

insegnamento	Strumenti di Game Development I
articolazione in moduli	nessuna
cfu	12
ore di didattica frontale	72
anno di corso	I anno
semestre	annuale
docente	Andrea Leganza
e-mail	a.leganza@unilink.it
ricevimento	Alla fine delle lezioni o per appuntamento da concordarsi via mail

RISULTATI DI APPRENDIMENTO ATTESI

L'insegnamento ha l'obiettivo di far conseguire i seguenti risultati di apprendimento:

1. **CONOSCENZA E CAPACITÀ DI COMPrensIONE:** lo studente dovrà essere in grado di comprendere i diversi aspetti e fasi legati alle fasi di sviluppo di un videogioco.
2. **CONOSCENZA E CAPACITÀ DI COMPrensIONE APPLICATE:** lo studente dovrà essere in grado di progettare e realizzare un progetto videoludico.
3. **AUTONOMIA DI GIUDIZIO:** lo studente dovrà essere in grado di valutare quali aspetti tecnici e tecnologici saranno fondamentali affinché il prodotto realizzato sia valido tecnicamente e coerente esteticamente.
4. **ABILITÀ COMUNICATIVE:** lo studente dovrà essere in grado di comunicare con il team di sviluppo in modo da fornire indicazioni chiare sulle specifiche necessità.
5. **ABILITÀ AD APPRENDERE:** lo studente dovrà comprendere quali aree di interesse dovrà approfondire per migliorare il prodotto software che vorrà realizzare.

PREREQUISITI

Nessuno

PROGRAMMA DETTAGLIATO

Il programma è strutturato in tre aree tematiche: strumenti e tecniche di sviluppo, fondamenti di programmazione e sviluppo tramite programmazione in un game engine,

- *Ciclo di vita di un software: Cascata, Spirale, Agile, Scrum, Develop, Kanban*
- *Cause del fallimento dei progetti software*
- *Verifica della fattibilità*
- *Budget di un progetto*
- *Struttura di un progetto software: GUI e codice*
- *Interfacce utente: diegetiche, non diegetiche, spaziali e meta*
- *Il team di sviluppo*
- *Strumenti software e licenze*
- *Formati di immagine Raster e Vettoriale*
- *Dispositivi hardware*
- *Gantt*
- *Unity 3D e Unreal*
- *Progettazione*
- *Tipologie di videogiochi*
- *GIT e sistemi di controllo versione distribuito*

- *Discord, Slack, Trello, e altre piattaforme di collaborazione e comunicazione.*
 - *API2D e 3D*
 - *Graphics pipeline*
 - *FBX, GLTF e formati 3D*
 - *Shaders*
 - *Metallic Roughness e Specular Glossiness*
 - *Maps: Diffuse, Normal, bump, height, displacement, Emissive, AO*
 - *Camera*
 - *Culling*
 - *Camera e Frustum*
 - *CPU e GPU bound*
 - *FPS, refresh rate, vsync.*
-
- *Fondamenti di programmazione in C#*
 - *Diagrammi di flusso*
 - *Buone pratiche di stile*
 - *Tipi di dato*
 - *Casting*
 - *Camel case*
 - *Integer Overflow*
 - *Costanti*
 - *Magic numbers*
 - *Operatori*
 - *Stringhe di testo*
 - *Strutture condizionali*
 - *Tabelle di verità*
 - *Cicli (for, while, do-while)*
 - *Array*
-
- *Unity Hub*
 - *Installare Unity3D*
 - *Realizzazione di progetti 2D e 3D*
 - *L'interfaccia grafica di Unity*
 - *Procedure per la creazione e distribuzione*
 - *La scena*
 - *GameObject e Componenti*
 - *Il componente Transform*
 - *La telecamera*
 - *Illuminazione*
 - *Flussi di lavoro*
 - *Importazione di modelli*
 - *Shaders, materiali e texture*
 - *Le geometrie standard*
 - *Creare un componente/script*
 - *Visual Studio e Visual Studio Code*
 - *I metodi Start, Awake, Update*
 - *Modificare la posizione*
 - *Modificare la rotazione*
 - *Modificare la scala*
 - *Interpolazione lineare*
 - *Creare variabili ed esporle nell'inspector*

- *Fisica di Unity*
- *La forza di gravità*
- *Collider e trigger*
- *Gestire le collisioni di tipo Collision e Trigger*
- *Gestire il click su un elemento della scena*
- *Materiali fisici*
- *Fixed Update*
- *Gestire più scene*
- *Passaggio tra Scene*
- *Coroutine*
- *UI system*
- *Post processing*
- *Conclusioni*

METODOLOGIE DIDATTICHE

Le lezioni sono erogate con approccio frontale, oltre agli aspetti prettamente teorici vengono mostrate simulazioni tecniche in tempo reale ed esposti casi reali.

MATERIALI DIDATTICI

Per la preparazione dell'esame è sufficiente consultare il materiale fornito dal docente.

MODALITÀ DI SVOLGIMENTO DELL'ESAME

Valutazione finale: esame scritto con domande a risposta multipla.

CRITERI DI VALUTAZIONE

Nell'esame finale, la/o studente dovrà dimostrare:

1. **CONOSCENZA E CAPACITÀ DI COMPrensIONE:** Lo studente dovrà essere in grado di comprendere sia i termini tecnici e le tecniche di sviluppo necessarie per realizzare un videogioco.
2. **CONOSCENZA E CAPACITÀ DI COMPrensIONE APPLICATE:** Lo studente dovrà essere in grado di descrivere i passi necessari per sviluppare tramite programmazione le dinamiche di programmazione richieste..
3. **AUTONOMIA DI GIUDIZIO:** Lo studente dovrà essere in grado di valutare le performance del prodotto realizzato in termini di programmazione e funzionamento.
4. **ABILITÀ COMUNICATIVE:** Lo studente dovrà acquisire padronanza del linguaggio specialistico della materia e maturare capacità di argomentare le scelte fatte.
5. **ABILITÀ AD APPRENDERE:** Lo studente dovrà dimostrare di aver compreso gli aspetti legati all'intera produzione di un videogioco, dagli aspetti teorici a quelli pratici.

CRITERI DI ATTRIBUZIONE DEL VOTO FINALE

Il voto si attribuisce in trentesimi, a cui si aggiunge possibilità di lode. Il voto finale rispecchia la preparazione dello studente come di seguito indicato:

- < 18 insufficiente: Conoscenze frammentarie e superficiali dei contenuti, errori nella comprensione dei concetti.
- 18-20: Conoscenze dei contenuti sufficienti ma generali, numerose incertezze nella comprensione di concetti teorici.
- 21-24: Conoscenze dei contenuti appropriate ma non approfondite.
- 25-30/lode: Conoscenze dei contenuti appropriate e approfondite.

CONSIGLI DEL/LA DOCENTE

Nessuno