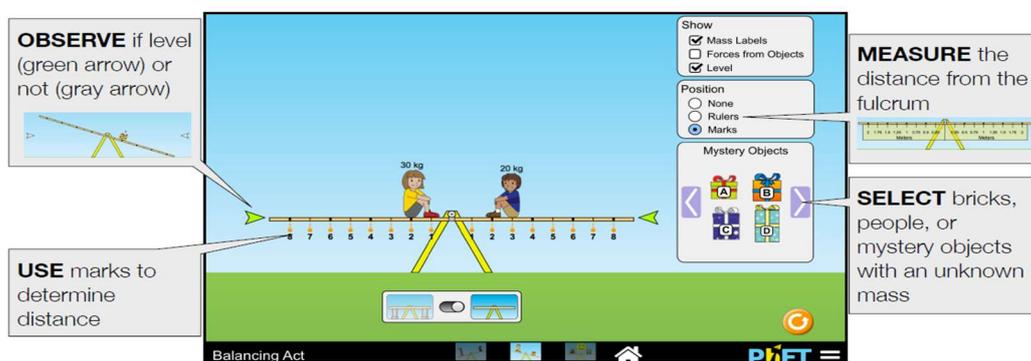


Figura 22 Balance Lab

In questa sezione gli studenti sono liberi di fare previsioni su come oggetti di diversa massa possano essere utilizzati per bilanciare l'asse e su come un cambiamento di disposizione delle masse sull'asse ne influenzi il movimento. Infine, gli studenti possono scrivere una regola per prevedere come si inclinerà l'asse ponendo sopra diversi oggetti.

Nella terza sezione "Game" ci sono dei mini-giochi in cui gli studenti possono mettersi alla prova e testare il loro apprendimento.

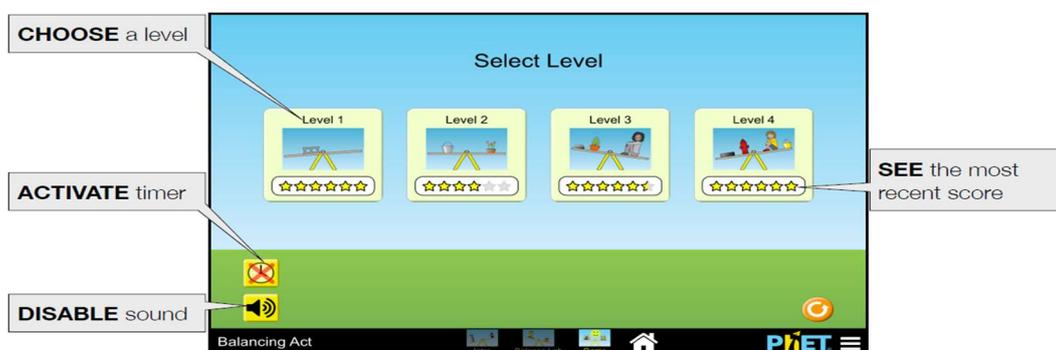


Figura 23 Game

Ci sono quattro livelli ognuno composto da sei quesiti a difficoltà crescente.

3.2 Analisi di simulazioni NetLogo di sistemi complessi

Le simulazioni che verranno prese in considerazione sono tutte disponibili sul sito NetLogo.

3.2.1 Predazione

La simulazione “Wolf Sheep Predation” si rifà al modello Lotka-Volterra descritto nel capitolo 2 ed esplora la stabilità degli ecosistemi predatori-preda. Questo è un sistema instabile e tende a provocare l'estinzione di una o più specie interessate. La simulazione in esame prevede due varianti: “sheep-wolves” e “sheep-wolves-grass”.

Nella prima variante, “sheep-wolves”, lupi e pecore vagano casualmente per il paesaggio.

```
to reproduce-wolves ;; wolf procedure
  if random-float 100 < wolf-reproduce [ ;; throw "dice" to see if you will reproduce
    set energy (energy / 2) ;; divide energy between parent and offspring
    hatch 1 [ rt random-float 360 fd 1 ] ;; hatch an offspring and move it forward 1 step
  ]
end

to catch-sheep ;; wolf procedure
  let prey one-of sheep-here ;; grab a random sheep
  if prey != nobody ;; did we get one? if so,
    [ ask prey [ die ] ;; kill it
      set energy energy + wolf-gain-from-food ] ;; get energy from eating
end

to death ;; turtle procedure
  ;; when energy dips below zero, die
  if energy < 0 [ die ]
end

to display-labels
  ask turtles [ set label "" ]
  if show-energy? [
    ask wolves [ set label round energy ]
  ]
end
```

Codice 1 Gestisce il comportamento dei lupi

Come si può osservare nelle linee di codice la probabilità di riproduzione dei lupi è randomica (si veda la funzione “random-float”). Se avviene la riproduzione l’energia iniziale viene divisa tra il genitore e la progenie. Ogni passo costa ai lupi energia, quindi devono mangiare per rifornire la loro energia che quando si esaurisce causa la loro morte (descritto nelle ultime tre linee di codice riportate in Cod. 1).

```

; Check model-version switch
; if we're not modeling grass, then the sheep don't need to eat to survive
; otherwise each grass' state of growth and growing logic need to be set up
ifelse model-version = "sheep-wolves-grass" [
  ask patches [
    set pcolor one-of [ green brown ]
    ifelse pcolor = green
      [ set countdown grass-regrowth-time ]
      [ set countdown random grass-regrowth-time ] ; initialize grass regrowth clocks randomly for brown patches
  ]
]
[
  ask patches [ set pcolor green ]
]

```

Codice 2 Gestisce l'erba

In questa versione l'erba è programmata in modo da essere infinita, infatti, nel codice riportato si può osservare un ciclo if-else che viene lanciato a seconda del modello scelto. Avendo scelto il modello "sheep-wolves" il ciclo non viene eseguito e l'erba viene impostata infinita tramite il comando "ask patches [set pcolor green]" questo fa in modo che le pecore non guadagnino o perdano energia.

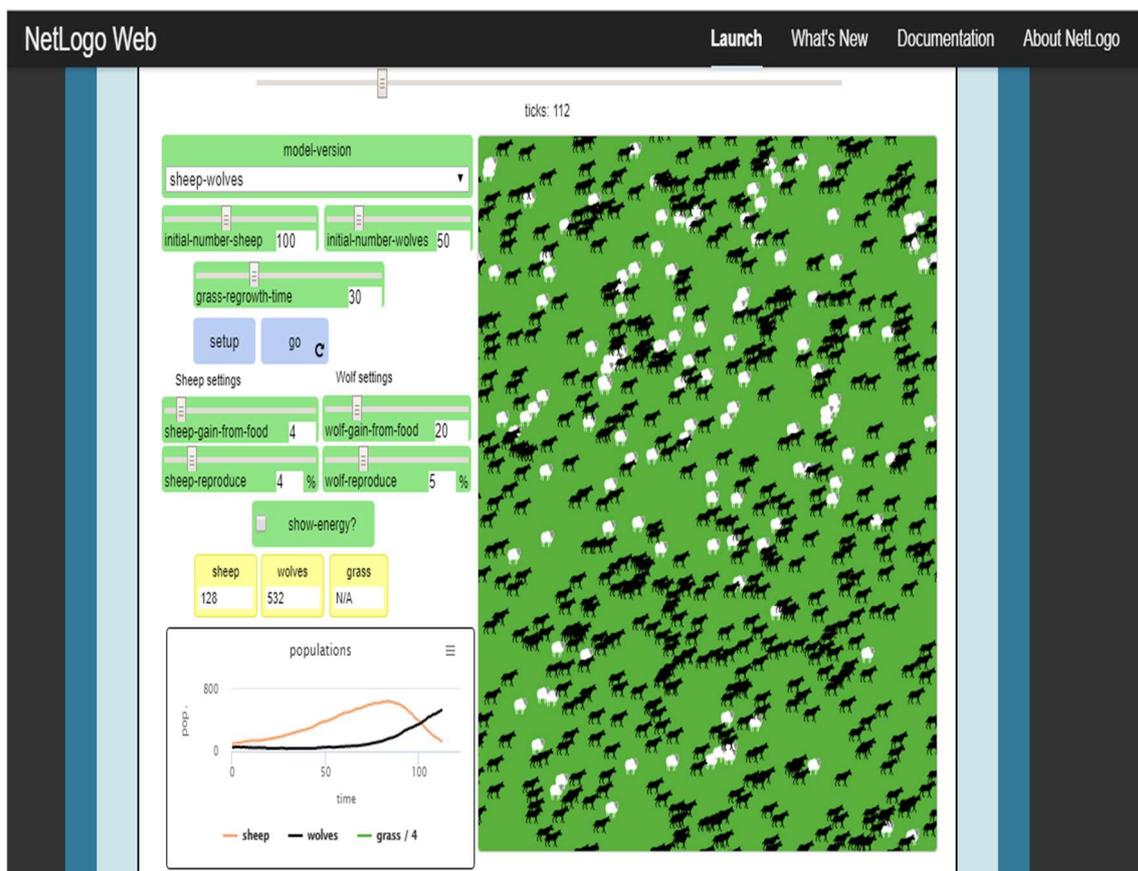


Figura 24 Sheep-Wolves”

Nella seconda versione "sheep-wolves-grass" si introduce come variabile l'erba che, come si vede nelle linee di codice della Cod. 2 , viene gestita grazie al colore della zolla.

Nelle linee di codice riportate in Cod. 3 si può osservare che c'è un countdown sulle zolle marroni che gestisce la ricrescita dell'erba tramite un ciclo if-else.

```
to grow-grass ; patch procedure
; countdown on brown patches: if you reach 0, grow some grass
if pcolor = brown [
  ifelse countdown <= 0
  [ set pcolor green
    set countdown grass-regrowth-time ]
  [ set countdown countdown - 1 ]
]
end
```

Codice 3 Countdown ricrescita dell'erba

Il comportamento dei lupi resta identico alla versione precedente. Il codice che gestisce il comportamento delle pecore è molto simile a quello che gestisce il comportamento dei lupi. Le pecore devono mangiare erba per mantenere alta la loro energia che viene gestita tramite un'istruzione condizionale (if) sul colore delle zolle, infatti, quando la zolla è verde l'energia della pecora aumenta. La riproduzione delle pecore è randomica e l'energia viene ripartita tra il genitore e la prole, come avviene per i lupi.

```
to eat-grass ; sheep procedure
; sheep eat grass and turn the patch brown
if pcolor = green [
  set pcolor brown
  set energy energy + sheep-gain-from-food ; sheep gain energy by eating
]
end

to reproduce-sheep ; sheep procedure
if random-float 100 < sheep-reproduce [ ; throw "dice" to see if you will reproduce
  set energy (energy / 2) ; divide energy between parent and offspring
  hatch 1 [ rt random-float 360 fd 1 ] ; hatch an offspring and move it forward 1 step
]
end
```

Codice 4 Gestisce il comportamento delle pecore

La morte di una preda o di un predatore per cause naturali è gestita dall'energia, infatti, come si vede nel codice riportato in Cod. 5 l'esaurimento dell'energia provoca la morte del soggetto.

```
to death ; turtle procedure (i.e. both wolf and sheep procedure)
; when energy dips below zero, die
if energy < 0 [ die ]
end
```

Codice 5 Estinzione di un soggetto causata dall'esaurimento dell'energia

Questa variante è più complessa della precedente ma allo stesso tempo è più stabile.



Figura 25 Sheep-Wolves-Grass

La seconda versione della simulazione corrisponde ai classici modelli di oscillazione della popolazione Lotka-Volterra.

Dopo aver scelto il modello possiamo regolare i vari parametri attraverso i cursori a disposizione. È possibile scegliere la dimensione iniziale della popolazione dei lupi e delle pecore, la quantità di energia che le pecore ottengono dall'erba (SHEEP-GAIN-FROM FOOD non utilizzabile nel modello pecore-lupi), la quantità di energia che i lupi ottengono per ogni pecora mangiata (WOLF-GAIN-FROM-FOOD), la probabilità che una pecora si riproduca ad ogni passo (SHEEP-REPRODUCE), la probabilità che un lupo si riproduca ad ogni passo (WOLF-REPRODUCE) e il tempo di rigenerazione dell'erba (GRASS-REGROWTH-TIME non utilizzabile nella versione pecore-lupi). Spuntando il comando SHOW-ENERGY viene mostrata l'energia di ogni animale. Una volta impostati i parametri premendo il tasto GO si avvia la simulazione.

È importante far notare agli studenti che utilizzano la variante pecore-lupi come fluttuano le due popolazioni e come la dimensione delle due sia collegata. Nella seconda variante è importante notare che viene aggiunta la popolazione dell'erba.

Per poter apprendere al meglio come funziona il modello, il sito mette a disposizione dei quesiti come ad esempio “Provare a regolare i parametri sotto diverse impostazioni. Quanto è sensibile la stabilità del modello ai parametri particolari? È

possibile trovare dei parametri che generano un ecosistema stabile nella variante del modello ovino-lupo?”

3.2.2 Segregazione

Questo modello è ispirato dagli studi che Thomas Schelling fece alla fine degli anni Sessanta per spiegare la comparsa di aree di segregazione urbana, i cosiddetti ghetti. Si suppone che un ambiente sia costituito da due tipologie di individui diversi tra loro, ciascun individuo è autonomo nelle proprie scelte, cioè decide il suo comportamento in relazione alla situazione in cui si trova. La regola alla base di questo modello è che ogni individuo decide di trasferirsi solo se meno di un terzo dei vicini sono del suo stesso tipo, cioè se il numero degli individui diversi è maggiore al 67%.

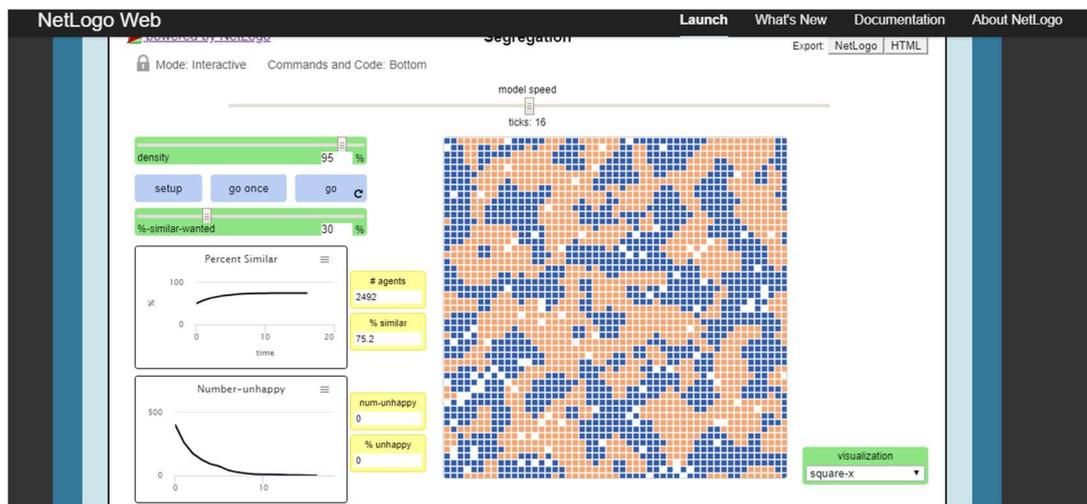


Figura 26 Segregation

“Segregation” simula il comportamento di due tipi di agenti (quadrantini arancione e blu) in un quartiere abitativo. Ogni agente arancione vuole assicurarsi di avere vicino una certa percentuale di agenti simili (%-similar-wanted), lo stesso vale per gli agenti blu. La simulazione mostra come questa preferenza individuale si diffonda nel quartiere, portando a modelli su larga scala. Facendo clic sul pulsante SETUP si impostano gli agenti, che di default sono circa di numero uguale. Cliccando GO si avvia la simulazione che agirà spostando gli agenti in caso vicino non avessero abbastanza vicini simili. Il cursore DENSITY controlla la densità di occupazione del quartiere, il cursore %-SIMILAR-WANTED controlla la percentuale di agenti dello stesso colore che ciascun agente vuole accanto. Nel grafico “Percent Similar” si può osservare l’andamento della

percentuale in funzione del tempo e viene anche mostrata la percentuale media di vicini dello stesso colore per ogni agente. Invece, nel grafico “Number-unhappy” si può osservare l’andamento degli agenti infelici in funzione del tempo, inoltre, viene mostrato anche la percentuale degli agenti che hanno meno vicini dello stesso colore di quello che vogliono. Infine, il selettore VISUALIZATION offre due opzioni per visualizzare gli agenti OLD e SQUARE-X queste due opzioni influiscono sulla simulazione solo a livello estetico.

Per capire meglio come funziona la simulazione si può osservare il codice che viene riportato in Cod. 6.

```

; run the model for one tick
to go
  if all? turtles [ happy? ] [ stop ]
  move-unhappy-turtles
  update-turtles
  update-globals
  tick
end

; unhappy turtles try a new spot
to move-unhappy-turtles
  ask turtles with [ not happy? ]
  [ find-new-spot ]
end

; move until we find an unoccupied spot
to find-new-spot
  rt random-float 360
  fd random-float 10
  if any? other turtles-here [ find-new-spot ] ; keep going until we find an unoccupied patch
  move-to patch-here ; move to center of patch
end

```

Codice 6 Corpo del codice della simulazione Segregation

Il codice opera tramite un ciclo condizionale, esso valuta inizialmente se tutti i soggetti sono felici, in caso affermativo viene interrotto, in caso negativo utilizza la funzione “move-unhappy-turtles” che muove il soggetto infelice in un posto vuoto. Questo avviene finché tutti i soggetti sono felici.

3.2.3 Cambiamento climatico

La simulazione “Climate Change” mostra un modello del flusso di energia termica nella Terra. Nella simulazione la Terra è colorata di rosa e la superficie del pianeta è una striscia verde che rappresenta l’albedo della terra, cioè indica il potere riflettente della superficie e può essere regolato grazie all’apposito cursore (albedo terrestre 0,37-0,39). Sopra la

superficie terrestre c'è l'atmosfera rappresentata in scala dal blu al bianco fino ad arrivare allo spazio aperto rappresentato in nero. È possibile aggiungere o rimuovere nuvole e anidride carbonica cliccando a destra sul corrispettivo comando. Le molecole di CO₂ rappresentano i gas serra che bloccano la luce infrarossa emessa dal pianeta. Le nuvole, invece, bloccano i raggi solari in entrata o in uscita influenzando il riscaldamento o il raffreddamento del pianeta.



Figura 27 Climate Change

L'energia della luce solare è rappresentata dalle linee gialle che “cadono” sulla terra. Alcuni raggi vengono riflessi dalle nuvole o dalla superficie terrestre, altri, invece, vengono assorbiti, trasformandosi in puntini rossi, causando l'aumento dell'energia termica. A volte l'energia termica (i puntini rossi) si trasformano in infrarossi (IR) causando un'emissione di energia da parte del pianeta. La probabilità che un puntino rosso diventi luce IR dipende dalla temperatura della terra, infatti più bassa è la temperatura della terra e più bassa è la probabilità che un puntino rosso generi luce IR e viceversa. L'energia IR è rappresentata con linee rosse, questa passa attraverso le nuvole ma può rimbalzare sulle molecole di CO₂.

La simulazione a prima vista può sembrare molto semplice, ma analizzando il codice si può apprezzare la complessità delle funzioni che lo compongono. Nelle impostazioni iniziali il colore del cielo e il colore dell'interno della terra sono predefiniti, l'unica variabile che l'utente può inserire e modificare è l'albedo.

```

to setup-world
  set sky-top max-pycor - 5
  set earth-top 0
  ask patches [ ;; set colors for the different sections of the world
    if pycor > sky-top [ ;; space
      set pcolor scale-color white pycor 22 15
    ]
    if pycor <= sky-top and pycor > earth-top [ ;; sky
      set pcolor scale-color blue pycor -20 20
    ]
    if pycor < earth-top
      [ set pcolor red + 3 ] ;; earth
    if pycor = earth-top ;; earth surface
      [ update-albedo ]
  ]
end

```

Codice 7 Impostazioni del mondo

L'albedo è descritta tramite una funzione che utilizza la scala di colore del verde come si può osservare in Cod. 8.

```

to update-albedo ;; patch procedure
  set pcolor scale-color green albedo 0 1
end

```

Codice 8 Funzione Albedo

```

to go
  ask clouds [ fd cloud-speed ] ; move clouds along
  run-sunshine ;; step sunshine
  ;; if the albedo slider has moved update the color of the "earth surface" patches
  ask patches with [pycor = earth-top]
    [ update-albedo ]
  run-heat ;; step heat
  run-IR ;; step IR
  run-CO2 ;; moves CO2 molecules
  tick
end

```

Codice 9 Corpo del codice della simulazione Climate Change

Nella Cod. 9 è riportato il “cuore” del codice, in queste linee di comando vengono eseguite le varie funzioni che permettono il corretto funzionamento della simulazione. La funzione “run-sunshine” controlla i raggi solari, “run-heat” controlla i raggi assorbiti dalla terra e l’aumento di temperatura, “run-IR” controlla i raggi infrarossi emessi dalla terra e “run-CO₂” controlla il movimento delle molecole di CO₂.

```

to run-sunshine
  ask rays [
    if not can-move? 0.3 [ die ] ;; kill them off at the edge
    fd 0.3 ;; otherwise keep moving
  ]
  create-sunshine ;; start new sun rays from top
  reflect-rays-from-clouds ;; check for reflection off clouds
  encounter-earth ;; check for reflection off earth and absorption
end

```

Codice 10 Funzione run-sunshine

```

to run-heat ;; advances the heat energy turtles
;; the temperature is related to the number of heat turtles
set temperature 0.99 * temperature + 0.01 * (12 + 0.1 * count heats)
ask heats
[
  let dist 0.5 * random-float 1
  ifelse can-move? dist
  [ fd dist ]
  [ set heading 180 - heading ] ;; if we're hitting the edge of the world, turn around
  if ycor >= earth-top [ ;; if heading back into sky
    ifelse temperature > 20 + random 40
    ;; heats only seep out of the earth from a small area
    ;; this makes the model look nice but it also contributes
    ;; to the rate at which heat can be lost
    and xcor > 0 and xcor < max-pxcor - 8
    [ set breed IRs ;; let some escape as IR
      set heading 20
      set color magenta ]
    [ set heading 100 + random 160 ] ;; return them to earth
  ]
]
end

```

Codice

11 Funzione run-heat

```

to run-IR
  ask IRs [
    if not can-move? 0.3 [ die ]
    fd 0.3
    if ycor <= earth-top [ ;; convert to heat if we hit the earth's surface again
      set breed heats
      rt random 45
      lt random 45
      set color red - 2 + random 4
    ]
    if any? CO2s-here ;; check for collision with CO2
    [ set heading 180 - heading ]
  ]
end

```

Codice 12 Funzione run-IR

È importante ricordare che la simulazione rappresenta un modello della realtà quindi vi sono molte semplificazioni importanti da sottolineare. Ad esempio, nella simulazione l'albedo terrestre, nonostante sia regolabile tramite l'apposito cursore che può avere un valore massimo di 1 (quando tutta la luce incidente viene riflessa) a un valore minimo di 0 (quando tutta la luce incidente viene assorbita), viene considerato costante, ma esso in realtà dipende dal tipo di materiale (l'albedo della neve fresca è circa 0.80-0.90, quello della sabbia di un deserto è 0.40 e quello dell'asfalto è circa 0.12).

Nessun modello è completamente preciso ma l'importante è che agisca allo stesso modo del sistema che dovrebbe modellare ed è quello che fa questa simulazione mostrando come l'effetto serra causato dalla CO₂ produca un aumento della temperatura terrestre.

3.2.4 Traffico

La simulazione "Traffic Basic" modella il movimento delle automobili su un'autostrada. Ogni vettura segue due semplici regole: decelera se vede un'auto più avanti e accelera se non vede un'auto più avanti. Infatti, osservando il codice della simulazione si può notare come questi due comportamenti siano controllati da un ciclo "if-else" che valuta la presenza di auto in prossimità dell'auto rossa: se davanti ci sono auto ("car-ahead != nobody") l'auto rossa viene decelerata tramite la funzione "slow-down-car" altrimenti viene accelerata tramite la funzione "speed-up-car".

```
to go
;; if there is a car right ahead of you, match its speed then slow down
ask turtles [
  let car-ahead one-of turtles-on patch-ahead 1
  ifelse car-ahead != nobody
    [ slow-down-car car-ahead ]
    [ speed-up-car ] ;; otherwise, speed up
  ;; don't slow down below speed minimum or speed up beyond speed limit
  if speed < speed-min [ set speed speed-min ]
  if speed > speed-limit [ set speed speed-limit ]
  fd speed
]
tick
end
```

Codice 13 Procedura principale di Traffic Basic

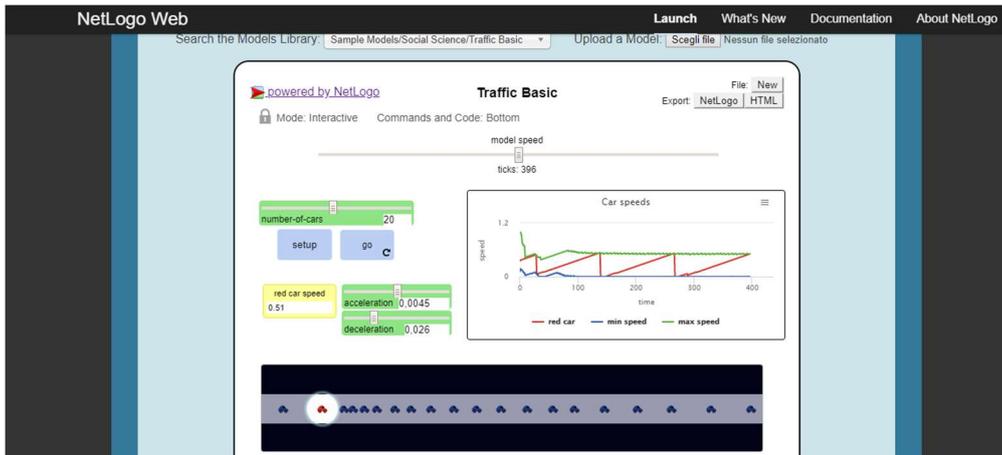


Figura 28 "Traffic Basic"

Questo modello dimostra come si possano creare ingorghi anche senza incidenti. In questa simulazione possiamo impostare il numero di macchine usando il cursore “NUMBER-OF-CARS”, la velocità con cui le auto accelerano quando non ci sono auto davanti con il cursore “ACCELERATION” e quanto un’auto è più lenta rispetto alla macchina di fronte attraverso il cursore “DECELERATION”. Va sottolineato che quando un’auto ne vede un’altra davanti eguaglia la velocità e poi rallenta un po’ di più. La posizione delle auto e le velocità iniziali sono generate in modo randomico, se alcune auto sono raggruppate insieme si muoveranno lentamente causando il rallentamento delle auto che le seguono fino a formare un ingorgo. Bisogna notare che anche se tutte le auto si muovono in avanti, gli ingorghi tendono a muoversi all’indietro questo tipo di comportamento è comune nei fenomeni d’onda. Il comportamento del gruppo è diverso dal comportamento dei singoli individui che lo compongono.

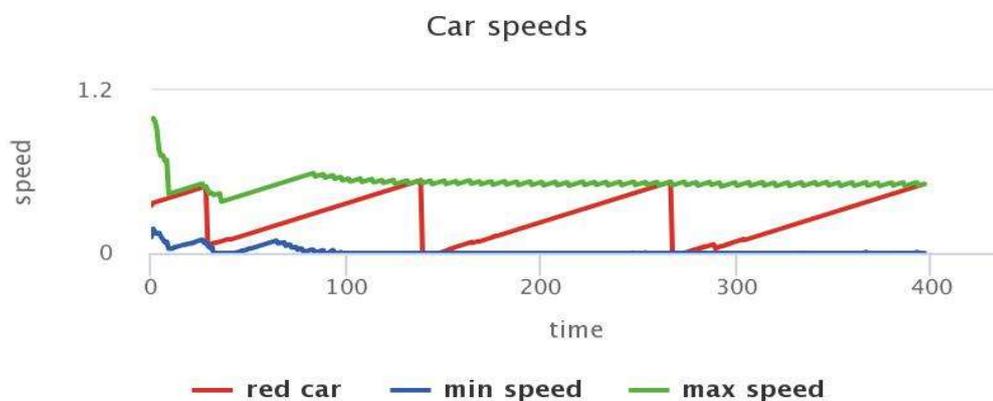


Figura 29 Grafico della velocità

Il grafico mostra tre valori: la velocità massima (rispettando i limiti) di qualsiasi auto in verde, la velocità minima di ogni auto in blu e la velocità dell'auto rossa. L'andamento dell'auto rossa mette in risalto l'andatura a singhiozzo.

3.3 Analisi comparativa

Ora che abbiamo descritto alcuni esempi di simulazioni PhET e NetLogo possiamo confrontarle tra loro per esaltarne le differenze. Per fare ciò abbiamo costruito una tabella (Tabella 1) in cui abbiamo inserito.

- Nella prima colonna il nome della simulazione
- Nella seconda le variabili che essa prende in input, ovvero tutte le variabili che l'utente può impostare a suo piacimento o che vengono fornite di default;
- Nella terza colonna ci sono le variabili di output, ovvero tutte le informazioni che restituisce la simulazione in diversa forma;
- Nella quarta colonna è descritto il modo in cui l'output delle simulazioni viene visualizzato graficamente;
- Nella quinta colonna sono raggruppate tutte le formule, per le simulazioni classiche (PhET), e le regole, per le simulazioni dei sistemi complessi (NetLogo), che connettono le variabili input con le variabili output;
- Nella sesta colonna sono indicati gli elementi probabilistici della simulazione, se presenti.

Nella tabella abbiamo raggruppato le simulazioni descritte in precedenza per sottolineare la differenza tra le simulazioni classiche (quelle del PhET) e quelle dei sistemi complessi (quelle di NetLogo).

La tabella 1 mette in grande rilievo la due principali differenze che desideravamo sottolineare; i diversi sistemi "causali" su cui si basano le simulazioni e il diverso significato di determinismo (legato alla presenza o meno di elementi probabilistici).

Nelle simulazioni classiche, tra grandezze di input (colonna 2) e grandezze di output (colonna 3) c'è un chiaro rapporto di causa-effetto che permette di prevederne gli sviluppi anche con carta e penna, senza necessità di utilizzare la simulazione per vedere il risultato. Tale rapporto è espresso dalle formule presenti in colonna 5 che connettono linearmente le grandezze in input e quelle in output. Ciò non è vero per le simulazioni NetLogo di

sistemi complessi, in cui le regole imposte agli agenti non connettono direttamente il comportamento del singolo agente all'output osservato su scala macroscopica. Questa caratteristica è tipica dei sistemi complessi che mostrano le cosiddette proprietà emergenti che sono collegate ai comportamenti individuali ma, più che dal comportamento dei singoli agenti, dipendono dalla loro interazione ed evoluzione nel tempo.

È, inoltre, possibile notare che nelle simulazioni dei sistemi meccanici classici (PhET) sono assenti elementi probabilistici, dunque le simulazioni sono dette deterministiche e forniscono, ogni volta che vengono eseguite, a partire dalle stesse condizioni iniziali, gli stessi risultati, esattamente come succederebbe se, con carta e penna, ponessimo gli stessi dati e calcolassimo il risultato basandoci su formule note. Ciò non avviene per le simulazioni NetLogo dove gli elementi randomici, espressi diversamente in ogni simulazione, costituiscono la base della simulazione. Difatti, osservando i codici delle simulazioni, si può notare che molte delle funzioni che gestiscono gli agenti sono di tipo random. Inoltre, nella tabella è stato messo in rilievo come, nelle simulazioni NetLogo, ad ogni settaggio e avvio della simulazione si ottiene una nuova configurazione iniziale e, di conseguenza, un differente output (colonna 4). Ad esempio, nella simulazione "Segregation" vengono visualizzati i grafici in funzione del tempo degli agenti soddisfatti e insoddisfatti; essi sono diversi ad ogni esecuzione anche se il parametro di soddisfazione (colonna di input) è mantenuto uguale.

Questa tabella, secondo noi, può diventare una guida per un'attività didattica con la quale guidare gli studenti a riflettere sulle strutture esplicative che stanno alla base delle simulazioni e sulla differenza tra sistemi fisici classici lineari e sistemi complessi.

Tabella 1 Analisi comparativa

	Grandezze di input	Grandezze di output	Visualizzazione dell'output	Connessione tra input e output	Elementi probabilistici
Masse e molle (PhET)	Massa appesa (m)	Lunghezza finale della molla (l_f)	Indicatori della lunghezza a riposo, della posizione di equilibrio e del centro di oscillazione	Equazione di bilancio forza peso e forza elastica $mg=k*\Delta l$	Assenti
	Gravità del pianeta (g)				
	Costante elastica (k)	Periodo (T)	Traccia del periodo	Periodo del moto $T=2\pi/\omega$ con $\omega^2=m/k$	
	Lunghezza a riposo della molla (l_i)	Posizione della massa nel moto oscillatorio $x(t)$	Posizione della massa nell'interfaccia grafica	Equazione del moto per la posizione $x(t)=A\cos(\omega t)$	
		Velocità della massa $x'(t)$	Vettore velocità	Equazione del moto per la velocità $x'(t)=-\omega*A\sin(\omega t)$	
	Accelerazione della massa $x''(t)$	Vettore accelerazione	Equazione del moto per la accelerazione $x''(t)=-\omega^2x(t)$		
Energia con lo skateboard (PhET)	Massa (m)	Velocità (v)	Tachimetro	Energia cinetica $K=1/2mv^2$	Assenti
	Coefficiente di attrito (μ)	Energia cinetica del sistema (K)	Istogramma o diagramma a torta	Energia cinetica $K=1/2mv^2$	
	Lunghezza della pista (l)	Energia potenziale del sistema (U)	Istogramma o diagramma a torta	Energia potenziale $U=mgh$	
	Switch per skater attaccato alla pista o no	Energia termica del sistema (Q_a)	Istogramma o diagramma a torta	Energia termica $Q_a=K-U-L_{attrito}=K-U-f_s*1$	

		Energia totale del sistema	Istogramma o diagramma a torta	Energia per un sistema isolato (senza attrito) $E_{mecc}=K+U$ Energia per un sistema non isolato (con attrito) $E_{sist}=E_{int}+E_{mecc}$	
Piano Inclinato (PhET)	Massa (m)	Forza di attrito (F_D)	Vettore	Forza di attrito $F_D=\mu_D * F_L$	Assenti
	Coefficiente di attrito dinamico (μ_d)				
	Coefficiente di attrito statico (μ_s)	Forza peso (F_P)	Vettore	Forza peso $P=mg$	
	Forza applicata (F_a)	Forza normale (F_N)	Vettore	Forza normale $F_N=mg \cos(\alpha)$	
Inclinazione piano (h)					
Attrito					
Parete elastica					
Posizione iniziale dell'oggetto	Forza risultante (F_{tot})	Vettore	Forza risultante $F_{TOT}=\sum F_i=mg * [\sin(\alpha)-\mu \cos(\alpha)]$		
Leve (PhET)	Massa (m)	Forza oggetto 1 (F_1)	Vettore	Condizione di equilibrio delle leve $F_1 * b_1 = F_2 * b_2$	Assenti
	Lunghezza braccio oggetto 1 (b_1)				
	Lunghezza braccio oggetto 2 (b_2)	Forza oggetto 2 (F_2)	Vettore		
		Livello di equilibrio	Indicatore livello di equilibrio		
			Righello per misurare la lunghezza del braccio		

Woolf Sheep Predation (NetLogo)	Numero iniziale di pecore	Numero finale di pecore	Grafico popolazione delle pecore in funzione del tempo	Gli agenti (turtles) si muovono casualmente nella mappa (patch), ad ogni clock compiono uno spostamento casuale. Ogni passo costa energia, se gli agenti non ricaricano la loro energia mangiando muoiono. Se due agenti dello stesso tipo si incontrano hanno una certa probabilità di riproduzione. Se avviene la riproduzione l'energia iniziale viene divisa tra il genitore e la progenie.	Ad ogni nuova esecuzione della simulazione, il setup viene fatto randomicamente e gli agenti disposti a caso secondo la densità stabilita.
	Numero iniziale di lupi				
	Tempo di rigenerazione dell'erba	Numero finale di lupi	Grafico popolazione dei lupi in funzione del tempo		
	Guadagno di energia dal cibo per le pecore				
	Guadagno di energia dal cibo per i lupi				
	Probabilità di riproduzione per le pecore	Numero di zolle d'erba	Grafico popolazione dell'erba in funzione del tempo		
	Probabilità di riproduzione per i lupi				
Segregation (NetLogo)	Densità degli agenti sulla griglia	Percentuale degli agenti soddisfatti	Grafico in funzione del tempo	Per ogni agente, ad ogni clock, se l'agente è circondato da meno della percentuale stabilita di vicini simili a lui, si sposta in una location random.	Ad ogni nuova esecuzione della simulazione, il setup viene fatto randomicamente e gli agenti disposti a caso secondo la densità stabilita.
	Percentuale di vicini simili che ogni agente vuole accanto				

		Percentuale degli agenti insoddisfatti	Grafico in funzione del tempo		La funzione di spostamento di ogni turtle (agente) in uno spazio libero ha due componenti randomiche: "rt random-float 360" la turtle ruota a destra di un numero di gradi random compreso tra 0 e 360; "fd random-float 10" la turtle si muove in avanti di un numero di passi random compreso tra 0 e 10.
Climate Change (NetLogo)	Intensità raggi solari Albedo Nuvole Anidride carbonica	Temperatura terrestre	Grafico aumento della temperatura terrestre in funzione del tempo	Gli agenti si muovono randomicamente per la mappa e tutte le interazioni tra loro sono di tipo probabilistico. Ad esempio, più è alta la temperatura terrestre e più è alta la probabilità che si emetta un raggio IR	La funzione di conversione dei raggi in calore se colpiscono di nuovo la superficie ha due componenti randomiche: "rt random 45" che ruota la turtle a destra di un numero di gradi random compreso tra 0 e 45 "rt random 45" che ruota la turtle a sinistra di un numero random di gradi compreso tra 0 e 45
Traffic Basic (NetLogo)	Numero di auto Accelerazione Decelerazione	Velocità dell'auto rossa	Grafico velocità delle auto in funzione del tempo	Se un agente si trova in prossimità di un altro agente decelera altrimenti accelera	La velocità delle auto aumenta di un numero random compreso tra 0 e 0.9 infatti: "set speed 0.1+random-float 0.9"

Conclusioni

Abbiamo iniziato questo viaggio nel mondo delle simulazioni con l'obiettivo di delineare i vantaggi del loro utilizzo didattico e di individuare un criterio di confronto tra le simulazioni PhET e NetLogo in modo da poter indicare nuovi strumenti, più accattivanti e coinvolgenti, per l'insegnamento delle scienze e, in particolare, della fisica.

La ricerca bibliografica di articoli e report ci ha consentito di delineare i vantaggi dell'utilizzo delle simulazioni nella didattica della fisica e delle scienze evidenziandone anche le criticità che sorgono quando gli studenti entrano in contatto con esse e di come sia possibile superarle, così da promuovere cambiamenti concettuali e progressi cognitivi. La sfida del cambiamento concettuale negli studenti si fa più ardua quando si ha a che fare con i sistemi complessi e le loro simulazioni. Infatti, gli studenti non riescono a rinunciare al "deterministic-centralized mindset" a favore di modelli stocastici.

Il focus della tesi è la ricerca di criteri di confronto tra le simulazioni di sistemi classici del PhET e le simulazioni dei sistemi complessi di NetLogo. Abbiamo scelto quattro simulazioni di meccanica classica PhET e quattro simulazioni NetLogo per poi studiarle e descriverle a fondo, in alcuni casi entrando nei codici sorgente. I risultati dello studio sono stati inseriti in una tabella in modo da poter visualizzare schematicamente le profonde differenze che contraddistinguono le diverse simulazioni. Questa analisi ci ha permesso di concludere che i due tipi di simulazioni non possono essere affrontate allo stesso modo. Infatti, le simulazioni PhET adottano un approccio deterministico, già molto consolidato negli studenti. In opposizione, le simulazioni NetLogo incarnano un approccio probabilistico e non deterministico e questo va esplicitato a priori nella didattica, altrimenti creerebbe delle criticità negli utenti.

Vorremmo concludere con un'osservazione personale. Le simulazioni sono diventate uno strumento sempre più diffuso anche nella cultura "pop" vedi la realtà virtuale o il film *Interstellar* (2014) in cui, grazie alla collaborazione tra il regista C. Nolan e il premio Nobel K. Thorne, viene mostrata un'immagine di buco nero (Gargatua) e del suo disco di accrescimento. Quest'ultima, infatti, è una simulazione senza precedenti che ha fornito una delle immagini più accurate di buco nero e ha indotto a continuare le ricerche in questo senso.

Riteniamo cruciale, quindi, continuare la ricerca sull'utilizzo delle simulazioni nella didattica per preparare le future generazioni all'utilizzo di questo strumento, affinché possano conoscere molti aspetti della realtà non sperimentabili empiricamente e affrontare con maggiore consapevolezza e competenza un futuro che, presumibilmente, farà largo uso degli ambienti simulativi.

Bibliografia

- Adams, W., Paulson, A., & Wieman, C. (2008). *What Levels of Guidance Promote Engaged Exploration with Interactive Simulations?* 1064. <https://doi.org/10.1063/1.3021273>
- Akpan, J. P., & Andre, T. (2000). Using a Computer Simulation Before Dissection to Help Students Learn Anatomy. *Journal of Computers in Mathematics and Science Teaching, 19*(3), 297–313.
- Ametller, J., & Pintó, R. (2002). Students' reading of innovative images of energy at secondary school level. *International Journal of Science Education, 24*(3), 285–312. <https://doi.org/10.1080/09500690110078914>
- Andaloro, G., Donzelli, V., & Sperandio-Mineo, R. M. (1991). Modelling in physics teaching: The role of computer simulation. *International Journal of Science Education, 13*(3), 243–254. <https://doi.org/10.1080/0950069910130303>
- Barelli, E., Branchetti, L., & Ravaioli, G. (2019). High school students' epistemological approaches to computer simulations of complex systems. *Journal of Physics: Conference Series, 1287*, 012053. <https://doi.org/10.1088/1742-6596/1287/1/012053>
- Bar-Yam, Y. (1997). *Dynamics of complex systems*. Addison-Wesley.
- Chang, K.-E., Chen, Y.-L., Lin, H.-Y., & Sung, Y.-T. (2008). Effects of learning support in simulation-based physics learning. *Computers & Education, 51*(4), 1486–1498. <https://doi.org/10.1016/j.compedu.2008.01.007>
- Chinn, C. A., & Malhotra, B. A. (2002). Epistemologically authentic inquiry in schools: A theoretical framework for evaluating inquiry tasks. *Science Education, 86*(2), 175–218. <https://doi.org/10.1002/sce.10001>

- Cilliers, P. (1998). *Complexity and Postmodernism: Understanding Complex Systems*. Routledge. <https://doi.org/10.4324/9780203012253>
- Feynman, R. P. (2002). Simulating Physics with Computers. In *Feynman and Computation*. CRC Press.
- Fougères, A.-J. (2012). Modelling and simulation of complex systems: An approach based on multi-level agents. *arXiv:1201.3880 [cs]*. <http://arxiv.org/abs/1201.3880>
- Greca, I., Seoane, M., & Arriasecq, I. (2014). Epistemological Issues Concerning Computer Simulations in Science and Their Implications for Science Education. *Science and Education*, 23. <https://doi.org/10.1007/s11191-013-9673-7>
- Grüne-Yanoff, T., & Weirich, P. (2010). The Philosophy and Epistemology of Simulation: A Review. *Simulation & Gaming*, 41(1), 20–50. <https://doi.org/10.1177/1046878109353470>
- Hmelo-Silver, C., Liu, L., Gray, S., & Jordan, R. (2015). Using Representational Tools to Learn About Complex Systems: A Tale of Two Classrooms. *Journal of Research in Science Teaching*, 52. <https://doi.org/10.1002/tea.21187>
- Holec, S., & Pfefferová, M. (2006). *Using simulations in physics education*.
- Jacobson, M. (2000). Problem Solving About Complex Systems: Differences Between Experts and Novices Complex Systems and Learning Issues. *Fourth International Conference of the Learning Sciences*.
- Jacobson, M. J., & Wilensky, U. (2006). Complex Systems in Education: Scientific and Educational Importance and Implications for the Learning Sciences. *Journal of the Learning Sciences*, 15(1), 11–34. https://doi.org/10.1207/s15327809jls1501_4

- Landriscina, F. (2009). Simulazione e apprendimento: Il ruolo dei modelli mentali. *Journal of e-Learning and Knowledge Society - Italian Version (until 2012)*, 5(2), Article 2. http://www.je-lks.com/ojs/index.php/Je-LKS_IT/article/view/209
- Liu, H.-C., Andre, T., & Greenbowe, T. (2008). The Impact of Learner's Prior Knowledge on Their Use of Chemistry Computer Simulations: A Case Study. *Journal of Science Education and Technology*, 17, 466–482. <https://doi.org/10.1007/s10956-008-9115-5>
- López, V., & Pintó, R. (2017). Identifying secondary-school students' difficulties when reading visual representations displayed in physics simulations. *International Journal of Science Education*, 39(10), 1353–1380. <https://doi.org/10.1080/09500693.2017.1332441>
- Nerdel, C., & Precht, H. (s.d.). *LEARNING COMPLEX SYSTEMS WITH SIMULATIONS IN SCIENCE EDUCATION*. 12.
- Sakellariou, I. (s.d.). Teaching Intelligent Agents using NetLogo. *Proceedings of Informatics ... Recuperato 1 luglio 2021*, da https://www.academia.edu/2316285/Teaching_Intelligent_Agents_using_NetLogo
- Schelling, T. C. (1971). Dynamic models of segregation. *The Journal of Mathematical Sociology*, 1(2), 143–186. <https://doi.org/10.1080/0022250X.1971.9989794>
- Sklar, E. (2007). NetLogo, a Multi-agent Simulation Environment. *Artificial Life*, 13(3), 303–311. <https://doi.org/10.1162/artl.2007.13.3.303>
- Smetana, L. K., & Bell, R. L. (2012). Computer Simulations to Support Science Instruction and Learning: A critical review of the literature. *International Journal*

of Science Education, 34(9), 1337–1370.

<https://doi.org/10.1080/09500693.2011.605182>

Tisue, S., & Wilensky, U. (2004). *NetLogo: A simple environment for modeling complexity*. 16–21.

Wieman, C. E., Adams, W. K., Loeblein, P., & Perkins, K. K. (2010). Teaching Physics Using PhET Simulations. *The Physics Teacher*, 48(4), 225–227.

<https://doi.org/10.1119/1.3361987>

Wilensky, U., & Reisman, K. (2006). Thinking Like a Wolf, a Sheep, or a Firefly: Learning Biology Through Constructing and Testing Computational Theories—An Embodied Modeling Approach. *Cognition and Instruction*, 24(2), 171–209.

https://doi.org/10.1207/s1532690xci2402_1

Zaharija, G., Grubač, A., & Granić, A. (2014). LEARN – LEgo Robot and Netlogo. *Proceedings of CIET 2014*, 209.