

# EduCode

G. Giorgino, A. Maretti

Corso di Applicazioni e Servizi web

**Sommario** Sviluppo di una web application che supporti la crescita di coding skill nei bambini attorno ai 7-9 anni, attraverso risoluzioni di semplici problemi utilizzando i principali concetti dei linguaggi di programmazione.

## Indice

Sommario .....	1
Introduzione .....	1
1 Design .....	2
1.1 Storyboard, interfaccia ed interazioni .....	2
1.2 User Experience .....	7
1.3 Usabilità .....	7
1.4 Layout Responsive .....	7
1.5 Tecnologie utilizzate .....	9
1.6 API di HTML5 e CSS utilizzate .....	9
1.7 Librerie utilizzate .....	10
2 Test .....	12

## Introduzione

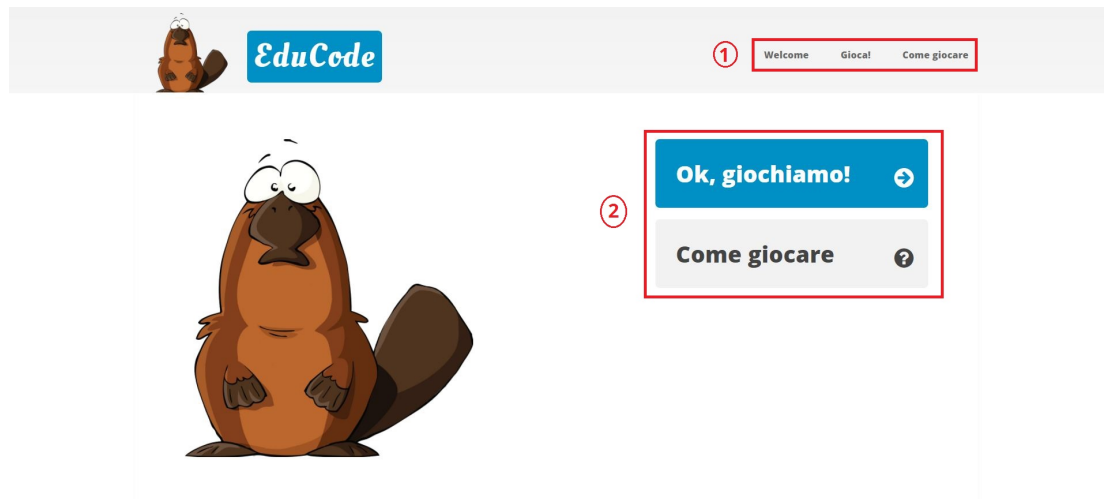
L'obiettivo del progetto consiste nel rendere disponibile anche ai più piccoli, una piattaforma in grado di aiutarli nell'introduzione alla programmazione, tramite la risoluzione di problematiche sfruttando concetti riconducibili alle principali nozioni di coding. Per rendere il tutto più semplice e a portata dei bambini, sono stati apportati tutta una serie di accorgimenti, descritti nei capitoli successivi, tali da rendere questa attività divertente e giocosa.

## 1 Design

Nella fase di design della piattaforma sono stati presi in esame tutti gli elementi necessari per rendere ogni componente il più semplice ed intuibile possibile, così da poter far eseguire gran parte dell'applicazione in modo autonomo ai bambini. In questo modo il supporto e l'impegno necessario da parte di un adulto nell'esecuzione e nella spiegazione dei vari esercizi sarà limitato, utile in particolar modo nel caso di gruppi di bambini impegnati in tale attività.

### 1.1 Storyboard, interfaccia ed interazioni

La prima schermata, in versione desktop, visualizzata all'utente sarà quella in Figura 1.



**Figura 1.** Home - Prima sezione

Scorrendo verso il basso sarà invece visibile la seconda parte della schermata home (Figura 2), concludendo con un semplice footnote contenente solamente i nomi e cognomi e il corso di riferimento.

Gli elementi interagibili sono stati evidenziati con riquadri rossi e contrassegnati con un numero. Ogni interazione sarà quindi brevemente analizzata. Iniziamo dalla schermata visualizzata in Figura 1:

- Il riquadro 1 corrisponde alla navbar, ogni componente quindi è collegato ad una pagina del sito, più nello specifico "Welcome" rimanderà alla home, "Gioca!", essendo una drop list, ci permetterà di selezionare una delle scelte previste, più specificatamente il livello che si desidera affrontare, ed infine "Come giocare" permetterà di visionare le istruzioni complete (Figura 5).

- Il riquadro 2 evidenzia due pulsanti: il primo, colorato di uno sfondo azzurro, ci reindirizzerà direttamente al primo livello del nostro gioco istruttivo, mentre il secondo, con la scritta "Come giocare", porterà alla stessa pagina del medesimo elemento sulla navbar, cioè quella dove vengono riportate le informazioni base per giocare (Figura 5).



**Figura 2.** Home - Prima sezione

Facciamo quindi ora riferimento alla schermata in Figura 2:

- Le tre figure sono elementi cliccabili, ognuno dei quali condurrà ad un capitolo differente della pagina del sito contenente le principali informazioni. La prima immagine fa riferimento ad una serie di informazioni riguardante il concetto di programmazione e le abilità che lo studio di questa disciplina permette di affinare (capitolo di riferimento dopo il click: Figura 3).

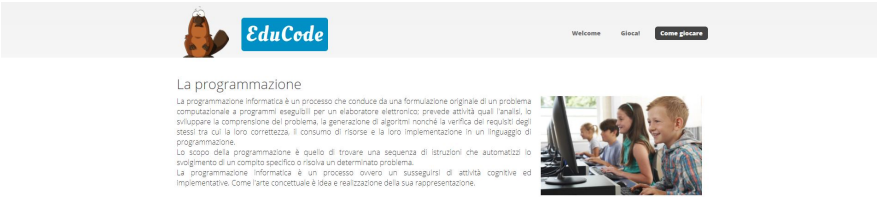


Figura 3. Info - Programmazione

La seconda immagine esplicita le principali motivazioni che possono spingere allo studio della programmazione (capitolo di riferimento dopo il click: Figura 4).

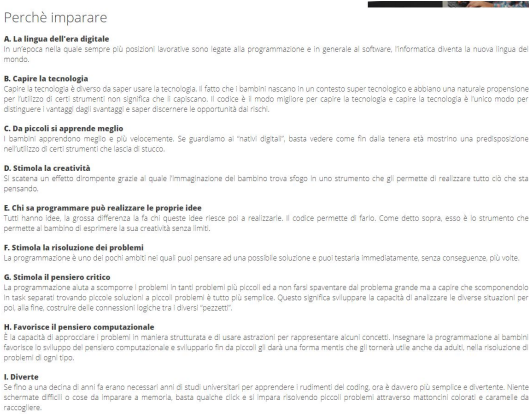


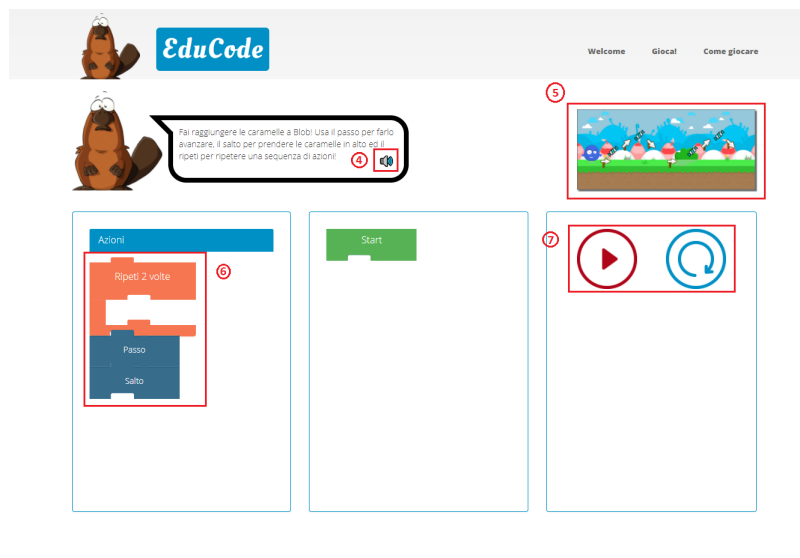
Figura 4. Info - Motivazioni

Infine avremo nuovamente un riferimento alle principali istruzioni del gioco (capitolo di riferimento dopo il click: Figura 5)



**Figura 5.** Info - Come giocare

Passiamo ora alla pagina mostrata in Figura 6:

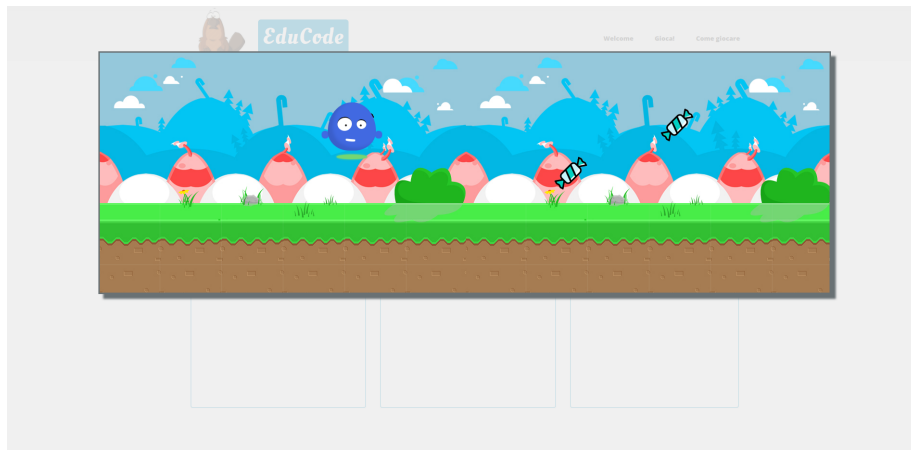


**Figura 6.** Schermata di gioco

- Il riquadro 4 rappresenta l'icona per l'avvio del text to speech delle istruzioni per l'esercizio corrente. Tale pulsante è stato implementato semplicemente utilizzando una funzione della libreria gratuita ResponsiveVoice.js<sup>1</sup>.
- Il riquadro 5 rappresenta un'immagine del livello corrispondente all'esercizio corrente, inoltre è possibile passarci sopra con il cursore (hover) per ottenere un leggero zoom affinché sia di più facile visione.

<sup>1</sup> <https://responsivevoice.org/>

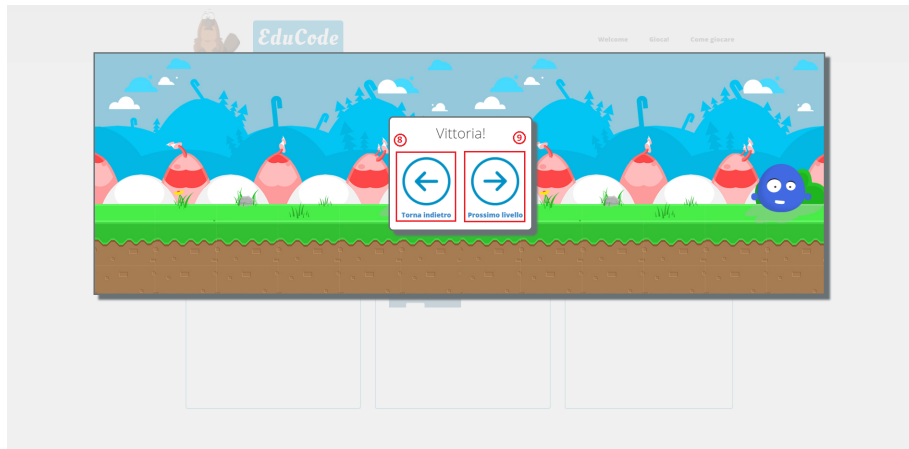
- Il riquadro 6 corrisponde all'area di gioco: essa è divisa in tre sezioni verticali, nella prima, partendo da sinistra, sono presenti i mattoncini essenziali per il completamento dell'esercizio ed ognuno dei quali avrà la caratteristica di essere drag and drop, in particolare dovrà essere posto nella seconda sezione dell'area di gameplay, sotto all'elemento "Start" (non spostabile o selezionabile). Gli elementi aventi la dicitura "Ripeti" potranno contenere altri mattoncini di qualsiasi tipo al loro interno, e adegueranno la loro dimensione automaticamente in base alla quantità di elementi contenuti.
- Il riquadro 7 corrisponde alla terza sezione della schermata di gioco e contiene due pulsanti, il primo avente il simbolo di play, permetterà di controllare l'esercizio e di avviare l'animazione corrispondente alla soluzione proposta. Esso sarà di colore rosso e non cliccabile finché tutti gli elementi saranno stati utilizzati e posti nella seconda sezione. Solamente in quel momento assumerà il colore verde e sarà utilizzabile. Il secondo pulsante rappresenta invece il riavvio dell'esercizio.



**Figura 7.** Animazione dell'esercizio

La schermata dell'animazione (Figura 7) visualizza la soluzione proposta dall'utente. Tale finestra rappresenta un breve percorso che il personaggio protagonista dell'animazione dovrà compiere. In caso di soluzione corretta, riuscirà ad accedere a tutte le caramelle poste lungo il percorso. Durante l'animazione non sarà possibile interagire con essa e sarà necessario pazientare il suo termine.

La Figura 8 fa riferimento al dialog che appare all'utente al termine dell'animazione descritta poco sopra. Sia in caso di vittoria che di sconfitta, saranno possibili due scelte: la prima (elemento 8), rappresentata da un tasto che punta verso sinistra, reindirizzerà alla pagina di gioco precedente, mentre l'altro pulsante (elemento 9) permetterà di accedere direttamente all'esercizio successivo.



**Figura 8.** Dialog al termine dell'animazione

## 1.2 User Experience

L'UX (User Experience) è in relazione agli aspetti e alle caratteristiche dell'applicazione utilizzate dall'utente. Da questo termine deriva la "user friendliness", cioè rendere la propria applicazione il più semplice ed intuibile possibile. Sicuramente una prerogativa quando il target d'utenza è molto giovane come nel nostro caso. Per questa ragione abbiamo reso il più giocoso, semplice e colorato ogni elemento importante dell'applicazione. L'esperienza d'uso è soggettiva e si concentra sull'atto dell'utilizzo, per questo motivo non essendo stato possibile far utilizzare tale applicazione a bambini dell'età target, non possiamo avere un riscontro diretto ma ugualmente abbiamo cercato di rendere il tutto adatto ai più piccoli.

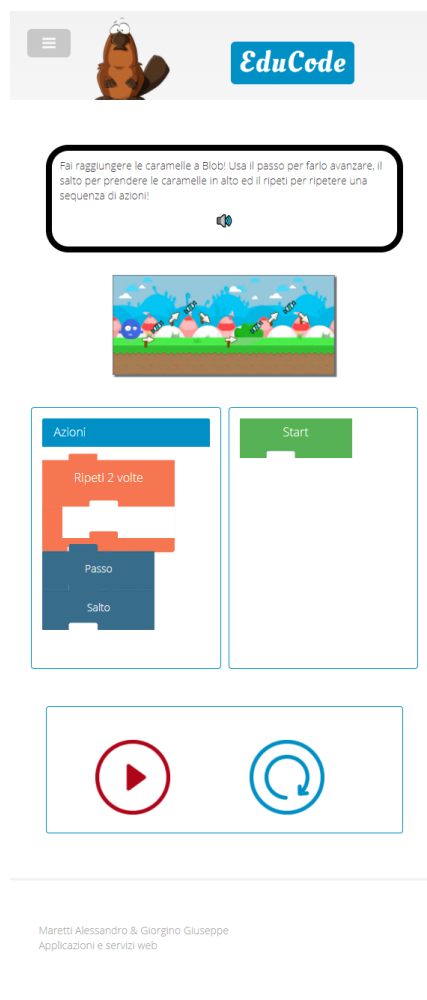
## 1.3 Usabilità

La definizione di usabilità, cioè: "Il grado in cui un prodotto può essere usato da particolari utenti per raggiungere certi obiettivi con efficacia, efficienza, soddisfazione in uno specifico contesto d'uso", esprime esattamente la motivazione di ogni scelta importante di design. Alcuni esempi sono stati la preferenza di una modalità di gioco ben nota per tutti, cioè l'abbinamento di pezzi di puzzle, rendendolo il gameplay immediato, oppure l'utilizzo di pochi elementi, chiari e semplici, infatti gli unici elementi colorati sono quelli fondamentali per il gioco e per la sua spiegazione. Ogni testo è stato scritto in modo semplice ed anche le azioni sono molto intuibili, come quelle di compiere passi e salti.

## 1.4 Layout Responsive

Per quanto concerne lo sviluppo del layout responsive abbiamo scelto di definire, tramite le media query, quattro step di risoluzione massima in larghezza (1680px,

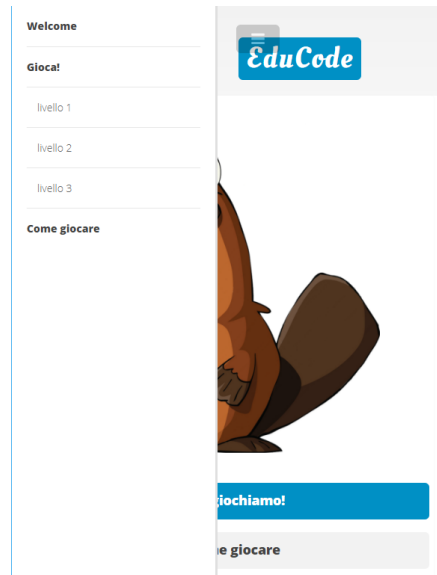
1280px, 980px e 736px) e due in altezza (618px e 525px) per poter gestire la grafica nei device mobili più diffusi. La gestione della larghezza ci è servita per poter fare incollare i contenuti nelle pagine principalmente testuali ma anche per ristrutturare la schermata di gioco che, seppur sempre scomoda per un cellulare in posizione portrait, può essere utilizzata. Abbiamo inoltre scelto di forzare l'orientamento landscape nel momento in cui si avvia l'animazione per agevolare la vista sui dispositivi mobili, sarà poi ripristinato l'orientamento precedente al termine dell'animazione. Le media query riguardanti l'altezza del display vengono utilizzate per centrare il riquadro in cui avvengono le animazioni nei vari dispositivi.



**Figura 9.** Schermata di gioco - Responsive



Infine è stato gestito il menu inserendo in alto a sinistra (per evitare di toccarlo mentre si effettua lo scroll della pagina) un pulsante che fa comparire lateralmente l'elenco delle voci disponibili come si può vedere in Figura 10



**Figura 10.** Menù responsive

### 1.5 Tecnologie utilizzate

Le principali tecnologie adoperate per questo progetto sono state:

- HTML5
- CSS3
- Javascript

### 1.6 API di HTML5 e CSS utilizzate

In particolare per HTML5 è stato fondamentale servirsi dell'API Drag and Drop. Più specificatamente abbiamo utilizzato la libreria [1] che si basa sulla suddetta API ma che la estende con caratteristiche, come quella di poter facilmente innestare più elementi con le proprietà drag e drop, essenziali per la riuscita del progetto.

Per quanto riguarda CSS3 sono state largamente utilizzate animazioni e key-frame. Le animazioni CSS3 permettono animazioni della maggior parte degli elementi HTML, senza utilizzare JavaScript o Flash. Quando vengono specificati

stili CSS all'interno delle regole keyframes, l'animazione gradualmente cambierà dallo stile corrente a quello nuovo in un certo intervallo di tempo. L'animazione per poter funzionare, dovrà essere abbinato all'elemento da animare.

Per la nostra applicazione viene modificato un keyframe in base alla tipologia di animazione necessaria, ad esempio una transformation sull'asse X di un certo numero di px (anch'essi calcolati dinamicamente) per il passo mentre per la parabola, corrispondente ad un salto del personaggio, avviene una modifica su entrambi gli assi equivalente alla funzione della parabola  $x = -ay^2$ . Dopodichè avviene l'assegnamento dell'animazione, aggiungendola a quelle eventualmente già presenti, all'elemento oggetto dell'animazione. Inoltre sarà impostato il delay che tale animazione deve avere, per avviarsi col giusto tempismo.

```
function addJumpRule(frameJump, start, stop, height){

    var tick = (stop - start) / 10;
    var j = 1;
    var mult = 5;
    var rapporto = 5;
    var inc = 0;
    for(var i = 0; i <= 10; i++){
        if(i <= 5){
            inc = (rapporto*(mult*mult));
            mult--;
        }else{
            inc = (rapporto*(j*j));
            j++;
        }
        frameJump.appendRule(""+i*10+"% {transform: translate("+
            (start+(tick*i))+ "px, "+ (-height+inc) + "px);}");
    }
}
```

### 1.7 Librerie utilizzate

La funzionalità di drag e drop dei vari elementi di gioco è stata sfruttata attraverso l'impiego di una libreria [1], la quale si basa sulle API drag e drop di HTML5 ma estendendole attraverso angular. In particolare permette di ottenere elementi con proprietà drag e drop che siano innestabili tra loro. Un esempio esemplificativo è il mattoncino di gameplay rappresentate un ciclo, infatti all'interno di esso è possibile inserire altri elementi. In particolare ne facciamo utilizzo nel file level.html, al cui interno sono definiti tutti i template che definisco il rendering degli elementi utilizzati, come il container ("mattoncino del loop"), l'elemento singolo ("mattoncino del passo"). Inoltre sono espressi i markup per le liste di elementi all'interno dalla zona di drop. Si trova in un template separato poichè verrà utilizzato ricorsivamente. La lista direttiva dnd-list permette di rilasciare gli elementi all'interno. Se un elemento viene spostato fuori, verrà eliminato

dalla lista corrente, per essere inserito in quella nuova; ciò è possibile attraverso l'attributo `dnd-moved`.

Verrà inoltre definito un modulo angular, in cui viene associato l'url del template e il controller di riferimento. Tale modulo sarà richiamato nel codice html all'interno dello spazio assegnato per il gameplay.

Il controller sarà definito in modo differente in base al livello di gioco scelto. Esso infatti contiene la struttura iniziale della lista di elementi. Sfruttando l'attributo `"dropzones"` viene generato il Json corrispondente ogni qual volta avviene una modifica alle liste di elementi.

Esempio di controller angular per il livello 3:

```
angular.module("demo").controller("NestedListsDemoController",
function($scope) {

$scope.models = {
    selected: null,
    templates: [
        {type: "item", id: 2},
        {type: "container", id: 1, columns: [[], []]}
    ],
    dropzones: {
        "Azioni": [
            {
                "type": "container",
                "id": 1,
                "name": "Ripeti 2 volte",
                "nRepeats": 2,
                "columns": [

                    ]
            ]
        ],
        {
            "type": "item",
            "name": "Passo",
            "id": "5"
        },
        {
            "type": "item",
            "name": "Salto",
            "id": "4"
        },
    ],
}
```

```

        "Start": [
            ]
        }
    };

    $scope.$watch('models.dropzones', function(model) {
        $scope.modelAsJson = angular.toJson(model, true);
        json = $scope.modelAsJson;
        checkPlay();
    }, true);
});

```

Affinchè il drag and drop fosse utilizzato anche da mobile, caratteristica non disponibile nativamente dall'API di HTML5, è stato utilizzato un polyfill<sup>2</sup> [2]. Per installarlo è sufficiente aggiungere lo script JavaScript, presente nella pagina GitHub dello sviluppatore, al proprio progetto.

## 2 Test

I test che sono stati eseguiti comprendono l'esecuzione della web application con diversi device fisici, due tablet Android da 7 e 10 pollici, uno smartphone da 5 pollici e i nostri pc personali. Grazie al layout responsive definito, l'esperienza d'uso è risultata piacevole in tutti i casi. Ovviamente sono stati effettuati altri test simulando altri device, in particolare quelli Apple, sfruttando alcuni plugin per Chrome o Firefox. Altri test sono stati eseguiti per controllare la compatibilità con diversi browser, in particolare Chrome, Firefox, ed Edge. Su tutti e tre non sono emerse problematiche, nè differenze nelle funzionalità.

---

<sup>2</sup> Nella programmazione web, un polyfill (o polyfiller) è del codice scaricabile che fornisce dei servizi che non fanno parte di un browser web. Esso implementa tecnologie che uno sviluppatore si aspetta il browser metta a disposizione nativamente, fornendo un ventaglio più uniforme di API.

## Bibliografia

- [1] Angular drag and drop. <https://github.com/marceljuenemann/angular-drag-and-drop-lists/tree/master/demo>.
- [2] Dragdroptouch. <https://github.com/Bernardo-Castilho/dragdroptouch>.