

## REPORT SERIES

A. Repetto, C.E. Catalano

A decorative graphic on the left side of the page consisting of a grid of colorful puzzle pieces in shades of red, green, yellow, and blue, arranged in a stepped pattern that tapers to the right.

**Studio e sviluppo di componenti  
semantiche riusabili per la  
progettazione di serious game.  
Applicazioni all'area edutainment**

# IMATI REPORT Series

Nr. 16-12 – July 2016

## **Managing Editor**

Paola Pietra

## **Editorial Office**

Istituto di Matematica Applicata e Tecnologie Informatiche “E. Magenes”

Consiglio Nazionale delle Ricerche

Via Ferrata, 5/a

27100 PAVIA (Italy)

Email: [reports@imati.cnr.it](mailto:reports@imati.cnr.it)

<http://www.imati.cnr.it>

Follow this and additional works at: <http://www.imati.cnr.it/reports>

---

Copyright © CNR-IMATI, 2016.

IMATI-CNR publishes this report under the Creative Commons Attributions 4.0 license.

Studio e sviluppo di componenti semantiche riusabili per la progettazione di serious game.  
Applicazioni all'area edutainment

*A. Repetto, C.E. Catalano*

IMATI-CNR, Genova, Italy

*Corresponding author:* [andrea.repetto@ge.imati.cnr.it](mailto:andrea.repetto@ge.imati.cnr.it)

---

Copyright © CNR-IMATI, July 2016.



**Abstract.**

I serious game sono videogiochi che hanno una finalità primaria diversa da quella del puro divertimento, ad esempio quella di trasmettere dei contenuti educativi, oppure di sensibilizzare il pubblico rispetto a determinate tematiche. Con particolare riferimento all'area edutainment, questi giochi contengono al loro interno una grande quantità di conoscenze relative al dominio applicativo. Perché il gioco sia realistico dal punto di vista delle interazioni, tali conoscenze vanno tradotte anche nei comportamenti veri e propri dei personaggi del gioco. In questo lavoro è stata sviluppata una metodologia e opportuni strumenti software che consentono di tradurre tali conoscenze formalizzate usando OWL e RDF in comportamenti veri e propri all'interno del motore di gioco.

Il lavoro svolto rappresenta un primo passo nella creazione di componenti riusabili, in cui gli oggetti di gioco sono arricchiti con la conoscenza del dominio e possono essere riutilizzati in contesti differenti, con un notevole abbattimento dei costi per gli attori coinvolti nella realizzazione del gioco stesso (esperti del dominio e sviluppatori di giochi).

**Keywords:** *Semantics, Object interaction, Ontologies, Edutainment, Natural heritage*

# Studio e sviluppo di componenti semantiche riusabili per la progettazione di serious game. Applicazioni all'area edutainment.

Andrea Repetto, Chiara Eva Catalano

## Abstract

---

I serious game sono videogiochi che hanno una finalità primaria diversa da quella del puro divertimento, ad esempio quella di trasmettere dei contenuti educativi, oppure di sensibilizzare il pubblico rispetto a determinate tematiche. Con particolare riferimento all'area edutainment, questi giochi contengono al loro interno una grande quantità di conoscenze relative al dominio applicativo. Perché il gioco sia realistico dal punto di vista delle interazioni, tali conoscenze vanno tradotte anche nei comportamenti veri e propri dei personaggi del gioco. In questo lavoro è stata sviluppata una metodologia e opportuni strumenti software che consentono di tradurre tali conoscenze formalizzate usando OWL e RDF in comportamenti veri e propri all'interno del motore di gioco.

Il lavoro svolto rappresenta un primo passo nella creazione di componenti riusabili, in cui gli oggetti di gioco sono arricchiti con la conoscenza del dominio e possono essere riutilizzati in contesti differenti, con un notevole abbattimento dei costi per gli attori coinvolti nella realizzazione del gioco stesso (esperti del dominio e sviluppatori di giochi).

# 1. Introduzione

---

In questo rapporto tecnico descriviamo il lavoro svolto durante l'attività dell'assegno nell'ambito del progetto regionale dal titolo "Serious Game per la fruizione di contenuti e l'apprendimento delle tematiche educative, culturali e ambientali in strutture museali di tipo edutainment". Lo scopo del progetto è stato di studiare e sviluppare tecniche innovative per l'analisi e gestione di contenuto digitale mirato alla progettazione di giochi educativi (serious game): l'apprendimento è riferito a tematiche di interesse edutainment nell'ambito delle strutture gestite da Costa Edutainment spa (Acquario di Genova, Città dei Bambini e dei Ragazzi, Galata Museo del Mare, Acquario di Livorno).

Il concetto di edutainment portato avanti da Costa si basa sull'educare "attraverso le emozioni". In alternativa e come complemento alle classiche lezioni scolastiche, le tematiche proposte da Costa, quali la preservazione dell'ambiente marino e dei suoi abitanti, possono essere apprese dai ragazzi in maniera più efficace tramite esperienze che cercano di educare e divertire insieme.

I serious game possono essere dunque funzionali a questo scopo: con *serious game* [1] si intendono videogiochi che hanno come scopo primario quello di informare, educare, o generare consapevolezza verso una certa tematica, al contrario dei giochi che hanno come scopo primario il divertimento in se stesso. Nel nostro caso, la tematica proposta è quella della salvaguardia dell'ambiente marino e dei suoi abitanti.

La realizzazione di un serious game, tuttavia, pone delle difficoltà aggiuntive rispetto al gioco di intrattenimento. In primo luogo, è necessario che il gioco risponda a precisi obiettivi educativi o comunque che veicoli un messaggio al giocatore; l'avanzamento nel gioco è subordinato al raggiungimento di questo obiettivo e questo spesso influisce direttamente sui meccanismi di gioco. In secondo luogo, nello sviluppo di un serious game vi sono almeno due attori coinvolti: l'educatore/divulgatore e l'azienda che sviluppa giochi. Questi due attori hanno competenze differenti (una più pedagogica e relativa al dominio di interesse, l'altra più tecnologica), e dovrebbero collaborare strettamente per far sì che il contenuto educativo sia espresso nella maniera corretta all'interno del gioco. Ciò non accade di solito, minando l'efficacia del gioco finale. Una direzione di ricerca recente prevede lo sviluppo di componenti riusabili all'interno del gioco [2]; per semplificare la collaborazione tra sviluppatori e pedagoghi/divulgatori, in particolare si potrebbe pensare allo sviluppo di "moduli educativi" che lo sviluppatore di giochi possa riutilizzare per dare una valenza educativa al proprio gioco.

Il nostro lavoro di ricerca si inserisce proprio in questo ambito, con una metodologia per lo sviluppo di componenti riusabili (oggetti di gioco) dotati di una loro *semantica* che, secondo la definizione data in [3] è l'insieme di dati che va oltre l'aspetto grafico, e che quindi influenza il significato di un oggetto all'interno del mondo di gioco.

Nello specifico siamo partiti dall'osservazione che i mondi virtuali a cui siamo abituati sono realistici dal punto di vista grafico, ma lo sono meno dal punto di vista dell'interazione: questo è dovuto al fatto che tipicamente le uniche interazioni possibili sono quelle codificate esplicitamente dallo sviluppatore. Codificare le interazioni è un lavoro dispendioso in termini di tempo, e man mano che il mondo virtuale viene arricchito di nuovi elementi, diventa difficile mantenerli coerenti fra di loro. Questo porta a situazioni di gioco poco realistiche, che limitano l'immersione del giocatore.

Il nostro lavoro va dunque nella direzione di rendere le interazioni più naturali e spontanee [4], con l'idea che introducendo la semantica all'interno dei mondi di gioco si possono instaurare interazioni realistiche fra gli elementi di gioco senza che lo sviluppatore del gioco le abbia codificate esplicitamente una per una.

Un primo passo in questa direzione è stato quello di codificare la conoscenza relativa al mondo di gioco e ai suoi elementi in modo formale: una scelta che abbiamo trovato appropriata è data dal formalismo delle ontologie [5]. Esse codificano la conoscenza tramite assiomi logici, definiti su un sottoinsieme della logica del prim'ordine. Un vantaggio delle ontologie, essendo basate su un formalismo logico, è dato dalla possibilità di inferire conseguenze logiche a partire da un insieme di assiomi. Esistono dei software, detti reasoner semantici, adibiti proprio a questo scopo. In questo modo, le informazioni codificate implicitamente nell'ontologia (ad esempio, una relazione di tipo transitivo) possono essere rese esplicite. Tipicamente, in un videogioco le interazioni sono programmate dallo sviluppatore del gioco, utilizzando

dei frammenti di codice detti “script”. Usare le ontologie per codificare le interazioni ci pone dunque davanti alla necessità di tradurre le proprietà semantiche espresse dalle ontologie in interazioni vere e proprie all’interno del gioco. Per lo sviluppatore, codificare nel gioco tale traduzione può diventare un lavoro complesso, fino a risultare infattibile: per questo motivo, abbiamo scelto di rendere questo processo il più automatico possibile. Abbiamo sviluppato una libreria software che, utilizzata come dipendenza del motore di gioco, permetta di tradurre automaticamente le istanze contenute nell’ontologia, comprese le loro proprietà semantiche, in oggetti di gioco intelligenti, che si comportino in maniera consistente. La libreria che abbiamo sviluppato prende il nome di *SemanticLayer* (strato semantico) poiché costituisce un livello intermedio fra la conoscenza formalizzata nell’ontologia, e il motore di gioco che sfrutta tale conoscenza per produrre le interazioni corrispondenti. I dettagli sullo sviluppo della libreria sono descritti nella sezione 2.

Nell’ambito del progetto abbiamo validato il nostro approccio in un caso reale. Grazie alla collaborazione con Costa Edutainment, è stato possibile definire un concept di serious game ed arrivare infine alla realizzazione di un prototipo che potesse essere utilizzato come banco di prova del nostro lavoro di ricerca. L’attività ha dunque visto una fase di analisi dei requisiti, che è stata portata avanti parallelamente all’attività di ricerca, a cui è seguita la realizzazione del prototipo vero e proprio. Esso rappresenta anche un primo passo per l’eventuale realizzazione di un gioco completo da offrire al pubblico; l’implementazione del prototipo è descritta nella sezione 3. L’attività di sviluppo del prototipo è stata infine validata dal punto di vista del contenuto educativo da Costa Edutainment (sezione 4); la sezione **Errore. L’origine riferimento non è stata trovata.** infine descrive le conclusioni e i possibili sviluppi futuri di questo lavoro.

## 2. Formalizzazione della conoscenza per lo sviluppo di componenti riusabili nella progettazione di serious game

---

Nell’introduzione abbiamo evidenziato i limiti dell’attuale metodologia con cui si sviluppano le interazioni nei mondi virtuali. Ci siamo dunque proposti di migliorare le attuali metodologie, con possibili applicazioni sia nel campo dei serious game che dei giochi per l’intrattenimento.

Un elemento chiave che abbiamo cercato di sfruttare è la possibilità di codificare le proprietà semantiche di un oggetto virtuale, in modo che il gioco possa utilizzarle per derivare possibili interazioni, dinamiche e dipendenti dal contesto. Tali oggetti, dotati di geometria e di semantica, sarebbero riutilizzabili in contesti differenti, aprendo interessanti prospettive sia nel campo dei serious game, in cui vi è una forte domanda di soluzioni a basso costo per produrre giochi di qualità che nel campo dei giochi per l’intrattenimento.

Infatti, le tecniche normalmente utilizzate per codificare la semantica degli oggetti virtuali (ad esempio, script di comportamento) risultano limitate al contesto del singolo gioco: la portabilità degli oggetti virtuali tra giochi differenti è dunque limitata alla sola componente grafica. L’utilizzo degli standard legati alle tecnologie della conoscenza, come OWL e RDF, potrebbero rendere anche la parte semantica più facile da riutilizzare in applicazioni differenti, e in futuro permettere la creazione di veri e propri moduli riusabili, in cui gli oggetti virtuali, dotati di semantica, possono interagire con altri in maniere che lo sviluppatore non aveva inizialmente previsto. Questo potrebbe avere un forte impatto sui serious game: la semantica degli oggetti di gioco in questo caso risponde a un preciso fine educativo: l’uso di moduli riusabili, con contenuto educativo “certificato”, porterebbe dunque grandi vantaggi in termini di velocità dello sviluppo e abbattimento dei costi di realizzazione.

Viste le premesse qui esposte, usare le ontologie come metodo per la codifica della conoscenza relativa alle entità di gioco ci sembrava un buon punto di partenza. Abbiamo dunque cercato un modo di introdurre la semantica espressa in questo modo nella pipeline di sviluppo del gioco, in modo che possa essere tradotta in comportamenti e interazioni all’interno del gioco stesso.

Codificare i comportamenti degli oggetti virtuali che compongono il gioco è uno dei compiti tipici dello sviluppatore. Per spiegare meglio come questo avvenga, occorre chiarire il funzionamento di un motore di gioco data-driven [6], come lo è ad esempio Unity. L’ambiente costruito tramite il motore di gioco è

composto da “oggetti di gioco” (*game object*). Essi hanno diverse componenti, per esempio il loro aspetto grafico (una palla da calcio avrà la forma di una sfera), e la loro interazione con il motore fisico (il pallone si muoverà se una forza viene applicata su di esso). L’interazione con altri oggetti avviene associando ad essi degli script, ovvero dei pezzi di codice che influenzano il comportamento dell’oggetto stesso. Gli script sono un modo di definire la semantica degli oggetti di gioco, nell’accezione descritta in [7].

Vogliamo che vi sia una corrispondenza tra la conoscenza espressa nell’ontologia e il comportamento che l’oggetto esprime. Tipicamente è lo sviluppatore del gioco che si occupa di tradurre il contenuto educativo in oggetti di gioco che si comportino di conseguenza: tuttavia, questo processo richiede molti sforzi da parte dello sviluppatore che deve lavorare a stretto contatto con l’educatore per far sì che il gioco risulti valido dal punto di vista didattico. I nostri obiettivi di ricerca si sono concentrati sul rendere tale processo più semplice per lo sviluppatore: la nostra idea è stata quella di rendere in parte automatizzato tale processo, effettuando la traduzione delle istanze dell’ontologia in oggetti di gioco attraverso un opportuno “*livello semantico*”, che faccia da tramite fra la conoscenza formalizzata nell’ontologia e il motore di gioco che deve utilizzare tale conoscenza.

Per tradurre la conoscenza espressa nell’ontologia in comportamenti eseguibili dagli oggetti di gioco ci siamo ispirati ad un paradigma esistente, quello dei *servizi* definito in [8]. Con *servizio* si intende la capacità da parte di un oggetto virtuale, di produrre alcuni effetti se si verificano determinati prerequisiti. Un servizio può essere ad esempio la capacità di un distributore automatico di erogare una bibita quando vi si inserisce una moneta, oppure può essere la capacità di un fuoco acceso di scaldare gli oggetti circostanti.

Tale paradigma è stato sviluppato a partire dal concetto di *smart object* [9], ovvero oggetti virtuali “intelligenti”, che possono essere interrogati riguardo alle loro funzionalità: allo stesso modo, gli oggetti virtuali che forniscono dei servizi possono essere interrogati dal giocatore umano o dall’*intelligenza artificiale* del gioco ed essere manipolati di conseguenza.

Come si può intuire, tale modello è abbastanza generale da risultare indipendente da uno specifico linguaggio di programmazione o motore di gioco: le interazioni così codificate dunque non dipendono da una specifica tecnologia di sviluppo giochi, e possono essere conservate in un repository con maggiori possibilità di riuso.

Osservando inoltre che il paradigma dei servizi è basato anch’esso sulle ontologie<sup>1</sup>: risulta dunque semplice definire un mapping automatico fra un dominio di conoscenza arbitrario, che esprime il contenuto educativo, e quello più specifico dei servizi. Il nostro contributo scientifico si inserisce proprio in questa fase.

Il **livello semantico (semantic layer)** è stato realizzato in termini pratici come una libreria software in linguaggio C#, in modo da essere compatibile con Unity 3D, ma anche con qualsiasi altro motore di gioco basato sulla piattaforma .NET di Microsoft. L’implementazione che abbiamo prodotto del semantic layer non è stato utilizzato solo a fini di ricerca, ma anche per la successiva realizzazione del prototipo.

La libreria contiene le funzionalità necessarie a istanziare delle entità di gioco il cui comportamento in fase di esecuzione corrisponde alla concettualizzazione formale data dall’ontologia: essa automatizza la fase di inizializzazione delle istanze come oggetti di gioco corrispondenti, lasciando poi allo sviluppatore il compito di utilizzarle nel modo che ritiene più appropriato.

---

<sup>1</sup> Infatti, il suo funzionamento si basa sul classificare gli oggetti virtuali secondo una gerarchia (anche multipla), ed associare ad ogni classe una o più proprietà.



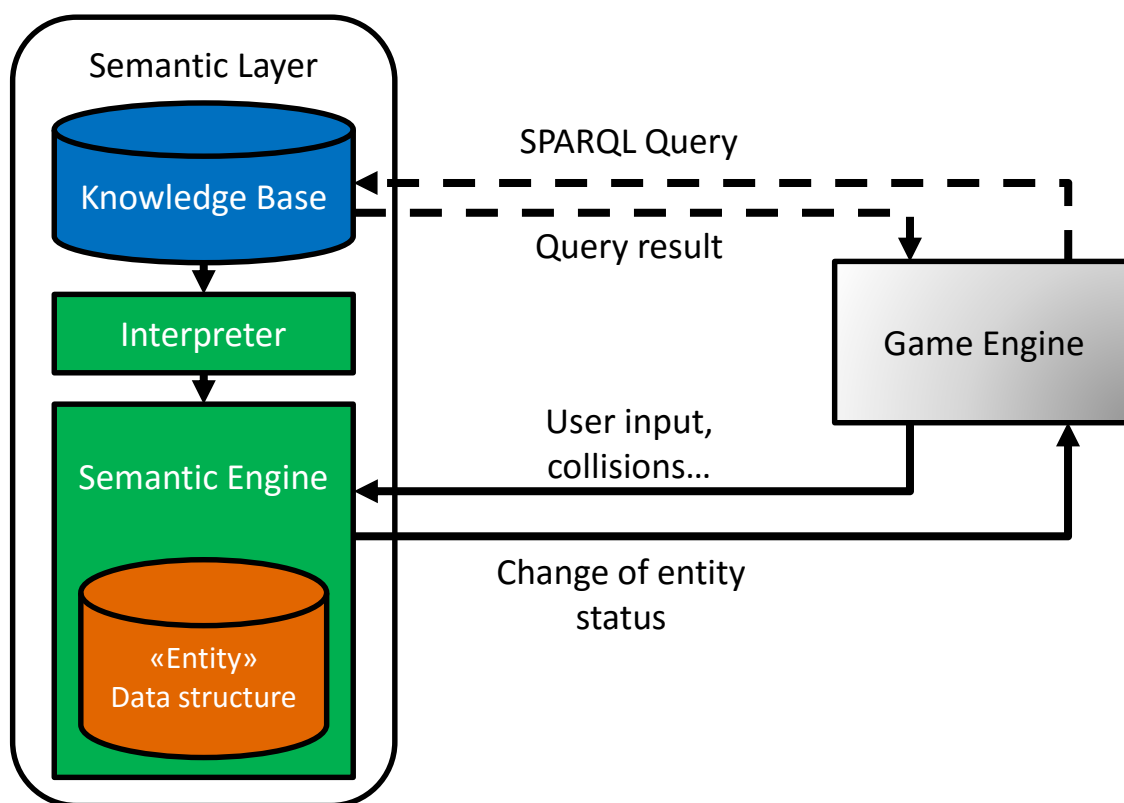


Figura 1: Architettura del semantic layer, e le modalità con cui esso può comunicare con il motore di gioco.

La Figura 1 rappresenta l'architettura del semantic layer. La **base di conoscenza** (Knowledge Base) è l'ontologia più un insieme di istanze, le quali hanno una corrispondenza diretta con i game object del motore di gioco. Tale base di conoscenza viene letta da un **interprete** per produrre una struttura dati specializzata di supporto, che ci serve per aggiornare in tempo reale le proprietà semantiche delle entità di gioco: infatti, accedere frequentemente alla base di conoscenza durante l'esecuzione del gioco porterebbe a un consumo di risorse di calcolo eccessive. Il motore semantico, che si appoggia a questa struttura dati, riceve gli input dal motore di gioco e cambia di conseguenza o stato interno delle entità.

Il mapping di cui abbiamo parlato avviene durante l'inizializzazione da parte dell'interprete, con l'applicazione alla base di conoscenza di determinate regole d'inferenza, espresse in linguaggio SWRL<sup>2</sup>. Queste regole fanno sì che determinate proprietà appartenenti all'ontologia del dominio applicativo siano mappate a specifici servizi, e di fare questo per ogni istanza che possiede tale proprietà.

Il **motore semantico** (Semantic Engine) controlla ad intervalli di tempo regolari se i prerequisiti per un certo servizio si verificano, e in caso affermativo provvede ad applicare gli effetti corrispondenti sulle entità coinvolte. Quando ciò avviene, ogni variazione allo stato delle entità di gioco è comunicata al motore di gioco, tramite un sistema ad eventi. Gli oggetti di gioco creati dal motore di gioco verranno modificati di conseguenza (per esempio modificando il loro aspetto grafico) in modo da riflettere tali cambiamenti.

La comunicazione tra il motore di gioco e il motore semantico è stata realizzata in modo da risultare più disaccoppiata possibile: essa può essere vista sostanzialmente come uno scambio di messaggi. Tale scambio di messaggi ha luogo attraverso un sistema ad eventi. Oltre a questo, è possibile da parte del motore di gioco interrogare direttamente la base di conoscenza, usando delle query SPARQL, che rappresentano uno standard approvato dal W3C<sup>3</sup> nel campo delle tecnologie della conoscenza. Questo è stato permesso per garantire allo sviluppatore la massima flessibilità nell'utilizzo della libreria.

<sup>2</sup> Semantic Web Rule Language

<sup>3</sup> <http://www.w3.org>

### 3. Concettualizzazione e sviluppo di un prototipo di serious game in area edutainment

---

Per dare maggior valore al nostro approccio, e dimostrarne l'utilità in termini pratici, abbiamo cercato di concettualizzare un'applicazione reale, ovvero un serious game, che facesse da banco di prova per la libreria che abbiamo sviluppato. Nell'ambito del progetto, era richiesto che il serious game veicolasse le tematiche d'interesse edutainment proposte da Costa Edutainment. Grazie a ripetuti incontri con il personale didattico di Costa abbiamo elaborato l'analisi dei requisiti: da tale analisi è stato stabilito il target di riferimento, che è quello dei ragazzi di età compresa tra gli 8 e i 12 anni, e abbiamo schematizzato una prima concettualizzazione degli scenari di gioco, che sono i seguenti:

1. **Popolamento dell'habitat:** ogni specie possiede delle caratteristiche fisiche che la rendono più adatta a vivere in un ambiente piuttosto che in un altro: in letteratura scientifica ci si riferisce a questo concetto come *convergenza evolutiva*, ovvero la capacità di specie diverse di sviluppare caratteristiche simili per adattarsi allo stesso ambiente, arrivando ad assomigliarsi moltissimo. Fra tali caratteristiche, quelle che risultano più evidenti sono la forma e il colore. All'inizio del gioco, viene data la possibilità di selezionare un habitat fra quelli disponibili<sup>4</sup>. Il ragazzo, dopo averne ricevuto una descrizione con elencate le caratteristiche principali, dovrà selezionare le specie più adatte a vivere in quell'habitat.
2. **Esplorazione:** in questo scenario il bambino sarà libero di muoversi nell'ambiente virtuale. Il suo compito in questo caso sarà di svolgere delle *quest* (o missioni) assegnate da un assistente virtuale. Tali quest consistono nel "fotografare" (ovvero catturare delle schermate di gioco) le specie che incontra mentre manifestano un comportamento tipico.

Costa Edutainment ci ha fornito anche molte informazioni associate alle specie marine, come le loro caratteristiche fisiche, l'ambiente in cui vivono (e come riconoscere dal loro aspetto l'habitat ideale), i comportamenti tipici, e le specie con cui interagiscono: tutte queste informazioni costituiscono il contenuto educativo su cui si basa il serious game.

Riguardo all'approccio educativo seguito, abbiamo scelto quello costruttivista, che si può sintetizzare con "imparare facendo" [10, 11]: infatti, le conoscenze del giocatore sono ampliate tramite la sperimentazione diretta e attiva, ogni volta che egli prova ad aggiungere un animale all'ambiente nel caso del primo scenario, o quando scatta una fotografia nel caso del secondo scenario. I due scenari possono essere visti come giochi a sé stanti, ma anche come due fasi dello stesso gioco.

Ogni specie<sup>5</sup> che abbiamo selezionato per il concept ha almeno un comportamento tipico da mostrare. A titolo di esempio:

- Il **pesce palla** si gonfia quando è stressato o spaventato;
- Il **delfino comune** (tursiope) allatta la propria prole, essendo un mammifero, e individua altri animali a grande distanza grazie al biosonar;
- Il **pesce angelo** cambia colorazione da giovane ad adulto;
- Il **pesce pagliaccio** vive in simbiosi con l'**anemone di mare**, essendo immune alle sue tossine;
- Il **pesce pulitore** ripulisce da parassiti cutanei o branchiali pesci quali la **cernia** e la **murena**.

---

<sup>4</sup> Nel nostro caso di studio ci siamo limitati a due ambienti, molto diversi tra loro: le scogliere coralline e il mare aperto.

<sup>5</sup> Le specie che abbiamo selezionato per il prototipo sono descritte nell'allegato; ognuna di esse è corredata con una breve scheda testuale che, oltre ad informazioni di carattere generale, descrive i suoi comportamenti tipici.

Avendo deciso di formalizzare la conoscenza relativa al contenuto educativo con le ontologie, abbiamo definito una nostra concettualizzazione dell'ambiente marino, di cui si può vedere la tassonomia delle

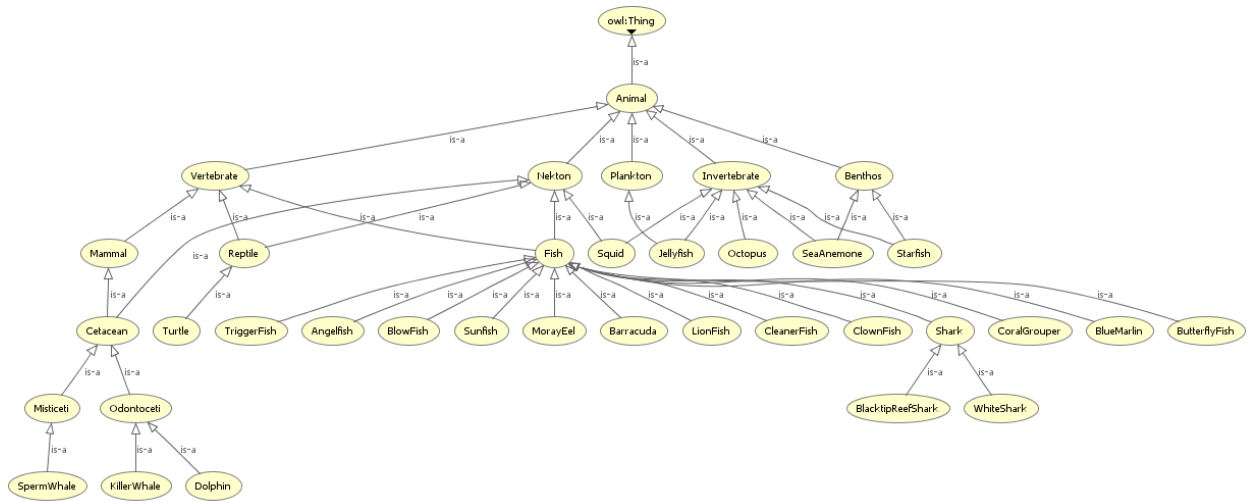


Figura 2: La tassonomia delle specie marine nel nostro caso di studio.

specie in Figura 2. Per ogni animale sono fornite alcune informazioni come il nome comune, e quello scientifico, le caratteristiche fisiche (forma, colore) e le relazioni con altre specie, ad esempio di simbiosi.

Tali caratteristiche, essendo codificate in modo formale, possono essere utilizzate dal reasoner semantico per derivare ulteriori informazioni: per esempio, se un pesce è molto colorato e di dimensioni piccole, il sistema può dedurre che è tipico delle scogliere coralline. Allo stesso modo, animali di dimensioni grandi o molto grandi (squalo bianco, orca, balena) saranno classificati come abitanti del mare aperto. Tali deduzioni sono possibili tramite un apposito insieme di regole, che legano l'aspetto dell'animale a un determinato ambiente di appartenenza.

Esistono però conoscenze che non riguardano tanto l'aspetto dell'animale, quanto il suo comportamento, ad esempio se è un animale diurno o notturno, qual è la sua dieta, e quali relazioni ha con le altre specie. Abbiamo quindi interesse a mostrare tali comportamenti al giocatore.

Dopo una fase di test preliminari, la libreria è stata applicata allo sviluppo del prototipo vero e proprio. In esso abbiamo fatto confluire il lavoro di codifica formale dell'ambiente marino, compresa la traduzione dei comportamenti, che nell'ontologia sono definiti in maniera astratta, a un livello più concreto grazie al meccanismo di mapping.

Il prototipo realizza in termini pratici i due scenari descritti in precedenza (popolamento dell'ambiente ed esplorazione): essi avvengono come due fasi consecutive dello stesso gioco, infatti gli animali selezionati nella prima fase sono quelli che effettivamente popolano l'ambiente nella seconda. Il prototipo è stato realizzato con il motore di gioco Unity<sup>6</sup> e ha una grafica 3D; i modelli 3D utilizzati, dotati di animazioni, sono stati acquistati tramite il servizio Asset Store<sup>7</sup> di Unity.

<sup>6</sup> <http://www.unity3d.com>

<sup>7</sup> <http://www.assetstore.unity3d.com>

Il gioco comincia con la selezione di un ambiente: al momento è possibile scegliere tra le scogliere coralline o il mare aperto. Una volta selezionato l'ambiente, verrà caricato lo scenario di popolamento (Figura 3), in cui lo scopo è di selezionare un certo numero di specie (fissato a cinque nel nostro caso) che si vuole inserire all'interno dell'ambiente selezionato. Il bambino è guidato nella scelta da un testo iniziale, che descrive le caratteristiche dell'ambiente e degli animali ospitati al suo interno, e da una scheda informativa per ogni animale. Le descrizioni degli ambienti e degli animali ci sono state fornite da Costa Edutainment.

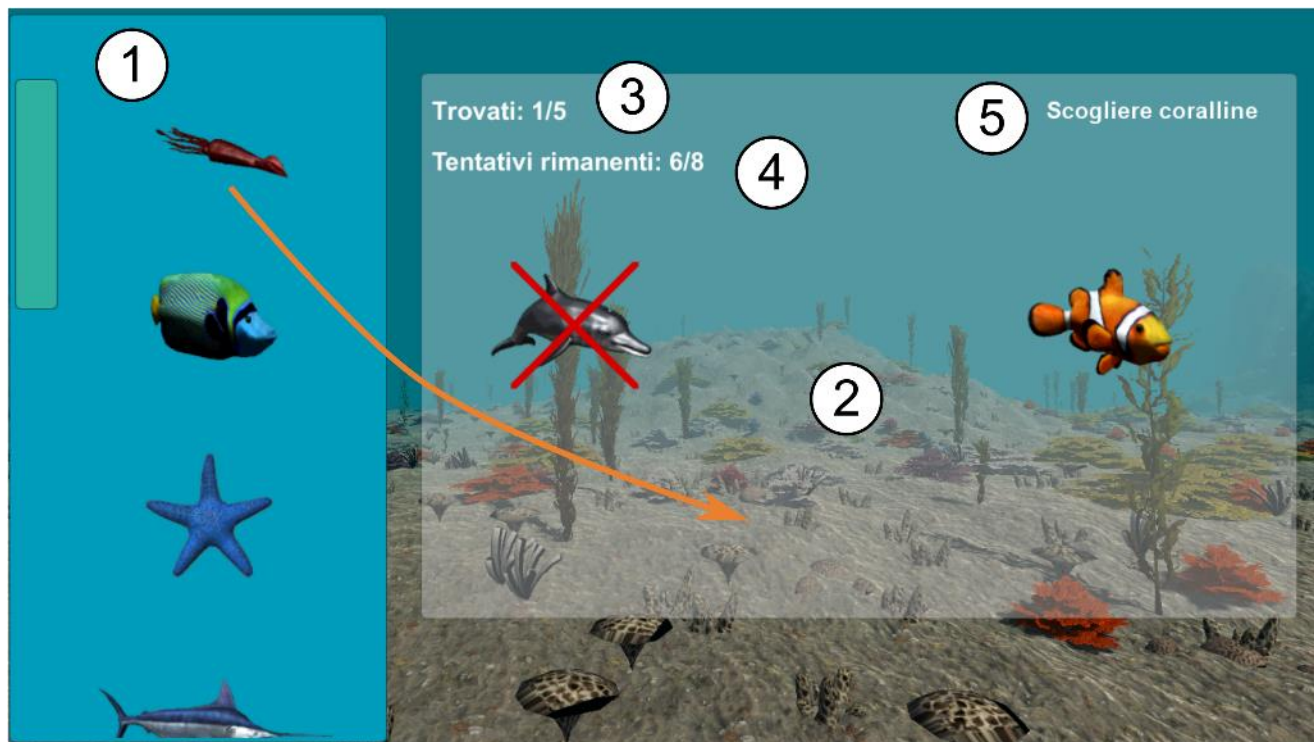


Figura 3: Il gioco implementa un'interfaccia drag-and-drop per l'inserimento degli animali nell'ambiente. Per provare ad inserire un nuovo animale nell'ambiente, occorre trascinare la sua icona sulla parte principale. Gli animali non adatti all'ambiente selezionato sono marcati con una croce.

Nella parte sinistra (1) vi è una finestra contenente le icone delle specie incluse nel gioco; esse possono essere trascinate sulla parte principale (2).

Sempre nella parte principale abbiamo:

- Il numero di animali indovinati (3);
- Il numero di tentativi a disposizione su quelli totali (4);
- Il nome dell'ambiente considerato (5).

Passando il puntatore del mouse sull'icona di un animale, si aprirà una finestra che fornisce informazioni aggiuntive. Tale finestra mostra, per ogni animale, il nome comune e quello scientifico, oltre a una breve descrizione.

Ogni volta che il giocatore seleziona un animale, ovvero lo trascina sull'area principale, viene mostrata una finestra che mostra l'esito (giusto o sbagliato), insieme alle motivazioni. Abbiamo già descritto come la classificazione delle specie sia ottenuta tramite il ragionamento automatico su un insieme di assiomi logici. Ad ogni regola è inoltre associata una spiegazione testuale che viene mostrata a video ogni volta che il giocatore seleziona una specie: il risultato è che quando il giocatore seleziona una specie, oltre alla risposta (giusto/sbagliato) riceverà una serie di spiegazioni relative al perché quell'animale può o non può stare nell'ambiente considerato (v. Figura 4).

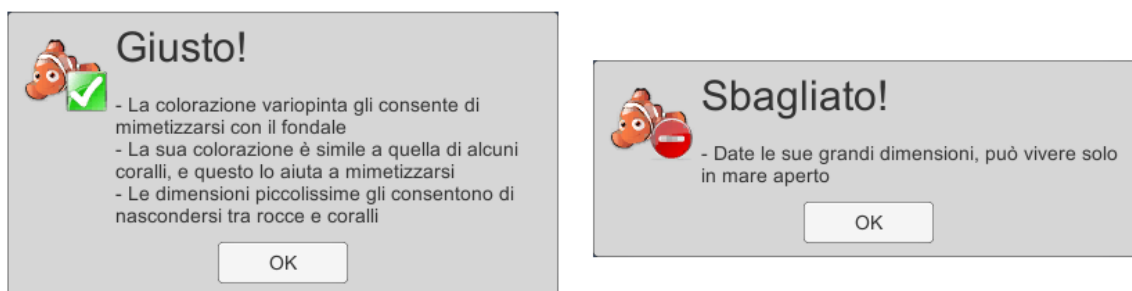


Figura 4: La finestra che viene mostrata in seguito all'inserimento di un animale nella barriera corallina: nel primo caso (pesce pagliaccio), la risposta è affermativa, nel secondo caso l'animale selezionato è lo squalo bianco, tipico animale da mare aperto, e abbiamo dunque una risposta negativa.

Quando il giocatore ha selezionato cinque risposte “giuste”, ovvero cinque specie che appartengono all'ambiente scelto, il primo scenario termina e si passa al secondo. In questo scenario il giocatore deve portare a termine una serie di missioni, o *quest*, in cui è richiesto di fotografare le specie marine che si incontrano (Figura 5) in particolare quando mostrano qualche comportamento specifico o interazione. Il movimento avviene lungo un percorso predefinito, con la possibilità fermarsi in determinati punti “chiave”; i pesci si muoveranno dunque in prossimità di tali punto.

Il semantic layer ci aiuta a controllare se la quest è stata superata: infatti è possibile catturare non solo l'immagine, ma anche lo stato attuale di un'entità di gioco (valore degli attributi, servizi attivati) in un dato momento. Tali informazioni vengono utilizzate dal motore di gioco per decidere se la quest è stata superata.

Inizialmente sono state realizzate delle quest “tutorial”, utili ad insegnare al giocatore il funzionamento dello scenario di gioco. Una delle prime quest richiede ad esempio di scattare una foto ad un animale qualsiasi (istanza della classe Animal). Dopo aver completato le quest iniziali, verrà richiesto di cercare specie con caratteristiche più specifiche: un esempio è il pesce palla, che si gonfia quando stressato o spaventato.

Le quest successive sono più elaborate, per esempio si richiede di fotografare cinque animali differenti; altre dipendono dallo scenario di gioco; per esempio, abbiamo aggiunto una quest in cui si richiede di fotografare il pesce pagliaccio mentre è in simbiosi con l'anemone di mare.



Un aspetto interessante è che, grazie alla concettualizzazione formale delle specie marine, le quest possono essere risolte in più modi: per esempio, se si richiede di trovare un mammifero, il giocatore potrà fotografare a propria scelta un capodoglio, un delfino o un'orca; eventuali altri mammiferi aggiunti all'ontologia saranno considerati una risposta valida, senza modificare minimamente il codice sorgente del gioco.

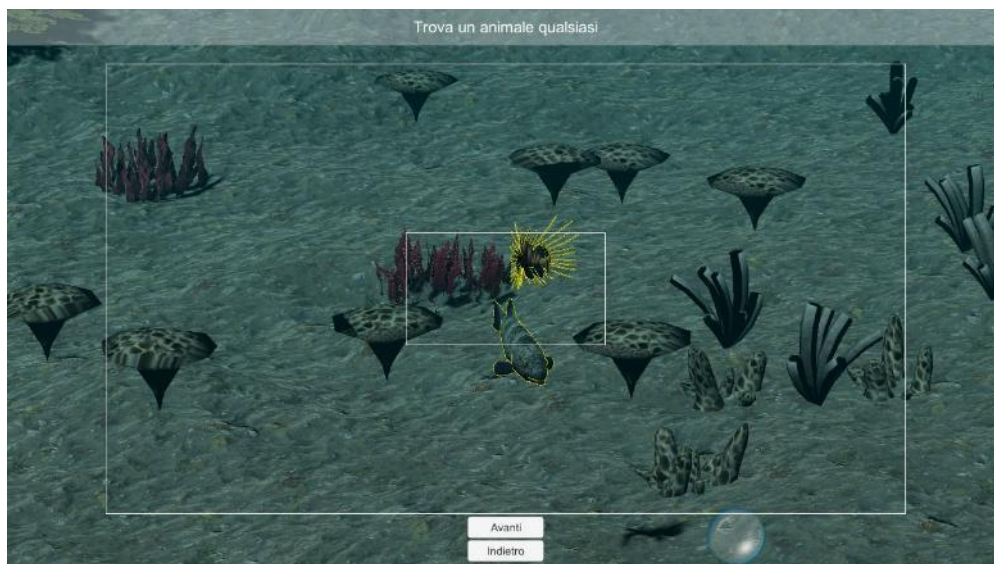


Figura 5: Schermata del secondo scenario (esplorazione), nel quale sono visibili la cernia e il pesce scorpione. Entrambi gli animali sono contornati con un bordo giallo, in modo da essere individuati più facilmente.

## 4. Validazione

Il prototipo di gioco è stato presentato a Costa Edutainment e a Testaluna e discusso in modo da valutare l'efficacia dell'approccio proposto.

È stato apprezzato il sistema di codifica formale della conoscenza tramite ontologie e il livello semantico che abbiamo sviluppato in [12] poiché esso consente la creazione di un modulo educativo riusabile ed estendibile.

Il gioco è stato ritenuto funzionale alla salvaguardia dell'ambiente, in quanto qualunque attività che informi e renda consapevoli i bambini delle caratteristiche dell'ambiente marino aiuta anche a sviluppare la sensibilità verso queste tematiche. I serious game rispondono a questa richiesta e dunque proporre ai bambini un videogioco educativo a tema marino può effettivamente aiutare a creare in loro consapevolezza e rispetto per l'ambiente. Una volta che il bambino si è avvicinato al "mondo sommerso" tramite il gioco, è invogliato ad approfondire la conoscenza..

Per quanto riguarda il prototipo che abbiamo realizzato, molte delle osservazioni hanno riguardato funzionalità o contenuti che dovrebbero essere aggiunte per rendere il gioco più divertente e coinvolgente per il pubblico a cui si rivolge, i ragazzi dagli 8 ai 12 anni. In effetti, l'obiettivo principale del lavoro di ricerca svolto è stato rivolto alla codifica formale del contenuto educativo in modo da renderlo interpretabile e integrabile in modo "automatico" all'interno di un videogioco. Pertanto, alcuni aspetti ludici sono stati lasciati in secondo piano nello sviluppo del prototipo e riguardano principalmente l'aggiunta di nuovi contenuti o funzionalità proprie del gioco. Di seguito, sono descritte con maggior dettaglio, diversi fattori che potrebbero rendere il serious game proposto più efficace.

**Arricchimento con ulteriori contenuti.** Si potrebbero includere nuove specie marine per rendere l'ambiente più variamente popolato. Ad esempio, si potrebbero aggiungere banchi di sardine per rendere più dinamico l'ambiente di mare aperto, o la seppia nell'ambiente delle scogliere. Anche la base di conoscenza andrebbe estesa di conseguenza, con nuove classi e nuovi comportamenti/interazioni a seconda

dei casi. Essendo il sistema organizzato in maniera modulare, sarebbe anche possibile aggiungere in modo semplice nuovi ambienti, ad esempio le scogliere mediterranee. In generale, sarebbe auspicabile rendere gli scenari più dinamici e movimentati, in modo che i bambini siano invogliati a giocare.

**Miglioramenti estetici.** Si potrebbe rendere la grafica più giocosa e meno realistica, simile a quella di un cartone animato. Ciò è possibile grazie a una tecnica di rendering detta cel-shading<sup>8</sup>; è stato anche osservato che alcuni pesci sono troppo piccoli per essere fotografati agevolmente. In questi casi, occorre trovare un compromesso tra realismo e semplicità d'uso: infatti, nei videogiochi è abbastanza tipico trovare oggetti con proporzioni diverse da quelle reali, per evitare di avere oggetti troppo grandi o troppo piccoli insieme. Si potrebbe anche dare maggiore libertà di movimento al giocatore, ad esempio permettendogli di nuotare come farebbe un subacqueo, consentendogli di avvicinarsi di più agli animali.

**Integrazione di ulteriori funzionalità.** Si potrebbe aggiungere un sistema di punteggi che renda il gioco più divertente e aggiunga una componente competitiva; includere delle finestre pop-up informative che forniscano ulteriori informazioni relativamente al contesto (per esempio, fornire notizie e curiosità sugli animali che si incontrano mentre si esplora lo scenario); aggiungere una mappa che mostri la collocazione geografica degli ambienti. Anche l'interfaccia utente potrebbe essere rivista e resa più intuitiva, specialmente in vista di una possibile conversione su piattaforma mobile.

**Sviluppo di nuovi scenari.** Un maggiore dinamismo nel gioco aiuterebbe a mantenere vivo l'interesse dei giocatori. Utilizzando la libreria SemanticLayer che abbiamo realizzato, l'implementazione di nuovi scenari dovrebbe risultare semplice, in quanto la parte relativa al contenuto educativo e all'interazione fra le specie marine è stata resa riutilizzabile.

**Sviluppo di un prodotto finale.** Perché il prototipo possa evolvere in un videogioco completo, si dovrebbe riorganizzare e ripulire il codice sorgente (refactoring) per migliorare l'estensibilità e il riuso delle funzioni già implementate, aggiungere effetti sonori, animazioni e in generale migliorare l'estetica dell'interfaccia grafica e degli ambienti, e l'usabilità dell'interfaccia. Sarebbe inoltre vantaggioso proporre il gioco per tablet: questo perché attualmente le piattaforme mobili quali Android e iOS stanno diventando quelle di riferimento per i giocatori più giovani. Tuttavia, questo presuppone un miglioramento dell'interfaccia utente e dell'aspetto ludico, poiché se si vuole che il gioco abbia successo su un mercato di questo tipo, la modalità d'interazione dev'essere immediatamente comprensibile al giocatore e, senza l'utilizzo di lunghi tutorial, deve risultare coinvolgente e appagante sin dal primo momento.

## 5. Conclusioni

---

Nel lavoro di ricerca proposto, abbiamo sviluppato una metodologia per definire componenti semantiche riusabili nella progettazione di serious game. In particolare, lo sviluppo prototipale ha riguardato l'area edutainment in cui il realismo delle interazioni è un aspetto molto importante da considerare per veicolare correttamente al giocatore la conoscenza contenuta nel gioco.

Il nostro specifico caso di studio era quello dell'ambiente marino: per prima cosa abbiamo espresso la conoscenza riguardo a questo campo applicativo in modo formale, poi abbiamo realizzato un insieme di strumenti software per tradurre tale conoscenza in comportamenti veri e propri all'interno del gioco, infine abbiamo testato la nostra libreria su un caso reale, con un prototipo di serious game che utilizzasse la libreria stessa.

Disaccoppiando la conoscenza del contenuto educativo dalla parte ludica, ma mantenendo uno scambio diretto e real-time tra le due componenti, abbiamo delineato una nuova metodologia di progettazione dei serious game. È opportuno sottolineare, infatti, che i benefici del nostro approccio non si limitano ai soli

---

<sup>8</sup> [https://en.wikipedia.org/wiki/Cel\\_shading](https://en.wikipedia.org/wiki/Cel_shading)

serious game, in quanto lo sviluppo di componenti riusabili torna a vantaggio anche dei giochi d'intrattenimento. Come già discusso più diffusamente in [12], esistono alcuni generi di videogiochi per l'intrattenimento che potrebbero beneficiare in modo particolare di tale soluzione. Ad esempio, nel caso delle avventure grafiche, si potrebbe avere un repository di oggetti dotati di funzionalità da cui attingere; nei giochi di ruolo, si potrebbero definire le meccaniche del gioco stesso (sviluppo del personaggio, abilità, equipaggiamento) a partire da una componente base riusabile; infine nei giochi d'azione stealth l'intelligenza artificiale<sup>9</sup> che controlla i personaggi, potendo "interrogare" gli oggetti circostanti riguardo alle loro funzionalità, potrebbe esprimere un maggiore dinamismo nella scelta delle proprie azioni, risultando dunque più realistica.

## 6. Riferimenti

---

- [1] M. Ulicsak e M. Wright, Games in Education: Serious Games: A Futurelab Literature Review, FutureLab, 2010.
- [2] GaLA Consortium, «Roadmap on Serious Games,» 2014. [Online]. Available: <http://www.seriousgamesociety.org/download/GALA%20Roadmap%203.pdf>.
- [3] C. E. Catalano, M. Mortara, M. Spagnuolo e B. Falcidieno, «Semantics and 3D media: Current issues and perspectives,» *Computers & Graphics*, vol. 35, n. 4, pp. 869-877, 2011.
- [4] M. Mortara, C. E. Catalano e C.-I. Ge, «Computer Graphics Recipes for Endowing 3D Shapes with Semantics in Virtual Worlds,» *3D Advanced Media In Gaming And Simulation (3AMIGAS)*, p. 5.
- [5] T. R. Gruber, «A translation approach to portable ontology specifications,» *Knowledge acquisition*, vol. 5, n. 2, pp. 199-220, 1993.
- [6] S. Bilas, «A data-driven game object system,» in *Game Developers Conference Proceedings*, 2002.
- [7] T. Tutenel, R. Bidarra, R. M. Smelik e K. J. D. Kraker, «The role of semantics in games and simulations,» *Computers in Entertainment (CIE)*, vol. 6, n. 4, p. 57, 2008.
- [8] J. Kessing, T. Tutenel e R. Bidarra, «Services in game worlds: A semantic approach to improve object interaction,» in *Entertainment Computing--ICEC 2009*, Springer, 2009, pp. 276-281.
- [9] M. Kallmann e D. Thalmann, «Direct 3D Interaction with Smart Objects,» in *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, New York, NY, USA, 1999.
- [10] G. French, «Children's early learning and development,» *The Framework for Early Learning*, 2007.
- [11] J. Piaget, M. Cook e W. Norton, The origins of intelligence in children, vol. 8, International Universities Press New York, 1952.
- [12] A. Repetto e C. E. Catalano, «A Semantic Layer for Knowledge-Based Game Design in Edutainment Applications,» *EAI Endorsed Transactions on Future Intelligent Educational Environments*, vol. 15, n. 5, 2015.

---

<sup>9</sup> Nei videogiochi, per intelligenza artificiale si intende un insieme di algoritmi e tecniche che guida il comportamento dei personaggi non giocanti; essa non rappresenta un'intelligenza artificiale nel senso più ampio del termine, il cui scopo sarebbe quello di massimizzare le proprie performance rispetto a quelle del giocatore; lo scopo è invece di rendere massimo il divertimento del giocatore, il che comprende la possibilità di commettere errori, di tanto in tanto.



**Recent titles from the IMATI-REPORT Series:**

- 16-01:** *Optimal strategies for a time-dependent harvesting problem*, G.M. Coclite, M. Garavello, L.V. Spinolo.
- 16-02:** *A new design for the implementation of isogeometric analysis in Octave and Matlab: GeoPDEs 3.0*, R. Vázquez.
- 16-03:** *Defect detection in nanostructures*, D. Carrera, F. Manganini, G. Boracchi, E. Lanzarone.
- 16-04:** *A study of the state of the art of process planning for additive manufacturing*, M. Livesu, M. Attene, M. Spagnuolo, B. Falcidieno.
- 16-05:** *Calcolo di descrittori ibridi geometria-colore per l'analisi di similarità di forme 3D*, A. Raffo, S. Biasotti.
- 16-06:** *An appointment scheduling framework to balance the production of blood bags from donation*, Seda Baş, Giuliana Carello, Ettore Lanzarone, Semih Yalçındağ.
- 16-07:** *From lesson learned to the refactoring of the DRIHM science gateway for hydro-meteorological research*, D. D'Agostino, E. Danovaro, A. Clematis, L. Roverelli, G. Zereik, A. Galizia.
- 16-08:** *Algorithms for the implementation of adaptive isogeometric methods using hierarchical splines*, E.M. Garau, R. Vázquez.
- 16-09:** *Feature curve identification in archaeological fragments using an extension of the Hough transform*, M.L. Torrente, S. Biasotti, B. Falcidieno.
- 16-10:** *Comparing methods for the approximation of rainfall fields in environmental applications*, G. Patané, A. Cerri, V. Skytt, S. Pittaluga, S. Biasotti, D. Sobrero, T. Dokken, M. Spagnuolo.
- 16-11:** *Persistence-based tracking of rainfall field maxima*, S. Biasotti, A. Cerri, S. Pittaluga, D. Sobrero, M. Spagnuolo.
- 16-12:** *Studio e sviluppo di componenti semantiche riusabili per la progettazione di serious game. Applicazioni all'area edutainment*, Andrea Repetto, Chiara Eva Catalano.