

Master Semipresenziale

Programmazione di Videogiochi





tech università
tecnologica

Master Semipresenziale Programmazione di Videogiochi

Modalità: Semipresenziale (Online + Tirocinio)

Durata: 12 mesi

Titolo: TECH Università Tecnologica

Accesso al sito web: www.techtute.com/it/videogiochi/master-semipresenziale/master-semipresenziale-programmazione-videogiochi

Indice

01

Presentazione

pag. 4

02

Perché iscriversi a questo
Master Semipresenziale?

pag. 8

03

Obiettivi

pag. 12

04

Competenze

pag. 16

05

Struttura e contenuti

pag. 20

06

Tirocinio

pag. 34

07

Dove posso svolgere
Il tirocinio?

pag. 40

08

Metodologia di studio

pag. 44

09

Titolo

pag. 54

01 Presentazione

Il settore della creazione di videogiochi è ormai consolidato e attrae sempre più giovani. Uno dei pilastri che sostengono un grande titolo nel settore dei videogiochi è la programmazione. Il processo che crea le istruzioni di base e ne detta il funzionamento generale è essenziale per far funzionare perfettamente la storia e lo sviluppo del gioco. Questa qualifica in videogiochi offre ai professionisti del gioco una specializzazione che gli permetterà di avanzare nella loro carriera nel settore del *Gaming*. Una qualifica che si svolgerà al 100% online, adattata alle esigenze e che ha un tirocinio in uno studio di creazione e sviluppo di videogiochi. Tutto ciò permetterà di avanzare nella carriera professionale insieme ai migliori esperti.





“

Diventa uno dei migliori programmatori del settore dei videogiochi con questo Master Semipresenziale"

L'industria dei videogiochi ha un grande potenziale. L'aumento della domanda e le esigenze dei *Gamers* stessi ha portato questo settore in una corsa per la perfezione nei loro titoli. L'alto livello di qualità in ciascuna delle creazioni porta dietro a un gruppo di professionisti della programmazione che possiedono una qualifica eccellente.

Questo Master Semipresenziale in Programmazione di Videogiochi risponde alle esigenze attuali del mercato, che richiede professionisti sempre più specializzati e con un alto grado di coinvolgimento nelle creazioni. La creatività gioca un ruolo importante, ma senza una solida conoscenza non sarebbe possibile ottenere videogiochi di alto livello.

Per questo motivo, questo titolo universitario offre agli studenti una conoscenza approfondita dei fondamenti della programmazione e dell'ingegneria del software, approfondisce la struttura dei dati e l'algoritmo, oltre a insegnare la programmazione orientata agli oggetti e le specifiche dei motori. Inoltre, questo programma affronta la programmazione in tempo reale per offrire ai professionisti dei videogiochi un Master Semipresenziale completo.

Per raggiungere l'obiettivo di progresso nella carriera professionale della programmazione di videogiochi, gli studenti avranno un personale docente esperto in questo settore, che li guiderà e li istruirà in ogni momento. Inoltre, il contenuto interattivo con video riassunti, casi pratici e letture aggiuntive completeranno l'ampio curriculum messo a disposizione da TECH in questo titolo 100% online e con tirocinio in azienda.

Così, gli studenti potranno completare questo percorso accademico con un soggiorno presenziale e intensivo di 3 settimane in uno studio all'avanguardia nella programmazione di videogiochi. Un ambiente professionale ideale, dove potrai verificare in loco i metodi di lavoro, i programmi e le tecniche più sofisticate per la creazione di titoli di alta qualità. Un'opportunità unica che solo TECH, la più grande università digitale del mondo, può offrire.

Questo **Master Semipresenziale in Programmazione di Videogiochi** possiede il programma più completo e aggiornato del mercato. Le caratteristiche principali del programma sono:

- ♦ Sviluppo di oltre 100 casi di Programmazione di Videogiochi presentati da professionisti di programmazione della e professori universitari di una vasta esperienza nell'industria dei videogiochi
- ♦ Contenuti grafici, schematici ed eminentemente pratici che forniscono informazioni scientifiche e pratiche sulle discipline essenziali per l'esercizio della professione
- ♦ Esercizi pratici che offrono un processo di autovalutazione per migliorare l'apprendimento
- ♦ Particolare enfasi sulle metodologie innovative
- ♦ Lezioni teoriche, domande all'esperto, forum di discussione su questioni controverse e compiti di riflessione individuale
- ♦ Disponibilità di accesso ai contenuti da qualsiasi dispositivo fisso o portatile dotato di connessione a Internet
- ♦ Lezioni teoriche, domande all'esperto, forum di discussione su questioni controverse e compiti di riflessione individuale
- ♦ Contenuti disponibili da qualsiasi dispositivo fisso o mobile dotato di connessione a internet
- ♦ Possibilità di svolgere un tirocinio clinico all'interno di uno dei migliori centri ospedalieri



Padroneggia la tecnica del Backtracking e crea i migliori videogiochi d'avventura con questo Master Semipresenziale"

“

Questo programma 100% online ti dà l'opportunità di fare un tirocinio in uno studio e metterti alla prova con i migliori programmatori"

In questa proposta di Master, di carattere professionale e modalità semipresenziale, il programma è diretto all'aggiornamento dei professionisti dei videogiochi che svolgono le loro funzioni in grandi studi creativi e che richiedono un alto livello di qualificazione. I contenuti sono basati sulle più recenti prove scientifiche, e orientati in modo didattico per integrare le conoscenze teoriche nella pratica del settore *gaming*, e gli elementi teorico-pratici faciliteranno l'aggiornamento delle conoscenze e consentiranno il processo decisionale nella Programmazione di Videogiochi.

Grazie al contenuto multimediale del programma realizzato con le più recenti tecnologie educative, permetteranno al professionista dei videogiochi un apprendimento localizzato e contestuale, cioè un ambiente simulato che fornirà un apprendimento immersivo programmato per allenarsi in situazioni reali. La creazione di questo programma è incentrata sull'Apprendimento Basato su Problemi, mediante il quale lo studente deve cercare di risolvere le diverse situazioni che gli si presentano durante il corso. Lo studente potrà usufruire di un innovativo sistema di video interattivi creati da esperti di rinomata fama.

Questo Master Semipresenziale ti permetterà di essere un punto di riferimento nel settore della programmazione di videogiochi. Iscriviti subito.

Sviluppa applicazioni in modo efficiente applicate ai motori di videogiochi grazie a questo Master Semipresenziale.

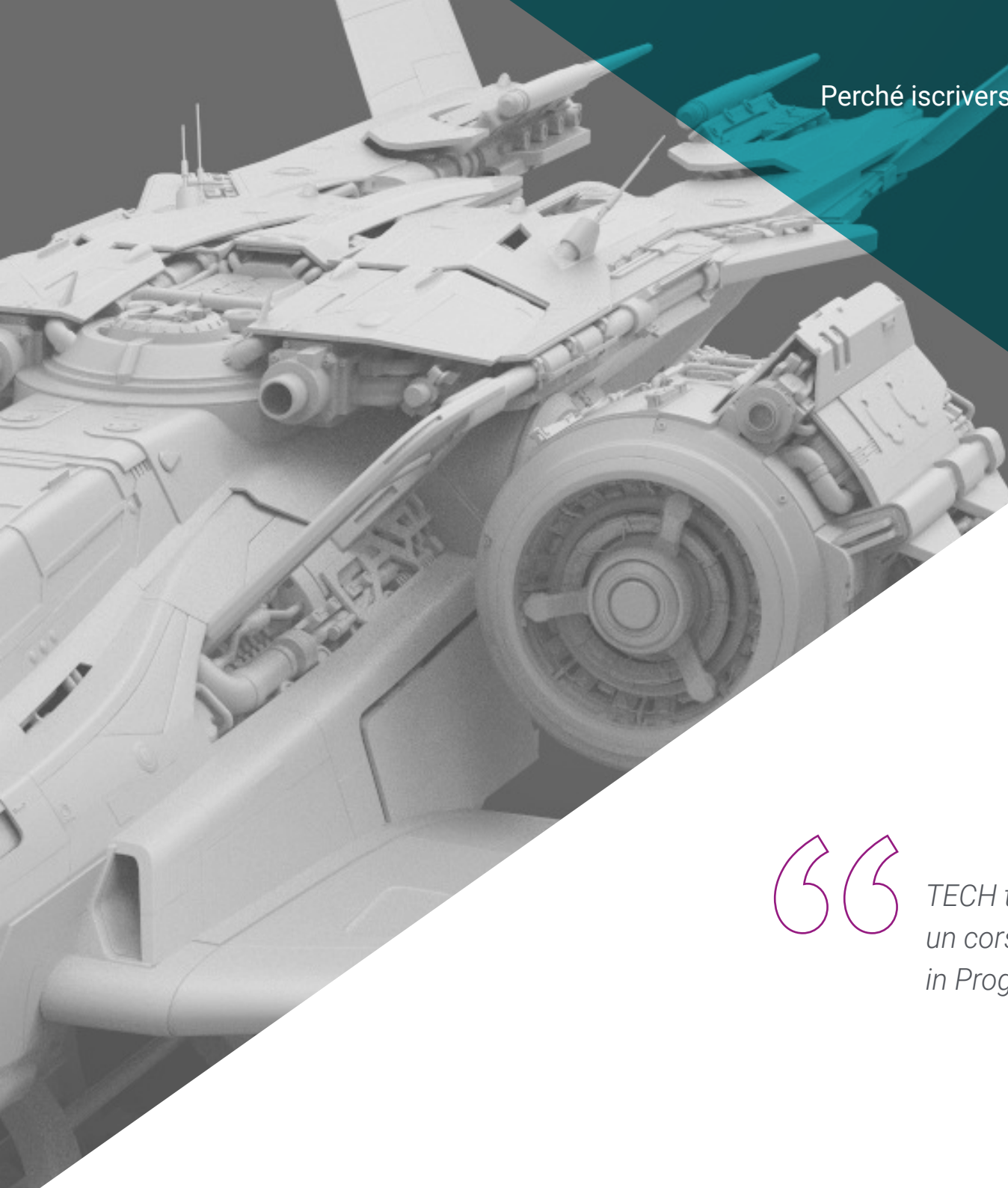


02

Perché iscriversi a questo Master Semipresenziale?

La programmazione richiede una solida conoscenza teorica, ma senza dubbio, la pratica quotidiana è ciò che perfeziona la padronanza del professionista nel settore dei videogiochi. Ecco perché TECH ha creato questo titolo, che combina perfettamente il più avanzato programma online in aree come la progettazione di algoritmi, sistemi intelligenti e programmazione in tempo reale con un tirocinio pratico in uno studio di programmazione all'avanguardia. In questo modo, gli studenti avranno una visione molto più completa della Programmazione di Videogiochi, dei programmi utilizzati e delle tecniche più recenti. Tutto questo, sempre guidato dai migliori esperti in materia.





“

TECH ti dà l'opportunità di frequentare un corso pratico efficace e intensivo in Programmazione di Videogiochi"

1. Aggiornarsi sulla base delle più recenti tecnologie disponibili

La tecnologia ha rivoluzionato negli ultimi anni l'area della Programmazione di Videogiochi, favorendo la creazione di titoli di maggiore qualità e realismo. Per questo, e al fine di avvicinare gli studenti a questa tecnologia, TECH ha creato questo Master Semipresenziale, dove il professionista si immergerà nei motori dei videogiochi, le sfide tecnologiche attuali e nei software per la creazione di ontologie. In questo modo si otterrà un aggiornamento sulla più recente tecnologia disponibile.

2. Approfondire nuove competenze dall'esperienza dei migliori specialisti

In questo percorso accademico, gli studenti saranno sempre guidati dai migliori specialisti in Programmazione di Videogiochi. Così, lo studente avrà un eccellente gruppo di insegnanti con una vasta esperienza nel settore, che lo accompagnerà durante la fase teorica, mentre, durante il tirocinio presenziale, sarà accanto a veri esperti, che fanno parte del team dello studio dove si svolgerà la fase pratica.

3. Entrare in ambienti clinici di prim'ordine

TECH svolge un rigoroso processo di selezione di tutti gli studi e le aziende in cui vengono effettuati i tirocini. In questo modo, gli studenti hanno accesso garantito a un ambiente professionale di primo livello nella Programmazione di Videogiochi. In questo modo, possono verificare di prima mano come è la vita quotidiana di un programmatore specializzato, le tecniche e i metodi impiegati per ottenere titoli di qualità.





4. Combinare la migliore teoria con la pratica più avanzata

TECH con questo titolo si adatta al lavoro quotidiano dei programmatori ed è per questo che adotta l'approccio teorico-pratico, lontano, inoltre, dalle lunghe ore di studio, per concentrarsi sui concetti chiave. Questo istituto accademico offre un modello di apprendimento innovativo che consente agli studenti di acquisire le competenze necessarie per essere all'avanguardia nella programmazione di videogiochi di alta qualità.

5. Ampliare le frontiere della conoscenza

TECH offre la possibilità di svolgere il Tirocinio non solo in centri di importanza nazionale, ma anche internazionale. In questo modo, lo specialista potrà allargare le proprie frontiere e confrontarsi con i migliori professionisti, che esercitano in ospedali di prim'ordine e in diversi continenti. Un'opportunità unica che solo TECH, la più grande università digitale del mondo, poteva offrire.

“ Avrai l'opportunità svolgere il tuo tirocinio all'interno di un centro di tua scelta”

03

Obiettivi

La progettazione del programma di questo Master Semipresenziale permetterà agli studenti di acquisire le competenze necessarie per avanzare in una programmazione altamente competitiva e che richiede personale qualificato. Quindi, l'obiettivo principale di questo insegnamento è quello di rendere lo studente un eccellente sviluppatore di videogiochi a partire da tutti gli strumenti che questo insegnamento fornisce. Per raggiungere questo obiettivo, questo titolo universitario fornisce contenuti multimediali aggiornati con letture aggiuntive e un sistema di apprendimento *Relearning*, basato sulla ripetizione dei contenuti, che faciliterà la fondazione dei concetti.





“

Vuoi lanciare un gioco multigiocatore? Impara a trasformare la tua idea in un vero linguaggio di programmazione per questo tipo di videogiochi”



Obiettivo generale

- Far conoscere agli studenti i diversi linguaggi e metodi di programmazione applicati al videogioco. Per questo si approfondirà il processo di produzione di un titolo e la sua integrazione nelle diverse fasi. Il professionista dei videogiochi imparerà i fondamenti della progettazione di videogiochi e le principali conoscenze teoriche. Inoltre, al termine di questo corso sarà in grado di comprendere il ruolo della programmazione e sviluppare videogiochi web e multigiocatore



Questo programma ti permetterà di avanzare professionalmente. Sarai in grado di padroneggiare qualsiasi struttura di dati e algoritmi con questo Master Semipresenziale"



Obiettivi specifici

Modulo 1. Fondamenti di programmazione

- Comprendere la struttura di base di un computer, il software e i linguaggi di programmazione di uso generale
- Analizzare gli elementi essenziali di un programma informatico, come i diversi tipi di dati, gli operatori, le espressioni, le dichiarazioni, le istruzioni di I/O e di controllo
- Interpretare gli algoritmi, che sono la base necessaria per lo sviluppo del software

Modulo 2. Struttura dei dati e algoritmi

- Imparare le principali strategie per la progettazione di algoritmi, nonché i diversi metodi e misure per il loro calcolo
- Distinguere il funzionamento degli algoritmi, la loro strategia ed esempi del loro utilizzo nei principali problemi noti
- Comprendere la tecnica di *Backtracking* e i suoi principali utilizzi

Modulo 3. Programmazione orientata agli oggetti

- Conoscere i diversi modelli di progettazione per i problemi orientati agli oggetti
- Comprendere l'importanza della documentazione e dei test nello sviluppo del software
- Gestire l'uso dei thread e della sincronizzazione, nonché la risoluzione di problemi comuni nell'ambito della programmazione concorrente

Modulo 4. Console e dispositivi per videogiochi

- ♦ Conoscere il funzionamento di base delle principali periferiche di input e output
- ♦ Comprendere le principali esigenze progettuali delle diverse piattaforme
- ♦ Studiare la struttura, l'organizzazione, il funzionamento e l'interconnessione di dispositivi e sistemi
- ♦ Comprendere il ruolo del sistema operativo e dei kit di sviluppo per i dispositivi mobili e le piattaforme videoludiche

Modulo 5. Ingegneria del software

- ♦ Distinguere le basi dell'ingegneria del software, nonché il suo processo e i diversi modelli di sviluppo che comprendono le tecnologie agili
- ♦ Riconoscere l'ingegneria dei requisiti, il suo sviluppo, l'elaborazione, la negoziazione e la convalida per comprendere i principali standard relativi alla qualità del software e alla gestione dei progetti

Modulo 6. Motori grafici per videogiochi

- ♦ Scoprire il funzionamento e la struttura di un motore grafico per videogiochi
- ♦ Comprendere le caratteristiche di base dei motori di gioco esistenti
- ♦ Programmare applicazioni in modo corretto ed efficiente applicate ai motori di videogiochi
- ♦ Scegliere il paradigma e i linguaggi di programmazione più appropriati per la programmazione di applicazioni applicate ai motori dei videogiochi

Modulo 7. Sistemi intelligenti

- ♦ Stabilire i concetti relativi alla teoria e all'architettura dei vari elementi e il loro processo di ragionamento
- ♦ Assimilare la teoria e la pratica dei concetti di informazione e conoscenza, nonché i diversi modi di rappresentare la conoscenza
- ♦ Comprendere il funzionamento dei ragionatori semantici, dei sistemi basati sulla conoscenza e dei sistemi esperti

Modulo 8. Programmazione in tempo reale

- ♦ Analizzare le caratteristiche principali di un linguaggio di programmazione in tempo reale che lo differenziano da un linguaggio di programmazione tradizionale
- ♦ Comprendere i concetti di base dei sistemi informatici
- ♦ Acquisire la capacità di applicare i principali fondamenti e le tecniche di programmazione in tempo reale

Modulo 9. Design e sviluppo di giochi online

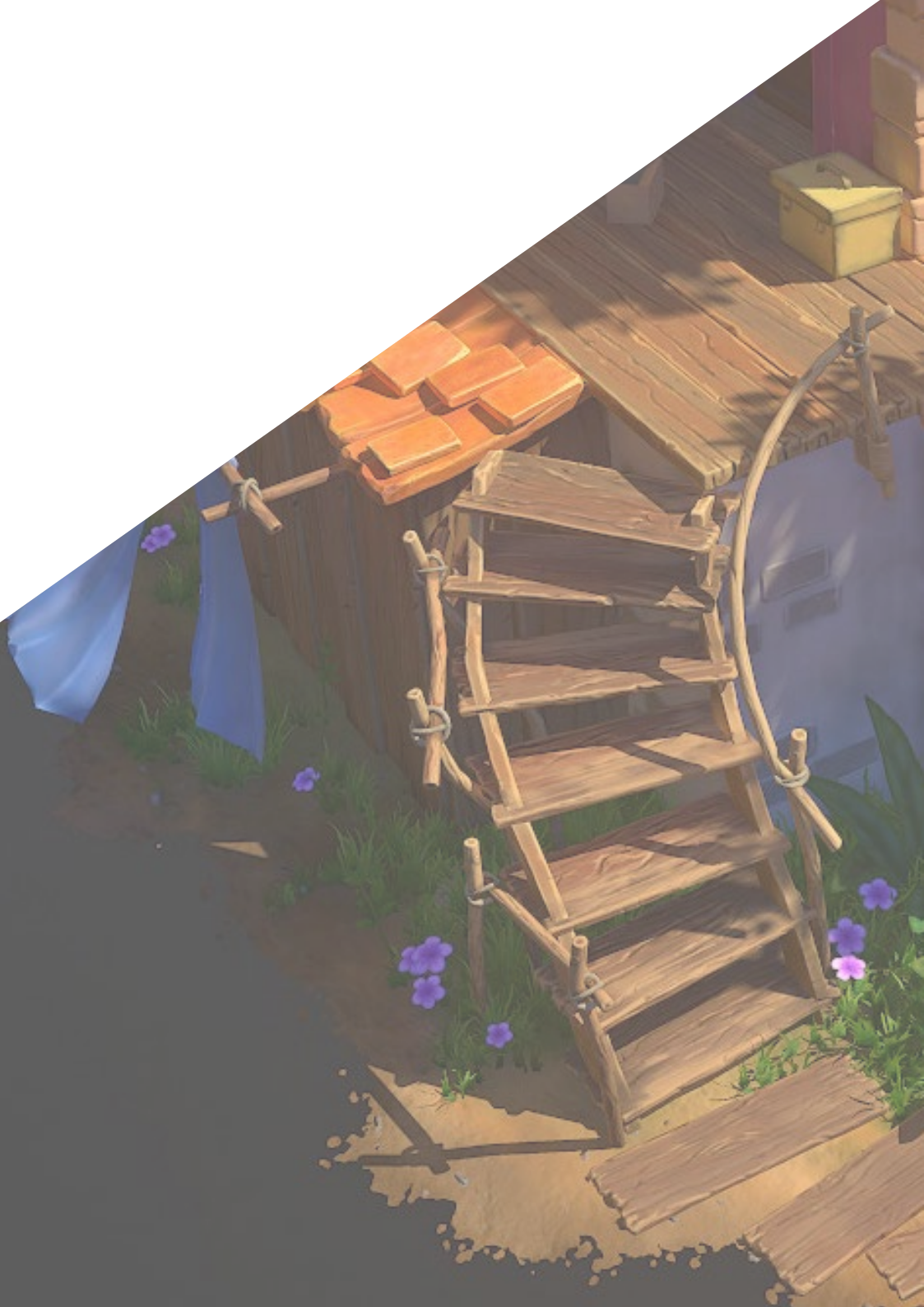
- ♦ Essere in grado di progettare giochi e applicazioni online interattive con la relativa documentazione
- ♦ Valutare le caratteristiche principali di giochi e applicazioni online interattive per comunicare in modo professionale e corretto

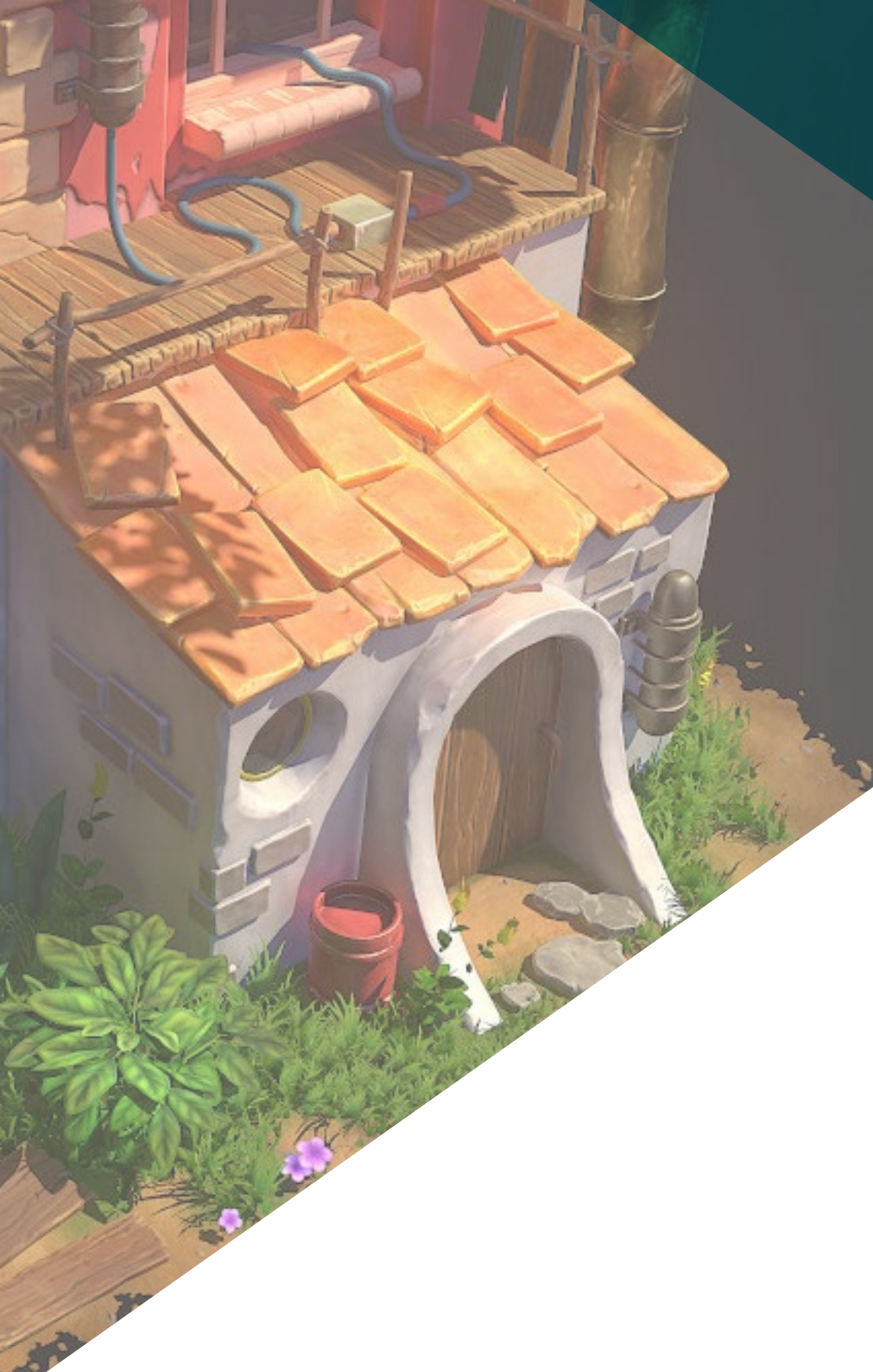
Modulo 10. Reti e sistemi multiplayer

- ♦ Descrivere l'architettura del protocollo di controllo della trasmissione/protocollo Internet (TCP/IP) e il funzionamento di base delle reti wireless
- ♦ Analizzare la sicurezza applicata ai videogiochi
- ♦ Acquisire la capacità di sviluppare giochi online multiplayer

04 Competenze

Il professionista del videogioco che studia questo Master Semipresenziale acquisirà una serie di competenze che gli permetteranno di entrare in qualsiasi tipo di studio sviluppatore di videogiochi. Così, gli studenti al termine di questo corso avranno le competenze necessarie per programmare in diversi linguaggi utilizzati nel settore del *gaming* e acquisiranno le competenze essenziali per eseguire un progetto dall'inizio alla fine su diverse piattaforme e motori di videogiochi.





“

*Acquisisci le competenze necessarie
per programmare videogiochi per
Xbox One, PS4 e Wii U-Switch”*



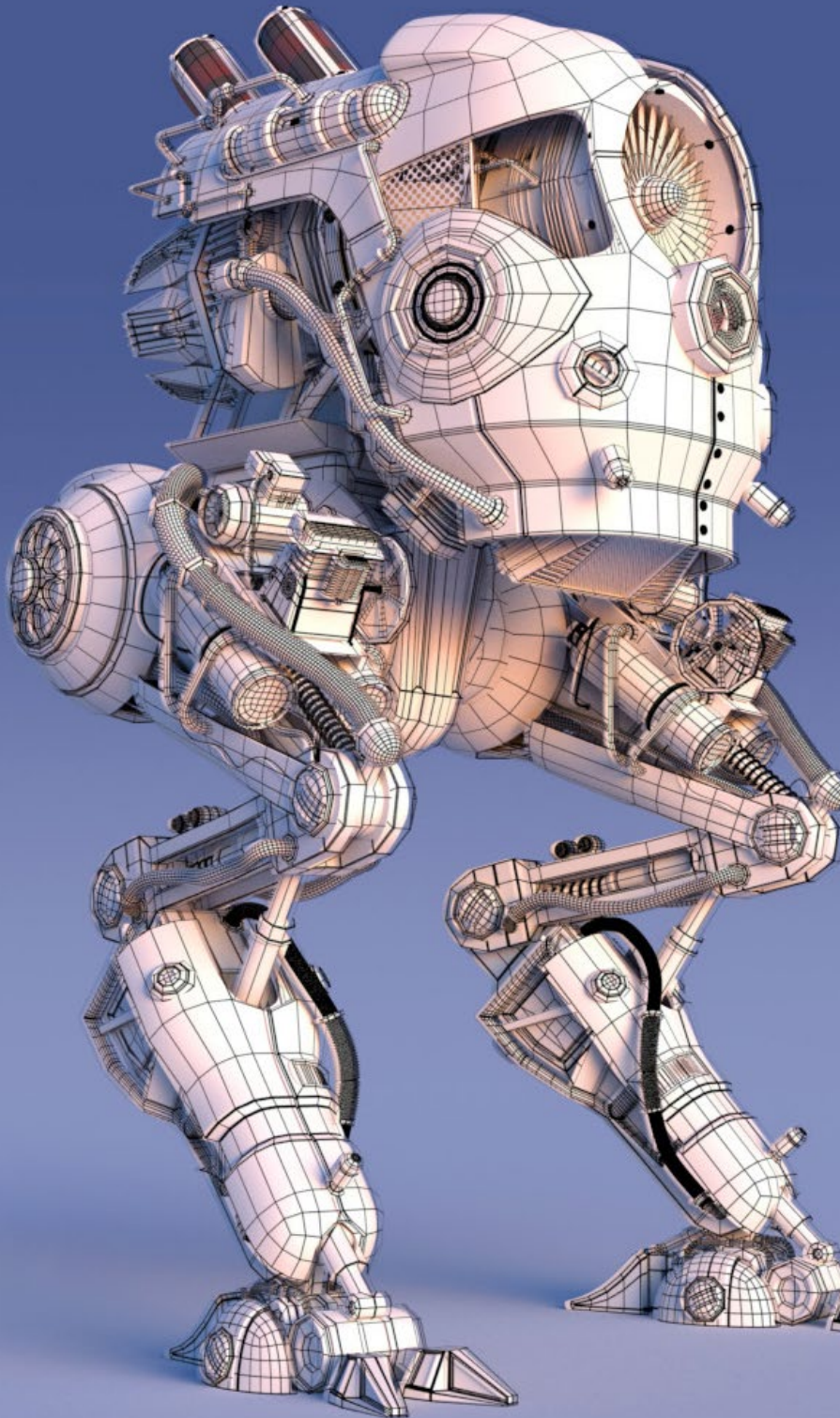
Competenze generali

- ◆ Progettare tutte le fasi di un videogioco, dall'idea iniziale al lancio sul mercato
- ◆ Specializzarsi come programmatore di videogiochi
- ◆ Approfondire tutte le parti dello sviluppo, a partire dall'architettura iniziale, la programmazione del personaggio del giocatore e tutti gli elementi coinvolti nel processo di gioco
- ◆ Ottenere una visione complessiva del progetto, potendo fornire soluzioni ai diversi problemi e difficoltà che si presentano nella progettazione di un videogioco

“

Prima di programmare, conosci l'esperienza del giocatore e analizza la giocabilità del videogioco. Con questo Master Semipresenziale imparerai a raggiungere il successo”





Competenze specifiche

- ◆ Conoscere il software necessario per essere uno sviluppatore di videogiochi professionista
- ◆ Capire l'esperienza utente e saper analizzare il gameplay
- ◆ Comprendere l'intera procedura teorica e pratica del processo di programmazione dei videogiochi
- ◆ Padroneggiare i linguaggi di programmazione più utili per il mondo dei videogiochi
- ◆ Integrare la programmazione appresa con diversi tipi di console e piattaforme
- ◆ Programmare videogiochi online e multiplayer
- ◆ Assimilare il concetto di motore videoludico per poter programmare correttamente
- ◆ Applicare le conoscenze di ingegneria del software alla programmazione di videogiochi

05

Struttura e contenuti

Questo Master Semipresenziale in Programmazione di Videogiochi offre agli studenti un contenuto completo e ampio nello sviluppo di videogiochi strutturato in 10 moduli specializzati. Il personale docente che integra questo titolo universitario potrà approfondire i fondamenti della programmazione e acquisire allo stesso tempo una conoscenza aggiornata e orientata alla pratica. In questo modo, gli studenti ricevono un'istruzione di qualità, con contenuti innovativi in formato multimediale e scaricabile per essere consultati in qualsiasi momento. La formazione teorica 100% online dà la libertà di distribuire il carico didattico in base alle proprie esigenze.





“

*Un piano di studi progettato per offrirti
il contenuto più avanzato e aggiornato
in Programmazione di Videogiochi”*

Modulo 1. Fondamenti di programmazione

- 1.1 Introduzione alla programmazione
 - 1.1.1. Struttura di base di un computer
 - 1.1.2. Software
 - 1.1.3. Linguaggio di programmazione
 - 1.1.4. Ciclo di vita di un'applicazione informatica
- 1.2. Progettazione di algoritmi
 - 1.2.1. Risoluzione dei problemi
 - 1.2.2. Tecniche descrittive
 - 1.2.3. Elementi e struttura di un algoritmo
- 1.3. Elementi di un programma
 - 1.3.1. Origini e caratteristiche del linguaggio C++
 - 1.3.2. L'ambiente di sviluppo
 - 1.3.3. Concetto di programma
 - 1.3.4. Tipi di dati fondamentali
 - 1.3.5. Operatori
 - 1.3.6. Espressioni
 - 1.3.7. Frasi
 - 1.3.8. Input e output di dati
- 1.4. Sentenze di controllo
 - 1.4.1. Frasi
 - 1.4.2. Biforcazioni
 - 1.4.3. Loop
- 1.5. Astrazione e modularità: funzioni
 - 1.5.1. Design modulare
 - 1.5.2. Concetto di funzione e utilità
 - 1.5.3. Definizione di una funzione
 - 1.5.4. Flusso di esecuzione in una chiamata di funzione
 - 1.5.5. Prototipo di una funzione
 - 1.5.6. Restituzione dei risultati
 - 1.5.7. Chiamata di una funzione: parametri
 - 1.5.8. Passaggio di parametri per riferimento e per valore
 - 1.5.9. Ambito identificatore
- 1.6. Strutture dati statiche
 - 1.6.1. *Array*
 - 1.6.2. Matrici: Poliedri
 - 1.6.3. Ricerca e ordinamento
 - 1.6.4. Catene: Funzioni di I/O per le stringhe
 - 1.6.5. Strutture: Unioni
 - 1.6.6. Nuovi tipi di dati
- 1.7. Strutture dati dinamiche: puntatori
 - 1.7.1. Concetto. Definizione di puntatore
 - 1.7.2. Operatori e operazioni con i puntatori
 - 1.7.3. *Array* di puntatori
 - 1.7.4. Puntatori e *array*
 - 1.7.5. Puntatori a stringhe
 - 1.7.6. Puntatori a strutture
 - 1.7.7. Indirizzi multipli
 - 1.7.8. Puntatori a funzioni
 - 1.7.9. Passaggio di funzioni, strutture e *array* come parametri di funzione
- 1.8. File
 - 1.8.1. Concetti di base
 - 1.8.2. Operazioni con i file
 - 1.8.3. Tipi di file
 - 1.8.4. Organizzazione dei file
 - 1.8.5. Introduzione ai file C++
 - 1.8.6. Gestione dei file
- 1.9. Ricorsività
 - 1.9.1. Definizione di ricorsività
 - 1.9.2. Tipi di ricorsione
 - 1.9.3. Vantaggi e svantaggi
 - 1.9.4. Considerazioni
 - 1.9.5. Conversione ricorsiva-iterativa
 - 1.9.6. Lo stack di ricorsione



- 1.10. Test e documentazione
 - 1.10.1. Test del programma
 - 1.10.2. Test della scatola bianca
 - 1.10.3. Test della scatola nera
 - 1.10.4. Strumenti per realizzare i test
 - 1.10.5. Documentazione del programma

Modulo 2. Struttura dei dati e degli algoritmi

- 2.1. Introduzione ai modelli di progettazione di algoritmi
 - 2.1.1. Ricorsività
 - 2.1.2. Dividi e conquista
 - 2.1.3. Altre strategie
- 2.2. Efficienza e analisi degli algoritmi
 - 2.2.1. Misure di efficienza
 - 2.2.2. Misurare l'ingresso di input
 - 2.2.3. Misurare il tempo di esecuzione
 - 2.2.4. Caso peggiore, migliore e medio
 - 2.2.5. Notazione asintotica
 - 2.2.6. Criteri di analisi matematica per algoritmi non ricorsivi
 - 2.2.7. Analisi matematica per algoritmi ricorsivi
 - 2.2.8. Analisi empirica degli algoritmi
- 2.3. Algoritmi di ordinamento
 - 2.3.1. Concetto di ordinamento
 - 2.3.2. Ordinamento delle bolle
 - 2.3.3. Ordinamento per selezione
 - 2.3.4. Ordinamento per inserimento
 - 2.3.5. Ordinamento per miscela (Merge_Sort)
 - 2.3.6. Ordinamento rapido (Quick_Sort)

- 2.4. Algoritmi con alberi
 - 2.4.1. Concetto di albero
 - 2.4.2. Alberi binari
 - 2.4.3. Percorsi degli alberi
 - 2.4.4. Rappresentare le espressioni
 - 2.4.5. Alberi binari ordinati
 - 2.4.6. Alberi binari bilanciati
- 2.5. Algoritmi con *heaps*
 - 2.5.1. Gli *heaps*
 - 2.5.2. L'algoritmo *heapsort*
 - 2.5.3. Code prioritarie
- 2.6. Algoritmi con grafi
 - 2.6.1. Rappresentazione
 - 2.6.2. Percorso in larghezza
 - 2.6.3. Percorso in profondità
 - 2.6.4. Ordinamento topologico
- 2.7. Algoritmi *greedy*
 - 2.7.1. La strategia *greedy*
 - 2.7.2. Elementi della strategia *greedy*
 - 2.7.3. Cambio valuta
 - 2.7.4. Il problema del viaggiatore
 - 2.7.5. Problema dello zaino
- 2.8. Ricerca del percorso minimo
 - 2.8.1. Il problema del percorso minimo
 - 2.8.2. Archi e cicli negativi
 - 2.8.3. Algoritmo di Dijkstra
- 2.9. Algoritmi *greedy* sui grafi
 - 2.9.1. L'albero a sovrapposizione minima
 - 2.9.2. Algoritmo di Prim
 - 2.9.3. Algoritmo di Kruskal
 - 2.9.4. Analisi della complessità
- 2.10. *Backtracking*
 - 2.10.1. Il *Backtracking*
 - 2.10.2. Tecniche alternative

Modulo 3. Programmazione orientata agli oggetti

- 3.1. Introduzione alla programmazione orientata agli oggetti
 - 3.1.1. Introduzione alla programmazione orientata agli oggetti
 - 3.1.2. Progettazione delle lezioni
 - 3.1.3. Introduzione a UML per la modellazione dei problemi
- 3.2. Relazioni tra lezioni
 - 3.2.1. Astrazione ed ereditarietà
 - 3.2.2. Concetti avanzati di ereditarietà
 - 3.2.3. Polimorfismo
 - 3.2.4. Composizione e aggregazione
- 3.3. Introduzione ai design pattern per i problemi orientati agli oggetti
 - 3.3.1. Cosa sono i design pattern?
 - 3.3.2. Pattern *Factory*
 - 3.3.3. Pattern *Singleton*
 - 3.3.4. Pattern *Observer*
 - 3.3.5. Pattern *Composite*
- 3.4. Eccezioni
 - 3.4.1. Quali sono le eccezioni?
 - 3.4.2. Gestione e acquisizione delle eccezioni
 - 3.4.3. Avvio delle eccezioni
 - 3.4.4. Creazione di eccezioni
- 3.5. Interfacce utente
 - 3.5.1. Introduzione a Qt
 - 3.5.2. Posizionamento
 - 3.5.3. Cosa sono gli eventi?
 - 3.5.4. Eventi: definizione e acquisizione
 - 3.5.5. Sviluppo di interfacce utente
- 3.6. Introduzione alla programmazione concorrente
 - 3.6.1. Introduzione alla programmazione concorrente
 - 3.6.2. Il concetto di processo e di thread
 - 3.6.3. Interazione tra processi o thread
 - 3.6.4. Thread in C++
 - 3.6.5. Vantaggi e svantaggi della programmazione concorrente

- 3.7. Gestione e sincronizzazione dei thread
 - 3.7.1. Ciclo di vita di un thread
 - 3.7.2. La classe *Thread*
 - 3.7.3. Pianificazione del thread
 - 3.7.4. Gruppi di thread
 - 3.7.5. Thread di tipo demoniaco
 - 3.7.6. Sincronizzazione
 - 3.7.7. Meccanismi di bloccaggio
 - 3.7.8. Meccanismi di comunicazione
 - 3.7.9. Monitor
- 3.8. Problemi comuni nella programmazione concorrente
 - 3.8.1. Il problema dei produttori-consumatori
 - 3.8.2. Il problema dei lettori e degli scrittori
 - 3.8.3. Il problema della cena dei filosofi
- 3.9. Documentazione e test del software
 - 3.9.1. Perché è importante documentare il software?
 - 3.9.2. Documentazione di progettazione
 - 3.9.3. Utilizzo di strumenti per la documentazione
- 3.10. Test di software
 - 3.10.1. Introduzione al test del software
 - 3.10.2. Tipi di test
 - 3.10.3. Test dell'unità
 - 3.10.4. Test di integrità
 - 3.10.5. Test di convalida
 - 3.10.6. Test del sistema

Modulo 4. Console e dispositivi per videogiochi

- 4.1. Storia della programmazione dei videogiochi
 - 4.1.1. Periodo Atari (1977-1985)
 - 4.1.2. Periodo NES e SNES (1985-1995)
 - 4.1.3. Periodo PlayStation/PlayStation 2 (1995-2005)
 - 4.1.4. Periodo Xbox 360, PS3 e Wii (2005-2013)
 - 4.1.5. Periodo Xbox One, PS4 e Wii U- Switch (2013-oggi)
 - 4.1.6. Il futuro
- 4.2. Storia del gameplay nei videogiochi
 - 4.2.1. Introduzione
 - 4.2.2. Contesto sociale
 - 4.2.3. Diagramma strutturale
 - 4.2.4. Futuro
- 4.3. Adattamento ai tempi moderni
 - 4.3.1. Giochi basati sul movimento
 - 4.3.2. Realtà virtuale
 - 4.3.3. Realtà aumentata
 - 4.3.4. Realtà mista
- 4.4. Unity: *Scripting I* ed esempi
 - 4.4.1. Che cos'è uno *script*?
 - 4.4.2. Il nostro primo *script*
 - 4.4.3. Aggiunta di uno *script*
 - 4.4.4. Apertura di uno *script*
 - 4.4.5. *MonoBehaviour*
 - 4.4.6. *Debugging*
- 4.5. Unity: *Scripting II* ed esempi
 - 4.5.1. Input da tastiera e mouse
 - 4.5.2. *Raycast*
 - 4.5.3. Istanziamento
 - 4.5.4. Variabili
 - 4.5.5. Variabili pubbliche e in serie

- 4.6. Unity: *Scripting* III ed esempi
 - 4.6.1. Ottenere i componenti
 - 4.6.2. Modificare i componenti
 - 4.6.3. Test
 - 4.6.4. Oggetti multipli
 - 4.6.5. *Collider e trigger*
 - 4.6.6. Quaternioni
- 4.7. Periferiche
 - 4.7.1. Evoluzione e tipologie
 - 4.7.2. Periferiche e interfacce
 - 4.7.3. Periferiche attuali
 - 4.7.4. Nel prossimo futuro
- 4.8. Videogiochi: prospettive future
 - 4.8.1. Giochi basati sul cloud
 - 4.8.2. Assenza di controller
 - 4.8.3. Realtà immersiva
 - 4.8.4. Alternative
- 4.9. Architettura
 - 4.9.1. Esigenze particolari dei videogiochi
 - 4.9.2. Evoluzione dell'architettura
 - 4.9.3. Architettura contemporanea
 - 4.9.4. Differenze tra architetture
- 4.10. Kit di sviluppo e la loro evoluzione
 - 4.10.1. Introduzione
 - 4.10.2. Kit di sviluppo di terza generazione
 - 4.10.3. Kit di sviluppo di quarta generazione
 - 4.10.4. Kit di sviluppo di quinta generazione
 - 4.10.5. Kit di sviluppo di sesta generazione

Modulo 5. Ingegneria software

- 5.1. Introduzione all'ingegneria del software e alla modellazione
 - 5.1.1. La natura del software
 - 5.1.2. La natura unica del *Webapps*
 - 5.1.3. Ingegneria software
 - 5.1.4. Il processo del software
 - 5.1.5. La pratica dell'ingegneria del software
 - 5.1.6. Miti del software
 - 5.1.7. Come nasce il tutto?
 - 5.1.8. Concetti orientati agli oggetti
 - 5.1.9. Introduzione a UML
- 5.2. Il processo del software
 - 5.2.1. Un modello generale di processo
 - 5.2.2. Modelli di processo prescrittivi
 - 5.2.3. Modelli di processo specializzati
 - 5.2.4. Il processo unificato
 - 5.2.5. Modelli di processo personali e di gruppo
 - 5.2.6. Che cos'è l'agilità?
 - 5.2.7. Che cos'è un processo agile?
 - 5.2.8. *Scrum*
 - 5.2.9. Kit di strumenti per i processi agili
- 5.3. Principi che guidano la pratica dell'ingegneria del software
 - 5.3.1. Principi che guidano il processo
 - 5.3.2. Principi che guidano la pratica
 - 5.3.3. Principi di comunicazione
 - 5.3.4. Principi di pianificazione
 - 5.3.5. Principi di modellazione
 - 5.3.6. Principi di costruzione
 - 5.3.7. Principi di implementazione

- 5.4. Comprendere i requisiti
 - 5.4.1. Ingegneria dei requisiti
 - 5.4.2. Porre le basi
 - 5.4.3. Indagine sui requisiti
 - 5.4.4. Sviluppo di casi d'uso
 - 5.4.5. Elaborazione del modello dei requisiti
 - 5.4.6. Negoziazione dei requisiti
 - 5.4.7. Convalida dei requisiti
- 5.5. Modellazione dei requisiti: scenari, informazioni e classi di analisi
 - 5.5.1. Analisi dei requisiti
 - 5.5.2. Modellazione basata su scenari
 - 5.5.3. Modelli UML che forniscono il caso d'uso
 - 5.5.4. Concetti di modellazione dei dati
 - 5.5.5. Modellazione basata sulle classi
 - 5.5.6. Diagrammi di classe
- 5.6. Modellazione dei requisiti: flusso, comportamento e modelli
 - 5.6.1. Requisiti che modellano le strategie
 - 5.6.2. Modellazione orientata al flusso
 - 5.6.3. Diagrammi di stato
 - 5.6.4. Creare un modello comportamentale
 - 5.6.5. Diagrammi di sequenza
 - 5.6.6. Diagrammi di comunicazione
 - 5.6.7. Schemi per la modellazione dei requisiti
- 5.7. Concetti di design
 - 5.7.1. Il design nel contesto dell'ingegneria del software
 - 5.7.2. Processo di progettazione
 - 5.7.3. Concetti di design
 - 5.7.4. Concetti di design orientati agli oggetti
 - 5.7.5. Il modello di design
- 5.8. Design architettonico
 - 5.8.1. Architettura del software
 - 5.8.2. Generi architettonici
 - 5.8.3. Stili architettonici
 - 5.8.4. Design architettonico
 - 5.8.5. Evoluzione dei design alternativi per l'architettura
 - 5.8.6. Mappatura dell'architettura con l'uso di flussi di dati
- 5.9. Design a livello di componente e basato su pattern
 - 5.9.1. Che cos'è un componente?
 - 5.9.2. Design dei componenti basato sulle classi
 - 5.9.3. Realizzazione del progetto a livello di componenti
 - 5.9.4. Design dei componenti tradizionali
 - 5.9.5. Sviluppo basato su componenti
 - 5.9.6. Modelli di progettazione
 - 5.9.7. Design del software basato su modelli
 - 5.9.8. Modelli architettonici
 - 5.9.9. Modelli di design a livello di componenti
 - 5.9.10. Modelli di design dell'interfaccia utente
- 5.10. Qualità del software e gestione dei progetti
 - 5.10.1. Qualità
 - 5.10.2. Qualità del software
 - 5.10.3. Il dilemma della qualità del software
 - 5.10.4. Raggiungere la qualità del software
 - 5.10.5. Garanzia di qualità del software
 - 5.10.6. Lo spettro amministrativo
 - 5.10.7. Il personale
 - 5.10.8. Il prodotto
 - 5.10.9. Il processo
 - 5.10.10. Il progetto
 - 5.10.11. Principi e pratiche

Modulo 6. Motori grafici per videogiochi

- 6.1. Videogiochi e TIC
 - 6.1.1. Introduzione
 - 6.1.2. Opportunità
 - 6.1.3. Difficoltà
 - 6.1.4. Conclusioni
- 6.2. Storia dei motori grafici per videogiochi
 - 6.2.1. Introduzione
 - 6.2.2. Epoca Atari
 - 6.2.3. Epoca anni '80
 - 6.2.4. Primi motori grafici: Epoca anni '90
 - 6.2.5. Motori grafici attuali
- 6.3. Motori grafici per videogiochi
 - 6.3.1. Tipi di motori grafici
 - 6.3.2. Parti che compongono un motore grafico
 - 6.3.3. Motori grafici attuali
 - 6.3.4. Selezione di un motore grafico per il nostro progetto
- 6.4. *Game Maker*
 - 6.4.1. Introduzione
 - 6.4.2. Progettazione degli scenari
 - 6.4.3. *Sprite* e animazioni
 - 6.4.4. Collisioni
 - 6.4.5. *Scripting* in GML
- 6.5. Motore grafico Unreal Engine 4: Introduzione
 - 6.5.1. Che cos'è Unreal Engine 4? Qual è la sua filosofia?
 - 6.5.2. Materiali
 - 6.5.3. UI
 - 6.5.4. Animazioni
 - 6.5.5. Sistema di particelle
 - 6.5.6. Intelligenza artificiale
 - 6.5.7. FPS



- 6.6. Motore grafico Unreal Engine 4: *Visual Scripting*
 - 6.6.1. Filosofia dei *Blueprint* e *Visual Scripting*
 - 6.6.2. *Debugging*
 - 6.6.3. Tipi di variabili
 - 6.6.4. Controllo del flusso di base
- 6.7. Motore grafico Unity 5
 - 6.7.1. Programmazione in C# e Visual Studio
 - 6.7.2. Creazione di *Prefabs*
 - 6.7.3. Utilizzo di *Gizmos* per il controllo dei videogiochi
 - 6.7.4. Motore grafico adattivo: 2D e 3D
- 6.8. Motore grafico Godot
 - 6.8.1. Filosofia progettuale di Godot
 - 6.8.2. Progettazione e composizione orientata agli oggetti
 - 6.8.3. Tutto in un unico pacchetto
 - 6.8.4. Software gratuito e promosso dalla comunità
- 6.9. Motore grafico RPG Maker
 - 6.9.1. Filosofia di RPG Maker
 - 6.9.2. Prendere come riferimento
 - 6.9.3. Creare un gioco con personalità
 - 6.9.4. Giochi commerciali di successo
- 6.10. Motore grafico Source 2
 - 6.10.1. Filosofia di Source 2
 - 6.10.2. Source e Source 2: evoluzione
 - 6.10.3. Uso da parte della comunità: contenuti audiovisivi e videogiochi
 - 6.10.4. Futuro del motore grafico Source 2
 - 6.10.5. *Mod* e giochi di successo

Modulo 7. Sistemi intelligenti

- 7.1. Teoria degli agenti
 - 7.1.1. Storia del concetto
 - 7.1.2. Definizione di agente
 - 7.1.3. Agenti nell'intelligenza artificiale
 - 7.1.4. Agenti nell'Ingegneria del Software
- 7.2. Architetture di agenti
 - 7.2.1. Il processo di ragionamento dell'agente
 - 7.2.2. Agenti reattivi
 - 7.2.3. Agenti deduttivi
 - 7.2.4. Agenti ibridi
 - 7.2.5. Confronto
- 7.3. Informazione e conoscenza
 - 7.3.1. Distinzione tra dati, informazioni e conoscenza
 - 7.3.2. Valutazione della qualità dei dati
 - 7.3.3. Metodi di raccolta dei dati
 - 7.3.4. Metodi di acquisizione dei dati
 - 7.3.5. Metodi di acquisizione della conoscenza
- 7.4. Rappresentazione della conoscenza
 - 7.4.1. L'importanza della rappresentazione della conoscenza
 - 7.4.2. Definire la rappresentazione della conoscenza attraverso i suoi ruoli
 - 7.4.3. Caratteristiche di una rappresentazione della conoscenza
- 7.5. Ontologie
 - 7.5.1. Introduzione ai metadati
 - 7.5.2. Concetto filosofico di ontologia
 - 7.5.3. Concetto informatico di ontologia
 - 7.5.4. Ontologie di dominio e di livello superiore
 - 7.5.5. Come costruire un'ontologia?

- 7.6. Linguaggi ontologici e software per la creazione di ontologie
 - 7.6.1. Triple RDF, Turtle e N3
 - 7.6.2. Schema RDF
 - 7.6.3. OWL
 - 7.6.4. SPARQL
 - 7.6.5. Introduzione ai diversi strumenti per la creazione di ontologie
 - 7.6.6. Installazione e utilizzo di Protégé
- 7.7. Web semantico
 - 7.7.1. Lo stato attuale e il futuro del web semantico
 - 7.7.2. Applicazioni del web semantico
- 7.8. Altri modelli di rappresentazione della conoscenza
 - 7.8.1. Vocabolari
 - 7.8.2. Panoramica
 - 7.8.3. Tassonomie
 - 7.8.4. Thesauri
 - 7.8.5. Folksonomie
 - 7.8.6. Confronto
 - 7.8.7. Mapped mentali
- 7.9. Valutazione e integrazione delle rappresentazioni della conoscenza
 - 7.9.1. Logica dell'ordine zero
 - 7.9.2. Logica di prim'ordine
 - 7.9.3. Logica descrittiva
 - 7.9.4. Relazione tra i diversi tipi di logica
 - 7.9.5. Prolog: programmazione basata sulla logica del primo ordine
- 7.10. Ragionatori semantici, sistemi basati sulla conoscenza e sistemi esperti
 - 7.10.1. Concetto di ragionatore
 - 7.10.2. Applicazioni di un ragionatore
 - 7.10.3. Sistemi basati sulla conoscenza
 - 7.10.4. MYCIN, storia dei sistemi esperti
 - 7.10.5. Elementi e architettura dei sistemi esperti
 - 7.10.6. Creazione di sistemi esperti

Modulo 8. Programmazione in tempo reale

- 8.1. Nozioni di base di programmazione concorrente
 - 8.1.1. Concetti fondamentali
 - 8.1.2. Concorrenza
 - 8.1.3. Vantaggi della concorrenza
 - 8.1.4. Concorrenza e hardware
- 8.2. Strutture di supporto alla concorrenza di base in Java
 - 8.2.1. Concorrenza in Java
 - 8.2.2. Creazione di *Thread*
 - 8.2.3. Metodi
 - 8.2.4. Sincronizzazione
- 8.3. *Thread*, ciclo di vita, priorità, interrupt, stati, esecutori
 - 8.3.1. *Thread*
 - 8.3.2. Ciclo di vita
 - 8.3.3. Priorità
 - 8.3.4. Interrupt
 - 8.3.5. Stati
 - 8.3.6. Esecutori
- 8.4. Esclusione reciproca
 - 8.4.1. Che cos'è l'esclusione reciproca?
 - 8.4.2. Algoritmo di Dekker
 - 8.4.3. Algoritmo di Peterson
 - 8.4.4. Esclusione reciproca in Java
- 8.5. Dipendenze di stati
 - 8.5.1. Iniezione di dipendenze
 - 8.5.2. Implementazione del modello in Java
 - 8.5.3. Modi per immettere le dipendenze
 - 8.5.4. Esempio

- 8.6. Modelli di progettazione
 - 8.6.1. Introduzione
 - 8.6.2. Modelli di creazione
 - 8.6.3. Modelli di struttura
 - 8.6.4. Modelli di comportamento
- 8.7. Uso delle librerie Java
 - 8.7.1. Cosa sono le librerie in Java?
 - 8.7.2. Mockito-all, mockito-core
 - 8.7.3. Guava
 - 8.7.4. Commons-io
 - 8.7.5. Commons-lang, commons-lang3
- 8.8. Programmazione degli *shader*
 - 8.8.1. Pipeline 3D e rendering
 - 8.8.2. Vertex Shading
 - 8.8.3. Pixel Shading: illuminazione I
 - 8.8.4. Pixel Shading: illuminazione II
 - 8.8.5. Post-effetti
- 8.9. Programmazione in tempo reale
 - 8.9.1. Introduzione
 - 8.9.2. Elaborazione degli interrupt
 - 8.9.3. Sincronizzazione e comunicazione tra processi
 - 8.9.4. Sistemi di pianificazione in tempo reale
- 8.10. Pianificazione in tempo reale
 - 8.10.1. Concetti
 - 8.10.2. Modello di riferimento per i sistemi in tempo reale
 - 8.10.3. Politiche di pianificazione
 - 8.10.4. Pianificatori ciclici
 - 8.10.5. Pianificatori con proprietà statiche
 - 8.10.6. Pianificatori con proprietà dinamiche

Modulo 9. Design e sviluppo di giochi online

- 9.1. Origini e standard dell'online
 - 9.1.1. Le origini di Internet
 - 9.1.2. Creazione del *World Wide Web*
 - 9.1.3. Nascita degli standard web
 - 9.1.4. L'ascesa degli standard web
- 9.2. HTTP e struttura client-server
 - 9.2.1. Ruolo client-server
 - 9.2.2. Comunicazione client-server
 - 9.2.3. Storia recente
 - 9.2.4. Informatica centralizzata
- 9.3. Programmazione web: introduzione
 - 9.3.1. Concetti di base
 - 9.3.2. Preparazione di un server web
 - 9.3.3. Nozioni di base di HTML5
 - 9.3.4. Moduli HTML
- 9.4. Introduzione all'HTML ed esempi
 - 9.4.1. Storia di HTML5
 - 9.4.2. Elementi HTML5
 - 9.4.3. APIS
 - 9.4.4. CCS3.
- 9.5. Modello a oggetti del documento
 - 9.5.1. Che cos'è il Modello a Oggetti del Documento?
 - 9.5.2. Uso di *DOCTYPE*
 - 9.5.3. L'importanza della validazione dell'HTML
 - 9.5.4. Accesso agli elementi
 - 9.5.5. Creare elementi e testi
 - 9.5.6. Uso di innerHTML
 - 9.5.7. Eliminazione di un elemento di testo o di un nodo
 - 9.5.8. Lettura e scrittura degli attributi di un elemento
 - 9.5.9. Manipolazione degli stili degli elementi
 - 9.5.10. Allegare più file contemporaneamente

- 9.6. Introduzione al CSS ed esempi
 - 9.6.1. Sintassi CSS3
 - 9.6.2. Fogli di stile
 - 9.6.3. Etichette
 - 9.6.4. Selezionatori
 - 9.6.5. Web design con i CSS
- 9.7. Introduzione a JavaScript ed esempi
 - 9.7.1. Che cos'è JavaScript?
 - 9.7.2. Breve storia del linguaggio
 - 9.7.3. Versioni JavaScript
 - 9.7.4. Visualizzare una finestra di dialogo
 - 9.7.5. Sintassi di JavaScript
 - 9.7.6. Capire gli *Script*
 - 9.7.7. Spazi
 - 9.7.8. Commenti
 - 9.7.9. Funzioni
 - 9.7.10. JavaScript esterno e sulla pagina
- 9.8. Funzioni in JavaScript
 - 9.8.1. Dichiarazioni di funzione
 - 9.8.2. Espressioni di funzione
 - 9.8.3. Chiamare le funzioni
 - 9.8.4. Ricorsività
 - 9.8.5. Funzioni e chiusure annidate
 - 9.8.6. Conservazione delle variabili
 - 9.8.7. Funzioni multinanellate
 - 9.8.8. Conflitti di denominazione
 - 9.8.9. Chiusure
 - 9.8.10. Parametri di una funzione
- 9.9. PlayCanvas per lo sviluppo di giochi online
 - 9.9.1. Che cos'è PlayCanvas?
 - 9.9.2. Configurazione del progetto
 - 9.9.3. Creare un oggetto
 - 9.9.4. Aggiunta della fisica
 - 9.9.5. Aggiunta di un modello
 - 9.9.6. Modifica delle impostazioni di gravità e della scena
 - 9.9.7. Esecuzione di *Script*
 - 9.9.8. Controlli della telecamera
- 9.10. Phaser per lo sviluppo di giochi online
 - 9.10.1. Che cos'è Phaser?
 - 9.10.2. Ricarica delle risorse
 - 9.10.3. Costruire il mondo
 - 9.10.4. Le piattaforme
 - 9.10.5. Il giocatore
 - 9.10.6. Aggiungere la fisica
 - 9.10.7. Usare la tastiera
 - 9.10.8. Raccogliere i *Pickup*
 - 9.10.9. Punti e punteggi
 - 9.10.10. Pompe rimbalzanti

Modulo 10. Reti e sistemi multiplayer

- 10.1. Storia ed evoluzione dei videogiochi multiplayer
 - 10.1.1. Anni '70: i primi giochi multiplayer
 - 10.1.2. Anni '90: Duke Nukem, Doom, Quake
 - 10.1.3. L'ascesa dei videogiochi multiplayer
 - 10.1.4. Multiplayer locale e online
 - 10.1.5. Giochi di società
- 10.2. Modelli di business multiplayer
 - 10.2.1. Origine e funzionamento dei modelli di business emergenti
 - 10.2.2. Servizi di vendita online
 - 10.2.3. Gioco gratuito
 - 10.2.4. Microtransazioni
 - 10.2.5. Pubblicità
 - 10.2.6. Abbonamento con pagamento mensile
 - 10.2.7. Pay to play
 - 10.2.8. Provare prima di acquistare

- 10.3. Giochi multiplayer locale e online
 - 10.3.1. Giochi multiplayer locale: gli inizi
 - 10.3.2. Giochi di società: Nintendo e unione familiare
 - 10.3.3. Giochi online: gli inizi
 - 10.3.4. Evoluzione dei giochi online
- 10.4. Modello OSI: livelli I
 - 10.4.1. Modello OSI: introduzione
 - 10.4.2. Livello fisico
 - 10.4.3. Livello di collegamento dati
 - 10.4.4. Livello di rete
- 10.5. Modello OSI: livelli II
 - 10.5.1. Livello di trasporto
 - 10.5.2. Livello di sessione
 - 10.5.3. Livello di presentazione
 - 10.5.4. Livello applicativo
- 10.6. Reti informatiche e internet
 - 10.6.1. Che cos'è una rete di computer?
 - 10.6.2. Software
 - 10.6.3. Hardware
 - 10.6.4. Server
 - 10.6.5. Archiviazione in rete
 - 10.6.6. Protocolli di rete
- 10.7. Reti mobili e wireless
 - 10.7.1. Rete mobile
 - 10.7.2. Rete wireless
 - 10.7.3. Funzionamento delle reti mobili
 - 10.7.4. Tecnologia digitale
- 10.8. Sicurezza
 - 10.8.1. Sicurezza personale
 - 10.8.2. *Hacks* e *cheats* nei videogiochi
 - 10.8.3. Sicurezza anti-trappola
 - 10.8.4. Analisi dei sistemi di sicurezza anti-trappola
- 10.9. Sistemi multiplayer: server
 - 10.9.1. Hosting server
 - 10.9.2. Videogiochi MMO
 - 10.9.3. Server dedicati ai videogiochi
 - 10.9.4. *LAN Parties*
- 10.10. Design e programmazione di videogiochi multigiocatore
 - 10.10.1. Fondamenti di design di videogiochi multigiocatore in Unreal
 - 10.10.2. Fondamenti di design di videogiochi multigiocatore in Unity
 - 10.10.3. Come rendere divertente il gioco multigiocatore?
 - 10.10.4. Oltre il comando: Innovazione nei comandi multigiocatore



Grazie al metodo Relearning, basato sulla ripetizione dei concetti chiave, potrai ridurre le lunghe ore di studio"

06 Tirocinio

Una volta che il professionista dei videogiochi ha completato l'insegnamento teorico online, questa qualifica prevede lo svolgimento di un tirocinio in uno studio creativo e sviluppatore di videogiochi. In questo modo, gli studenti possono entrare direttamente in contatto con i gruppi di sviluppo del settore e dimostrare tutto ciò che hanno imparato. Durante questa fase di apprendimento lo studente avrà un tutor che lo accompagnerà fino al completamento di questo insegnamento.



“

*Siediti accanto a grandi sviluppatori
e lavora in squadra durante la fase
pratica di questo insegnamento”*

Il periodo di Tirocinio di questo programma di Programmazione di Videogiochi dura 3 settimane. Gli studenti si recheranno allo studio creativo e sviluppatore di videogiochi da lunedì a venerdì con una giornata di 8 ore consecutive di insegnamento pratico da uno specialista.

Questa fase pratica permetterà agli studenti di utilizzare tutte le conoscenze di programmazione acquisite in questo Master Semipresenziale. Insieme ad altri professionisti di questa azienda sarà in grado di coordinare con il resto del team e imparare i principali strumenti di programmazione e software. In questo modo, si ottiene un apprendimento legato alla realtà dell'industria dei videogiochi.

Durante il soggiorno in questo studio, il professionista non solo utilizzerà il solito linguaggio di programmazione, ma sarà anche in grado di progettare videogiochi multigiocatore. Una categoria di videogiochi in crescita che affascina i Gamers.

La parte pratica sarà svolta con la partecipazione attiva dello studente svolgendo le attività e le procedure di ogni area di competenza (imparare a imparare e imparare a fare), con l'accompagnamento e la guida degli insegnanti e altri compagni di formazione che facilitano il lavoro di squadra e l'integrazione multidisciplinare come competenze trasversali per la pratica della programmazione dei videogiochi (imparare a essere e imparare a relazionarsi).

“

*Partecipa a uno studio che ti darà
l'opportunità di dimostrare il tuo potenziale
come programmatore di videogiochi"*



Le procedure descritte di seguito saranno la base della parte pratica della formazione, e la loro realizzazione sarà soggetta alla disponibilità propria del centro e al suo volume di lavoro, essendo le attività proposte come segue:

Modulo	Attività Pratica	Quantità
Creazione di struttura di dati e algoritmi	Eseguire la tecnica di <i>Backtracking</i> , per la creazione di dati e algoritmi	1
	Partecipare all'analisi degli algoritmi per ottenere una maggiore efficienza	
	Eseguire attività di misurazione delle dimensioni dell'ingresso e del tempo di esecuzione	
	Creare Algoritmi di ordinamento con alberi con <i>Heaps</i> , con grafi e con <i>Greedy</i>	
Programmazione orientata agli oggetti	Usare il <i>Pattern Factory</i> , <i>Pattern Singleton</i> , <i>Pattern Observer</i> e <i>Pattern Composite</i> nella creazione di oggetti	1
	Creare, catturare e gestire le eccezioni nella creazione di oggetti	
	Eseguire la programmazione concorrente	
	Impiegare meccanismi di blocco e comunicazione	
	Creare documentazione e test del software	
Programmazione in tempo reale	Creare e sincronizzare <i>Threads</i>	1
	Programmare <i>Shader</i>	
	Implementare il modello in Java e utilizzare le librerie Java.	
	Creare post-effetti	
	Elaborare interruzioni, sincronizzazione e comunicazione tra processi.	
Design e sviluppo di giochi online	Programmare il web con i moduli HTML	1
	Usare DOCTYPE e innerHTML per lo sviluppo di giochi online	
	Fare uso di PlayCanvas per lo sviluppo di giochi online	
	Configurare il progetto di progettazione e sviluppo del gioco web	

Assicurazione di responsabilità civile

La preoccupazione principale di questa istituzione è quella di garantire la sicurezza sia dei tirocinanti sia degli altri agenti che collaborano ai processi di tirocinio in azienda. All'interno delle misure rivolte a questo fine ultimo, esiste la risposta a qualsiasi incidente che possa verificarsi durante il processo di insegnamento-apprendimento.

A tal fine, questa istituzione educativa si impegna a stipulare un'assicurazione di responsabilità civile per coprire qualsiasi eventualità che possa insorgere durante il seminario educativo presso il centro di tirocinio.

La polizza di responsabilità civile per i tirocinanti deve garantire una copertura assicurativa completa e deve essere stipulata prima dell'inizio del periodo di tirocinio. Grazie a questa garanzia, il professionista si sentirà privo di ogni tipo di preoccupazione nel caso di eventuali situazioni impreviste che possano sorgere durante il tirocinio e potrà godere di una copertura assicurativa fino al termine dello stesso.

“

L'Università stipulerà un'assicurazione di responsabilità civile che coprirà qualsiasi eventualità che possa sorgere durante lo svolgimento del soggiorno nel centro di tirocinio"



Condizioni Generali di Tirocinio

Le condizioni generali dell'accordo di tirocinio per il programma sono le seguenti:

1. TUTORAGGIO: durante il Master Semipresenziale agli studenti verranno assegnati due tutor che li seguiranno durante tutto il percorso, risolvendo eventuali dubbi e domande. Da un lato, lo studente disporrà di un tutor professionale appartenente al centro di inserimento lavorativo che lo guiderà e lo supporterà in ogni momento. Dall'altro lato, allo studente verrà assegnato anche un tutor accademico che avrà il compito di coordinare e aiutare lo studente durante l'intero processo, risolvendo i dubbi e fornendogli tutto ciò di cui potrebbe aver bisogno. In questo modo, il professionista sarà accompagnato in ogni momento e potrà risolvere tutti gli eventuali dubbi, sia di natura pratica che accademica.

2. DURATA: il programma del tirocinio avrà una durata di tre settimane consecutive di preparazione pratica, distribuite in giornate di 8 ore lavorative, per cinque giorni alla settimana. I giorni di frequenza e l'orario saranno di competenza del centro, che informerà debitamente e preventivamente il professionista, con un sufficiente anticipo per facilitarne l'organizzazione.

3. MANCATA PRESENTAZIONE: in caso di mancata presentazione il giorno di inizio del Master Semipresenziale, lo studente perderà il diritto allo stesso senza possibilità di rimborso o di modifica di date. L'assenza per più di due giorni senza un giustificato motivo/certificato medico comporterà la rinuncia dello studente al tirocinio e, pertanto, la relativa automatica cessazione. In caso di ulteriori problemi durante lo svolgimento del tirocinio, essi dovranno essere debitamente e urgentemente segnalati al tutor accademico.

4. CERTIFICAZIONE: lo studente che supererà il Master Semipresenziale riceverà un certificato che attesterà il tirocinio svolto presso il centro in questione.

5. RAPPORTO DI LAVORO: il Master Semipresenziale non costituisce alcun tipo di rapporto lavorativo.

6. STUDI PRECEDENTI: alcuni centri potranno richiedere un certificato di studi precedenti per la partecipazione al Master Semipresenziale. In tal caso, sarà necessario esibirlo al dipartimento tirocini di TECH affinché venga confermata l'assegnazione del centro prescelto.

7. NON INCLUDE: il Master Semipresenziale non includerà nessun elemento non menzionato all'interno delle presenti condizioni. Pertanto, non sono inclusi alloggio, trasporto verso la città in cui si svolge il tirocinio, visti o qualsiasi altro servizio non menzionato.

Tuttavia, gli studenti potranno consultare il proprio tutor accademico per qualsiasi dubbio o raccomandazione in merito. Egli fornirà tutte le informazioni necessarie per semplificare le procedure.

07

Dove posso svolgere il Tirocinio?

Questo titolo di Master Semipresenziale prevede un tirocinio pratico in uno studio di riferimento dove gli studenti metteranno in pratica tutto ciò che hanno imparato sulla programmazione di videogiochi. In questo modo, TECH offre agli studenti l'opportunità di avvicinare questo titolo a professionisti del videogioco che desiderano acquisire un insegnamento di qualità e in linea con le attuali esigenze del settore.

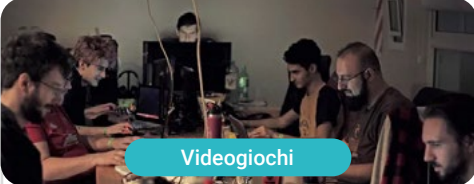


“

Raggiungi i tuoi obiettivi applicando tutte le tue conoscenze in uno studio di riferimento nel settore dei videogiochi”



Gli studenti potranno svolgere il tirocinio di questo Master Semipresenziale presso i seguenti centri:



Videogiochi

Startreming Games

Paese	Città
Argentina	Mendoza

Indirizzo: Ruta 160 esquina Buenos Aires 88,
San Rafael, Mendoza

Studio di sviluppo di videogiochi indipendente e operante da remoto

Tirocini correlati:

- Programmazione di Videogiochi
- Hard Surface 3D Modeling

Programmazione di Videogiochi





“

Cogli l'opportunità per conoscere gli ultimi sviluppi in questo campo per applicarli alla tua pratica quotidiana"

08

Metodologia di studio

TECH è la prima università al mondo che combina la metodologia dei **case studies** con il **Relearning**, un sistema di apprendimento 100% online basato sulla ripetizione diretta.

Questa strategia dirompente è stata concepita per offrire ai professionisti l'opportunità di aggiornare le conoscenze e sviluppare competenze in modo intensivo e rigoroso. Un modello di apprendimento che pone lo studente al centro del processo accademico e gli conferisce tutto il protagonismo, adattandosi alle sue esigenze e lasciando da parte le metodologie più convenzionali.



“

TECH ti prepara ad affrontare nuove sfide in ambienti incerti e a raggiungere il successo nella tua carriera"

Lo studente: la priorità di tutti i programmi di TECH

Nella metodologia di studio di TECH lo studente è il protagonista assoluto. Gli strumenti pedagogici di ogni programma sono stati selezionati tenendo conto delle esigenze di tempo, disponibilità e rigore accademico che, al giorno d'oggi, non solo gli studenti richiedono ma le posizioni più competitive del mercato.

Con il modello educativo asincrono di TECH, è lo studente che sceglie il tempo da dedicare allo studio, come decide di impostare le sue routine e tutto questo dalla comodità del dispositivo elettronico di sua scelta. Lo studente non deve frequentare lezioni presenziali, che spesso non può frequentare. Le attività di apprendimento saranno svolte quando si ritenga conveniente. È lo studente a decidere quando e da dove studiare.

“

*In TECH NON ci sono lezioni presenziali
(che poi non potrai mai frequentare)”*



I piani di studio più completi a livello internazionale

TECH si caratterizza per offrire i percorsi accademici più completi del panorama universitario. Questa completezza è raggiunta attraverso la creazione di piani di studio che non solo coprono le conoscenze essenziali, ma anche le più recenti innovazioni in ogni area.

Essendo in costante aggiornamento, questi programmi consentono agli studenti di stare al passo con i cambiamenti del mercato e acquisire le competenze più apprezzate dai datori di lavoro. In questo modo, coloro che completano gli studi presso TECH ricevono una preparazione completa che fornisce loro un notevole vantaggio competitivo per avanzare nelle loro carriere.

Inoltre, potranno farlo da qualsiasi dispositivo, pc, tablet o smartphone.

“

Il modello di TECH è asincrono, quindi ti permette di studiare con il tuo pc, tablet o smartphone dove, quando e per quanto tempo vuoi”

Case studies o Metodo Casistico

Il Metodo Casistico è stato il sistema di apprendimento più usato nelle migliori facoltà del mondo. Sviluppato nel 1912 per consentire agli studenti di Giurisprudenza non solo di imparare le leggi sulla base di contenuti teorici, ma anche di esaminare situazioni complesse reali. In questo modo, potevano prendere decisioni e formulare giudizi di valore fondati su come risolverle. Nel 1924 fu stabilito come metodo di insegnamento standard ad Harvard.

Con questo modello di insegnamento, è lo studente stesso che costruisce la sua competenza professionale attraverso strategie come il *Learning by doing* o il *Design Thinking*, utilizzate da altre istituzioni rinomate come Yale o Stanford.

Questo metodo, orientato all'azione, sarà applicato lungo tutto il percorso accademico che lo studente intraprende insieme a TECH. In questo modo, affronterà molteplici situazioni reali e dovrà integrare le conoscenze, ricercare, argomentare e difendere le sue idee e decisioni. Tutto ciò con la premessa di rispondere al dubbio di come agirebbe nel posizionarsi di fronte a specifici eventi di complessità nel suo lavoro quotidiano.



Metodo Relearning

In TECH i *case studies* vengono potenziati con il miglior metodo di insegnamento 100% online: il *Relearning*.

Questo metodo rompe con le tecniche di insegnamento tradizionali per posizionare lo studente al centro dell'equazione, fornendo il miglior contenuto in diversi formati. In questo modo, riesce a ripassare e ripete i concetti chiave di ogni materia e impara ad applicarli in un ambiente reale.

In questa stessa linea, e secondo molteplici ricerche scientifiche, la ripetizione è il modo migliore per imparare. Ecco perché TECH offre da 8 a 16 ripetizioni di ogni concetto chiave in una stessa lezione, presentata in modo diverso, con l'obiettivo di garantire che la conoscenza sia completamente consolidata durante il processo di studio.

Il Relearning ti consentirà di apprendere con meno sforzo e più rendimento, coinvolgendoti maggiormente nella specializzazione, sviluppando uno spirito critico, difendendo gli argomenti e contrastando opinioni: un'equazione diretta al successo.



Un Campus Virtuale 100% online con le migliori risorse didattiche

Per applicare efficacemente la sua metodologia, TECH si concentra sul fornire agli studenti materiali didattici in diversi formati: testi, video interattivi, illustrazioni, mappe della conoscenza, ecc. Tutto ciò progettato da insegnanti qualificati che concentrano il lavoro sulla combinazione di casi reali con la risoluzione di situazioni complesse attraverso la simulazione, lo studio dei contesti applicati a ogni carriera e l'apprendimento basato sulla ripetizione, attraverso audio, presentazioni, animazioni, immagini, ecc.

Le ultime prove scientifiche nel campo delle Neuroscienze indicano l'importanza di considerare il luogo e il contesto in cui si accede ai contenuti prima di iniziare un nuovo apprendimento. Poter regolare queste variabili in modo personalizzato favorisce che le persone possano ricordare e memorizzare nell'ippocampo le conoscenze per conservarle a lungo termine. Si tratta di un modello denominato *Neurocognitive context-dependent e-learning*, che viene applicato in modo consapevole in questa qualifica universitaria.

Inoltre, anche per favorire al massimo il contatto tra mentore e studente, viene fornita una vasta gamma di possibilità di comunicazione, sia in tempo reale che differita (messaggistica interna, forum di discussione, servizio di assistenza telefonica, e-mail di contatto con segreteria tecnica, chat e videoconferenza).

Inoltre, questo completo Campus Virtuale permetterà agli studenti di TECH di organizzare i loro orari di studio in base alla loro disponibilità personale o agli impegni lavorativi. In questo modo avranno un controllo globale dei contenuti accademici e dei loro strumenti didattici, il che attiva un rapido aggiornamento professionale.



La modalità di studio online di questo programma ti permetterà di organizzare il tuo tempo e il tuo ritmo di apprendimento, adattandolo ai tuoi orari"

L'efficacia del metodo è giustificata da quattro risultati chiave:

1. Gli studenti che seguono questo metodo non solo raggiungono l'assimilazione dei concetti, ma sviluppano anche la loro capacità mentale, attraverso esercizi che valutano situazioni reali e l'applicazione delle conoscenze.
2. L'apprendimento è solidamente fondato su competenze pratiche che permettono allo studente di integrarsi meglio nel mondo reale.
3. L'assimilazione di idee e concetti è resa più facile ed efficace, grazie all'uso di situazioni nate dalla realtà.
4. La sensazione di efficienza dello sforzo investito diventa uno stimolo molto importante per gli studenti, che si traduce in un maggiore interesse per l'apprendimento e in un aumento del tempo dedicato al corso.

La metodologia universitaria più apprezzata dagli studenti

I risultati di questo innovativo modello accademico sono riscontrabili nei livelli di soddisfazione globale degli studenti di TECH.

La valutazione degli studenti sulla qualità dell'insegnamento, la qualità dei materiali, la struttura del corso e i suoi obiettivi è eccellente. A conferma di ciò, l'istituto è diventato il migliore valutato dai suoi studenti sulla piattaforma di recensioni Trustpilot, ottenendo un punteggio di 4,9 su 5.

Accedi ai contenuti di studio da qualsiasi dispositivo con connessione a Internet (computer, tablet, smartphone) grazie al fatto che TECH è aggiornato sull'avanguardia tecnologica e pedagogica.

Potrai imparare dai vantaggi dell'accesso a ambienti di apprendimento simulati e dall'approccio di apprendimento per osservazione, ovvero Learning from an expert.



In questo modo, il miglior materiale didattico sarà disponibile, preparato con attenzione:



Materiale di studio

Tutti i contenuti didattici sono creati dagli specialisti che impartiranno il corso, appositamente per questo, in modo che lo sviluppo didattico sia realmente specifico e concreto.

Questi contenuti sono poi applicati al formato audiovisivo che supporterà la nostra modalità di lavoro online, impiegando le ultime tecnologie che ci permettono di offrirti una grande qualità per ogni elemento che metteremo al tuo servizio.



Capacità e competenze pratiche

I partecipanti svolgeranno attività per sviluppare competenze e abilità specifiche in ogni area tematica. Pratiche e dinamiche per acquisire e sviluppare le competenze e le abilità che uno specialista deve possedere nel mondo globalizzato in cui viviamo.



Riepiloghi interattivi

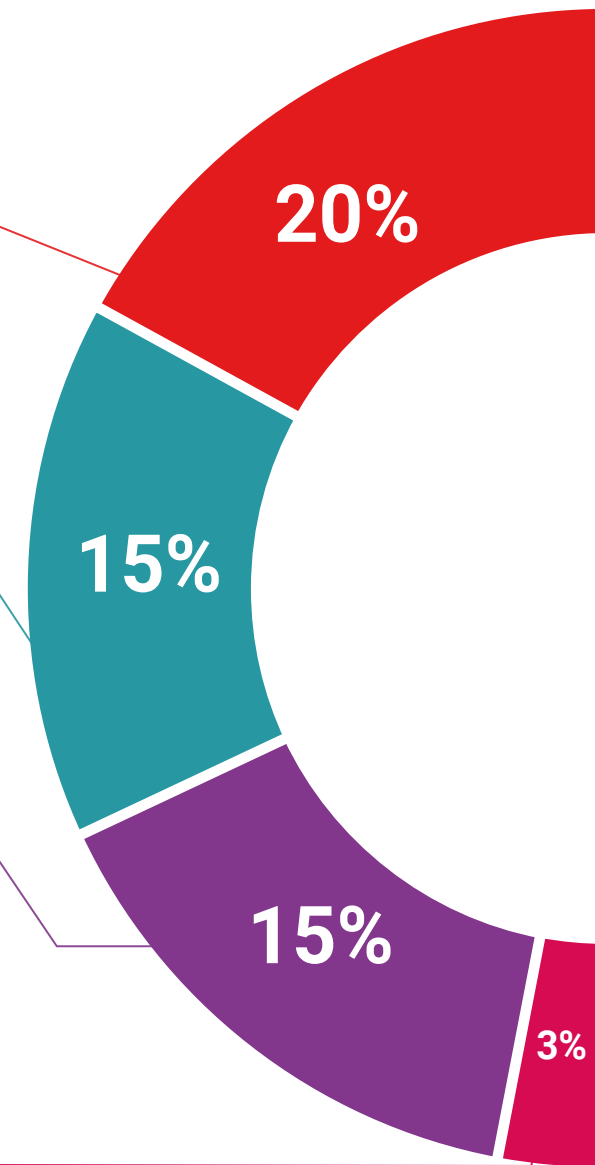
Presentiamo i contenuti in modo accattivante e dinamico tramite strumenti multimediali che includono audio, video, immagini, diagrammi e mappe concettuali per consolidare la conoscenza.

Questo esclusivo sistema di preparazione per la presentazione di contenuti multimediali è stato premiato da Microsoft come "Caso di successo in Europa".



Letture complementari

Articoli recenti, documenti di consenso, guide internazionali... Nella biblioteca virtuale di TECH potrai accedere a tutto il materiale necessario per completare la tua specializzazione.





Case Studies

Completerai una selezione dei migliori *case studies* in materia. Casi presentati, analizzati e monitorati dai migliori specialisti del panorama internazionale.



Testing & Retesting

Valutiamo e rivalutiamo periodicamente le tue conoscenze durante tutto il programma. Lo facciamo su 3 dei 4 livelli della Piramide di Miller.



Master class

Esistono prove scientifiche sull'utilità d'osservazione di terzi esperti. Il cosiddetto *Learning from an Expert* rafforza le conoscenze e i ricordi, e genera sicurezza nel futuro processo decisionale.



Guide di consultazione veloce

TECH offre i contenuti più rilevanti del corso sotto forma di schede o guide rapide per l'azione. Un modo sintetico, pratico ed efficace per aiutare a progredire nel tuo apprendimento.



09

Titolo

Il titolo di Master Semipresenziale in Programmazione di Videogiochi garantisce, oltre alla specializzazione più rigorosa e aggiornata, l'accesso ad una qualifica di Master Semipresenziale rilasciata da TECH Università Tecnologica



“

Porta a termine questo programma e ricevi il tuo titolo universitario senza spostamenti o fastidiose formalità”

Questo **Master Semipresenziale in Programmazione di Videogiochi** possiede il programma più completo e aggiornato del panorama professionale e accademico.

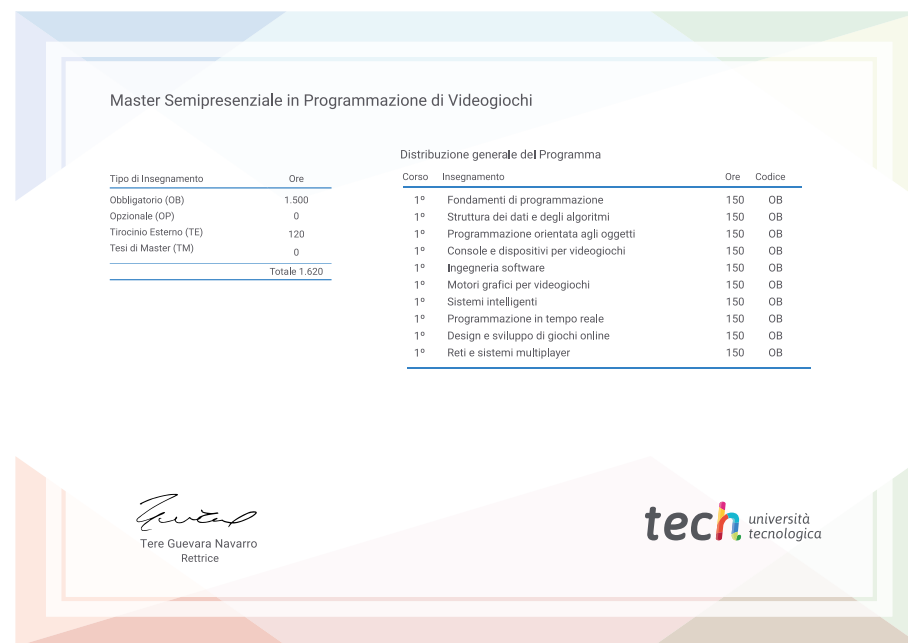
Dopo aver superato le valutazioni, lo studente riceverà mediante lettera certificata, con ricevuta di ritorno, la corrispondente qualifica di Master Semipresenziale rilasciata da TECH Università Tecnologica, che accrediterà il superamento delle valutazioni e l'acquisizione delle competenze del programma.

Oltre alla qualifica, sarà possibile ottenere un certificato e un attestato dei contenuti del programma. A tal fine, sarà necessario contattare il proprio consulente accademico, che fornirà tutte le informazioni necessarie. .

Titolo: **Master Semipresenziale in Programmazione di Videogiochi**

Modalità: **Semipresenziale (Online + Tirocinio)**

Durata: **12 mesi**



*Apostille dell'Aia. Se lo studente dovesse richiedere che il suo diploma cartaceo sia provvisto di Apostille dell'Aia, TECH EDUCATION effettuerà le gestioni opportune per ottenerla pagando un costo aggiuntivo.

futuro
salute fiducia persone
educazione informazione tutor
garanzia accreditamento insegnamento
istituzioni tecnologia apprendimento
comunità impegno
attenzione personalizzata innovazione
conoscenza presente qualità
formazione online
sviluppo istituzioni
classe virtuale lin

tech università
tecnologica

Master Semipresenziale
Programmazione di Videogiochi

Modalità: Semipresenziale (Online + Tirocinio)

Durata: 12 mesi

Titolo: TECH Università Tecnologica

Master Semipresenziale

Programmazione di Videogiochi

