

# *Flow your story*

Relazione seconda unità didattica

---

*METHODS IN COMPUTER SCIENCE  
EDUCATION: ANALYSIS*

Andrea Rosati  
1595834

Federica Spini  
1588482

## Revision History

<b>Revision</b>	<b>Date</b>	<b>Author(s)</b>	<b>Description</b>
1.0	03.06.19	AR, FS	Prima stesura documento

## Indice

<b>1</b>	<b>Introduzione</b>	<b>3</b>
1.1	Cenni storici: dai librigame ai videogiochi narrativi . . . . .	4
1.1.1	Librigame . . . . .	4
1.1.2	Avventure testuali . . . . .	6
1.1.3	I moderni giochi narrativi . . . . .	7
<b>2</b>	<b>Target e requisiti della la lezione</b>	<b>9</b>
2.1	Pre-requisiti didattici . . . . .	9
2.2	Pre-requisiti informatici . . . . .	9
2.3	Obiettivi didattici . . . . .	10
2.3.1	Obiettivi didattici interdisciplinari . . . . .	10
2.3.2	Obiettivi didattici informatici . . . . .	11
<b>3</b>	<b>Flowgorithm</b>	<b>12</b>
3.1	Vantaggi . . . . .	13
3.2	Limitazioni . . . . .	13
<b>4</b>	<b>La lezione</b>	<b>14</b>
4.1	Introduzione ai librigame e le avventure testuali . . . . .	14
4.2	Demo per gli studenti . . . . .	15
4.2.1	Edipo e la Sfinge . . . . .	15
4.2.2	Ulisse e Polifemo . . . . .	17
4.3	Valutazione . . . . .	19
<b>5</b>	<b>Altre possibilità per riadattare la lezione</b>	<b>21</b>

## Introduzione

Flow your story è un'attività interdisciplinare di informatica e letteratura in cui gli studenti, partendo da un racconto o un romanzo, progettano e sviluppano al computer un videogioco di genere avventura testuale (spiegato nella sezione 1.1.2) basato su di esso. Le avventure testuali, per definizione, sono prive di qualsiasi elemento di grafica e interfaccia utente e quindi sono molto facili da sviluppare poiché bisogna soltanto implementare la logica del gameplay<sup>1</sup> senza preoccuparsi di altri aspetti. Questa semplificazione nello sviluppo rende le avventure testuali adatte ad essere sviluppate da ragazzi dai 14 anni in su. Abbiamo riassunto quelli che secondo noi sono i punti di forza di questa attività interdisciplinare:

### Punti forza

- I videogiochi sono ormai un fenomeno di massa e un medium non soltanto appannaggio dei cosiddetti nerd. Crearne uno, anche soltanto testuale, può essere un'esperienza più coinvolgente per molti ragazzi, rispetto a fare un esercizio di informatica più fine a se stesso
- Molti adolescenti leggono di malavoglia, o non lo fanno affatto, i libri che gli vengono assegnati a scuola. Sapere in anticipo di dover sviluppare un gioco basato sul romanzo da leggere potrebbe stimolarli ad essere più attenti e coinvolti dalla lettura stessa
- Scegliendo un'opportuna tecnologia è possibile far lavorare gli studenti in gruppi
- Scegliendo un'opportuna tecnologia è possibile riunire i lavori dei ragazzi in un unico progetto più grande
- Si tratta di un progetto facilmente riadattabile per diverse materie oltre l'italiano come storia, filosofia, biologia, matematica ecc... Qualsiasi materia in cui sono presenti decisioni e conseguenze, in un rapporto causa-effetto, può essere affrontata utilizzando dei diagrammi di flusso

Per la nostra dimostrazione abbiamo scelto i poemi e miti greci ed in particolare: il mito di Edipo fino al suo scontro con la Sfinge, ed il canto dell'Odissea in cui Ulisse affronta Polifemo. Su questi due episodi abbiamo sviluppato due esempi di avventure testuali con diversi finali. Abbiamo scelto questi argomenti, perché, soprattutto per quanto riguarda l'Odissea, fanno parte del programma di studio

<sup>1</sup>La totalità dell'esperienza utente durante l'interazione con un videogame

statale, ed essendo ricchi di avventure si prestano perfettamente alla creazione di un'avventura testuale.

### 1.1 Cenni storici: dai librigame ai videogiochi narrativi

Le avventure testuali sono oggi un genere di nicchia per i giocatori moderni, questo perché sono state rimpiazzate dai videogiochi narrativi, che sono la loro naturale evoluzione (stesso gameplay ma con grafica e sonoro all'avanguardia). Le stesse avventure testuali sono state però la naturale evoluzione di un altro tipo di prodotto: il librogame.

#### 1.1.1 Librigame

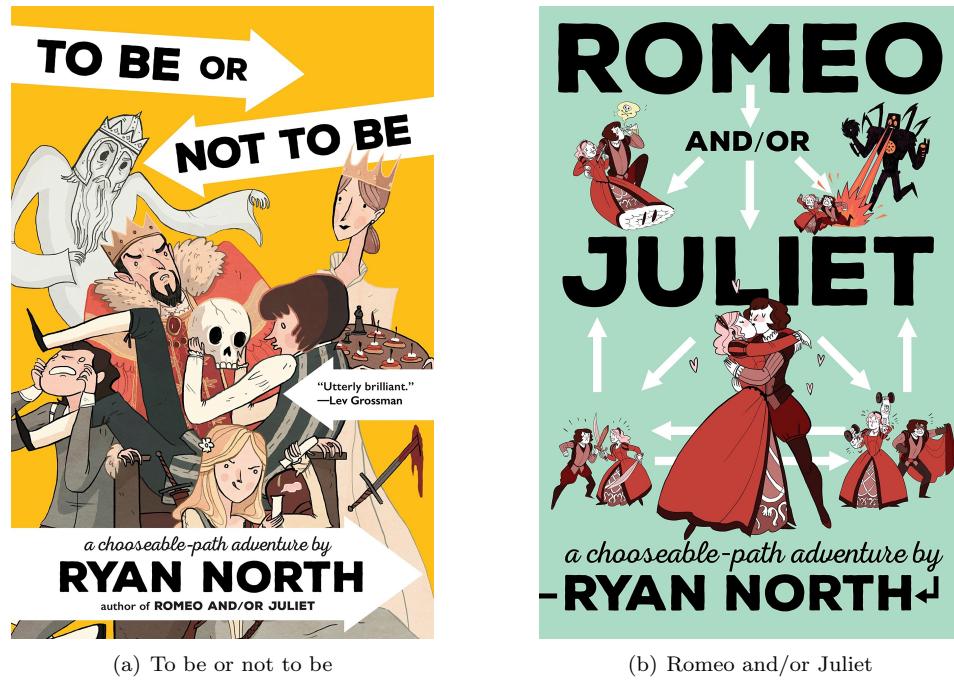


Figura 1: Due moderni librigame scritti da Ryan North

Il librogame non è un videogioco, ma un genere letterario in cui il lettore può prendere decisioni che cambiano il corso della storia narrata. I primi librogame possedevano un sistema "a bivi" [2] in cui a fine di ogni capitolo il libro obbligava il giocatore a prendere una decisione scegliendo una delle opzioni offerte, a ogni opzione era associata il numero della pagina del libro da cui il lettore doveva riprendere la lettura se aveva scelto quella opzione. Le storie spesso erano di genere fantasy o fantascientifico, e solitamente i libri erano ricchi di illustrazioni

per rendere l'esperienza più immersiva. Col tempo questi libri hanno offerto sistemi di gioco più complessi che li hanno resi sempre più simili a dei giochi di ruolo in stile *Dungeons and dragons* dove il ruolo del "game master"<sup>2</sup> è gestito dal libro stesso. In questa tipologia di librogame il giocatore oltre a prendere decisioni per la storia deve anche gestire i parametri che caratterizzano il proprio personaggio, e in alcuni può anche combattere dei nemici utilizzando dei dadi per risolvere i combattimenti.

Questi libri hanno goduto di una certa popolarità tra fine anni '70 e inizio anni '90 per poi scomparire per via della sempre più importante presenza dei video-giochi nel mercato ludico. Negli ultimi anni sia per un'operazione nostalgica, e, forse, anche per l'avvento degli ebook, alcuni editori stanno riproponendo dei librigame ottenendo anche un buon successo come nel caso dello scrittore canadese Ryan North [1] che basandosi su due delle opere più famose di Shakespeare ha pubblicato *To be or not to be* e *Romeo and/or Juliet* facendo un lavoro simile a quello che noi proponiamo di far fare agli studenti.

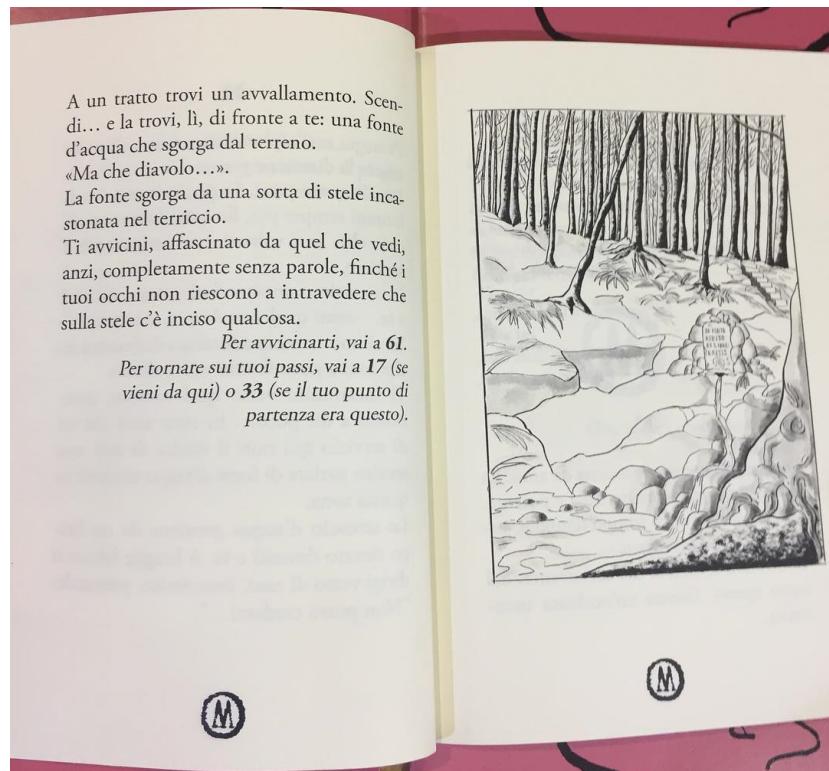


Figura 2: Pagina "bivio" in un librogame

<sup>2</sup>la persona che gestisce una partita di gioco di ruolo, descrivendo l'ambientazione ed elaborandone le mappe, proponendo le avventure che coinvolgono i personaggi giocanti, interpretando il ruolo dei personaggi non giocanti, fra cui i mostri, e decidendo i risultati delle azioni intraprese.

### 1.1.2 Avventure testuali

Le avventure testuali sono dei videogiochi composti da un'interfaccia testuale con cui il giocatore può interagire tramite comandi testuali. Come per i librigame, il fulcro del gameplay delle avventure testuali consiste nel prendere decisioni che influenzano lo svolgersi della storia, tuttavia avendo il software meno limiti rispetto alla carta stampata gli sviluppatori hanno potuto inserire in questi videogiochi meccaniche di gioco più avanzate come gestione di un inventario, sviluppo del personaggio e risoluzione di enigmi. I comandi testuali per interagire col gioco consistono spesso nella scelta di un'opzione da una lista, oppure, nei casi delle avventure più avanzate, nello scrivere delle frasi all'imperativo composte spesso dal solo predicato verbale e complemento oggetto, interpretate poi da dei rudimentali parser testuali.



Le prime avventure nacquero nel finire degli anni '70 e divennero molto popolari negli anni '80. Con il progredire della tecnologia e l'arrivo di videogiochi dalla grafica sempre più gradevole la popolarità andò a sciamare fino a rendere le avventure testuali un genere di nicchia.

**\* AVVENTURA NEL CASTELLO! \***

Stai pilotando il tuo monoposto sopra la desolata regione scozzese delle Highlands. Hai appena sorvolato il lago di Loch Ness... Improvvisamente il Motore perde colpi. I comandi non rispondono!

Stai precipitando!

Cosa devo fare ? GUARDA AEREO

C'e' troppo fumo.

Stai precipitando!

Cosa devo fare ? \_

Figura 3: L'interfaccia testuale di Avventura nel Castello

### 1.1.3 I moderni giochi narrativi

Il genere di videogiochi che oggi ha raccolto l'eredità delle avventure testuali è senza dubbio quello dei giochi narrativi. Considerato come un sotto-genere delle avventure grafiche, anche nei videogiochi narrativi il fulcro del gameplay è prendere decisioni che influenzano la storia ma a differenza delle avventure testuali questi videogiochi offrono sia una grafica all'avanguardia che un'interfaccia più intuitiva rispetto quella testuale. Solitamente le storie di questi giochi sono meno complesse rispetto quelle delle avventure testuali e dei librigame, offrendo quindi meno bivi percorribili, tuttavia spesso offrono piccoli parti di gameplay non tipicamente da avventura come per esempio l'espolarazione di ambienti o minigames.



(a) Life is strange



(b) Reigns: Game of thrones



(c) Heavy rain



(d) The Walking dead

Figura 4: Quattro tra i più famosi videogiochi narrativi usciti nell'ultima decade

Questo genere sta avendo un notevole successo negli ultimi anni a tal punto che ci sono state anche produzioni tripla A<sup>3</sup> da parte delle grandi software house

<sup>3</sup>Produzioni di videogiochi con budget milionari, l'equivalente del termine "blockbuster" per un film.

di videogiochi e si possono ormai trovare giochi di questo genere su qualsiasi piattaforma video-ludica presente sul mercato incluso anche il mobile. Ultimamente la piattaforma di streaming video **Netflix** sta sperimentando nuovi contenuti video in cui in alcuni determinati punti si chiede all'utente come far continuare la storia scegliendo un'opzione da quelle offerte; i primi contenuti offerti erano destinati per un pubblico molto giovane e in seguito è stato rilasciato "Bandersnatch"<sup>[5]</sup> della serie **Black Mirror** che possiamo considerare il primo contenuto interattivo prodotto da Netflix destinato per un pubblico adulto. Il network ha successivamente ammesso di esser soddisfatto di come il pubblico ha reagito a questi prodotti dichiarando poi di volerne produrre ulteriori.

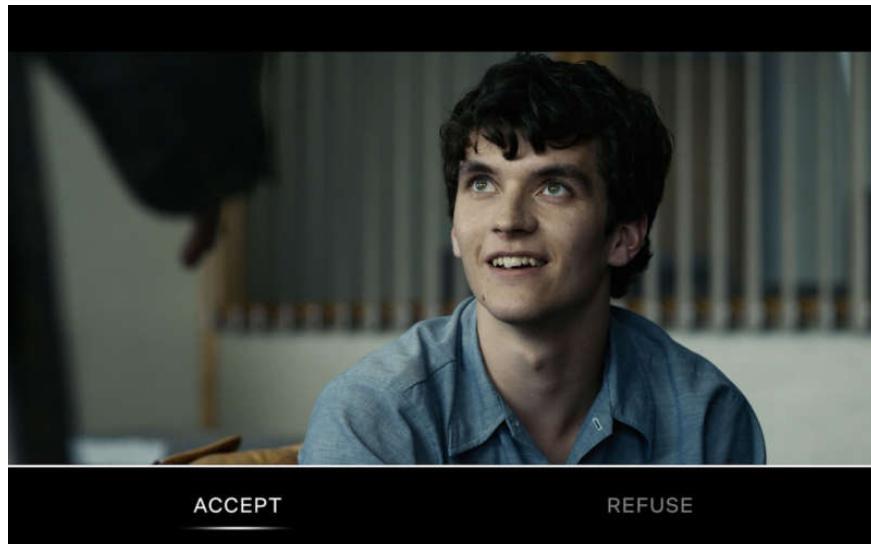


Figura 5: L'episodio speciale di black mirror "Bandersnatch"

Tutto questo cappello introduttivo è mirato a spiegare come i ragazzi possano essere coinvolti da un gioco, che è un revival degli anni '80 che stà tornando di moda. E' importante che vedano l'informatica come qualcosa di utile per creare software interessante anche da utilizzare, e metterli nelle condizioni di inventare qualcosa che si avvicini ai prodotti da loro conosciuti, anche con le conoscenze base che gli sono state date finora. Spesso il problema agli inizi è infatti che i ragazzi che non amano risolvere i problemi per fini logico-enigmistici, non capiscono il potenziale di ciò che gli viene insegnato, e si limitano a fare considerazioni che minimizzano la materia, pensando che cose come ordinare una lista non servano a niente. Fargli fare qualcosa che, se ben realizzato, potrebbero anche far utilizzare a terzi, potrebbe dar loro più soddisfazione.

## Target e requisiti della la lezione

Come già anticipato nell'introduzione, il target d'età per questa attività è dai 14 anni in su, perciò ci rivolgiamo a ragazzi dei primi tre anni delle scuole superiori, dove l'anno esatto dipende dall'ordine in cui affrontano gli argomenti dell'informatica a seconda dell'indirizzo.

### 2.1 Pre-requisiti didattici

Prima di creare e programmare la storia, bisogna ovviamente aver ben chiaro l'argomento di letteratura su cui si deve sviluppare la storia. I concetti che servono su come realizzarla possono essere "raccontati" sul momento, ovviamente l'aver letto un librogame o aver giocato un gioco narrativo potrebbero aiutare, anche se si faranno esempi a lezione. Per il resto, le conoscenze necessarie sono quelle per scrivere un buon tema: proprietà di linguaggio, utilizzo corretto di ortografia, sintassi, punteggiatura e così via.

### 2.2 Pre-requisiti informatici

Questa learning unit può esser realizzata sfruttando solamente concetti base di informatica, che tuttavia devono essere ben chiari ai ragazzi. Questo perché devono essere utilizzati tutti assieme in un lavoro più grande per ragazzi delle superiori a questo punto del programma, rispetto ai normali esercizi con cui sono abituati a trattare. È possibile scrivere l'intero programma senza utilizzare particolari strutture dati, pattern e senza che il codice diventi ingestibile, sfruttando quindi il paradigma procedurale. Le nozioni necessarie da conoscere obbligatorie per gli studenti sono:

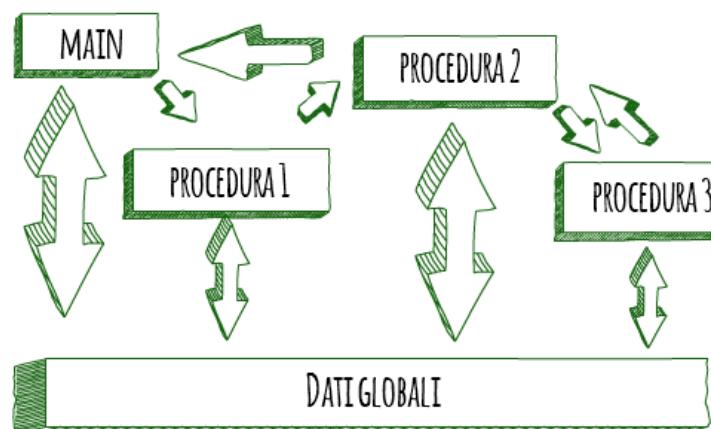


Figura 6: Paradigma procedurale

### Pre-requisiti informatici

- Concetto di variabile ed operazioni di lettura, scrittura, dichiarazione e assegnazione
- Conoscere e saper maneggiare i fondamentali tipi di dato presenti nei più comuni linguaggi di programmazione, quali interi, stringhe, booleani e decimali
- Saper utilizzare i controlli condizionali, anche annidati
- Saper utilizzare i cicli e conoscere la differenza fra while e for
- Raccogliere parti di codice ripetute in procedure
- Saper utilizzare gli array di dati
- Conoscere i diagrammi di flusso e saper rappresentare un algoritmo tramite essi

Molti concetti informatici appresi successivamente potrebbero essere facilmente inseriti nel progetto, magari per crearne versioni con un codice maggiormente strutturato o più vario (affronteremo meglio la questione nella sezione 5).

## 2.3 Obiettivi didattici

Dopo aver spiegato cosa i ragazzi devono conoscere prima di realizzare la Learning Unit, spieghiamo ora cosa dovrebbero aver imparato alla fine.

### 2.3.1 Obiettivi didattici interdisciplinari

Gli obiettivi didattici di questa learning unit sono principalmente legati alla materia letteraria. I ragazzi tramite questa esperienza imparano a leggere un racconto o un romanzo analizzandone la struttura della trama e la caratterizzazione dei personaggi. Il compito può essere un sostituto del classico "riassunto", con una richiesta che impone al ragazzo di aver dimostrato quali sono i punti chiave della vicenda e la conseguenza logica con cui essa si svolge. Inoltre è richiesta una parte di scrittura creativa, in cui il ragazzo è libero di dimostrare di avere altre conoscenze dello stesso contesto (parlando dell'Odissea potrebbe, ad esempio, citare altri episodi mitologici o parlare in modo competente del contesto storico), e di "divertirsi" nell'immaginare scenari alternativi.

Come obiettivo interdisciplinare aggiuntivo si può chiedere ai ragazzi di implementare un sistema di punteggio nei propri giochi in modo che gli altri studenti possano utilizzarli. In questo modo potrà essere ancora più coinvolgente giocare con i lavori dei compagni, ma soprattutto si dovrà dimostrare di aver studiato, perché le scelte più coerenti con la storia reale saranno quelle premiate. Quindi si impara anche come "valutare" perché alla fine si sta creando qualcosa che potrebbe essere utilizzato come una verifica leggera per i compagni di classe.

Anche il modo in cui viene effettuato questo sistema di punteggio può essere materiale utile per valutare, dal punto di vista delle conoscenze letterarie, il gruppo di ragazzi che ha realizzato il progetto.

### 2.3.2 Obiettivi didattici informatici

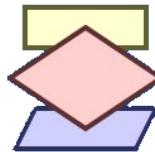
Dal punto di vista informatico invece, questa learning unit mira a consolidare le conoscenze che i ragazzi già possiedono, dandogli l'occasione di poter far pratica con un software completamente progettato da loro e che sia potenzialmente più corposo dei soliti esercizi a cui sono abituati. Lo scopo è dimostrare che anche con i semplici mezzi base che loro hanno a disposizione, è possibile creare un programma complesso che sia anche divertente da utilizzare. Gli obiettivi principali possono essere riassunti nella tabella sottostante:

Obiettivi didattici informatici
<ul style="list-style-type: none"><li>• Consolidare tutti i concetti acquisiti finora nel resto del corso di informatica e usarli tutti assieme: questo dovrebbe aumentare la confidenza e la manualità che i ragazzi hanno con i vari comandi appresi. Infatti facendo gli esercizi classici i ragazzi hanno la tendenza, già dalla lettura della traccia, a pensare cose come "D'accordo, questo esercizio è sul ciclo e dovrò usare un ciclo" pensando ad un altro esercizio precedentemente già visto. Questo fa in modo che appena escono dalla confort zone degli esercizi a cui sono abituati si sentano spaesati: dover usare tutte le conoscenze in un lavoro come quello che proponiamo potrebbe renderli più indipendenti</li><li>• Apprendere i vantaggi del pensare e schematizzare in anticipo il programma: in un progetto che è così più grande rispetto agli esercizi di informatica a cui i ragazzi sono abituati è molto importante pensare e schematizzare in anticipo, per programmare dopo</li><li>• Utilizzare in modo costruttivo gli esempi messi a disposizione da noi: non limitarsi ad un copia-incolla cambiando i testi, ma utilizzare strutture simili per creare qualcosa di originale</li><li>• Imparare ad orientarsi in un codice più vasto: se c'è un bug o qualcosa che va cambiato, bisogna trovare il punto giusto dove effettuare le modifiche e non "perdersi" non sapendo cosa viene fatto nelle varie parti del codice</li></ul>

# Flowgorithm

Si tratta di uno strumento didattico davvero molto utile per i ragazzi che studiano l'informatica alle scuole superiori. Difatti la scuola italiana mira in primo luogo a far apprendere ai ragazzi il pensiero computazionale e, solo successivamente, ad esprimere questo pensiero con dei linguaggi di programmazione. I flowchart sono il primo concetto che i ragazzi imparano al riguardo e crearli tramite un computer per poterli "avviare" chiarifica davvero il concetto di come vengono eseguiti i programmi. Il sito e la documentazione di Flowgorithm sono reperibili a questo [link](#).

Oltre al fatto che ai ragazzi si insegna ad usare i flowcharts, c'è il vantaggio che questa struttura è perfetta per modellare il tipo di problema che bisogna affrontare in questo progetto. Il corso di una storia il cui svolgimento dipende dalle scelte effettuate, è infatti un diagramma di flusso in cui il percorso effettuato è direzionato da cicli e controlli, che avranno esiti differenti a seconda degli input del giocatore.



## Flowgorithm



Figura 7: Un flowchart usato per il gioco "Analogue: a hate story"

### 3.1 Vantaggi

Flowgorithm è molto semplice e contiene l'insieme di tutti i comandi che sono alla base della programmazione. È tipato, e questo è buono perché i ragazzi possono abituarsi da subito ad utilizzare correttamente i tipi. L'altro vantaggio positivo è che è possibile copia e incollare codice, ovvero veri e propri "pezzi" di flowchart selezionati, tra diversi file. Questa non è una caratteristica che si può dare per scontata con gli strumenti di apprendimento: anche linguaggi famosi come Scratch non ne offrono la possibilità, ma che nell'ottica di un progetto con una classe è uno molto importante. Infatti, avere la possibilità di fare copia e incolla, permette di poter assegnare dei progettini a gruppi di studenti e "fonderli" in progetti più ampi. Per quello che chiediamo di fare in questa learning unit si potrebbe ad esempio assegnare un canto dell'Odissea ad ogni gruppo di loro e mettendosi d'accordo sui finali, si potrebbe fare un unico grande gioco. Oppure si potrebbe anche creare un libro in cui la prima decisione serve semplicemente per scegliere con quale delle storie dei ragazzi si vuole giocare.

### 3.2 Limitazioni

Probabilmente a causa del fatto che i diagrammi di flusso con Flowgorithm possono essere tradotti in una grandissima quantità di linguaggi di programmazione, sorgono necessariamente alcune limitazioni:

#### Limitazioni di Flowgorithm

- Non c'è la possibilità di utilizzare variabili globali
- La lunghezza degli array non è variabile
- Solo gli array vengono passati alle funzioni per referenza, le altre variabili per valore
- I tipi che possono essere utilizzati sono limitati intero, reale, stringa, booleano o vettore di uno di questi. Una struttura come dizionari o insiemi sarebbe potuta essere comoda
- Flowgorithm non gestisce i files
- Non è possibile inserire caratteri speciali all'interno delle stringhe, ciò implica che non è possibile inserire delle ASCII Art all'interno del gioco
- Non ci sono blocchi per aggiungere attesa tra un'istruzione e l'altra.

Queste limitazioni in alcuni casi rendono meno pratico lo sviluppo del software, in altri invece limitano la creatività degli studenti.

## La lezione

*Flow your story* è pensata per essere svolta in un periodo di tempo non breve, poiché gli studenti dovranno inizialmente leggere un libro, creare una storia a bivi, e infine programmarla. Per poter istruire gli studenti sul come procedere basterà una sola lezione di introduzione dove parteciperanno sia il docente di lettere che quello di informatica, idealmente sarebbe sicuramente utile una lezione di "round-up" a metà tra inizio e consegna del progetto in cui gli studenti possono discutere con i professori riguardo l'avanzamento del proprio lavoro. Pensiamo quindi che questa attività vada svolta come compito per casa, possibilmente sfruttando un periodo come quello delle vacanze di natale in cui solitamente il docente di lettere assegna agli studenti un libro da leggere per casa, si potrebbe quindi fare la lezione di introduzione prima delle vacanze e il round up al rientro, dando poi una o due settimane di tempo ai ragazzi per correggere e terminare il progetto. Da notare che se un ragazzo non dispone di un computer in casa può tranquillamente svolgere su carta il compito disegnando il flowchart della propria storia.

La possibilità di copia e incollare pezzi di codice in Flowgorithm permette di poter far svolgere l'attività in piccoli gruppi di studenti da due o tre elementi: crediamo fortemente che lavorare in gruppo in *flow your story* possa essere un'esperienza sia più coinvolgente, sia maggiormente didattica, poiché non capita spesso durante le superiori di poter lavorare in gruppo. Se ad ogni gruppo sarà assegnato un libro diverso da cui partire si potrebbe anche incentivare la collaborazione tra gruppi.

### 4.1 Introduzione ai librigame e le avventure testuali

Durante la prima lezione ai ragazzi verrà fatta un introduzione storica, simile a quella presente in questa relazione, sulle avventure testuali. In questo modo anche chi non sa di cosa si sta parlando potrà essere coinvolto. Gli si spiegherà che dovranno scrivere questo tipo di avventure, e in particolare gli si chiederà di farlo basandosi su un'opera (o una parte di essa) assegnatagli dall'insegnante di lettere, cercando di mantenere la loro storia coerente con l'opera letteraria di base.

Si consiglierà vivamente di organizzare bene il lavoro, la struttura a bivi della storia e di non esagerare troppo con le biforazioni spiegandogli che all'aumentare delle biforazioni aumentano in maniera esponenziale i finali da gestire. Verrà infine mostrato e consegnato un progetto di demo da cui poter prendere spunto per strutturare il codice, mentre un progetto più complesso che darà loro idee su eventuali meccaniche più creative da utilizzare verrà solo mostrato a lezione.

## 4.2 Demo per gli studenti

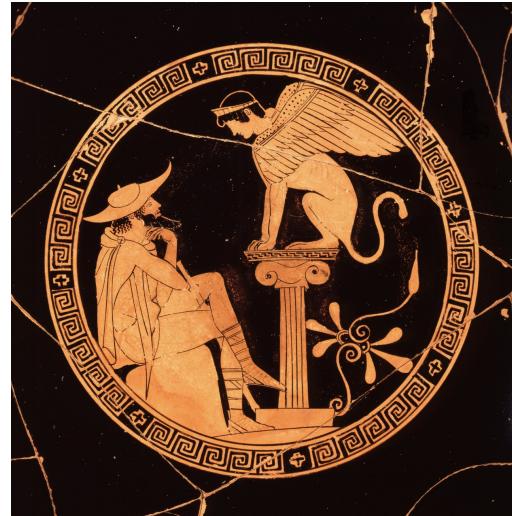
Abbiamo realizzato un mini-progetto da consegnare agli studenti, sia per stuzzicare la loro creatività riguardo la creazione della storia e del gameplay, sia per mostrare loro un possibile codice ben strutturato. Questo codice è quello che abbiamo consegnato di "Edipo e la Sfinge". In informatica è molto costruttivo utilizzare un codice già pronto, saperlo analizzare e prendere spunto dalle parti che si ritengono più interessanti. Anche un lavoro di banale copia e incolla non è una conoscenza che bisogna dare per scontata: se gli studenti imparano a selezionare in modo "critico" le parti di codice che gli interessano, imparano indirettamente a leggere il codice ed è davvero un buon lavoro da far fare. Per questo i ragazzi dovranno prima giocare con l'esempio e poi gli verrà spiegato il codice evidenziando gli stessi punti di cui parleremo qui di seguito. Un'altro esempio più complesso, "Ulisse e Polifemo", gli verrà mostrato e spiegato subito dopo, ma di questo non daremo il codice perché li indurrebbe a copiare troppo. Nelle seguenti sottosezioni vengono descritti in modo più approfondito i due progetti.

### 4.2.1 Edipo e la Sfinge

La storia è tratta dal mito greco di "Edipo e la Sfinge". Lo schema della storia è molto semplice e i meccanismi di gioco sono molto basilari, per i ragazzi è meglio confrontarsi con un diagramma di flusso più semplice all'inizio.

In questa demo il giocatore interpreta Edipo dal momento in cui scopre di esser stato adottato: il gioco pone diverse scelte al giocatore, alcune di queste porteranno ad un game over anticipato e alla fine verrà mostrato un punteggio che sarà tanto alto quanto le scelte del giocatore siano state fedeli alla storia originale.

In questo esempio l'interazione è basata esclusivamente sul rispondere a domande a risposta multipla ed è stato implementato un sistema di punteggio, qui di seguito abbiamo selezionato i concetti più interessanti da far notare e analizzare agli studenti.



### Cosa far notare nel codice di "Edipo e la Sfinge"

- La logica utilizzata per mostrare le opzioni e selezionarne una. Notare come possa essere utile usare funzioni per i meccanismi che si ripetono, come impostare e mostrare le diverse opzioni
- Come discernere su come continuare la storia, in funzione dell'ultima opzione selezionata
- Controllare il flusso della storia anche con dei valori booleani nel codice: è stato raggiunto un finale? E' stata fatta in precedenza una scelta piuttosto che un'altra? Infatti, ad esempio, il finale è diverso se Edipo va o no ad ascoltare la profezia, nonostante la storia sembri procedere nello stesso modo
- Come utilizzare un semplice sistema di punteggio
- La gestione dei testi, delle opzioni e dei finali della storia.

Questo esempio include alcune funzioni di comodo per gestire sia le domande a risposta multipla che quelle a risposta aperta. Non tutte queste funzioni sono utilizzate nella demo ma sono incluse per renderle disponibili agli studenti in modo così da potergliele far utilizzare nei loro progetti.

Tabella 1: Funzioni già pronte

Nome funzione	Descrizione
impostaDueOpzioni	Imposta e stampa due opzioni ad un quesito
impostaTreOpzioni <sup>4</sup>	Come la precedente ma con tre opzioni
tuttominuscolo	Data una stringa restituisce la stessa con tutti i caratteri in minuscolo
controllaRisposta	Dato un vettore di stringhe "opzioni", una stringa opzione e una stringa errore da stampare, controlla se l'opzione data è inclusa nel delle opzioni, in caso contrario stampa la frase d'errore
controllaRispostaAperta	Date due stringhe e una stringa d'errore, controlla se le prime due sono uguali, in caso contrario stampa la stringa d'errore
controllaRispostaIntero	Dato un intero legge un altro intero da tastiera, se l'intero inserito è minore o uguale a quello inserito la funzione termina, altrimenti stampa una frase d'errore e richiede il secondo intero in input.

<sup>4</sup>Le funzioni per impostare le opzioni sono diverse perché Flowgorithm non permette di variare la lunghezza di un vettore.

Edipo è un progetto base, che contiene tutta la storia in un unico diagramma di flusso: questa scelta è volutamente discutibile perché il diagramma risulta enorme e verrà spiegato agli studenti come la storia potrebbe essere "divisa" in funzioni con le varie parti poi composte insieme. Anche il fatto di assegnare i testi a delle variabili, in questo caso, potrebbe non essere comodo: la soluzione ottima resta usare variabili globali o leggere i testi da file, ma non offrendo Flowgorithm questa possibilità, se sono più comodi per orientarsi nel codice i ragazzi possono utilizzare direttamente le stringhe con i testi corretti, che non sarà una scelta penalizzata a meno che lo stesso testo non venga ripetuto più volte.

#### 4.2.2 Ulisse e Polifemo

Abbiamo realizzato un altro possibile compito che uno studente potrebbe consegnare come progetto di *flow your story*. La storia è tratta da uno dei più popolari canti dell'*Odissea*, quello in cui Ulisse sbarca nella terra dei ciclopi. A differenza della de-



mo, la storia non soltanto segue quella del poema ma ha anche un ramo completamente ideato da noi in cui la storia può finire con Ulisse che arruola Polifemo. Anche le meccaniche sono molte diverse, non c'è un punteggio ma ci sono domande a risposta aperta[8] con cui per procedere bisogna conoscere la risposta, loop e molte variabili per mantenere in memoria lo stato della partita. Essendo troppo grande il codice per venire mostrato in un unico diagramma di flusso, è stato diviso in funzioni che vengono chiamate dal diagramma di flusso eseguibile. Di questo progetto verrà mostrata l'esecuzione agli studenti che dovranno giocarci e vedere quali sono i meccanismi aggiuntivi rispetto ad Edipo, che schematicamente sono:

### Cosa far notare nell'esecuzione di "Ulisse e Polifemo"

- Alcune domande sono a risposta aperta, che bisogna conoscere per procedere nella storia. Questo serve per far dimostrare al giocatore le proprie conoscenze, come alternativa al sistema di punteggio
- Alcune delle risposte, come il nome che Ulisse dà a Polifemo, sul momento non cambiano il modo in cui procede la storia, ma possono influenzare il finale più avanti perché sono memorizzate
- I controlli sul procedere della storia possono essere effettuati su diversi tipi di dati: stringhe (il nome di Ulisse), gli interi (il numero di marinai in vita), booleani (Ulisse ha preso il vino), e così via
- Alcuni bivi possono essere ripetuti più volte ciclicamente e, tramite un contatore, solo ad un certo punto c'è una svolta della situazione
- E' possibile raggiungere un altro ramo della storia completamente differente dal primo: in questo caso è ancora più importante pensare a come gestire i finali con un diagramma di flusso

Verrà poi mostrato il diagramma di Ulisse e Polifemo che, al contrario di quello di Edipo e la Sfinge, è molto scarso. Questo perché la storia è gestita con funzioni che ne rappresentano i capitoli. Non verranno forniti altri dettagli implementativi agli studenti, che dovranno trarre solo ispirazione per la storia.

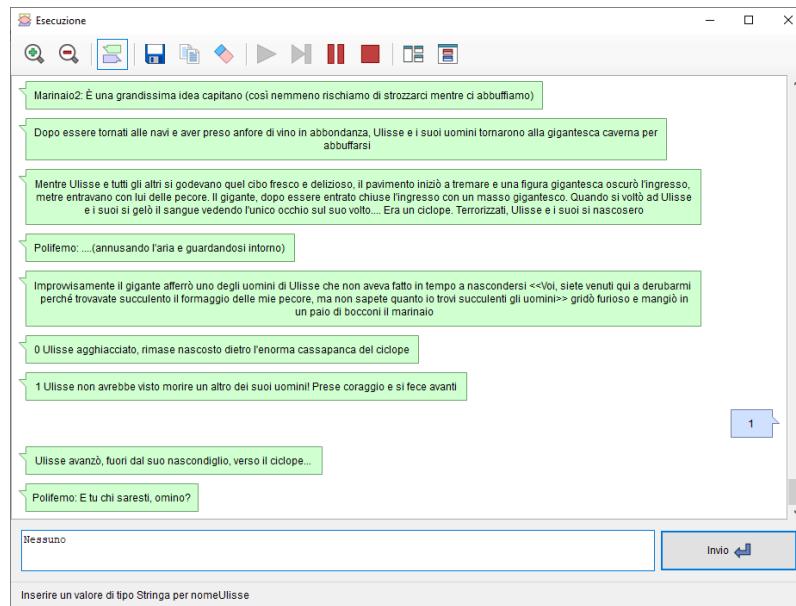
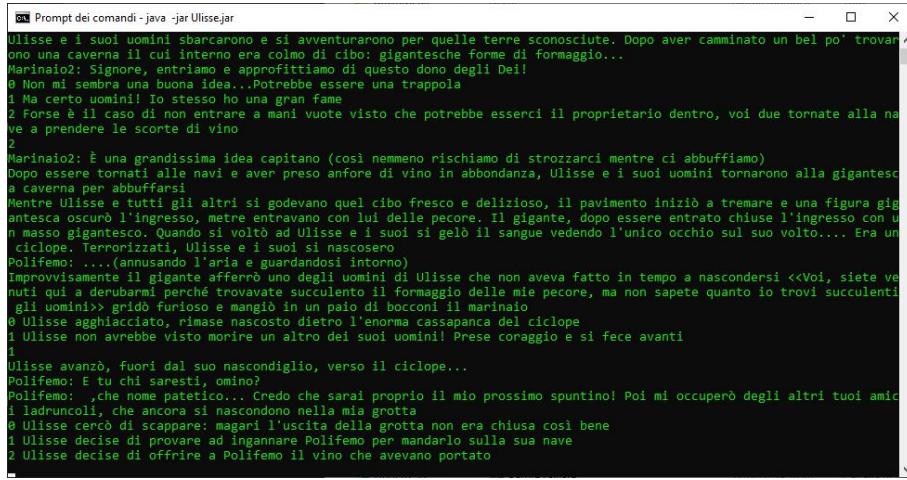


Figura 8: Qui il giocatore risponde apertamente ad una domanda

Di questa avventura testuale verrà fornito agli studenti anche un file jar per poter rigiocare a casa l'avventura. Il codice è stato prodotto con il tool di conversione di Flowgorithm e purtroppo dobbiamo segnalare che abbiamo dovuto correggere noi alcuni errori che non permettevano la compilazione del codice. Il jar è avviabile tramite console di comando, dal punto di vista della leggibilità rispetto al layout a "chat" di Flowgorithm abbiamo sicuramente un peggioramento ma consegnando un eseguibile abbiamo la sicurezza che gli studenti non copino il codice.



```

  Prompt dei comandi - java -jar Ulissejar
Ulisce e i suoi uomini sbarcarono e si avventurarono per quelle terre sconosciute. Dopo aver camminato un bel po' trovarono una caverna il cui interno era colmo di cibo: gigantesche forme di formaggio...
Marinaio2: Signore, entriamo e approfittiamo di questo dono degli Dei!
0 Non mi sembra una buona idea... Potrebbe essere una trappola
1 Ma certo uomini! Io stesso ho una gran fame
2 Forse è il caso di non entrare a mani vuote visto che potrebbe esserci il proprietario dentro, voi due tornate alla nave a prendere le scorte di vino
2
Marinaio2: È una grandissima idea capitano (così nemmeno rischiamo di strozzarci mentre ci abbuffiamo)
Dopo essere tornati alle navi e aver preso anfore di vino in abbondanza, Ulisse e i suoi uomini tornarono alla gigantesca caverna per abbuffarsi
Mentre Ulisse e tutti gli altri si godevano quel cibo fresco e delizioso, il pavimento iniziò a tremare e una figura gigantesca oscurò l'ingresso, metteva entravano con lui delle pecore. Il gigante, dopo essere entrato chiuse l'ingresso con un masso gigantesco. Quando si voltò ad Ulisse e i suoi si gelò il sangue vedendo l'unico occhio sul suo volto.... Era un ciclope. Terrorizzati, Ulisse e i suoi si nascosero
Polifemo: ....(annusando l'aria e guardandosi intorno)
Improvvisamente il gigante afferrò uno degli uomini di Ulisse che non aveva fatto in tempo a nascondersi <>Voi, siete venuti qui a derubarmi perché trovavate succulento il formaggio delle mie pecore, ma non sapete quanto io trovi succulenti gli uomini>
gridò furioso e mangiò in un paio di bocconi il marinai
0 Ulisse agghiacciato, rimase nascosto dietro l'enorme cassapanca del ciclope
1 Ulisse non avrebbe visto morire un altro dei suoi uomini! Prese coraggio e si fece avanti
1
Ulisse avanzò, fuori dal suo nascondiglio, verso il ciclope...
Polifemo: E tu chi saresti, omino?
Polifemo: ,che nome patetico... Credo che sarai proprio il mio prossimo spuntino! Poi mi occuperò degli altri tuoi amici laduncoli, che ancora si nascondono nella mia grotta
0 Ulisse cercò di scappare: magari l'uscita della grotta non era chiusa così bene
1 Ulisse decise di provare ad ingannare Polifemo per mandarlo sulla sua nave
2 Ulisse decise di offrire a Polifemo il vino che avevano portato
  
```

Figura 9: Ulisse e Polifemo dal prompt dei comandi di windows

### 4.3 Valutazione

Per quanto riguarda la valutazione ci sono due livelli da affrontare: l'ambito interdisciplinare, quindi letterario, e l'ambito informatico. Per una questione di competenza è giusto che la valutazione dell'ambito letterario sia affidato al professore di lettere che valuterà il lavoro come più ritiene opportuno. In questo modo poi, il lavoro che è impegnativo per i ragazzi farà ottenere loro sue valutazioni in due materie differenti.

A livello informatico volevamo valutare il lavoro dei ragazzi su più livelli: abbiamo ideato un sistema a punteggio a obiettivi i quali saranno suddivisi in principali e secondari, il punteggio massimo è di 100. Lo studente per raggiungere la sufficienza (60 punti) dovrà raggiungere tutti gli obiettivi principali, nel caso non ci riesca non verranno considerati i punti derivanti dagli obiettivi bonus, qui di seguito gli obiettivi principali:

### Obiettivi principali

I requisiti minimi da rispettare per ritenere il progetto concluso sono creare una storia con almeno:

- Quattro bivi decisionali ad opzioni
- Due finali principali differenti
- Due finali gameover anticipati
- Due variabili che il gioco raccoglie e utilizza in diversi momenti del gioco.
- Un controllo con dei valori booleani

Per quanto riguarda i punti bonus, dalla tabella 2 si può notare che ci sono sessanta punti bonus divisi in nove obiettivi secondari. Se sommati a quelli della sufficienza superano di venti il punteggio totale; questo è per permettere agli studenti di poter raggiungere il punteggio pieno in diverse maniere e senza sconvolgere troppo la loro storia solo per inseguire la piena valutazione. Gli obiettivi secondari che abbiamo scelto servono sia a vedere se lo studente ha manualità con i concetti base informatici, sia per spingerlo a organizzare bene il proprio codice.

Tabella 2: Valutazione obiettivi secondari

Obiettivo	Punti
Aggiungere altri due bivi	5
Raddoppiare i finali gameover	5
Inserire una domanda a risposta aperta	5
Aumentare a quattro le variabili che il gioco raccoglie e utilizza successivamente	5
Chiedere di inserire delle stringhe che poi verranno utilizzate nei testi della storia	5
Dividere i pezzi di storia in funzioni	5
Inserire un ciclo	5
Inserire un sistema di punteggio	10
Aggiungere un ulteriore finale principale	15

## Altre possibilità per riadattare la lezione

Creare storie con bivi è un'attività molto versatile dal punto di vista didattico poiché è possibile creare storie basandosi su molte altre materie scolastiche. Un esempio potrebbe essere quello di creare storie ambientate in determinati periodi storici, un po' come la Ubisoft<sup>5</sup> sta facendo con la popolare serie di videogiochi *Assassin's creed* i cui episodi sono ambientati in diversi periodi storici. Magari potrebbe essere un obiettivo didattico riuscire ad inserire almeno un certo numero di episodi storici studiati nel programma scolastico.



Figura 10: Alcuni giochi della serie "Assassin's creed"

Altre materie possono essere anche le lingue straniere, scrivendo in un'altra lingua o simulando cose come conversazioni telefoniche o altre tipologie di interazioni tipiche. Dal punto di vista informatico l'attività è facilmente riadattabile per classi con un livello di competenze informatiche più avanzato, utilizzando linguaggi o tool che permettono la creazione di strutture dati più complesse (oggetti, struct ecc...). In questo modo sarebbe possibile ricreare l'avventura modellando le parti del racconto come fossero nodi di un grafo, ottenendo quindi un codice molto più gestibile e mantenibile. Se la classe si dimostra particolarmente brava si potrebbe anche provare a far sviluppare l'avventura sotto forma di chat bot: alcune piattaforme di messaggistica istantanea come Telegram offrono gratuitamente la possibilità di poterne creare ed esistono molti framework che facilitano lo sviluppo di chat bot in tantissimi linguaggi di programmazione. Un'altra idea è far "ristrutturare" il progetto iniziato con Flowgorithm mano a mano che si va avanti con il programma.

<sup>5</sup>Azienda francese sviluppatrice e editrice di videogiochi.