

Claudio Gasparini

**Per le Scuole:**

Primaria

Secondaria 1° grado

Secondaria 2° grado

STEAM - Coding

Tinkering - Arduino

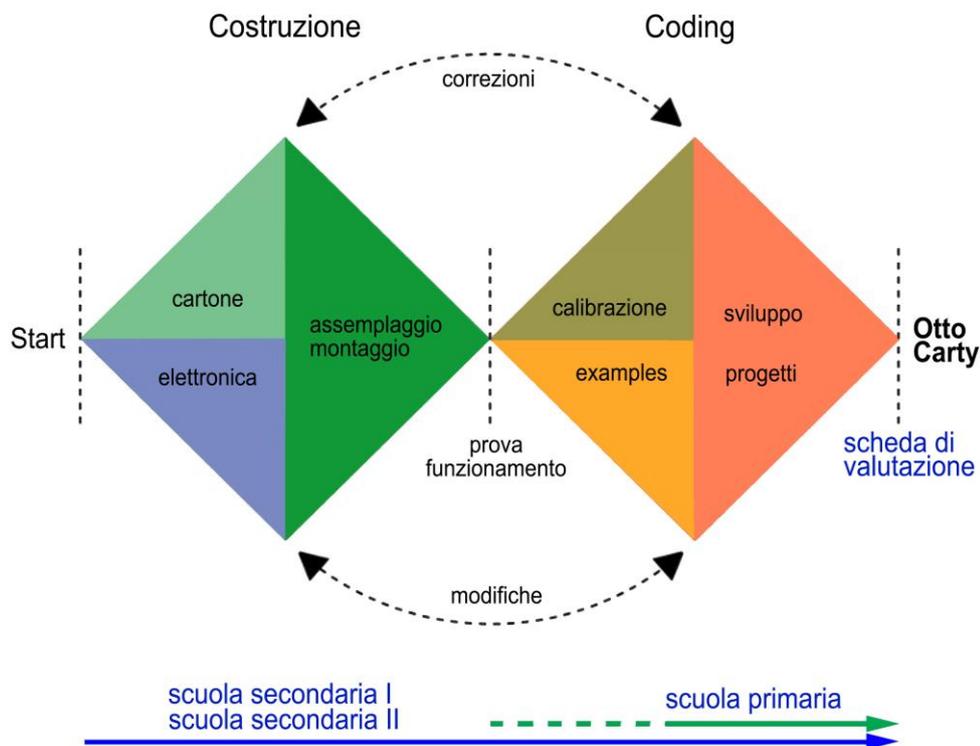
Robotica - Makers

# Otto Cardy Robot

Costruisci il tuo Robot in cartone

**Schede didattiche per il docente**

## Schema di Design Thinking applicato a Otto Cardy



Come costruire un robot in cartone che cammina con Arduino Nano e programmato con Google Blockly



# Otto Cardy **schede**

Costruisci un robot di cartone - Claudio Gasparini - [www.cad-tutor.com/otto\\_cardy\\_1](http://www.cad-tutor.com/otto_cardy_1)

---

Prima edizione: Febbraio 2022

Seconda edizione: Ottobre 2023

© Copyright 2023 Claudio Gasparini

[claudio.gasparini@gmail.com](mailto:claudio.gasparini@gmail.com)

Info del Progetto: [www.cad-tutor.com/otto\\_cardy/](http://www.cad-tutor.com/otto_cardy/)

Info dell'Autore: [www.cad-tutor.com/gasparini](http://www.cad-tutor.com/gasparini)

Hanno collaborato: Aurora Mangiarotti, Rosangela Mapelli

Questo eBook è visualizzato utilizzando il 100% di bytes riciclati. Nessun bit è stato maltrattato durante la produzione di questo libro 😊



# Otto Cardy **schede**

Costruisci un robot di cartone - Claudio Gasparini - [www.cad-tutor.com/otto\\_cardy\\_2](http://www.cad-tutor.com/otto_cardy_2)

## Schede didattiche di Otto Cardy

**Esercizi guidati** per apprendere l'uso del coding a blocchi per il controllo del movimento e dei suoni del robot di cartone **Otto Cardy**

### S 1.0 Piano della lezione insegnante

#### 1. Preparazione

- Puoi preparare alcuni esempi (*sketch*) di movimento scelti fra i file **Examples** di *OttoBlockly*.
- Individua delle varianti alle consegne da assegnare ai singoli gruppi

#### 2. Coinvolgimento (5 min.)

- Fai una panoramica dei movimenti e suoni forniti dal programma che è possibile aggiungere.
- Presenta la funzione del "sensore di profondità" e la distanza che rileva
- Fai notare l'altezza delle note e la lunghezza del suono.

#### 3. Esplorazione (15 min.)

- Invita gli studenti a lavorare a coppie o in gruppi
- Invita a elaborare una strategia di lavoro utilizzando le funzioni di **suoni-movimenti-danza-distanza**.

#### 4. Elaborazione (20 min.)

- Invita gli studenti a programmare dei movimenti con utilizzo dei **suoni** e dei **gesti**.
- Chiedi di inserire delle sequenze musicali di loro gradimento fra i movimenti del robot

#### 5. Valutazione

- Crea una scheda di valutazione dei lavori dei singoli studenti prendendo spunto dalle sezioni di valutazione fornite.

8 schede didattiche

primaria



secondaria

### Supporti per l'insegnante

#### Obiettivi principali

Gli alunni saranno in grado di controllare i movimenti ed aggiungere i suoni

**Tempo:** 20-30 min

**Gruppo di lavoro:** tutta la classe

**Materiali:** Otto Cardy + OttoBlockly

#### Rubrica delle competenze

Competenze di asse: **matematica**

- individuare le strategie appropriate per la soluzione di problemi

Competenze di asse: **informatica**

- scegliere strategie utili alla soluzione di un problema
- elaborare semplici istruzioni per controllare il comportamento del robot

**Sito del progetto Otto Cardy:**

[www.cad-tutor.com/otto\\_cardy](http://www.cad-tutor.com/otto_cardy)

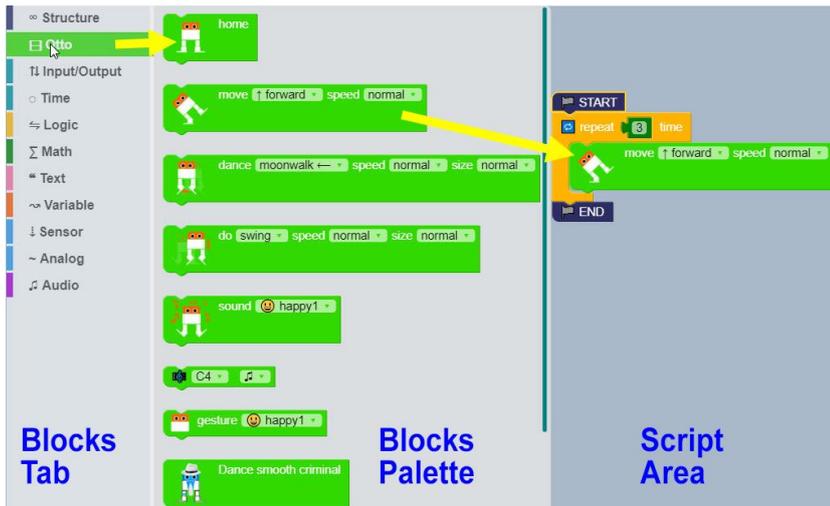


# Otto Cardy **schede**

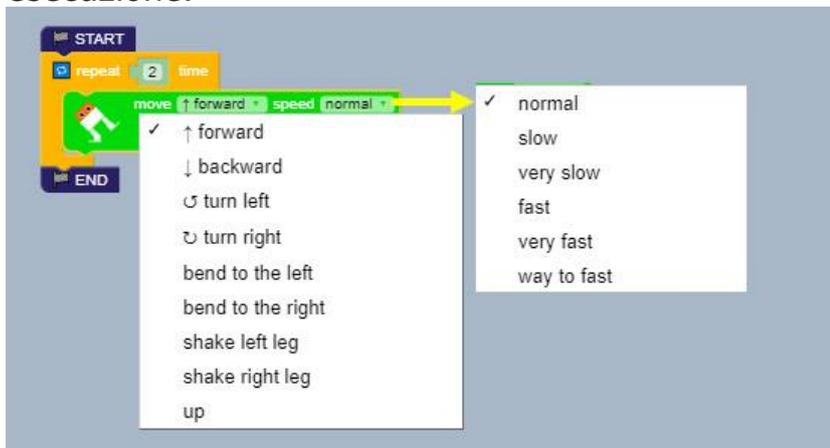
Costruisci un robot di cartone - Claudio Gasparini - [www.cad-tutor.com/otto\\_cardy\\_3](http://www.cad-tutor.com/otto_cardy_3)

## S1.1 Insegniamo a camminare al robot

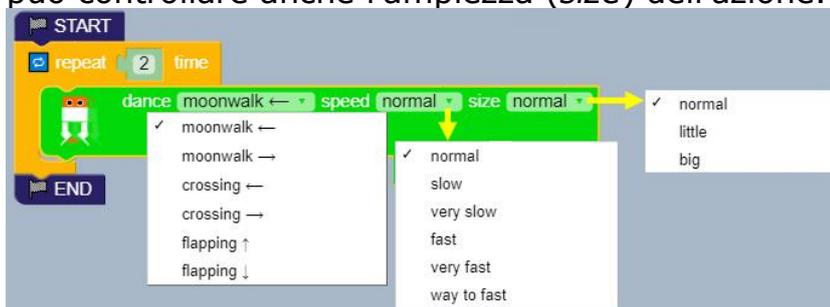
Nel menu *Blocks Tab* scegliere la voce **Otto** che fornisce, nella *Blocks Palette*, alcuni blocchi con movimenti già predisposti.



Movimenti associati al blocco **MOVE**: ad ogni movimento si può assegnare una velocità di esecuzione.



Movimenti del blocco **DANCE**: oltre alla velocità si può controllare anche l'ampiezza (*size*) dell'azione.



## Supporti per l'insegnante

### Movimenti di OttoBlockly

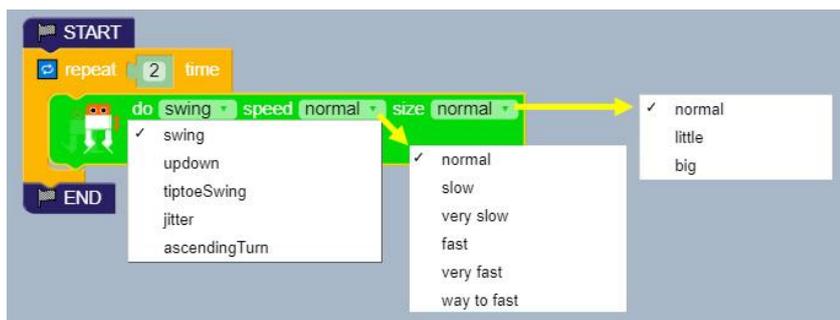
- **move**: si muove in avanti, in dietro, si piega e si solleva sulle punte;
- **dance**: sono n.3 tipi di danza quali *moonwalk*, *crossing* e *flapping*;
- **do**: muove le gambe con i seguenti movimenti di *swing*, *updown*, *tiptoeSwing*, *Jitter* e *ascendingTurn*;
- **gesture** e **sound**: sono due elenchi di *gesti* e *suoni* che possono essere usati anche combinati insieme.



# Otto Cardy **schede**

Costruisci un robot di cartone - Claudio Gasparini - [www.cad-tutor.com/otto\\_cardy\\_4](http://www.cad-tutor.com/otto_cardy_4)

Il blocco **DO** aggiunge dei semplici gesti come strisciare, alzarsi sulle punte ed altri ancora: ad ogni movimento si può assegnare una velocità e un'ampiezza di esecuzione.



I blocchi **SOUND** e **GESTURE** permettono di inserire alcuni *suoni* del cicalino e dei *gesti* che possono essere coordinati fra di loro.



Movimenti del blocco **dance**: oltre alla velocità si possono controllare anche l'ampiezza (*size*) dell'azione.

## Supporti per l'insegnante

### Suoni e movimenti coordinati

Associare dei movimenti a dei suoni può rappresentare una sfida e allo stesso tempo un'occasione di discussione sui significati da dare a suoni e movimenti.

Ad esempio, che movimento associare al suono "surprise" potendo disporre solo del movimento delle gambe e dei piedi?

Si potrebbe iniziare con la simulazione dal vero associando i suoni di Otto con i movimenti reali delle gambe dei ragazzi. Sarebbe divertente anche associare i suoni ai movimenti reali delle gambe e dei piedi.

E se il robottino avesse anche le braccia? Che movimenti si potrebbe associare ad ogni suono?



# Otto Cardy **schede**

Costruisci un robot di cartone - Claudio Gasparini

[www.cod-tutor.com/otto-cardy](http://www.cod-tutor.com/otto-cardy) 5

## S 1.2 Inserire i movimenti

Se si inserisce un solo blocco nell'area *Script*, esempio **MOVE**, il programma esegue l'istruzione all'infinito.



Per interrompere un'istruzione o una serie di istruzioni s'inserisce l'istruzione **END**. L'istruzione di blocco ha in genere una durata di un secondo.

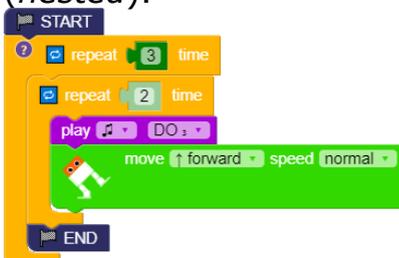


Il blocco **REPEAT** (LOOP) fornisce un ciclo di istruzioni contenute al suo interno per un numero determinato di volte. Alla fine passa all'istruzione successiva e se non la trova ripete all'infinito.

Per interrompere la sequenza va inserito il blocco **END**. Il blocco **START** non è necessario.



Si può anche inserire due loop uno dentro l'altro (*nested*).



### Codice Arduino

(senza le righe di testa)

```
void loop() {
  Otto.walk(1,1000,1); //
  FORWARD
}
```

```
void loop() {
  Otto.walk(1,1000,1); //
  FORWARD
  while(true);
}
```

```
void loop() {
  for (int count=0 ; count<3 ;
  count++) {
    Otto.walk(1,1000,1); //
  FORWARD
  }
  while(true);
}
```

### Esercizi 1.2

1. Inserire due movimenti con una pausa di 1 secondo fra i due;
2. Quante volte viene suonata la nota nello schetch riportato a fianco?

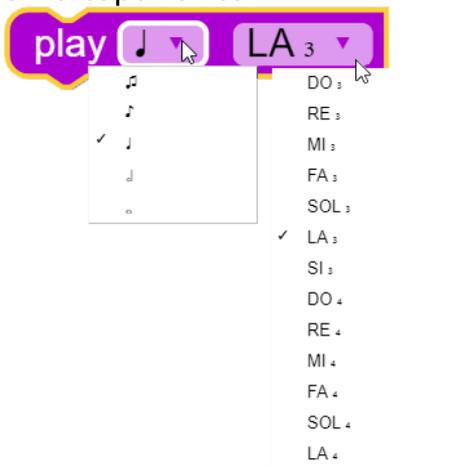


# Otto Cardy **schede**

Costruisci un robot di cartone - Claudio Gasparini - [www.cad-tutor.com/otto\\_cardy\\_6](http://www.cad-tutor.com/otto_cardy_6)

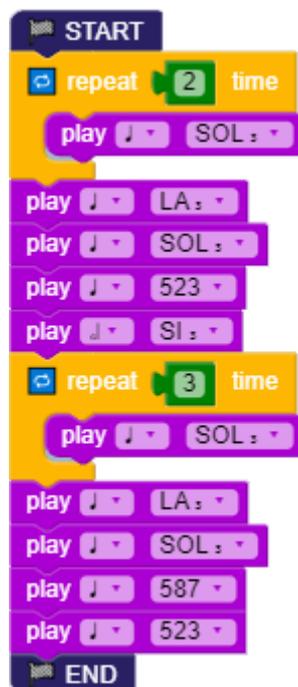
## S 1.3 Inserire i suoni

Il Menu Audio permette di inserire delle note attraverso il *cicalino*, quindi di non grande qualità sonora, oppure caricare un file mp3 se è installato un lettore con altoparlante.



Il blocco **PLAY** permette di inserire una nota controllando sia la sua durata che l'altezza. L'altezza delle note si realizza in due ottave.

Nell'esempio seguente sono visualizzate in sequenza le note corrispondenti alla motivo musicale **Happy Birthday**. Notare il blocco **REPEAT** utilizzato due volte per prolungare le note del ritornello.



## Codice Arduino

```
void loop() {
  tone(13,523,2000);
  delay(2000);
}
```

```
void loop() {
  for (int count=0 ; count<2 ;
count++) {
  tone(3,392,500);delay(500);}
  tone(13,440,500);delay(500);
  tone(13,392,500);delay(500);
  tone(13,523,500);delay(500);
  tone(13,493,1000);delay(1000);
```

```
for (int count=0 ; count<3 ;
count++) {
  tone(13,392,500);delay(500);}
  tone(13,440,500);delay(500);
  tone(13,392,500);delay(500);
  tone(13,587,500);delay(500);
  tone(13,523,500);delay(500);
  while(true);
}}
```

### Esercizi 1.3

1. Aggiungere le note musicali che completano il motivetto
2. Comporre un nuovo motivo fra quelli che ti piacciono



# Otto Cardy **schede**

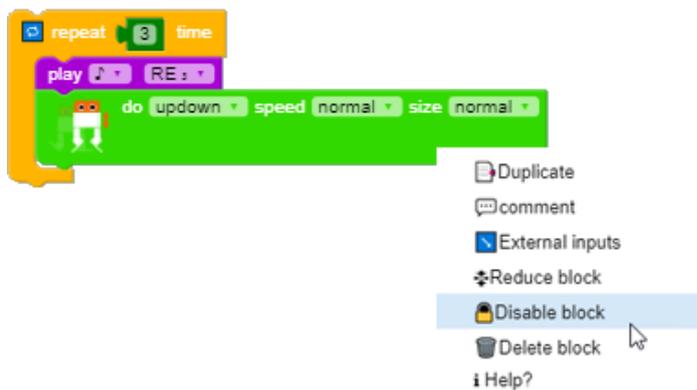
Costruisci un robot di cartone - Claudio Gasparini - [www.cad-tutor.com/otto\\_cardy\\_7](http://www.cad-tutor.com/otto_cardy_7)

## S 1.4 Regole dei blocchi

Per testare i diversi movimenti può essere utile inserire delle note all'inizio di ogni azione.

A differenza di **Scratch** e di altri programmi a blocchi, tutti i blocchi presenti nella **Script Area** vengono processati anche se non sono collegati tra di loro come in figura.

Infatti se osserviamo il codice Arduino a lato, tutte le linee del codice sono attive anche se i blocchi non sono collegati.



Cliccando il tasto destro del mouse sopra un blocco, si apre una finestra di selezione con alcune funzioni di scelta, come *Duplicate* o *Disable block* per disattivare le istruzioni di un blocco senza cancellarlo.

Per *duplicare* un blocco è sempre possibile usare le funzioni di copia di Windows con *Ctrl+C* e *Ctrl+V* per il *copia-incolla*.

## Codice Arduino

```
void loop() {  
  // primo blocco  
  tone(13,261,250); delay(250);  
  tone(13,293,250); delay(250);  
  Otto.swing(1, 1000, 25);  
  
  // secondo blocco  
  tone(13,392,250); delay(250);  
  Otto.updown(1, 1000, 25);  
  
  // terzo blocco  
  for (int count=0; count<3 ; count++) {  
    tone(13,440,250);delay(250);  
    Otto.jitter(1, 1000, 25);  
  }  
}
```

## Esercizi 1.4

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

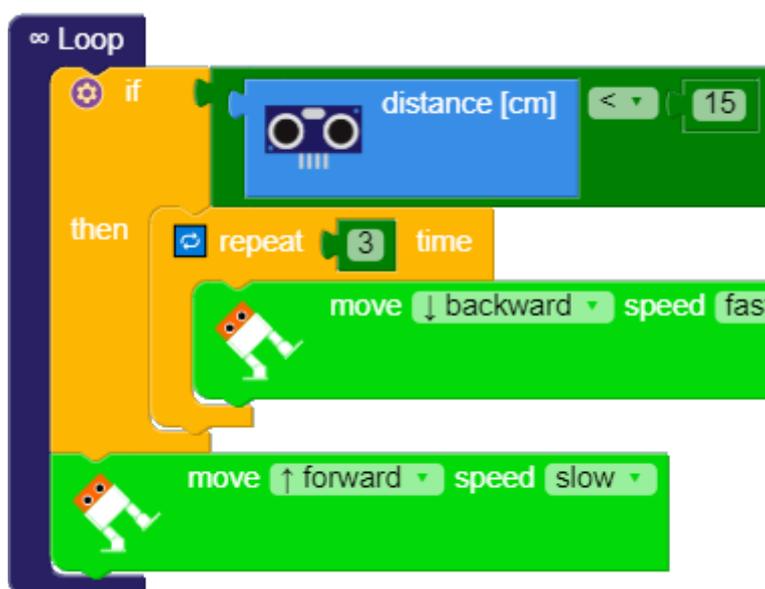


# Otto Cardy **schede**

Costruisci un robot di cartone - Claudio Gasparini - [www.cad-tutor.com/otto\\_cardy\\_8](http://www.cad-tutor.com/otto_cardy_8)

## S 1.5 Sensore di distanza

OttoBlockly fornisce anche il blocco **distance** per controllare la distanza, in centimetri, tramite il  **sensore di prossimità**  che rileva, mediante ultrasuoni, lo spazio fra il sensore e un oggetto.



Per controllare la **distanza**, si usa il blocco **IF** che verifica se la distanza è inferiore ad un valore dato, nella figura 15 cm. Se la condizione è vera, allora esegue le istruzioni interne altrimenti salta agli blocchi seguenti. In questo caso inverte la camminata andando indietro per 3 secondi (*time*).

Se non si verifica, cioè se non ci sono ostacoli, Otto prosegue la sua camminata.

A fianco viene fornito il codice dello sketch con due note diverse che avvisano quando rileva l'ostacolo e quando poi termina la marcia all'indietro.

### Codice Arduino

```
void setup() {  
  {  
    {for (int count=0 ; count<3  
    ; count++)  
      {Otto.walk(1,750,-1); //  
    BACKWARD }  
    }  
    Otto.walk(1,2000,1); //  
  FORWARD  
  }  
}  
  
void loop() {  
  if (Otto.getDistance() < 15)  
    {tone(13,261,1000);  
    delay(1000);  
    for (int count=0 ;  
    count<3 ; count++)  
      {Otto.walk(1,750,-1);//  
    BACKWARD  
    }  
    tone(13,349,1000);  
    delay(1000);}  
  Otto.walk(1,2000,1); //  
  FORWARD  
}
```

### Esercizi 1.5

1. Inserire un avviso sonoro (*nota*) quando Otto rileva un ostacolo.
2. Inserire un altro avviso (*nota diversa*) quando termina il blocco **Repeat** (*loop*)



# Otto Cardy **schede**

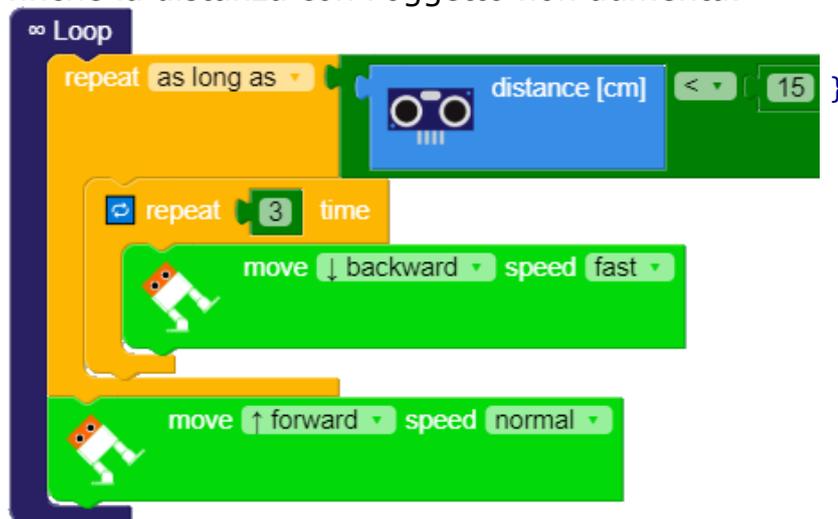
Costruisci un robot di cartone - Claudio Gasparini

## S 1.6 Condizione di distanza

Per **ripetere** una serie di blocchi a seconda di una condizione, come ad esempio fino a quando la distanza rimane minore di 15 cm, si può usare il blocco **repeat as long as** al posto del blocco precedente **IF**. Corrisponde alla comune funzione di programmazione **WHILE**.

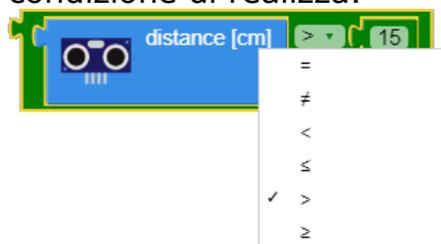


Nel nostro caso al posto del blocco **IF** possiamo utilizzare il blocco **REPEAT** per far arretrare Otto finché la distanza con l'oggetto non aumenta.



La struttura è simile alla precedente dove è stato modificato solo il blocco **REPEAT**. Nel codice a fianco notare che l'istruzione **IF** è stata sostituita con **WHILE**.

E' possibile modificare anche le **condizioni di distanza** per modificare il movimento quando la condizione di realizza.



### Codice Arduino

```
while (Otto.getDistance() < 15) {
  Otto.walk(1,1000,1); // FORWARD
}
```

```
void loop() {
  while (Otto.getDistance() < 15)
  {
    for (int count=0 ; count<3 ;
    count++) {
      Otto.walk(1,750,-1); // BACKWARD
    }
    Otto.walk(1,1000,1); // FORWARD
  }
}
```

### Esercizi 1.6

1. Modificare la distanza e cambiare la direzione di Otto
2. Se modifichiamo la condizione di distanza con **>15** cosa cambia?

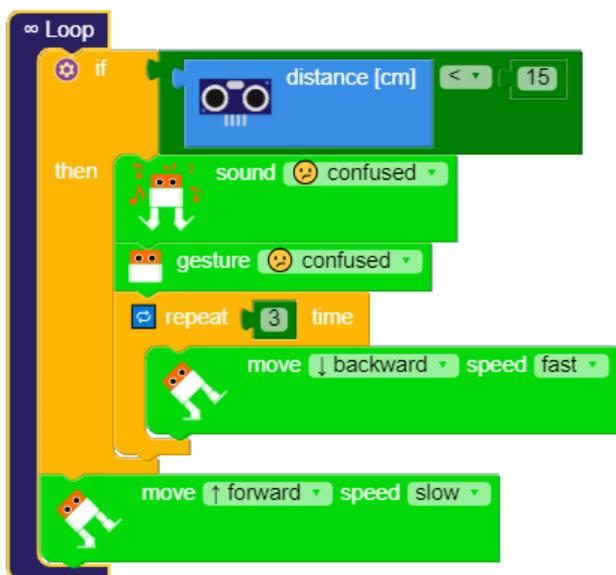


# Otto Cardy **schede**

Costruisci un robot di cartone - Claudio Gasparini - [www.cad-tutor.com/otto\\_cardy\\_10](http://www.cad-tutor.com/otto_cardy_10)

## S 1.7 Blocchi sound e gesture

Per enfatizzare l'incontro con un ostacolo si possono aggiungere dei suoni più articolati come ad esempio un *sound* fra quelli disponibili e un *gesture* di sorpresa per l'inconveniente dell'ostacolo.



Se vogliamo inserire due condizioni di distanza, ad esempio a 20 cm esegue un'azione mentre a 10 cm un'altra, allora dobbiamo inserire due blocchi IF come in figura:



### Codice Arduino

```
void loop()
{if (Otto.getDistance() < 15)
  {Otto.sing(S_confused);
  Otto.playGesture(OttoConfused);
  for (int count=0 ; count<3 ;
  count++)
    { Otto.walk(1,750,-1); //
    BACKWARD}}
  Otto.walk(1,2000,1); //
  FORWARD
}
```

```
void loop()
{if (Otto.getDistance() == 20)
  {for (int count=0 ; count<3 ;
  count++)
    {Otto.bend(1,750,1); }}
  if (Otto.getDistance() < 10)
    {for (int count=0 ; count<3
    ; count++)
      {Otto.walk(1,750,-1); //
      BACKWARD }}
    Otto.walk(1,2000,1); //
    FORWARD
}
```

### Esercizi 1.7

1. Perché nella seconda figura è stata inserita la condizione **=20** e non **<20**?
2. Proponi un'altra soluzione per avere **due** condizioni di distanza: a **10** e **20**



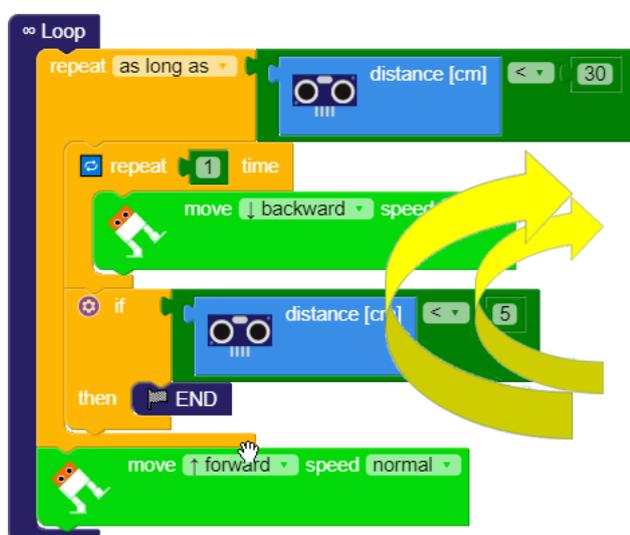
# Otto Cardy **schede**

Costruisci un robot di cartone - Claudio Gasparini - [www.cad-tutor.com/otto\\_cardy\\_11](http://www.cad-tutor.com/otto_cardy_11)

## S 1.8 Condizioni di doppio ciclo

E' possibile utilizzare due o più cicli uno interno all'altro (*cicli annidati*) per ripetere una serie di operazioni integrate.

Nel nostro caso abbiamo inserito un blocco interno di **END**: se la distanza è inferiore a 5 cm. il robot Otto si ferma.



L'esempio non è molto elegante dal punto di vista della programmazione ma è un esempio di come bloccare l'esecuzione del programma.

Le condizioni inserite sono le seguenti:

- **SE** la distanza è < (*minore*) di 30 cm. allora arretra;
- **SE** la distanza è < (*minore*) di 5 cm. allora si ferma (STOP).

Notare che la prima condizione è quella della distanza maggiore che viene verificata per prima e dopo, solo verifica se è vera anche la seconda condizione, distanza <5.

## Codice Arduino

```
void loop()
{ while
  (Otto.getDistance() < 30)
    {for (int count=0 ;
count<3 ; count++)
      {Otto.walk(1,750,-1); //
BACKWARD }
    if (Otto.getDistance() < 5)
      {while(true);}}
  Otto.walk(1,1000,1); //
FORWARD
}
```

## Esercizi 1.8

1. E' possibile inserire un altro blocco **repeat as long as**, come nella prima condizione, al posto del blocco **IF**?
2. Trova un'altra soluzione per le due condizioni annidate.



# Otto Cardy **schede**

Costruisci un robot di cartone - Claudio Gasparini - [www.cad-tutor.com/otto\\_cardy\\_12](http://www.cad-tutor.com/otto_cardy_12)

## S 2.0 PROGETTI FINALI

Il progetto finale riguarda la realizzazione di **alcuni sviluppi possibili** realizzati con **Otto Blockly** nel creare movimenti, suoni o gesti del robottino.

Vediamo alcune idee di progetto:

1. Aggiungere al robot la capacità di **rilevare** le distanze attraverso una serie di suoni diversi a seconda della distanza rilevata;
2. Associare dei nuovi **movimenti** e **suoni** ideati dagli studenti;
3. Creare delle **variabili** che controllano suoni e movimenti;
4. Aggiungere un **led** che si illumina quando il robot si muove in avanti;
5. Creare degli **abbigliamento** al robot con tessuti o carta secondo delle storie da inventare;
6. Lavorando a gruppi, sceneggiare delle **storie**, fantastiche o di soggetto storico, dove i personaggi sono i robot;
7. Aggiungere le **braccia** al robot con altri due servomotori;
8. Aggiungere un **display** 8x8 per visualizzare messaggi e testi;
9. Aggiungere un controllo **Bluetooth** per poter controllare i movimenti con il cellulare;
10. Sviluppare applicazioni che fanno svolgere al robot dei **movimenti complessi** interagendo con i vari sensori.

## Sviluppi del progetto

**Cardy** permette ai ragazzi di aver la conoscenza degli aspetti principali della costruzione di un **oggetto** funzionante con arduino.

Le possibilità di approfondimento sono molteplici: forniamo qui alcuni link di siti dove trovare informazioni tecniche, didattiche e idee per sviluppi di applicazioni varie.

Siti del progetto **OttoDIY**:

<https://www.ottodiy.com/>

- <https://ottoschool.com/scratch/>
- <https://wikifactory.com/+OttoDIY/otto-diy>

Sul sito si possono trovare i riferimenti per acquistare gli altri componenti da aggiungere al robot base.



# Otto Cardy **schede**

Costruisci un robot di cartone - Claudio Gasparini - [www.cad-tutor.com/otto\\_cardy\\_13](http://www.cad-tutor.com/otto_cardy_13)

## Scheda n. 2 - Progettazione del percorso

Titolo dell'attività	<b>Otto Cardy - Costruisci il tuo robot</b>
Ordine di scuola e classe	<b>Tutte</b>
Contesti disciplinari	Discipline STEAM e italiano
Finalità	<ul style="list-style-type: none"><li>• sviluppo del pensiero computazionale</li><li>• programmare con la programmazione visuale a blocchi;</li><li>• capacità di tinkering e creatività</li><li>• saper lavorare in gruppo</li><li>• stimolare la motivazione verso le discipline STEM e ridurre il gap di genere</li><li>• utilizzare linguaggi di programmazione per approcciarsi alla robotica</li></ul>
Tempi	Tempo necessario per sviluppare le schede didattiche proposte e la costruzione del robottino ( <i>Vedi schede didattiche</i> )
Prodotto dello studente o del gruppo	Robot Otto Cardy
Valutazione	Valutazione formativa e autovalutazione. ( <i>Vedi schede didattiche</i> )
Descrizione dell'attività	<p>Le fasi di realizzazione riguardano <b>tre attività</b> distinte che contribuiscono a creare il robot.</p> <p>La <b>prima</b> è la costruzione dei componenti di cartone a partire dall'interpretazione dei disegni tecnici forniti secondo queste fasi:</p> <ul style="list-style-type: none"><li>- A1: taglio dal cartoncino dei singoli componenti</li><li>- A2: piegatura del cartoncino</li><li>- A3: scelta delle migliori tecniche di incollaggio</li></ul>



## Otto Cardy **schede**

Costruisci un robot di cartone - Claudio Gasparini - [www.cad-tutor.com/otto\\_cardy\\_14](http://www.cad-tutor.com/otto_cardy_14)

	<p>La <b>seconda</b> attività riguarda l'assemblaggio dei componenti elettronici e la verifica del loro funzionamento:</p> <ul style="list-style-type: none"><li>- B1: studio dello schema di montaggio dei componenti</li><li>- B2: individuazione e verifica della funzione dei singoli componenti</li><li>- B3: connessione di tutti i componenti e verifica del loro corretto funzionamento</li></ul> <p>La <b>terza</b> attività riguarda la programmazione a blocchi del robot e la sperimentazione dei vari movimenti:</p> <ul style="list-style-type: none"><li>- C1: installazione del software OttoBlockly</li><li>- C2: taratura dei movimenti di base</li><li>- C3: caricamento di alcuni sketch di movimento e di danza</li><li>- C4: aggiunta di suoni e di musiche</li></ul>
Prerequisiti	Per sviluppare questo progetto non sono richieste specifiche conoscenze di informatica o di elettronica.



## Scheda n. 3 - Griglia di Valutazione del lavoro di gruppo

Competenze lavorative collaborative: **Lavoro di gruppo**

Nome insegnante: .....

Nome studente: .....

CATEGORIA	4	3	2	1
Contributi	Fornisce regolarmente idee utili durante la partecipazione al gruppo e alla discussione in classe. Un leader definito che contribuisce con molto impegno.	Di solito fornisce idee utili quando si partecipa al gruppo e alla discussione in classe. Un membro del gruppo forte che si sforza!	A volte fornisce idee utili durante la partecipazione al gruppo e alla discussione in classe. Un membro del gruppo soddisfacente che fa ciò che è richiesto.	Raramente fornisce idee utili quando si partecipa al gruppo e alla discussione in classe. Può rifiutarsi di partecipare.
Qualità del lavoro	Fornisce un lavoro di alta qualità.	Fornisce un lavoro di qualità.	Fornisce lavori che occasionalmente devono essere controllati / rifatti da altri membri del gruppo per garantire la qualità.	Fornisce lavori che di solito devono essere controllati / rifatti da altri per garantire la qualità.



## Otto Cardy **schede**

Costruisci un robot di cartone - Claudio Gasparini - [www.cad-tutor.com/otto\\_cardy\\_16](http://www.cad-tutor.com/otto_cardy_16)

Gestione del tempo	Utilizza regolarmente il tempo durante tutto il progetto per garantire che le cose vengano fatte in tempo. Il gruppo non deve modificare le scadenze o le responsabilità lavorative a causa della procrastinazione e di questa persona.	Di solito utilizza bene il tempo durante il progetto, ma potrebbe aver procrastinato su una cosa. Il gruppo non deve modificare le scadenze o le responsabilità lavorative a causa della procrastinazione di questa persona.	Tende a procrastinare, ma fa sempre le cose entro le scadenze. Il gruppo non deve modificare le scadenze o le responsabilità lavorative a causa della procrastinazione di questa persona.	Raramente le cose vengono fatte entro le scadenze. E il gruppo deve adeguare le scadenze o le responsabilità lavorative a causa della gestione inadeguata del tempo di questa persona.
Risoluzione dei problemi	Cerca attivamente e suggerisce soluzioni ai problemi.	Affina le soluzioni suggerite da altri.	Non suggerisce o perfeziona soluzioni, ma è disposto a provare soluzioni suggerite da altri.	Non cerca di risolvere problemi o aiuta gli altri a risolverli. Lascia che gli altri facciano il lavoro.
Concentrati sul compito	Rimane costantemente concentrato sull'attività e su ciò che deve essere fatto. Molto autodiretto.	Si concentra sull'attività e su ciò che deve essere fatto la maggior parte del tempo. Altri membri del gruppo possono contare su questa persona.	Si concentra sull'attività e su ciò che deve essere fatto alcune volte. Gli altri membri del gruppo a volte devono lamentarsi, stimolare e ricordare per mantenere questa persona in attività.	Raramente si concentra sull'attività e su ciò che deve essere fatto. Lascia che gli altri facciano il lavoro.



## Otto Cardy **schede**

Costruisci un robot di cartone - Claudio Gasparini - [www.cad-tutor.com/otto\\_cardy\\_17](http://www.cad-tutor.com/otto_cardy_17)

Lavorare con gli altri	Quasi sempre ascolta, condivide e sostiene gli sforzi degli altri. Cerca di far lavorare bene le persone insieme.	Di solito ascolta, condivide e supporta gli sforzi degli altri. Non causa "entusiasmo" nel gruppo.	Spesso ascolta, condivide e supporta gli sforzi degli altri, ma a volte non è un buon membro del team.	Raramente ascolta, condivide e sostiene gli sforzi degli altri. Spesso non è un buon giocatore di squadra.
------------------------	---	--	--	--



# Otto Cardy **schede**

Costruisci un robot di cartone - Claudio Gasparini - [www.cad-tutor.com/otto\\_cardy\\_18](http://www.cad-tutor.com/otto_cardy_18)

## Scheda n. 4 - Autovalutazione individuale

Rubrica di autovalutazione alunnoaborative: **Lavoro di gruppo**

Nome insegnante: .....

Nome studente: .....

<b>indicatori</b>	<b>descrittori</b>		
Mi sono impegnato nel mio lavoro	cercando di fare del mio meglio	lavorando in modo sufficiente	potevo fare di più
Ho partecipato alla discussione	dando il mio contributo	ascoltando	con scarso interesse
Ho accolto le idee degli altri	integrando le mie proposte con le loro	ascoltando, ma poi seguendo solo le mie idee	No
Ho collaborato con i compagni	mostrandomi molto disponibile	aiutando solo quelli che mi chiedevano aiuto	non sempre
Di fronte alla difficoltà	Ho risolto autonomamente	Ho chiesto aiuto	Mi sono scoraggiato
Sono riuscito a rispettare i tempi	si	in parte	No
L'attività svolta è stata interessante e istruttiva	si, ho appreso cose nuove	in parte	No