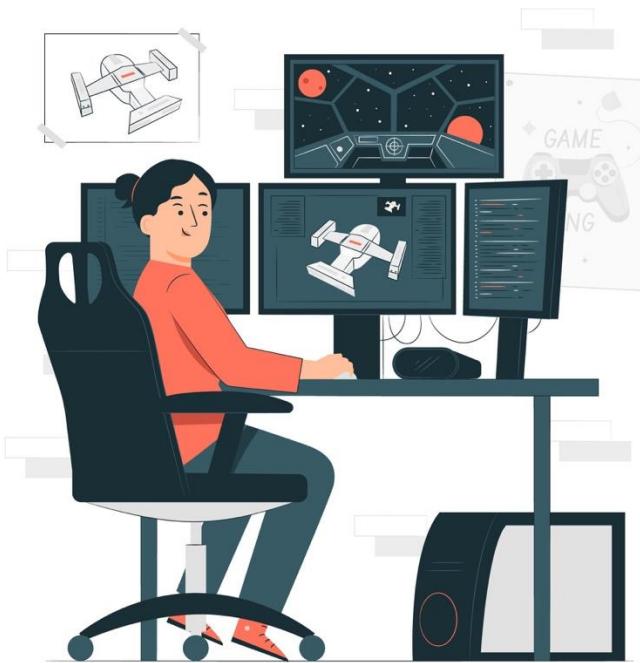


Game Development

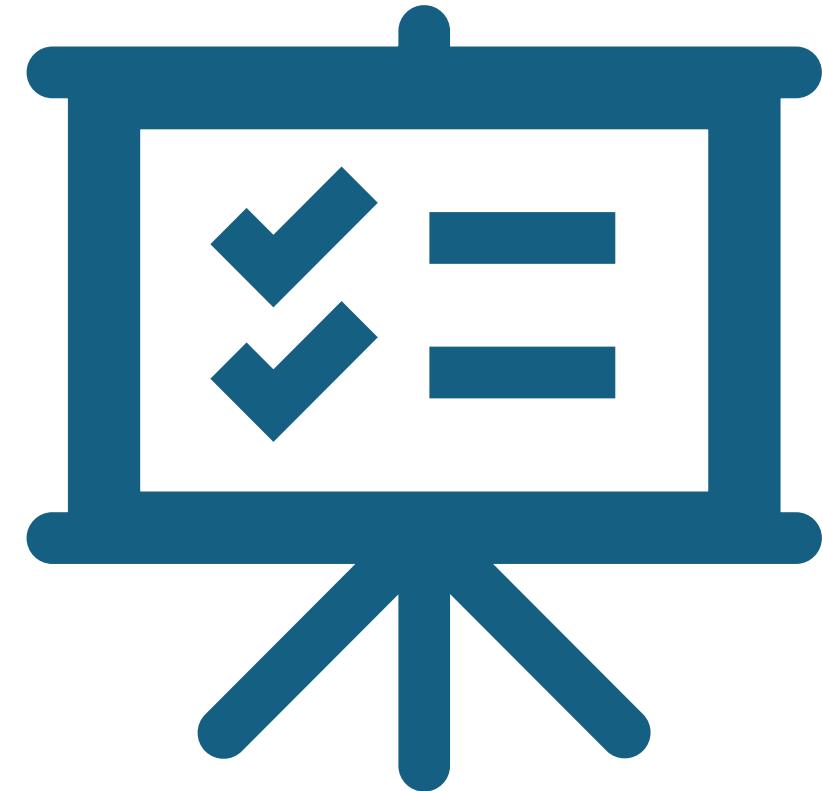
Final Project Introduction



Lecturer : Dr. VA Hongly

Session Objective

- ✓ Introduction
- ✓ Core Requirements & Scope
- ✓ Key Milestones & Timeline
- ✓ Grading Rubric
- ✓ List of proposed project
- ✓ Q&A
- ✓ Sharing Session



Course Agenda

Week	Activity	In-Class
1	Course Kick-off & Unity Setup	Syllabus, game-dev overview, install Unity + IDE, create first project
2	Exploring UI	Editor tour (Hierarchy/Inspector), GameObjects & Components, play mode
3	Scripting	Basic C#, Explore build-in function & lifecycle
4	Working with Assets & Physics	Import/organize assets, colliders & rigidbodies, forces/triggers, simple particles
5	Building a Simple 2D Game	2D workflow (sprites/tilemap), player controller, camera follow, basic UI, win/lose
<u>Project Introduction</u>		
6	Camera & Lighting	Cinemachine basics, ortho vs perspective, lights & shadows, mood
7	Building 3D Game	3D workflow, move, interactions
8	2D Animation	Introduction to 2D animation (sprite animation, flipbook animation, bone-based), Animator Controller
9	3D Animation	Introduction to 3D animation (keyframe animation, rigging, blend trees), animators for characters
10	<u>Preparation Week</u>	
Final Project Presentation		Present, Demo, Q/A, peer feedback

Project Introduction



A final project (**40% grade**)

Group project

- Apply:
 - Programming skills
 - Design
 - Teamwork
 - Problem-solving



Purpose

- Applying Theoretical Knowledge Practically
- Encouraging Collaboration and Teamwork
- Developing Creativity and Design Thinking
- Understanding Project and Time Management

Project Completion Process

Ideation: Form a team and define a game concept, genre, and core mechanics.

GDD: Create a short 1–2-page design document.

Prototype: Build basic gameplay loop and main scene.

Iteration: Add UI, extra features, assets, and audio.

Alpha Build: Produce a playable version with bug fixes.

Final Build: Polish visuals, controls, and performance.

Presentation: Submit build file, video demo, and documentation.

Define Group Members

- Form a team 2~4 members
- Role & Responsibility:
 - Leader Role
 - Each group must assign a leader responsible for coordinating tasks, managing deadlines, and communicating with the lecturer.
 - Member
 - Assign specific roles based on skills and interests
 - Suggested roles:
 - Coder/Programmer
 - writing and debugging code, implementing core mechanics
 - Designer
 - Focuses on the visual elements of the project such as Sprites and Textures, UI, Asset Usage etc.,
 - Audio, Animation, and Tester (can be included)



GDD

1. Game Overview

- Title, genre, target audience, game goal.

2. Story & Setting

- Short world description and main challenge.

3. Core Gameplay

- Player abilities, interactions, enemies/obstacles.
- Win/lose conditions.

4. Gameplay Flow

- Menu → Gameplay → End/Result.

5. Art Direction

- Style (pixel, low-poly, cartoon).
- Key assets list.

6. Audio

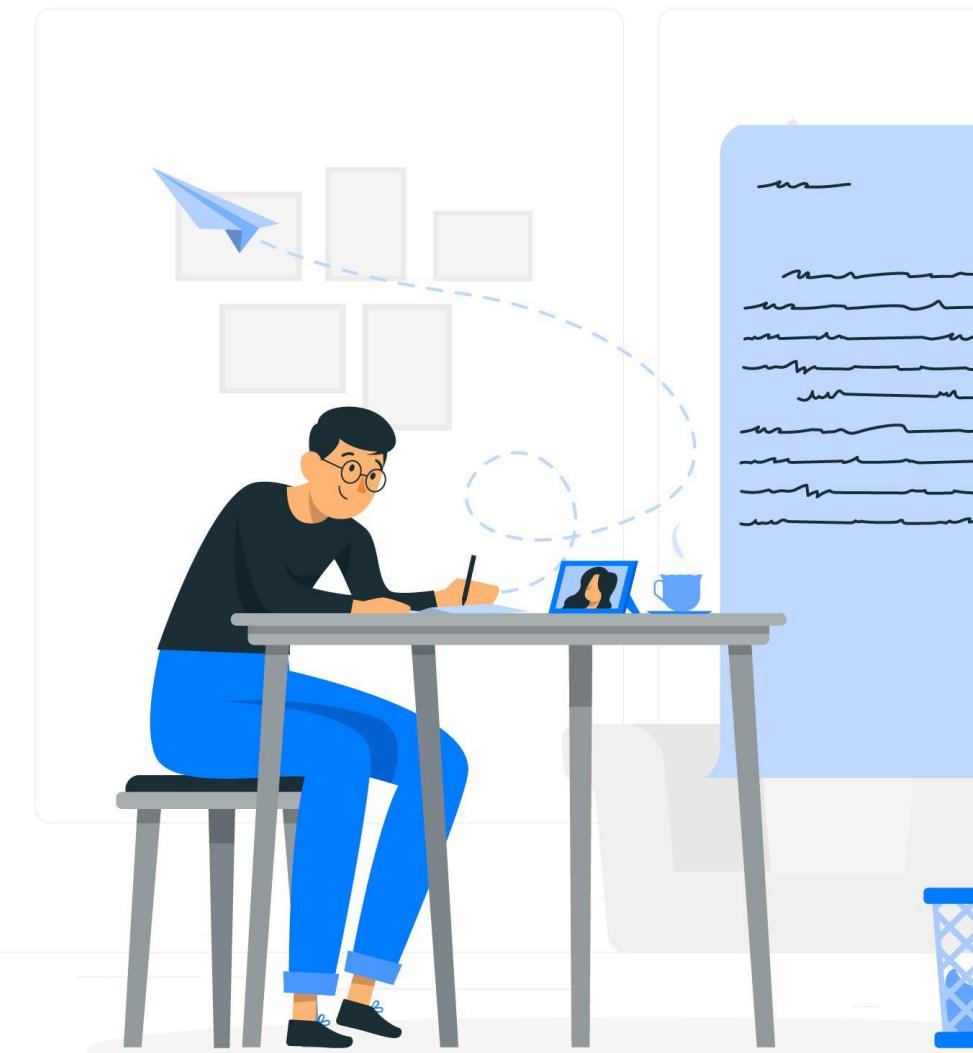
- Music style + required SFX.

7. Technical Requirements

- Engine (Unity 2D/3D), key scripts, platform, controls.

8. Scope

- Minimum 2 scenes, manageable mechanics, free assets allowed.



Project Development Phase

- **3–4 follow-up sessions** with instructor
- Weekly progress update from each team:
 - What have you **completed**?
 - What **challenges** are you facing?
 - What will you work on **next**?



Project Presentation

- Suggested outline
 - **Introduction:** Brief overview of the game (title, concept, genre).
 - **Gameplay & Core Mechanics:** Show how the game works and what makes it unique.
 - **Design & Art Direction:** Present visuals, style, UI, and animations.
 - **Technical Implementation:** Key scripts, systems, or features you developed.
 - **Conclusion & Future Improvement:** What you would add if you had more time.
 - **Demo Playthrough:** Short live/recorded gameplay showcase.
- Duration: **12 minutes** (7min presentation + 5 min Q/A)
- Committee(s): **TBD**
- Date: **24/12/2025 to 25/12/2025**



Submission Checklist

- ✓ **Slide Presentation**
 - As PDF or URL
 - Record demo of the game
- ✓ **GitHub Url:**
 - Include assets, scripts, and configuration files.
 - Provide instructions on installation, project information, setup requirements
- ✓ **Compiled Application (optional but recommend):**
 - Build the application to the target platform (e.g., executable file for PC, APK for Android).
- **Deadline: December 23, 2025**



Main Requirement



Do

- Build a **playable** 2D or 3D game.
- Include **core mechanics, UI, audio, and win/lose conditions**.
- Use **Unity** (any version).
- Create **at least 2 scenes**: Main Menu + Gameplay.
- Use legal/free assets or create your own.
- Ensure the game is **appropriate for an academic environment**.
- **Encouraged:**
 - **Educational games** (learning, awareness, problem-solving).
 - **Khmer-themed games** (culture, story, environment, folk tales).



Don't

- Include **inappropriate or sensitive content** (adult themes, explicit scenes).
- Add **real-money mechanics** or chance-based reward systems.
- Include **hate speech**, discrimination, or culturally offensive material.
- Use unlicensed or copyrighted paid assets without permission.

Bonus Graphic Requirement

No.	Optional Requirement	Explanation
1	Graphics Optimization	<ul style="list-style-type: none"> - Implement techniques to improve rendering performance and frame rates, such as reducing draw calls, optimizing textures, and using efficient shaders. - This helps ensure the game runs smoothly, especially on lower-end devices.
2	Physically-Based Rendering (PBR)	<ul style="list-style-type: none"> - Use PBR techniques to create materials that react realistically to lighting, improving the visual fidelity of surfaces. - This can enhance the immersion and realism of the game environment.
3	Level of Detail (LOD)	<ul style="list-style-type: none"> - Implement LOD techniques to reduce the complexity of 3D models based on their distance from the camera. - This optimization reduces the rendering load and improves performance without sacrificing visual quality.
4	Camera Culling	<ul style="list-style-type: none"> - Use culling techniques to prevent the rendering of objects that are not visible to the camera (e.g., objects behind the camera or out of view). - This optimizes performance by reducing the number of objects processed per frame.
5	Advanced Lighting	<ul style="list-style-type: none"> - Implement advanced lighting techniques, such as dynamic shadows, reflection probes, and global illumination. - These techniques can greatly enhance the realism and depth of the game environment.

- *These are **optional features** that you can include in your project.*
- ***Extra points will be awarded** to teams that successfully implement any of the following advanced features*

Bonus Game Requirement

No.	Optional Requirement	Explanation
1	AI and NPC Behavior	<ul style="list-style-type: none"> - Implement artificial intelligence for non-player characters (NPCs) to create realistic behaviors and interactions. - This adds depth to the gameplay experience and enhances immersion.
2	Procedural Generation	<ul style="list-style-type: none"> - Use procedural generation techniques to create random or semi-random levels or environments. - This can enhance replayability and provide a unique experience each time the game is played.
3	Integration with Social Media	<ul style="list-style-type: none"> - Allow players to share achievements, screenshots, or progress on social media platforms. - This can promote the game and engage a broader audience.
4	Multiplayer Capability	<ul style="list-style-type: none"> - Incorporate multiplayer functionality to allow multiple players to interact within the game. - This can involve creating a local or online multiplayer experience, requiring additional networking knowledge.
5	Customizable Character Options	<ul style="list-style-type: none"> - Allow players to customize characters or avatars (e.g., appearance, abilities, or equipment). - Customization adds a layer of engagement and personalization to the game.
6	Augmented Reality (AR)	Integrate AR features that allow users to interact with the game environment through their device's camera.
7	Virtual Reality (VR)	Develop a VR version of the game to provide an immersive experience using VR headsets.

- *These are **optional features** that you can include in your project.*
- ***Extra points will be awarded** to teams that successfully implement any of the following advanced features:*

Evaluation

Criteria	Ratio
Concept and Objective	15
Design	15
Technical Implementation	30
Development Process	10
Presentation	20
Project File	5
Bonus	5 + (extra point)

Game Idea

- Puzzle Game Based on Traditional Khmer Games
- CADT Campus Tour
- Khmer Language Learning Quest Game
- Physics Simulation for High School Student
- Environmental Awareness Game
- Khmer Folk Tale Adventure
- Traffic Safety Mini-Game
- Math Challenge Runner
- Food & Nutrition Quest
- Cultural Heritage Explorer

FAQ

Q1: Can we change our game idea after Week 1?

A1: Minor adjustments are allowed, but the core concept must remain consistent after the GDD submission.

Q2: Can we add features not listed in the requirements?

A2: Yes, extra features may earn bonus points if implemented correctly.

Q3: Can we use AI-generated art or sound?

A3: Yes, but only if the content is safe, original, and legally usable. You must credit the tool and ensure no copyrighted material is replicated.

Q4: What happens if some features are not finished by Week 5?

A4: Submit a playable core version. Unfinished features can be explained in the “Future Improvement” part of your presentation.

Q5: Can we reuse the concept from previous assignments?

A5: Definitely Yes. As long as it cover all the requirements