Game Design versus Game Development

Robert Bruce

CS-330: Introduction to Game Programming

Game Design vs. Game Development: Roles

- "Video game designer"
- "Graphic designer"
- "Game animator"
- "2D/3D artist"
- "Character designer"

- "Game programmer"
- "Gameplay engineer"
- "Artificial intelligence (AI) programmer"
- "Quality assurance (QA) game tester"
- "Systems designer"

Source: https://emeritus.org/blog/technology-video-game-designer-vs-developer/

Game Designer: Key Stages in Game Creation

- "Vision and Conceptualization"
  - In this stage, game developers produce concept art which may go through multiple changes based on current gaming trends and tastes.
- "Plot and Characters"
  - In this stage, the game designers write the story with a focus on engaging plot twists and memorable character personalities.
- "Rules and Gameplay"
  - In this stage, the game designers outline the game mechanics.
- "User Interface and Levels"
  - In this stage, the game designers create an intuitive visual user-interface (a wire diagram) and outline the progressive levels of gameplay through storyboards.
- "Teamwork and Collaboration"
  - This stage occurs in varying degrees throughout the game design and game development process.
  - Everyone should have a clearly identified role on game design and/or game development.

Source: https://olibr.com/blog/game-designer-vs-game-developer/

Game Developer: Key Stages in Game Creation

Note: all the stages listed below may occur to some extent concurrently depending on the size of the Game Developer team, budget to develop the game, and effective team communication and collaboration. Although Quality Assurance is usually applied after the Minimum Viable Product (MVP) is achieved, that is not always the case. For instance, the Quality Assurance (QA) could be testing portions of a custom-made Physics engine in stages before the Physics Engine is incorporated into the game's MVP. The QA team may also develop custom stress-testing tools during development phases of the game's MVP prototype as a means of saving time and working efficiently. Communication is a critical factor to efficient project management.

- "Coding and Programming"
  - In this stage, the game developers are codifying the game design into an interactive game based on a minimum viable product.
- "Artificial Intelligence and Logic"
  - In this stage, the game developers are developing the game logic to control NPC (non-playing characters) such as enemies.
- "Physics and Graphics"
  - In this stage, the game developers are programming their own physics engine or utilizing an existing physics game engine for robust and realistic game play.
- "Quality Assurance and Testing"
  - In this stage, the game developers are determining what, if any, errors or defects exist in the functionality and playability of the game. This may involve writing software to stress-test certain features within the game.

Source: https://olibr.com/blog/game-designer-vs-game-developer/

Elements of Successful Game Design

Note: These ideas are more deeply addressed in game mechanics (a future lecture).

1. "Optimizing the Player Experience"
   - User-interface is intuitive and easy to navigate.
2. "Building an Engaging Narrative"
   - Keep the player entertained.
   - Controversial: make the game addictive.
3. "Pay Attention to the Audio Effects"
   - Use music to impact player mood.
   - Use sound effects as hints of nearby treasure.
   - Use sound effects to warn players of eminent danger.
4. "Ensure a Seamless Control Scheme"
   - Game controls should be intuitive to control the player's character.
5. "Create a Powerful Visual Aesthetic"
   - Visual effects and graphics impact the immersive factor of the game (e.g. Fable series, Call of Duty series, Crysis).
   - Tip: simple computer graphics can also be immersive if the game is fun to play (e.g. vector graphics like Tempest and Asteroids; or 2D games like Pac Man, Donkey Kong, or Q*Bert).

Source: https://emeritus.org/blog/technology-game-design-mit-xpro/

Game Design: Responsibilities

- Outline: game story
- Outline: game goals, objectives, target audience
- Art: designing art assets for the game
  - Illustrating: 2D drawings of all characters, objects in game, and the user-interface in the game (e.g. heads-up display, game menus)
  - Storyboarding: describes the entire game visually through a series of scenes (modelling characters, objects, environment, etc.)
  - Texturing: defines the look for clothing, environment (bark on trees, leaf patterns, stones), game objects (jewels, sword handle, leather bags, weathered wood on doors or treasure chests), character's skin, etc.
  - Technical animating: character rigging and controls for characters and objects, developing bounding boxes around characters or objects for collision detection.
  - Animating: player characters (hero or heroine), enemies, and objects (crossbow shooting a bolt), etc.
  - Lighting: casting light on overall scene, characters, and objects.
  - Music composing: defines the vibe or feeling in the game
  - Voice acting: recording the actor's voice and facial expression (to be later applied to characters in the game)
  - Music composing: defining the musical themes throughout the game.
  - Sound and foley designing: defining the sounds that occur throughout the game such as walking sounds, clanging of two swords in battle, the whisp of a flying arrow, a gunshot, the casting of a spell, etc.
  - Motion capturing: this is more a technique that could be combined with traditional animation.
- Producers: keeps track of timelines, budget, and spending for the game project.

Game Development: Responsibilities

- Software development:
  - Controller event programming: which keys did the player press to manipulate a character or object?
  - Sound effects programming: which sounds are invoked during the game?
  - Visual effects programming: programming shaders for fluid simulation, smoke simulation, fire simulation, special lighting, etc. This can be programmed algorithmically rather than through traditional animation.
  - Artificial Intelligence (AI) programming: procedural animation (automation in animation) of enemy characters or objects (a flag that moves in the wind, trees that sway in the wind, etc.
  - Physics simulation programming: developing algorithms for ridged (sword or rocks), soft-body simulations (cloth), and particle simulations (smoke, fire, explosions, water).
  - Network programming: the infrastructure for massively multi-player game capability.
  - User-interface (UI) programming: programming the functionality when user clicks through menus or displaying and updating attributes in the heads-up display (HUD).
  - Quality Assurance (QA) testing: determines if there are any broken features within the game may also assess the gaming experience.\
  - Game Programmers: define the minimum viable product in coordination with Producers and Art team. Also responsible for the overall implementation of the game. Responsibilities may include developing a custom back-end physics engine and rendering engine with accompanying API (application programmer interface) or the use of an existing game engine (Godot, Unreal, Unity, Blender Game Engine, CryEngine, etc.).

What will be your role as a student in CS-330?

Creating a game is massively interdisciplinary!

- Your responsibilities within a team in CS-330 will likely span: Game Design and Game Development.

Game Design examples:

- Your team may create crude or detailed storyboards or illustrations for the game you intend to create. Examples: the layout of a Heads-Up Display (HUD) or the game board.
- Your team may create your own sound effects with creative recording with your smart-phone. *Note: your team is not required to make original sound effects though for CS-330.*
- Your team may use art assets with permissive licenses (such as MIT, Creative Commons, etc.) for characters, background, game field, sound effects, music, etc.

Game Programming examples:

- Your team will be programming events within Godot to read user-input (keyboard, mouse, joystick, etc.) to control the player's character.
- Your team will be programming events such as how the character interacts with other NPC (non-playing characters) or objects within the game.
- Your team will probably be testing your game for Quality Assurance in the testing phase.

Improving your future employability?

Can game development improve your candidacy as a job prospect?

- Absolutely yes!

You are demonstrating experience in:

- User-interface design and user-experience.
- Managing a project based on specifications your team defined (i.e. Minimum Viable Product)
- Team collaboration with effective oral and written communication (very important to have regardless of the product developed).

The next steps are thinking strategically about how you can showcase your skills to future employers.

- Reflect upon your project and think about ways it could be relevant to future employers (even if the future employer is not in the gaming industry).

Level-up your knowledge: a deep dive

For Game Design:

- *Animation Writing and Development: From Script Development to Pitch* by Jean Ann Wright.
- *Cartoon Animation* by Preston Blair.
- *Don't Make Me Think* by Steve Krug.
- *Exploring Storyboarding* by Wendy Tumminello.
- *Guerrilla Home Recording: How to get great sound from any studio (no matter how weird or cheap your gear is)* by Karl Coryat.
- *The Animator's Survival Kit* by Richard Williams.
- *The Illusion of Life* by Frank Thomas and Ollie Johnston.
- *Voice-Over for Animation* by Jean Ann Wright and M. J. Lallo.

Level-up your knowledge: a deep dive (continued)

For Game Development:

- *3D Math Primer for Graphics and Game Development (2nd edition)* by Fletcher Dunn and Ian Parberry
- *Advanced Character Physics* by Thomas Jakobsen.
- *Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behavior* by Xavier Provot.
- *Physics for game developers* (2nd edition) by David M. Bourg and Bryan Bywalec.
- *Real time physics class notes* by Matthias Müller, Jos Stam, Doug James, and Nils Thürey.

Level-up your knowledge: a deep dive (continued)

For Game Development:

- *3D Math Primer for Graphics and Game Development (2nd edition)* by Fletcher Dunn and Ian Parberry
- *Advanced Character Physics* by Thomas Jakobsen.
- *Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behavior* by Xavier Provot.
- *Physics for game developers* (2nd edition) by David M. Bourg and Bryan Bywalec.
- *Real time physics class notes* by Matthias Müller, Jos Stam, Doug James, and Nils Thürey.