

Game Development

David Adair, Tom Bielecki,
Rajvir Jhawar, Brandon Koepke,
Ian Macaulay, and Brett Pelletier

Outline

- Unique challenges in game development
 - Requirements
 - Architectures and Design
 - Testing
 - Creative Interaction
 - Frameworks
- Questions?

Requirements

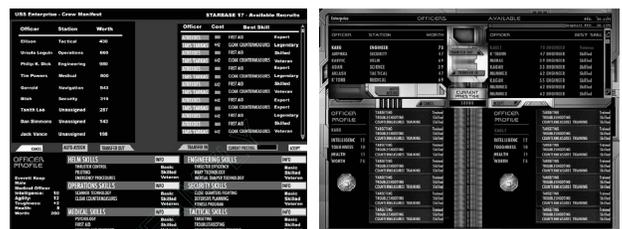
Requirements

- Requirements elicitation is difficult
- Three Activities
 - Elicitation of requirements from individual sources
 - Ensuring needs are consistent and feasible
 - Validate that the requirements are accurate
- Main Issues
 - Scope
 - Understanding
 - Volatility

Requirements

- Game design should describe the:
 - Game definition
 - Core Gameplay
 - Contextual Gameplay
 - Talk Story
 - Game Assets
- Requirements take several forms, examples:
 - Frame-rate requirements
 - Length of time to produce the product
 - Having few defects

Requirements

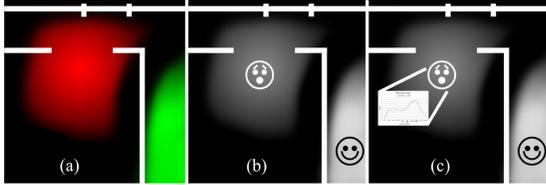


Mockup

Final

Requirements

- Emotional requirements
 - Designers are concerned about the player's game experience
 - Difficult to quantify 'fun' and 'engaging'
 - Must capture the intent of the designer



Architecture and Design

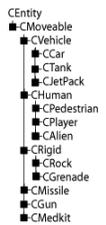
Architecture and Design

- Must facilitate functional and nonfunctional requirements
 - Adaptability (DLC, expansions)
 - Performance (load balancing, threading)
- Two primary design paradigms
 - Object-Hierarchy Trees (OOP)
 - Entity-Component Systems (EC)
- Industry shift from OOP to EC



Architecture and Design

- Primary development paradigm in the past
- Focus on classifying and modelling entities as they would look and behave in the real world
- Can become an obstacle with very deep object hierarchies



"...functionality can be added or changed in a single place in the code to affect many different game units. One such change inadvertently added line-of-sight checking for trees."

-Herb Marslas, Age of Empires II

Architecture and Design

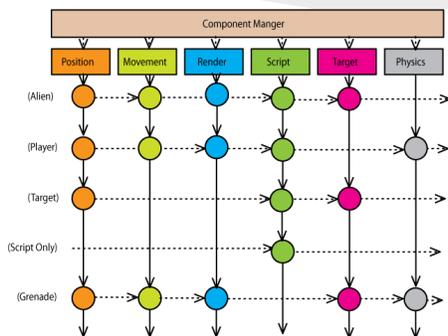


Figure 2: Object composition using components, viewed as a grid.

Architecture and Design

- Future is designer-level object creation
 - Developers moved almost entirely to engine, front/back-end development, complex AI behavior, and component managers

```

    ◦ Data-driven interface like Skrit
    [[:chicken_red,n:0x837FD928]]
    {
    [placement]
    {
    p position = 1.3,0,1.8,0x1738FFDB;
    q orientation = 0.3828,0.2384,-0.7772,0.98;
    }
    [common]
    {
    screen_name = "Super Chicken";
    }
    [body]
    {
    avg_move_velocity = 18.000000;
    }
    }
    
```

Architecture and Design

- Future is functional programming
 - Per-core processing power reaching plateau
 - Tech companies adding more cores
 - High performance games must take advantage of the additional cores
- Purely-functional game assets allow for iterative/agile development
 - Entities/components/assets have no side-effects, can be swapped/modified without concern
- Understanding strengths of functional can allow for:
 - existing design paradigms to be augmented

Game Testing



Functionality Testing

- The games has to be playable
- Tests for
 - Stability
 - Game mechanic
 - Integrity of the game
- Release



Compliance Testing

- Each platform has its own standard of testing.
- This could include
 - error messages
 - copyright material
 - trademarks
 - save data handler
- If not meet this can cause the game to be delayed or canceled on that platform



Compatibility Testing

- Tests the hardware of the game
 - Joysticks
 - Controllers
 - Mouse/Keyboard
- Test the performance of the minimum advertised requirements



Multiplayer Testing

- Used for games with multiplayer feature
- Tests the connectivity methods
 - modem
 - LAN
 - internet
- Play



Load Testing

- Test the limitation of a system
 - Number of players
 - Number of sprites active on the screen
- This test requires
 - Large group of testers
 - software to emulate heavy activity



Game Testing Continued...



Localization Testing

- Translation of game content
- Tester must understand regional differences between countries
 - Germany does not permit games having references to the Nazi party
 - Japanese and American cultures are very different



Itagaki was asked if he was fine with "sexualizing" a girl who is 17 years old, and he answered with: "In Japan, that's okay. Maybe it's 20 in America."

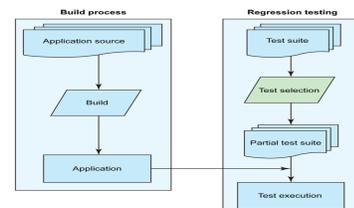
Soak Testing

- Testing how the game runs on the system for prolonged periods of time with significant load
- Different modes of operation
 - title screen
 - start menu
 - paused game
- Significant loads
 - mouse clicks
 - button presses
 - any other inputs
- Helps to developer find memory leaks and rounding errors



Regression Testing

- Testing to find any faults/bugs in the code after a patch, configuration change or bug fix
 - "Regression" meaning to look back and recheck
 - Determine if the change had detrimental effects



Beta Testing

- "Beta" versions of the game released to the general public or select individuals
- Cheap and generally effective way to find bugs in game
- "Hitting two birds with one stone"
 - Free advertising/ hype for game
 - Gamers will find bugs developers could not



Creative Interaction

Creative Interaction

Video games as a medium is almost entirely defined by interactivity

In gaming's "look and feel", interactivity describes the 'feeling', comprised of two parts:

- Control
- Feedback

Creative Interaction

Control Mechanisms and mashups



Creative Interaction

Control Gimmicks?



Creative Interaction

Feedback

Control is useless without feedback

Psychology and cognitive science take a big part

The goal is to make you feel physically and visually connected to the world that the game is trying to portray

Creative Interaction

Types of Feedback

- Audio
 - Virtual/physical surround-sound 3D soundscape
- Visual
 - 3D technologies
 - VR goggles
 - Projections / Augmented reality
- Tactile
 - Haptic feedback
 - Sensory substitution
 - Gaming chairs

Creative Interactions

Feedback Challenges

- Seamlessness
 - Very hard to reproduce 'realistic' game worlds in real-time!
- Performance
 - Lag (processing, network)
- "Uncanny Valley"
 - Interactions and feedback that skirt a fine line of being not quite real enough
- Intuitive Control/Feedback loop
 - Many nuanced ways for customers to not 'feel right'

Frameworks

Frameworks

- Provide functionalities such as Rendering, Physics, User Input, Collision Detection, and Animation
- Today, most frameworks are provided in the form of a Game Engine which provide the necessary APIs, and SDKs to develop games for various platforms



Types of Frameworks

- Low-Level
 - Developers combine different APIs to develop their own engine
 - Advantages:
 - Gives developers the flexibility of choosing their own components
 - Disadvantages:
 - Usually have higher costs and possible technical issues

Types of Frameworks

- **Mid-Level**
 - Have most features out of the box but still require programming behind the scenes
 - Features include rendering, physics, user interface, collision detection, and much more
 - Examples:
 - Unreal Engine 3
 - CryENGINE 3

Types of Frameworks

- **High-Level**
 - Point and Click Development
 - Most common today due to their ease of use and speed of development
 - Disadvantage:
 - Can be limiting in the types of games or the quality of graphics that can be produced
 - Examples:
 - Unity3D
 - Game Maker

Popular Frameworks

- **Unreal Development Kit**
 - Development Kit for Unreal Engine 3
 - Commercial and Free Non-Commercial Versions
- **Unity 3D**
 - Cross Platform Development Kit
 - Free version can be used to develop commercial games
- **Cocos 2D**
 - Open Source
 - Cross Platform Framework

Thank You!

Questions?