

ESEMPI DI METODI COMPOSITIVI

FARE MUSICA «IN REGOLA»

- L'applicazione di regole "algoritmiche" nella composizione musicale ha radici molto antiche
- Nel ventesimo secolo il concetto viene portato all'estremo nelle composizioni del serialismo di Schoenberg, Webern, Boulez e altri
- Nei tempi recenti c'è stato un cambio di rotta in alcune correnti (ad es. Arvo Pärt), che abbandonano gli estremismi per tornare alla musica tonale, in cui però alcuni parametri possono comunque essere decise da regole predefinite

MUSICA DAL TESTO

- Teorie di Noam Chomsky per definire le strutture del linguaggio
- Allo stesso modo ci si può aspettare che creando musica partendo dal testo si ottengano dei risultati strutturali analoghi
- Un primissimo esempio di processo algoritmico per scegliere le note musicali da utilizzare partendo dal testo è di Guido d'Arezzo (1000 d.C.)

ALGORITMO DI GUIDO D'AREZZO

- Le vocali presenti in un testo vengono associate a diversi pitch, mentre la distanza che le separa indica le figure ritmiche da utilizzare
- Guido d'Arezzo sosteneva comunque che la sua idea doveva fornire solo degli elementi di base per il compositore, che poi utilizzava questi elementi per la creazione dei suoi brani

ESEMPIO

(**GuidoWordMusic.java**)

Nell'applicazione:

- Viene creata un'associazione tra le vocali e i pitch, con una scelta casuale tra 3 pitch diversi per ogni vocale
- Viene usato una croma come figura ritmica di base: all'aumentare del numero di consonanti che separano le vocali aumenta la figura ritmica
- Ogni pitch successivo viene scelto ad una distanza massima di 7 semitoni dal precedente
- Viene suonata la melodia risultante (a 120 BPM) e viene mostrato lo score corrispondente

TINTINNABULI (ARVO PÄRT)

- Arvo Pärt (1935-) scrisse agli esordi musica dodecafonica
- Negli anni '70 coniò lo stile compositivo Tintinnabuli, in cui:
 - Ci sono 2 tipi di voce
 - La prima arpeggia la triade tonica
 - La seconda si muove diatonicamente in un movimento graduale

ESEMPIO

(Arvoish.java)

Nell'applicazione:

- Vengono usate 4 voci: la melodia e 3 voci per l'accompagnamento
- La composizione è basata sul modo eolio (scala minore naturale) e sulla triade di LA minore
- Vengono create melodie con dei random walk
- Vengono create armonie impostando un set di regole per la scelta dei pitch e delle durate

MUSICA ALEATORIA E STOCASTICA

- Per la scelta di parametri compositivi:
 - La musica aleatoria impiega processi casuali
 - La musica stocastica impiega metodi probabilistici

GIOCO DEI DADI DI MOZART

L'esempio più famoso di musica aleatoria è il metodo dei dadi di Mozart *Musikalisches Würfelspiel* (anche se non è certa la paternità), usato per generare dei valzer nello stile del compositore

- Vengono fornite 176 frasi musicali (battute)
- Vengono fornite regole di composizione delle frasi
- Vengono scelte le giustapposizioni delle frasi con il lancio di due dadi

L'aleatorietà entra in gioco in una fase della composizione: il materiale di base è definito

In generale, nella musica aleatoria spesso il caso entra in gioco durante l'esecuzione, mentre il compositore definisce regole e/o frammenti melodico/ritmici

ESEMPIO

(**MozartDiceGame.java**, **MozartPhraseList.java**)

- MozartPhraseList.java contiene l'insieme di tutte le frasi musicali da utilizzare per la generazione dei valzer
- MozartDiceGame.java si occupa della simulazione del lancio dei dadi e della creazione della composizione risultante
- La composizione avrà la forma ABAB, per un totale di $8*4=32$ battute
- Da notare che sia la battuta 8 che la battuta 16 hanno un numero limitato di scelte, per consentire di terminare le singole parti in modo consonante

LA MUSICA DI JOHN CAGE

- L'aleatorietà portata agli estremi caratterizza alcune composizioni di John Cage (1912-1992)
- In questo caso si arriva a far decidere al caso tutti i parametri di una composizione (pitch, timbri, durate, ecc.)
- Esempi di metodi utilizzati:
 - Lancio di dadi
 - Scelta di carte da un mazzo mischiato
 - Composizione di numeri di telefono casuali
 - Risposta ad azioni casuali compiute dal pubblico in sala

LA MUSICA DI IANNIS XENAKIS

- Anche se esempi di musica composta con metodi stocastici sono presenti da prima, con l'avvento dei calcolatori questo tipo di composizioni acquista nuova linfa, grazie alle capacità computazionali
- Iannis Xenakis (1922-2001), compositore, ingegnere e architetto, basa le sue tecniche compositive sulla *Teoria dei Giochi* di John von Neumann
- A differenza della musica aleatoria, il processo compositivo è definito con regole probabilistiche più elaborate

MATRICI STOCASTICHE

- Nei processi stocastici si può definire la probabilità di ottenere un certo risultato tra quelli possibili (meglio se il numero è finito!)
- Le matrici stocastiche elencano per ogni risultato la sua probabilità di accadere
- Es. (5 risultati possibili: 1-5):

1	2	3	4	5
0.05	0.15	0.5	0.2	0.1

- La somma delle probabilità sarà sempre 1!
- Come è possibile simulare un'estrazione che si basi sulle probabilità fornite?

ESEMPIO

(StochasticMatrix.java)

- Viene presa in considerazione la matrice stocastica della slide precedente:

1	2	3	4	5
0.05	0.15	0.5	0.2	0.1

- Si effettua un'estrazione tra 0 e 1, e si identifica il range in cui ricade l'estratto per trovare il risultato finale:

1	2	3	4	5
0 - 0.05	0.05 - 0.2	0.2 - 0.7	0.7 - 0.9	0.9 - 1

- Nell'esempio vengono effettuate 10 estrazioni
- A posteriori vengono calcolate le percentuali di scelta

CATENE DI MARKOV IN MUSICA

- Le catene di Markov sono processi di scelta in cui:
 - i valori ottenibili sono finiti
 - la probabilità di ottenere un risultato dipende dal risultato precedente (e solo da quello)
- In musica vengono usate in processi di analisi/generazione (v. Xenakis) in cui:
 - si studiano le occorrenze dei valori di un certo parametro in dipendenza dal suo valore precedente
 - si modellano delle matrici di transizione che indicano le probabilità di scelta
 - si applicano le matrici per ottenere una catena di Markov basata sull'analisi effettuata

CATENE DI MARKOV IN MUSICA

- Ad es. si prendono i valori dei pitch di una melodia:
64,62,60,62,64,64,64,62,62,62,64,67,67,64,62,60,62,64,64,64,64,62,62,64,62,60
- Si crea una matrice che indica il numero di volte in cui un certo pitch segue un certo altro:

		Stato seguente			
		60	62	64	67
Stato precedente	60	0	2	0	0
	62	3	3	4	0
	64	0	5	5	1
	67	0	0	1	1

CATENE DI MARKOV IN MUSICA

- Si converte la matrice in matrice di transizione calcolando le probabilità riga per riga:

	60	62	64	67
60	0.0	1.0	0.0	0.0
62	0.3	0.3	0.4	0.0
64	0.0	0.45454545	0.45454545	0.09090909
67	0.0	0.0	0.5	0.5

ESERCIZIO

(BasicMarkov.java)

- Basandosi sulla matrice di transizione definita nella slide precedente, creare una melodia in cui:
 - i pitch vengono calcolati con un processo markoviano, partendo dal pitch iniziale 64
 - le durate sono sempre quelle di default
 - vengano generate 30 note

VARIANTE DI ORDINE MAGGIORE DI 1

- Se la probabilità di uno stato dipende non solo dal precedente, ma dai precedenti m , la catena markoviana si dice di ordine m
- Ad es., sempre considerando la melodia presentata e una catena di ordine 2, la matrice di transizione diventa:

	60	62	64	67
64,62	0.6	0.4	0.0	0.0
62,60	0.0	1.0	0.0	0.0
60,62	0.0	0.0	1.0	0.0
62,64	0.0	0.25	0.50	0.25
64,64	0.0	0.4	0.6	0.0
62,62	0.0	0.3333333	0.6666666	0.0
67,67	0.0	0.0	1.0	0.0
64,67	0.0	0.0	0.0	1.0
67,64	0.0	1.0	0.0	0.0

ESEMPIO

(`MarkovTest.java`)

- jMusic mette a disposizione la classe `PhraseMatrix`, che modella un processo markoviano di ordine specificato, con la possibilità di gestire pitch, durate e dinamica
- Nell'esempio viene generata una melodia prendendo in considerazione il brano degli esempi precedenti
- Vengono considerati pitch, durate, e dinamica e si utilizza una catena di Markov di ordine 2

GENERAZIONE «NATURALE»

- La creazione artistica ha spesso preso ispirazione da fenomeni fisici, naturali, biologici
- Da sempre ciò ha portato alla creazione di opere d'arte che rappresentano questi fenomeni o li descrivono, ad esempio in forma di musica
- In anni recenti c'è stato un ritorno di questi concetti nella creazione artistica, con la possibilità – offerta dalla tecnologia – di modellare i fenomeni e usare questi modelli per creare opere d'arte

SIMULAZIONE DI PROCESSI BIOLOGICI

- Una branca dell'informatica si occupa di simulare processi biologici, creando vita artificiale
- Un famoso algoritmo sul comportamento degli stormi d'uccelli in volo (ma anche di banchi di pesci e altro) è l'algoritmo Boids (1986) di Craig Reynolds
- Simula il fatto che un gruppo di uccelli ha delle caratteristiche di uniformità anche in assenza di leader
- Si basa sull'interazione tra forme di vita artificiale, dette boid, che si muovono in 3D
- Le traiettorie scelte dalle singole AI si basano (nella forma più semplice) su tre regole:
 - *separazione*: il boid evita il sovraffollamento locale
 - *allineamento*: il boid cerca di allinearsi con i boid vicini
 - *coesione*: il boid si muove verso la posizione media rispetto ai boid vicini

ESEMPIO

(Swarm.java)

- Per applicare (in modo semplice) alla creazione musicale i concetti espressi si è deciso di tradurre le tre regole come segue (un boid è costituito da una nota):
 - *separazione*: la nota dovrà avere pitch diverso dalle altre
 - *allineamento*: la nota avrà lo stesso valore ritmico delle altre
 - *coesione*: la nota converge verso un pitch medio e fa parte di una scala pentatonica

AUTOMI CELLULARI

- Un formalismo usato per descrivere l'evoluzione di sistemi complessi è l'automa cellulare (Cellular Automaton, CA)
- Un automa è costituito da un insieme di celle (in 1D, 2D, 3D...) che assumono un insieme finito di stati (tipicamente «vivo» o «morto»)
- Viene definito il concetto di celle «vicine», ad es.:
 - in 1D, le 2 celle che affiancano una certa cella
 - in 2D con celle quadrate, le 24 celle che distano al max 2 passi da una certa cella
- Dato uno stato iniziale, l'automa si evolve modificando lo stato delle celle
- Lo stato delle celle si modifica secondo regole che dipendono dallo stato della cella stessa e da quello delle celle «vicine»

AUTOMI CELLULARI – 1D

- Un esempio di applicazione del caso monodimensionale alla musica consiste nel far evolvere un pattern ritmico
- Rappresentando un pattern in n suddivisioni, corrispondenti a n celle di un automa, la singola configurazione si può rappresentare anche come intero a n bit (1: «vivo», 0: «morto»); ad es: 0110
- Attenzione: ogni evoluzione avviene in un unico frammento di tempo (altrimenti dipenderebbe dall'ordine in cui vengono controllate le celle)

ESEMPIO

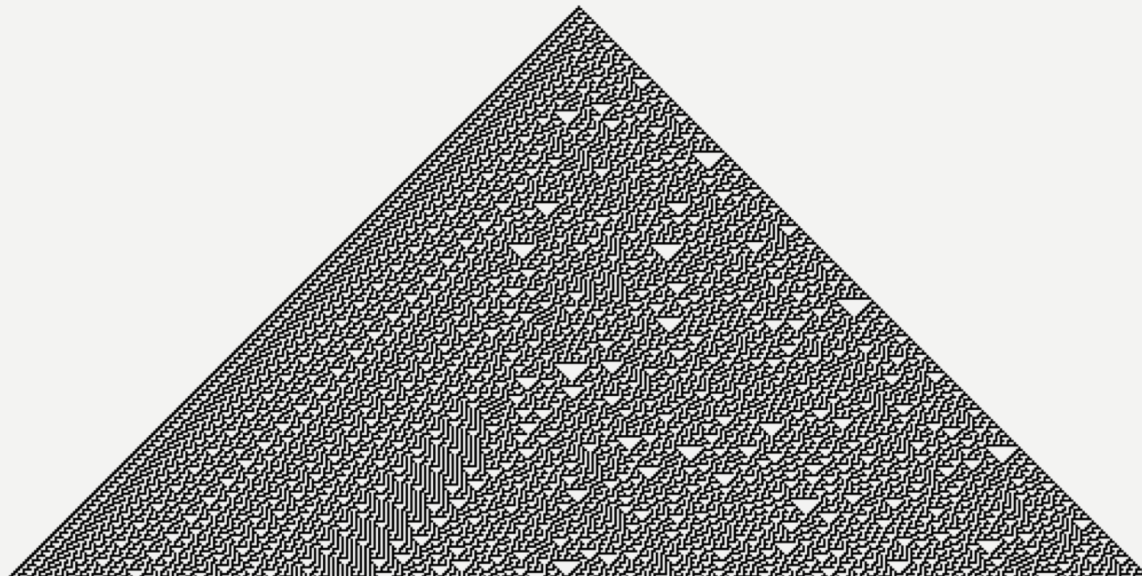
(**RhythmicAutomata.java**)

- Nell'esempio vengono create 24 battute da 4/4, ritmicamente divise in sedicesimi
- Vengono usati 3 timbri: Hats, Rim Shot, Claps
- Per ogni timbro, la battuta iniziale è creata casualmente, poi si evolve secondo le seguenti regole (vengono considerate «vicine» le 4 celle adiacenti ad una specifica):
 - se una cella ha 1 o 2 celle vicine vive, la cella vive
 - se una cella ha 4 celle vicine vive, la cella muore
 - se una cella ha 0 o 3 celle vicine vive, la cella cambia stato

AUTOMI CELLULARI – 1D

- Se si identificano come celle vicine le 2 adiacenti, per specificare le regole alla base dell'evoluzione si può utilizzare la notazione di Wolfram: il nome è indicato dal numero decimale che, in binario, fornisce la tabella delle regole. Ad es. (CA 30):

Configurazione	111	110	101	100	011	010	001	000
Nuovo stato cella centrale	0	0	0	1	1	1	1	0



ESERCIZIO

(`RhythmicAutomata2.java`)

Modificare l'esempio precedente

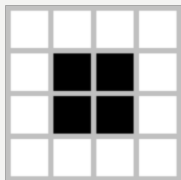
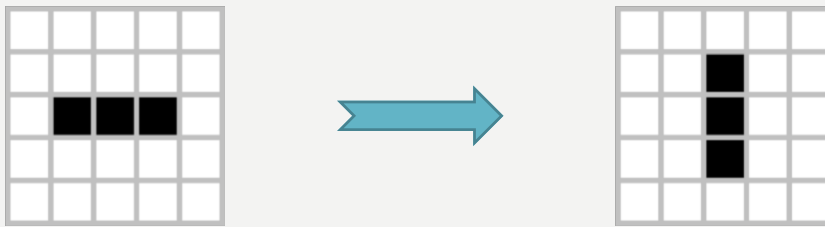
- Vengono considerate vicine le 2 celle adiacenti
- Le battute vengono suddivise in ottavi
- Specificare in una costante la regola di evoluzione (CA_n)
- Far evolvere le singole voci di conseguenza

GIOCO DELLA VITA

- Anche se i concetti di base sono molto semplici, gli automi cellulari presentano comportamenti che descrivono bene processi matematici, fisici, biologici
- La più famosa formulazione è il *Gioco della vita* di John Conway (anni '60)
- Il «mondo» è costituito da una griglia infinita di caselle quadrate in 2D
- Ogni cella ha 8 «vicini»: le celle immediatamente adiacenti
- Ogni «generazione» si evolve nel modo seguente:
 - Qualsiasi cella viva con meno di due celle vive adiacenti muore, come per effetto d'isolamento
 - Qualsiasi cella viva con due o tre celle vive adiacenti sopravvive
 - Qualsiasi cella viva con più di tre celle vive adiacenti muore, come per effetto di sovrappopolazione
 - Qualsiasi cella morta con esattamente tre celle vive adiacenti diventa una cella viva, come per effetto di riproduzione

GIOCO DELLA VITA – ESEMPI

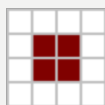
- Attenzione: si ricordi che ogni evoluzione avviene in un unico frammento di tempo (altrimenti dipenderebbe dall'ordine in cui vengono controllate le celle)



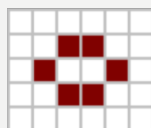
GIOCO DELLA VITA – ESEMPI

- Still lifes

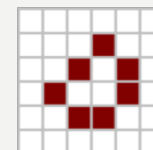
Block



Hive

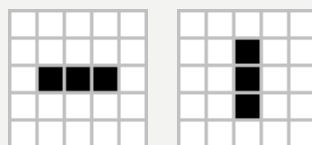


Loaf

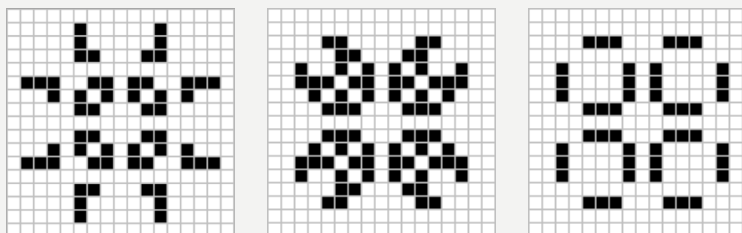


- Oscillators

Blinker (periodo 2)

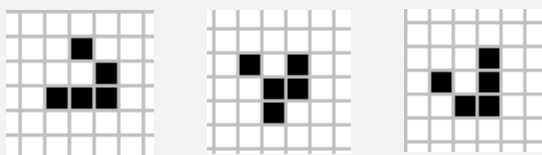


Pulsar (periodo 3)



- Spaceships

Glider



ESEMPIO

(AutomataMusic.java)

- Nell'esempio viene usata la classe CellularAutomata di jMusic per creare una griglia 10 x 10 di Gioco della vita
- Vengono eseguiti 100 step di evoluzione
- Per ogni step viene scelto un pitch da aggiungere alla melodia che dipende dal numero di celle complessivamente vive

Attenzione!

- il parametro di seed sembra avere problemi!
- le regole originali del Gioco sono modificate!
 - Qualsiasi cella viva con due celle vive adiacenti sopravvive
 - Qualsiasi cella viva con meno o più di due celle vive adiacenti muore
 - Qualsiasi cella morta con due o tre celle vive adiacenti diventa una cella viva

GIOCO DELLA VITA E MUSICA

- L'esempio precedente usa una regola semplicissima per generare dei pitch
- Più spesso:
 - o le regole sono molto più complesse
 - oppure l'automa viene usato per modellare suoni e non musica a livello simbolico
- Ad esempio, si utilizzano parametri dell'automa per controllare un suono ottenuto con sintesi granulare

SINTESI GRANULARE (IN BREVE)

- Nella sintesi granulare un suono (generato sinteticamente o molto più spesso un file audio) viene scomposto in *grani* da 1-100ms
- I grani vengono combinati e riprodotti a velocità, fase e ampiezze diverse
- Viene generata una nuvola di suoni che può avere risultati molto eterogenei e interessanti
- Iannis Xenakis adottò la sintesi granulare per creare alcune sue composizioni, però allora veniva usato il nastro magnetico, che veniva tagliato, rallentato, filtrato, ecc.

ESEMPIO

(GranularCA.java)

- Nell'esempio viene usata la sintesi granulare messa a disposizione da jSyn, elaborando i parametri con un automa cellulare creato in jMusic
- I grani avranno una durata insolitamente grande, usando la sintesi granulare in modo degenerare
- Anche il sample usato è insolitamente lungo (un frammento dei Pini di Roma di Respighi)
- Durata e frequenza di riproduzione dei grani dipendono dal numero di celle vive