Game Design Documents

CIS 487/587

Bruce R. Maxim

UM-Dearborn

Communication

- Documentation
  - Methods vary widely
  - Written, descriptive model of the game
    - Depth varies according to the needs of the game

Design Documentation Stages

- Design treatment or concept paper
  - Game feasibility
- Design summary/design documents
  - Pitch document or proposal
- Design specification/product specification/production document
  - Functional product specification

Game Treatment

- Game story
  - Abstract or “Reader’s Digest” type overview
- Game play and look
  - Focus on appearance
  - Player roles and actions
  - Strategies and motivations
- Development Specification
  - Hardware
  - Software
  - Algorithm style

Communication

- **Treatment**
  - A brief, general description of the game and the fundamental concepts
  - May include:
    - Concept statement
    - Goals and objectives
    - Core mechanics and systems
    - Competitive analysis
    - Licensing and IP information
    - Target platform and audience
    - Scope
    - Key features

Sample Development Specification

- This game uses a new 3D engine
- Backgrounds are animated
- Roughly 50 scenes will be rendered using 3D Studio
- Will be developed for Windows
- Programmed using C++, DirectX, and our in-house physics API
- Estimated development time 10-16 months

Communication

- Other document types may include:
  - Preliminary design document
  - Initial Design Document
  - Revised Design Document
  - General Design Document
  - Expanded Design Document
  - Technical Design Document
  - Final Design Document

Design Document

- More formal and complete than game treatment
  - What does the player do?
  - What is the interface?
  - What is the plot?
- Level Details
  - What are the levels?
  - Who are the characters?
  - How do characters interact?

Good Design Documents

- State the goals of the game explicitly
- Make the document itself readable
- Give priorities to ideas so that team members know what is important and what may be rejected
- List all details (e.g. behavioral model)
- Describe how you will do things

Design Document Content

- Game Overview
  - More detailed revision of game treatment
- Plotline detail
  - List player goals and achievements and work backwards
- Story outlines for each game section

Outlining Your Game

- Describe universal elements- common features to every part of the game
  - scoring rules
  - names
  - special powers
  - anything else?
- Details of every scene or game level
  - Name for scene
  - Resource details
  - Physical and audio appearance

Outlining Your Game

- Details of every scene (continued).
  - Background or playfield
  - Foreground objects and characters
  - Animations present for the scenes
  - Music and sound effects
  - Script for characters
  - Scenes and transitions
  - Flow charts for story branches
  - Miscellaneous elements (credits, saving games, setup, etc.

Game Design Document Sections

- Table of Contents
- Introduction/Overview
- Game Mechanisms
- Artificial Intelligence
- Game Elements
- Story Overview
- Game Progression
- User Interface

Product Specification

- Who is the production team?
- Target audience
- Gameplay
- Shelf-life?
- Production tools
- Schedule with milestones and deliverables

Game Specification

- What is it like to play the game?
- Interface mock-up
- Story-line summary
  - Major: final accomplishments
  - Minor: intermediate tasks
- Storyboards
  - Prototype artwork and screen sequences

Game Specification

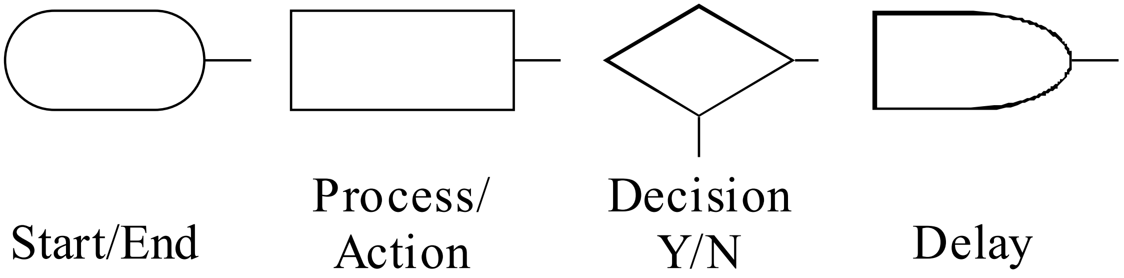
- Character bibles
  - Profiles and biographies for each character
- Flowcharting
  - What are the decision points and scene transitions?
- Scripts
  - What happens in each scene and during each level?

Storyboarding

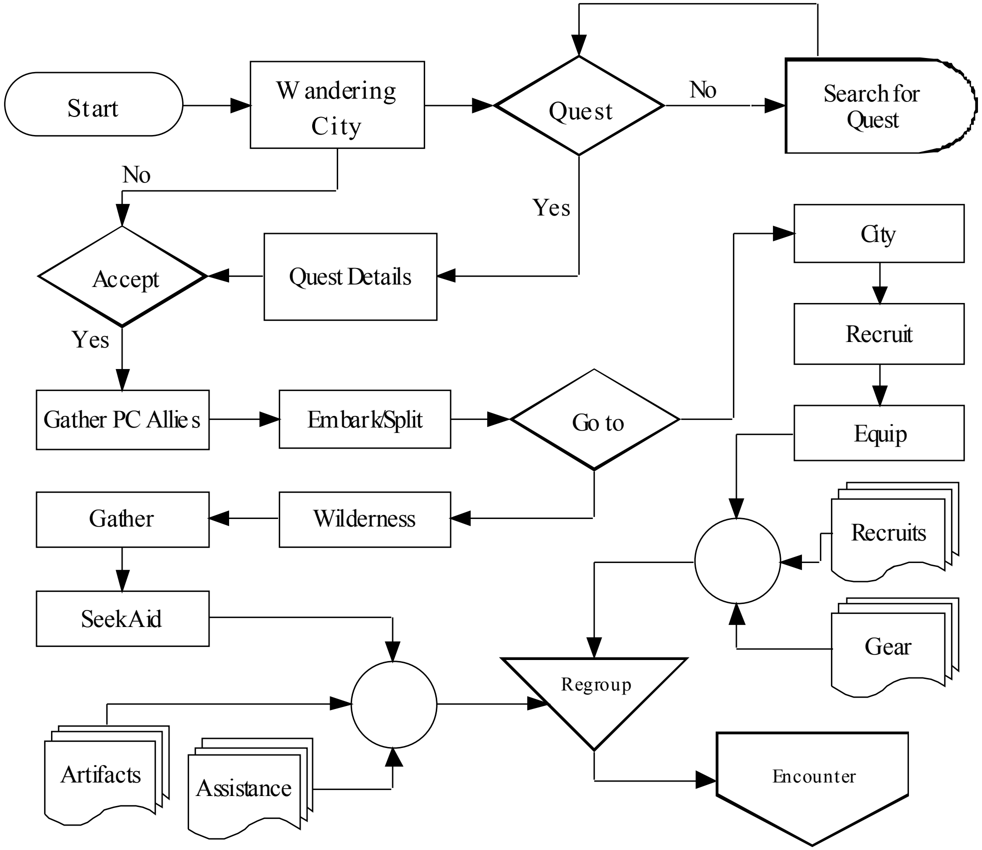
- Story outline
- Draw 6-12 scenes from gamer and assemble them like a comic strip
- Add some notes to each sketch describing the action, artwork, sounds

Communication

- Flowcharts
  - A typical technique for diagramming steps in a process
  - Most developers are familiar

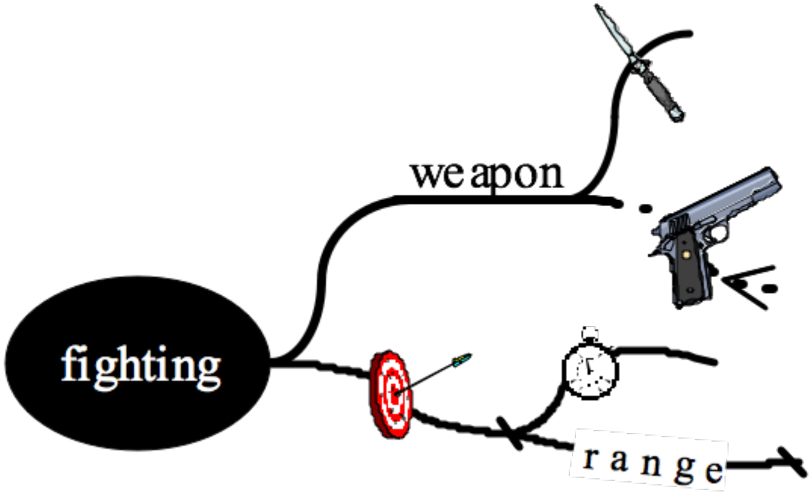


Communication



Communication

- **Associative diagram**
  - Drawing that helps manage and organize information visually
- Mind Map
  - A style of associative diagram
  - Key words and figures are placed on branches



Detail Questions

- What can characters do (fly,jump,invisible)?
- How many enemies does hero fight?
- What weapons are available?
- How does the player get rejuvenated?
- Multi-player stuff?
- Game perspective (side, tops, 3D, first person)?
- What kind of sound track?
- What about main character’s personality?

Level Outline

- Name of section, level, or scene
- Physical or audio appearance
- Foreground objects and charcters
  - Actions?
  - Animation?
  - Sound effects?
- Character scripts
- Transitions

Character Bible

- Journal in which the designer writes a profile and biography for characters used in the script
- Script may not be linear, so hypertext technology may need to be used to maintain continuity

Puzzle Types - 1

- Ordinary use of objects
- Unusual use of an ordinary object
- Creating new objects out of old?
- Information puzzles (e.g. find missing piece)
- Codes and word puzzles
- Excluded middle
  - (relies on cause and effect type relationships)
- People puzzles (outwit the guard)
- Timing puzzles

Puzzle Types - 2

- Sequence puzzles
- Logic puzzles (e.g. riddles)
- Trial and error
- Machinery puzzles
- Alternate interfaces
- Mazes

Bad Puzzles

- Unnecessary repetition
- Restore puzzle
  - find answer to puzzle when you die
- Arbitrary puzzles
  - cause should be linked to effects instead of random
- Designer puzzles
  - only designer can solve the puzzle
- Binary puzzle (e.g. wrong answer = death)
- Hunt the pixel
- Unnecessary interludes

Good Puzzles

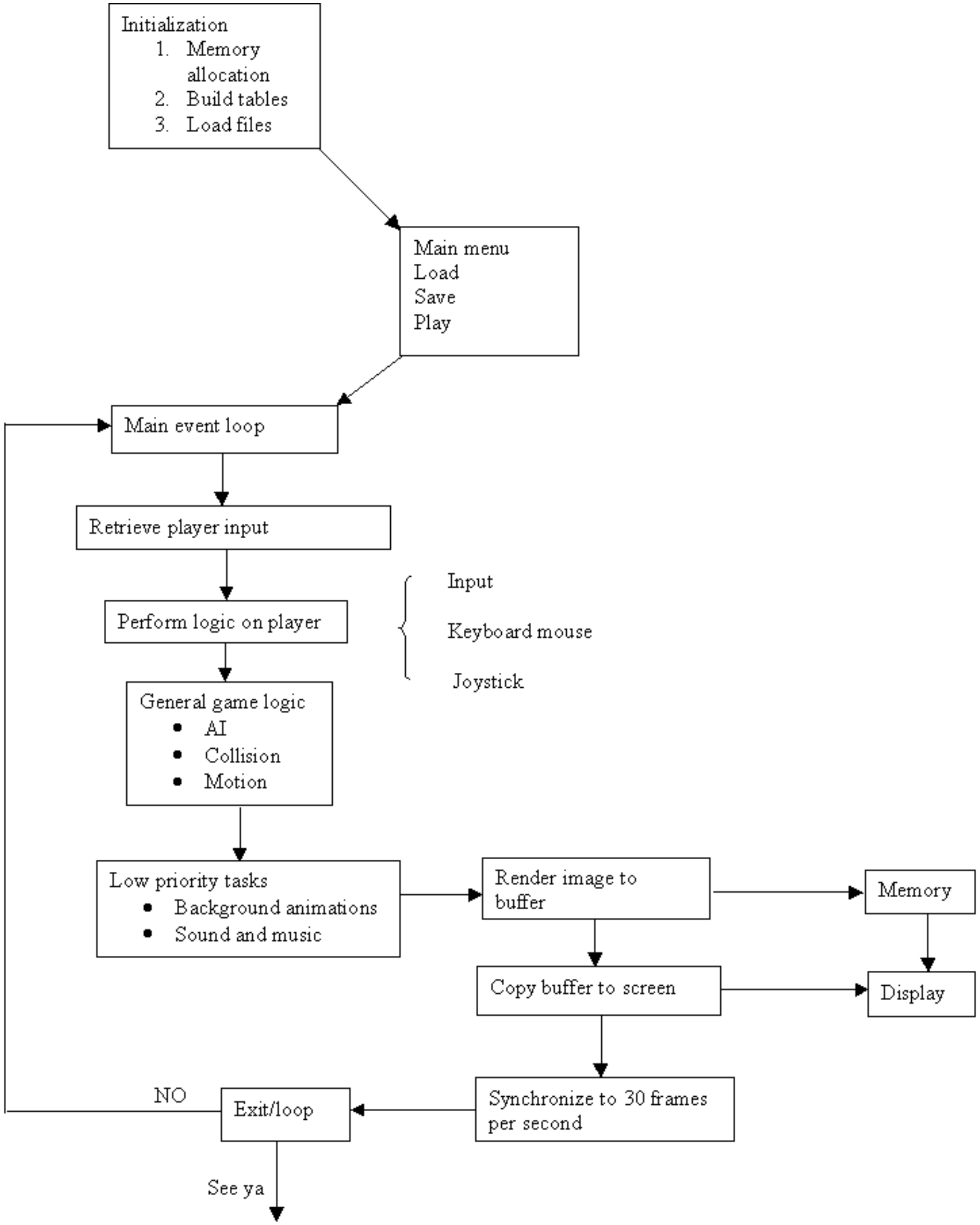
- Solvable
- Being fair
- No down time
- Some randomness – different each time you played
- Naturalness to environment
- Amplify a theme
- Principle of least astonishment

Hints

- Bread crumbs – at first everything works well and then give less direct help, if user struggles give more help
- Proximity of puzzle to solution – a fair game gives users everything they need to know
- Alternate solutions
- Red herrings (things that “don’t compute”)
- Steering a player

Designing Puzzles

- Break story into scenes
- Puzzles are obstacles to moving between scenes
- Trick is to make the puzzles match the story and setting
- Keep your character's abilities in mind
- Empathize with the player and what he or she will know when puzzle is encountered



Initialization

1. Memory allocation
2. Build tables
3. Load files

Main menu
Load
Save
Play

Main event loop

Retrieve player input

Perform logic on player

Input

Keyboard mouse

Joystick

General game logic

- AI
- Collision
- Motion

Low priority tasks

- Background animations
- Sound and music

Render image to buffer

Memory

Copy buffer to screen

Display

Synchronize to 30 frames per second

Exit/loop

NO

See ya

Typical Game Sections

1. Game startup
   - Initialize variables
   - Set up data structures
   - Allocate memory
   - Load graphics and sound files
2. Game enters main loop or exits to OS
3. User is prompted for input
4. User input retrieve

Game Sections - 2

5. Game state updated based on user’s last input
6. Based on last player action AI is applied, collisions processed, objects move
7. Once player logic processing is complete, background animation performed, music, sound effects,and housekeeping performed

Game Sections - 3

8. Current animation frame is rendered (drawn to virtual buffer)
9. Program displays frame by copying buffer to screen
10. Frame display rate locked to 30 fps
11. Exit section (game over)
    - Release resources
    - Restore system settings
    - Exit to OS

Why Use Prototypes?

- Minimize risk of starting over from scratch
- Involve client in development process early
- Prototypes can function as an animated storyboard

Prototypes Answer Questions

- What will the finished product look like?
- What do we need to do?
- Can we produce the product at all?
- Can we attract a publisher?

Prototyping

- Work with a group of 3 or 4 people
- You need to create a paper prototype for a first person shooter
- What rules will need for movement and shooting?
- Create a scoring system for your paper prototype.
- Why did your group choose this set of rules?

The next 5 slides come from the Rabin text

Iterating

- **Waterfall method**
  - – Development methodology
  - – Design and production are broken into phases
- **Iterative development**
  - – Practice of producing things incrementally
    - Refining and re-refining the product

Iterating

- **Prototypes**
  - Early working models of the product
  - Used to test ideas and techniques
- **Physical prototypes**
  - Non-electronic models; physical materials
- **Software prototypes**
  - Used regularly during iterative development

Iterating

- **Software testing**
  - Process of verifying performance and reliability of a software product
- **Tester**
  - Person trained in methods of evaluation
- **Bug**
  - Discrepancy between expected and actual behavior
- **Problem/Bug report**
  - Description of the behavior of the discrepancy

Iterating

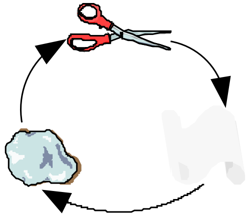
- **Focus test**
  - Testing session using play-testers
  - Testers represent the target audience
  - Lots of feedback at one time
  - Data can be compromised by group think

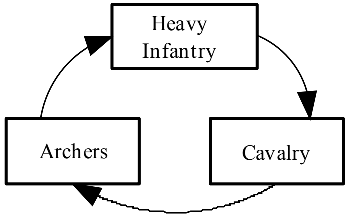
Iterating

- **Tuning**
  - Developing solutions by adjusting systems
  - Iterations are faster
  - Changes are less dramatic
- **Balance**
  - Equilibrium in a relationship
    - Player relationships, mechanics, systems, etc.

Iterating

- Intransitive relationships
  - Multiple elements offer weaknesses and strengths relative to each other as a whole
  - Balanced as a group
  - Example: Rock-Paper-Scissors (RPS)




Heavy Infantry
Archers
Cavalry