

Scheda didattica di DuploCar

Un robot che disegna fatto con Lego Duplo

S1. Uso di Kitronik Servo:Lite Board

In questo progetto ho usato la scheda di espansione *Kitronik Servo:Lite Board* perché facile da usare, dal costo molto contenuto e perché dotata di molti software di controllo completi. Il link è il seguente:

kitronik.co.uk/5623

E' necessario caricare il software di controllo della scheda di espansione **kitronik-servo-lite** dal menu **Extension** di **MakeCode**.

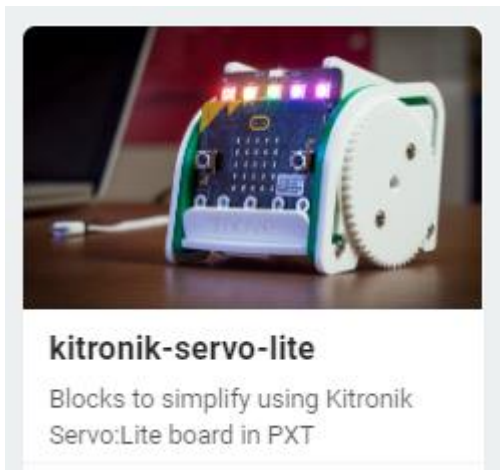


Figura 1 Scheda da caricare in Extension di MakeCode

S2. Come calibrare i servo motori

Per avere una rotazione continua delle ruote ho usato i servomotori (*motori*) a rotazione continua, definiti anche "servo a 360 gradi".

Il controllo dei servo a **rotazione continua** avviene definendo la **velocità** di rotazione e non la **distanza** angolare come per i normali servo di 180°.



Figura 2 Schema di definizione della velocità del servo 360°

Scheda didattica

primaria

secondaria



20-35 min

Supporti per docente

Obiettivi principali

Gli alunni saranno in grado di controllare i movimenti usando principi di geometria

Tempo: 20-30 min

Gruppo di lavoro: tutta la classe

Materiali: Lego Duplo + micro:bit con servo:lite e due servomotori

Rubrica delle competenze

Competenze di asse: **matematica**

- individuare le strategie appropriate per la soluzione di problemi

Sito del progetto Otto Cardy:

www.cad-tutor.com/duplocar



[Video su YouTube](#)



[File GitHub](#)

DuploCAR scheda 1

Nella figura segue con il **pulsante A** i servo P1 e P2 ruotano entrambi in avanti alla massima velocità per 1 secondo e poi si fermano. Con il **pulsante B** ruotano al contrario. Entrambi dopo Con il **pulsante B** ruotano al contrario. Entrambi dopo 1 sec. si fermano perché la velocità **90** significa STOP.

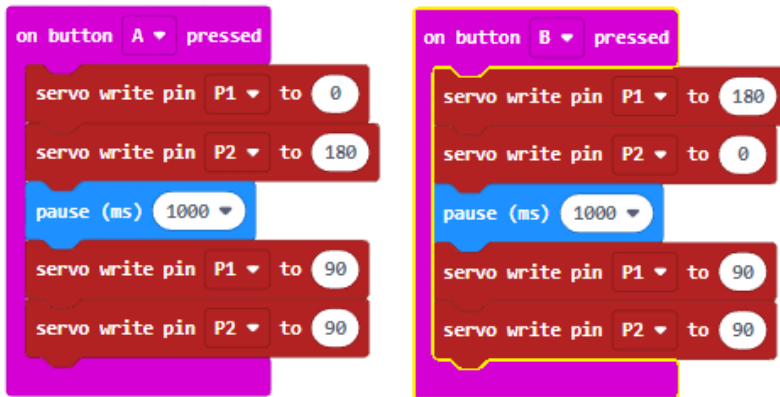


Figura 3 Rotazione in avanti e stop

Nota 1: Ricorda che i servo sono speculari quindi i valori vanno invertiti per avere la stessa direzione

NOTA 2: non si potrà mai avere una precisione assoluta fra i valori di angolo **indicati** nel codice rispetto a quelli ottenuti in concreto dal robot per diversi motivi quali: carica delle batterie, attrito non sempre costante delle ruote e velocità del movimento.

Coding

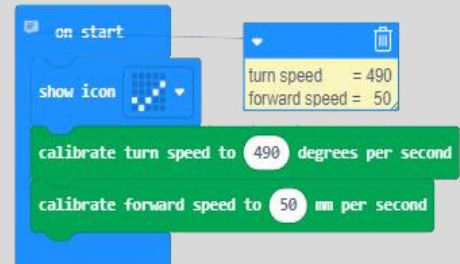


Figura 4 Come calibrare i servo

Con i servomotori a rotazione continua è necessario calibrare i motori per la **velocità** degli spostamenti **lineari** e **angolari**.

Nella mia configurazione con i tipo di servo che ho utilizzato, questi sono i valori **turn speed (angolo)** e **forward speed (avanti)** per avere un buona corrispondenza dei movimenti.

Queste istruzioni di "calibrate" andranno inserite in ogni programma in **"On start"**.

S 3. Disegnare un quadrato

Il disegno di un quadrato è un algoritmo molto semplice: vai avanti di **L** e gira di **90°**

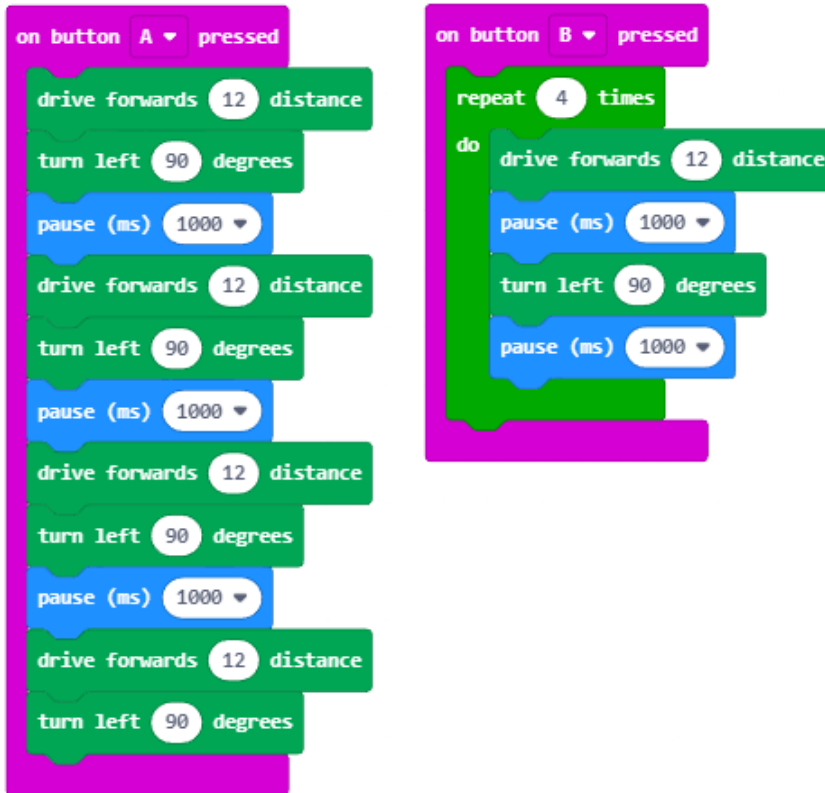


Figura 5 A sinistra sequenza estesa e a destra in un loop di repeat

Nella figura a sinistra, **pulsante A**, il disegno del quadrato è realizzato con tutte le singole istruzioni mentre a destra con il **pulsante B** le 2 principali istruzioni (*avanti di 12, gira a sinistra di 90°*) sono inserite in un loop di **repeat** 4 volte.

Notare che l'istruzione **pause (ms) 1000** (*pausa 1000 millisecondi cioè di 1 sec*) è necessaria perché permette al motore di posizionarsi. La pausa di 1 sec. è il minimo per avere un movimento adeguato.

A documentazione di queste schede ho inserito in YouTube e in GitHub gli esempi dei movimenti e del software.

Grafica con Scratch

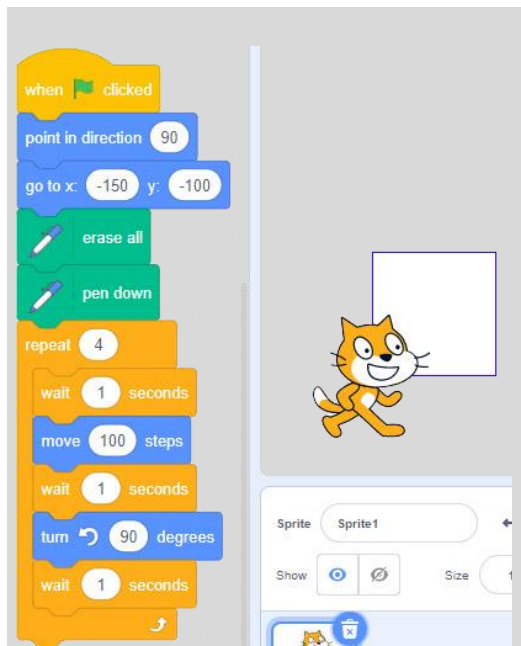


Figura 6 Quadrato con Scratch

Anche con **Scratch** si può usare la funzione **Repeat** per disegnare un quadrato in modo ridotto, con i seguenti comandi:

- **avanti** di 100
- **gira** di 90°

e ripetuti **4 volte**.



[Video su YouTube](#)



[File su GitHub](#)

S 4. Disegnare un poligono

Per disegnare un poligono si utilizza un **Funzione** che permette di ricevere in entrata due parametri: **sides** (*numero lati*) e **length** (*lunghezza dei lati*).

In questo modo si possono disegnare, con un'unica funzione, molti tipi di poligoni indicando in ingresso come variabili i valori numerici di **sides** e **length**.

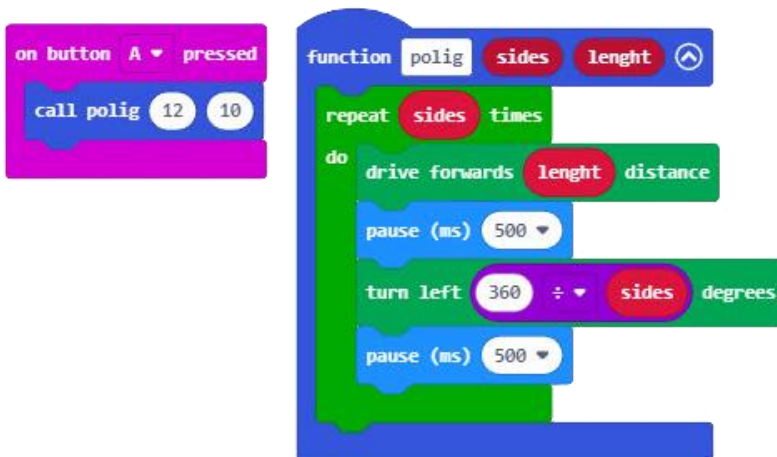


Figura 7 Funzione di Polig

La **Funzione polig** viene chiamata con il pulsante A dove si possono specificare **n.12 lati** con **lunghezza 10**.

Disegno con Scratch

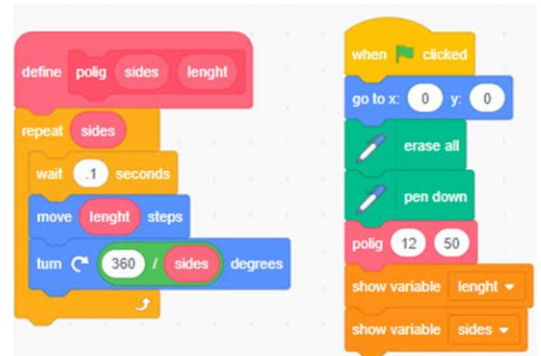


Figura 8 Disegno di un poligono

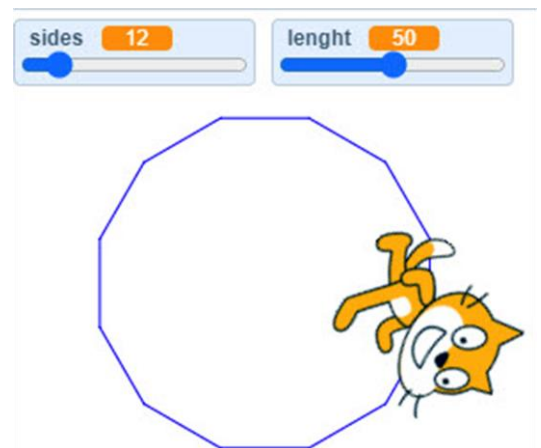


Figura 9 Il poligono di 12 lati

Il blocco **wait** in Scratch ha la funzione di rallentare il disegno per permettere di controllare il singolo movimento.

In Scratch la **Funzione polig** è creata con il comando **Make a Block** (*crea un Blocco*)

Esercizi 1.2

1. Inserire un terzo parametro corrispondente al tempo di pausa
2. E' possibile inserire l'angolo al posto del numero di lati?

S 5. Disegnare un cerchio

La formula per calcolare una circonferenza è: $2 \pi r$
 In figura X il cerchio che viene disegnato ha valori:
 raggio = 20 e circonferenza divisa in 30 segmenti.

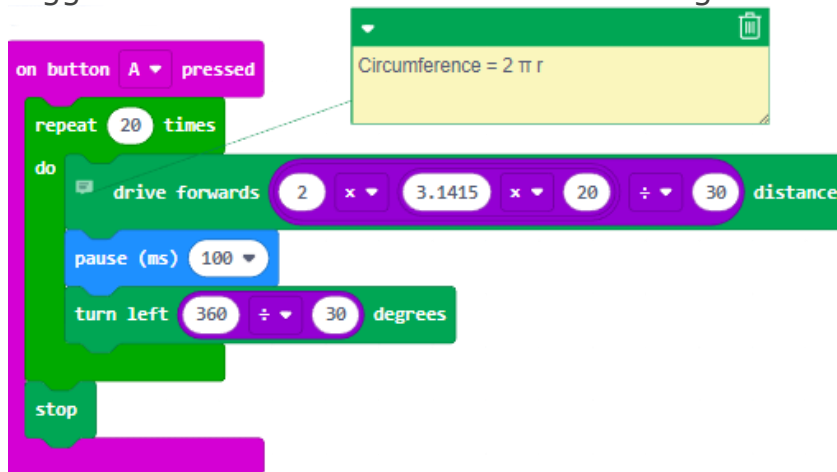


Figura 10 Circonferenza divisa in 20 segmenti

Qui non è usata **Funzione** alcuna e questo comporta di dover modificare i valori singolarmente.

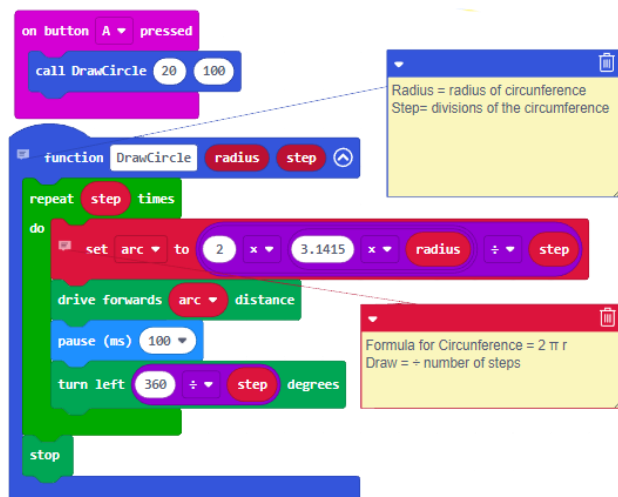


Figura 11 Funzione Circonferenza

Ho usato una **Funzione** con due parametri: **Radius** e **Step** dove *Radius* è il raggio e *Step* è il numero di segmenti della circonferenza. Maggiore è il valore di **Step** maggiore sarà la precisione di tracciamento.
 Nota: le dimensioni del disegno non possono essere precise come al plotter, ovviamente.

Disegno con Scratch

Lo stesso algoritmo viene usato anche in **Scratch** con una funzione con una sola variabile del raggio di 100. La circonferenza è suddivisa in 360 segmenti per avere una maggiore precisione del disegno.

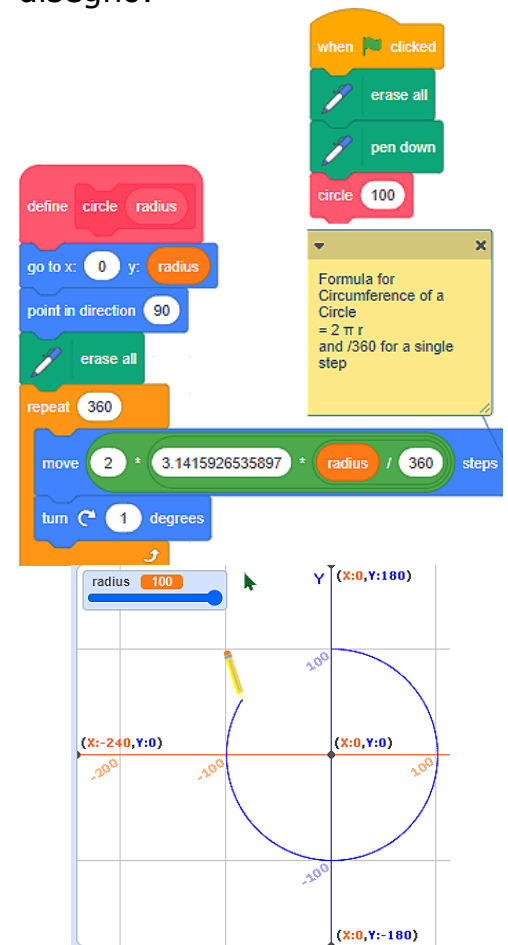


Figura 12 Disegno del Cerchio

S 6. Disegnare un frattale con MakeCode

E' molto divertente far disegnare un frattale al car robot DuploCAR perché il disegno che ne esce è assolutamente random e casuale in tutti i suoi aspetti.

Visto che non si può ottenere una precisione del disegno per i motivi accennati in precedenza, il disegno che se ne ricava è più simile ad un "disegno d'arte" che una grafica frattale.

Buon divertimento.

Allego i link al software in Hub e ad alcuni video pubblicati su YouTube



Video su YouTube



File su GitHub

Disegno con Scratch

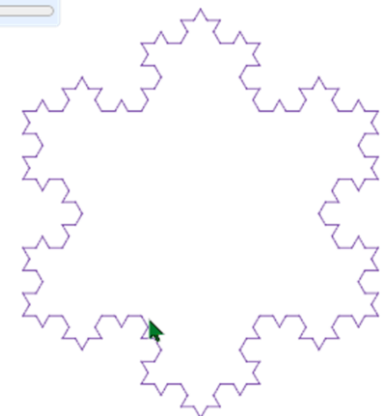


Figura 13 Curva di Koch

Esercizi 1.4

1. Inserire un commento in un blocco
2. Cosa modifica la funzione "External inputs"?

S 7. Disegnare un frattale con Scratch

Con Scratch il disegno frattale è molto più preciso e chiaro potendo controllare gli effetti diretti della variazione delle variabili.

Allego i link al software in Hub e ad alcuni video pubblicati su YouTube

Un mandala è una serie di disegni geometrici che si generano dalla rotazione concentrica di figure piane semplici.

In **Scratch** si possono trovare molti esempi di **Koch Snowflake** dalle forme e dai colori più svariati.

Un disegno frattale può essere riprodotto anche con la stampante 3D con una profondità e spessore vario.

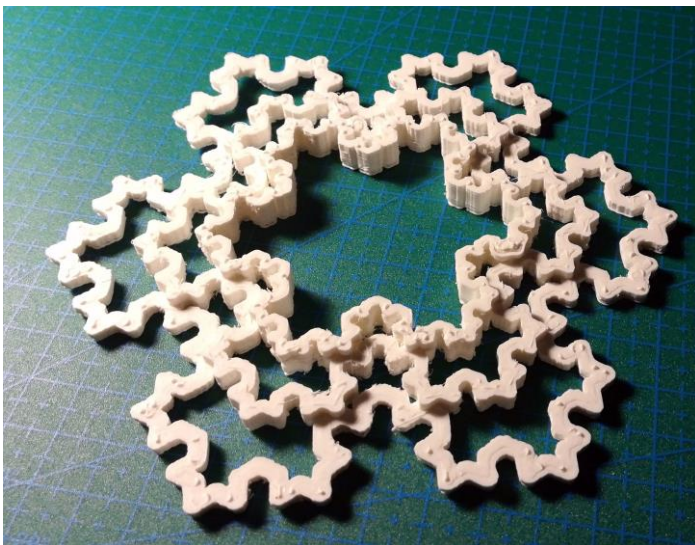


Figura 15 Curva di Koch ripetuta e stampata con stampante 3D

Disegno con Scratch

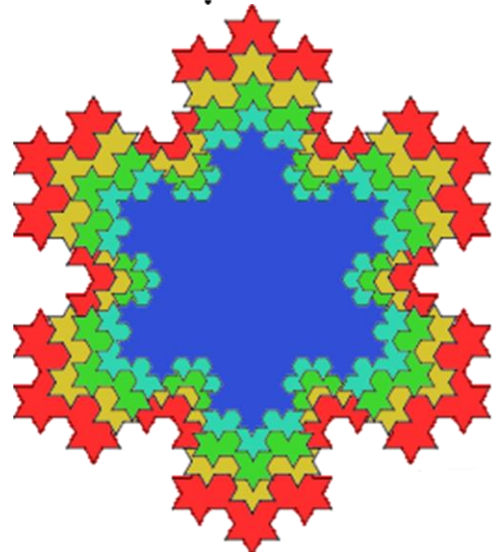
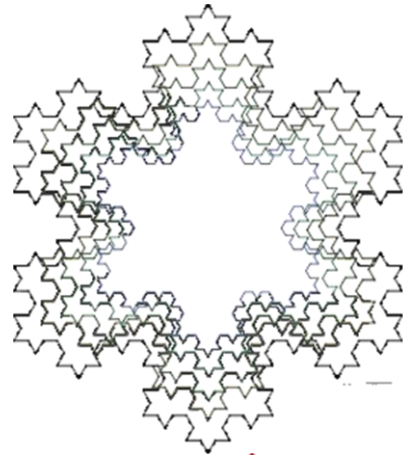


Figura 14 Curva di Koch ripetuta

Esercizi 1.5

1. Creare delle copie di frattale che poi si possono colorare con un programma di pittura
2. Che ne dici di riprodurre lo stesso disegno con la stampante 3D?

S 8. Disegnare un mandala con Scratch

Un *mandala* è un disegni concentrici decorati con Un mandala è un disegno composta dalla ripetizione di figure geometriche di base ripetute secondo uno schema circolare.

Con **Scratch** la costruzione è molto semplice perché si basa su due **loop (repeat)** inseriti l'uno dentro l'altro con la figura geometrica continuamente ruotata di un angolo limitato.

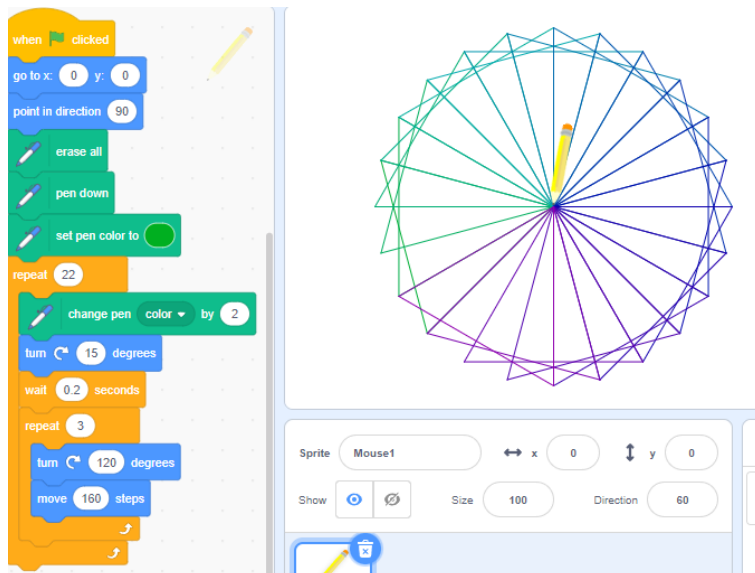
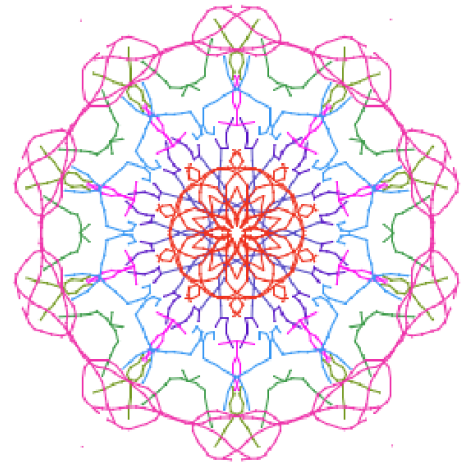


Figura 16 Semplice mandala