

Курс Game Creator Online

BASIC LEVEL



подробнее о курсе

ИЗУЧАЕМЫЕ ТЕХНОЛОГИИ



Godot Engine



VS Code



Github

ПРОГРАММА КУРСА

1. Game Creator

GAME CONCEPT + GDD: СТВОРЮЄМО ВЛАСНУ ГРУ

Теми:

- Що таке Game Development
- Game Designer, Developer, Artist, Sound Designer
- Жанри відеоігор
- 2D і 3D
- Game Loop
- Game Mechanics
- Ціль гравця
- Персонаж і світ
- Перешкоди й нагороди
- Умови перемоги
- Основи Game Design Document

Практика: створюємо концепцію власної гри й заповнюємо спрощений GDD.

Результат: власний Game Concept: жанр, герой, світ, основна механіка й ціль гри.

GODOT: ЗНАЙОМСТВО З GAME ENGINE

Теми:

- Що таке Godot
- Інтерфейс Godot
- Project

- Scene
- Node
- Inspector
- 2D-режим
- Sprite
- Camera
- Структура проєкту

Практика: створення першого проєкту та першої ігрової сцени.

Результат: перший власний Godot-проєкт.

GODOT: СЦЕНИ Й ОБ'ЄКТИ

Теми:

- Scene Tree
- Nodes та їх призначення
- Parent / Child
- Node Types
- Instancing
- Resources
- Організація Game Project

Практика: створення базового рівня з декількома об'єктами.

Результат: структурований ігровий рівень.

2. GDScript — програмуємо власну гру

GDSCRIPT: ЗМІННІ, ТИПИ ДАНИХ Й ОПЕРАТОРИ

Теми:

- Що таке код
- GDScript
- Змінні
- Типи даних
- Numbers
- Strings
- Boolean
- Оператори
- Зміна значень

Практика: створюємо параметри персонажа й об'єктів гри: health, speed, score тощо.

Результат: учень розуміє, як зберігати та змінювати дані в грі.

GDSCRIPT: УМОВИ Й GAME LOGIC

Теми:

- if / else

- Порівняння
- Логічні оператори
- Умови
- Game Logic
- Реакція об'єктів на різні ситуації

Практика: створюємо логіку — якщо гравець зробив X → відбувається Y. Наприклад: health = 0 → Game Over.

Результат: перша самостійно створена ігрова логіка.

GDSCRIPT: ЦИКЛИ, МАСИВИ ТА СЛОВНИКИ

Теми:

- Що таке цикл
- for
- while
- Array
- Елементи масиву
- Додавання та видалення елементів
- Dictionary
- Key / Value
- Перебір масивів за допомогою циклів

Практика: створюємо список предметів у грі й оброблюємо його за допомогою циклів.

Результат: учень розуміє, як зберігати й обробляти набори даних у грі.

GDSCRIPT: ФУНКЦІЇ ТА СТРУКТУРА КОДУ

Теми:

- Functions
- Parameters
- Arguments
- Return
- Повторне використання коду
- Організація коду
- Структура скрипта

Практика: створюємо власні функції для ігрових дій:

- take_damage()
- collect_item()
- add_score()
- open_door()

Результат: учень не просто копіює код, а розуміє його структуру й може створити власну функцію.

GDSCRIPT: INPUT, EVENTS I SIGNALS

Теми:

- Input
- Events
- Signals
- Реакція на дії гравця
- Взаємодія між об'єктами
- Передача подій між сценами

Практика: натискання кнопки → об'єкт реагує → відбувається дія.

Результат: перша повноцінна взаємодія між об'єктами.

MINI GAME: ЗБИРАЄМО ПЕРШУ ГРУ

Теми:

- Застосування GDScript на практиці
- Player Input
- Score
- Collectibles
- Timer
- Умови перемоги
- Game Over
- Об'єднання різних механік

Практика: створюємо просту гру, наприклад: «Збери якомога більше монет за 30 секунд».

Використовуємо змінні, умови, цикли, масиви, функції, Input і Signals.

Результат: перша завершена мінігра.

3. Від коду до справжньої гри

INTERACTION: ВЗАЄМОДІЯ ОБ'ЄКТІВ

Теми:

- Player Interaction
- Trigger
- Area
- Collision Detection
- Signals на практиці
- Взаємодія з предметами

Практика: герой підходить до предмета → натискає кнопку → предмет змінюється або запускає дію.

Результат: працююча система взаємодії.

SCENES & REUSABLE OBJECTS

Теми:

- Окремі сцени
- Reusable Objects
- Instancing

- Зв'язок сцен
- Організація Game Project

Практика: створюємо об'єкти, які можна багаторазово використовувати на рівні.

Результат: власний набір reusable game objects.

CHARACTER: СТВОРЕННЯ ГЕРОЯ

Теми:

- CharacterBody
- Sprite
- Animation
- Idle
- Walk
- Run
- Jump
- Input

Практика: створення керованого персонажа.

Результат: герой рухається по рівню.

CHARACTER: АНІМАЦІЇ ТА КЕРУВАННЯ

Теми:

- AnimationPlayer / AnimationTree
- Напрямок руху
- Зміна анімацій
- Стани персонажа
- Плавність руху
- Зв'язок руху й анімацій

Практика: пов'язуємо рух персонажа з анімаціями.

Результат: повноцінний animated character.

4. Physics + Level Design

PHYSICS & COLLISION

Теми:

- Game Physics
- Gravity
- Collision
- Static / Dynamic Objects
- Floor
- Walls
- Platforms
- Trigger Zones

Практика: створення фізики рівня.

Результат: герой ходить, стрибає та взаємодіє з фізичним світом.

LEVEL DESIGN

Теми:

- Структура рівня
- Spawn Point
- Перешкоди
- Секрети
- Нагороди
- Checkpoint
- Напрямок руху гравця
- Складність

Практика: створення повноцінного рівня.

Результат: Playable Level.

COLLECTIBLES & GAMEPLAY LOOP

Теми:

- Collectibles
- Предмети
- Бонуси
- Очки
- Timer
- Ціль рівня
- Умови перемоги
- Gameplay Loop

Практика: рух → збір → перешкоди → нагорода → завершення рівня

Результат: повноцінна ігрова петля.

5. Вороги й механіки

ENEMY & SIMPLE AI

Теми:

- NPC
- Enemy
- Основи AI
- Patrol
- Detection
- Переслідування
- Реакція на гравця

Практика: створення ворога з базовою поведінкою.

Результат: ворог самостійно реагує на гравця.

COMBAT SYSTEM

Теми:

- Attack
- Damage
- Health
- Death
- Invulnerability
- Player vs Enemy

Практика: створення бойової механіки.

Результат: гравець може атакувати ворогів і отримувати damage.

GAME MECHANICS & PROGRESSION

Теми:

- Core Gameplay Loop
- Rewards
- Challenges
- Progression
- Difficulty
- Комбінація механік
- Win / Lose Conditions

Практика: створення і розширення власної gameplay mechanic.

Результат: учень додає до гри власну механіку й систему прогресії.

6. Game Systems

UI & HUD

Теми:

- Game UI
- HUD
- Health Bar
- Score
- Timer
- Buttons
- Menu
- Pause
- Game Over
- Victory Screen

Практика: створення UI навчальної гри.

Результат: гра отримує повноцінний інтерфейс.

GAME STATES & GAME MANAGER

Теми:

- Game States
- Start Game
- Playing
- Pause
- Game Over
- Victory
- Restart
- Game Manager
- Перехід між станами

Практика: Start → Play → Pause → Win/Lose → Restart

Результат: повний цикл гри.

INVENTORY, SAVE & GAME SYSTEMS

Теми:

- Inventory
- Items
- Checkpoints
- Save / Load
- Збереження прогресу
- Game Data

Практика: додаємо до гри одну або декілька систем залежно від складності проекту.

Результат: гра отримує базову систему прогресу та збереження.

GAME AUDIO, ART & AI

Теми:

- Game Audio
- Music
- Sound Effects
- Game Assets
- Sprite / Tileset
- Background
- Visual Style
- AI для створення концептів й assets

Практика:

- додаємо звук;
- підбираємо assets;

- створюємо або адаптуємо візуальний стиль;
 - за бажанням використовуємо AI для окремих елементів.
- Результат: гра отримує власний візуальний і звуковий стиль.
-

7. Власний Game Project

GAME STUDIO: СТВОРЮЄМО ТА ПОЛІРУЄМО ВЛАСНУ ГРУ

Теми:

- Концепція
- Персонаж
- Рівень
- Механіки
- Вороги
- UI
- Assets
- Звук
- Testing
- Debugging
- Balance
- Difficulty
- Feedback
- Polish
- Анімації
- Ефекти
- Screen Shake / Transitions
- GitHub

Практика: учень завершує власну гру, тестує її, знаходить і виправляє баги, додає фінальні покращення та завантажує проєкт на GitHub.

Результат: готова фінальна версія власної гри.

HILLEL GAME CREATOR SHOWCASE: ФІНАЛЬНЕ ЗАНЯТТЯ — ПРЕЗЕНТАЦІЯ ВЛАСНИХ ІГОР

Кожен учень:

- запускає свою гру;
- показує gameplay;
- демонструє головну механіку;
- показує персонажа та рівень;
- розповідає про найцікавіше рішення;
- показує, що було найскладнішим.

Без окремої презентації — головне запустити гру й показати її наживо.

Результат: власна готова гра й досвід її публічної демонстрації.



В конце курса выполняется дипломный проект.