game development documentation

• why?
  team of developers need an agreed-upon design, preferably documented in the same place

• documents:
  – concept document
  – competitive analysis
  – design document
  – flowcharts
  – story bible
  – script
  – art bible
  – game minute
  – storyboard
  – technical design document
  – marketing plan
  – development schedule

game development documentation: concept document

• “pitch” or proposal for the game
• audience is someone who is going to fund the development, e.g., funder, manager, publisher
• content should be short
• should discuss high-level design issues
• should detail game story
• should sound exciting but tractable
• can include some rough art (e.g., sketches), screen mock-ups
game development documentation: competitive analysis

- reviews the field, the competitors
- describes the “related work” within the same game genre
- highlights what is unique about the new game

game development documentation: design document

- functional specification
- fully describes gameplay in detail
- should be a reference for the implementation of the game
- includes:
  - game mechanics
  - storyline, context
  - “worlds” and/or “levels” and/or “stages” in the game
  - characters, objects and items in the game
- covers the “who”, “what”, “when”, “where”, “why”, “how” of the game
- should NOT include detailed content that is in the other documents described here (though there is necessarily some repetition at a high level in order for the design document to make sense on its own)
- design document discussed in detail in second section of these notes (below)

game development documentation: flowcharts

- can be part of the design document, or can be separate
- two purposes:
  - to track out-of-game menu options (e.g., start new game, load old game, set preferences, etc).
  - to track players through game levels
- can also be used to explain players’ decision-making processes during the game

game development documentation: story bible

- overview of game narrative
- can be in the design document, or can be separate (should be separate if story is long and detailed)
- full details of game’s back story and context go in the story bible
- includes background, character details, history of characters, context, etc.—whatever is necessary for the game designers and implementers to know so they can build the game
- acts as a guide for consistency during game development
game development documentation: script

• like a movie-script with dialogues for games that have “cut scenes”
• also includes “stage directions”
• can be contain pseudo-code
  – *IF-THEN-ELSE* or *SWITCH* branching statements to show options between playing different scenes or selection based on what the player does
  – *OR* statements showing possible selection between clauses in dialogue or scenes which can be selected randomly by the game engine
• result is a “branching dialogue tree”
• any text that appears in the game should be in the script
• keep in mind when writing the script that this is a game and not a movie!
  – scenes should not be too long
  – so player can get back to doing stuff instead of watching stuff

game development documentation: art bible

• contains art “assets”
  – drawings of characters, objects, items
  – animations of characters and objects doing standard things (e.g., walking, door opening, etc.)
  – background scenery
• contains sketches and screen shots
• details the “look and feel” of the game

game development documentation: game minute

• detailed description of a short session of game play
• contains everything player sees and does
• audience for the “game minute” is typically managers, but also developers can use it to keep team members on the same page in terms of a unified game experience

game development documentation: storyboard

• linear sequence of scenes, but with branching
• should be drawn like a state/action graph
• “nodes” in the graph are scenes
• “links” are actions that player can take
• links can also be optional directions that the game engine can take without player input, in which case a probability should accompany the links indicating how likely it is that each branch would be taken
• remember state diagram from the midterm—storyboard should show the different possible actions and what scene in the game would result if each action were taken
game development documentation: technical design document

• contains implementation specifications
• includes details about:
  – code (software) structure, e.g., class design
  – implementation of artificial intelligence
• describes the rendering architecture
• describes any hardware specifications (e.g., game is designed to run on a tablet, on a phone, in a browser, etc.)

game development documentation: marketing plan

• the marketing plan is a business document
  – who is the target customer?
  – what is the estimated number of games that will be sold?
  – what does market analysis show about pricing?
  – what is the competition like from a business standpoint (as opposed to a technical/design comparison, which is done in the “competitive analysis” section, detailed above)

game development documentation: development schedule

• development schedule
• what are the steps in the development process?
• what are the key development goals?
• when will a complete prototype be ready?
• when will alpha-testing begin?
• when will beta-testing begin?
• when will the first version be released for sale?

design document

• writing style is important
• sections
  – table of contents
  – introduction/overview
  – game mechanics: WHAT and HOW
  – artificial intelligence
  – game elements
  – story overview
  – game progression
  – system menus
**Design Document: Table of Contents**

- Ends up as tool people use to navigate your document
- Must include subsections, sub-subsections, etc

**Design Document: Introduction/Overview/Executive Summary**

- Good starting point for understanding the game
- And for new team members who join
- And for anyone reading about the game for the first time (e.g., funder, marketer, producer, etc.)
- Should be ONE PAGE long

**Design Document: Game Mechanics**

- Most important part of your document
- What the game play really is
- Avoid detailing specifics about game-world objects or characters (do that later)
- Be abstract/general enough so that many different games could be made from this section of the document
- Introduce players’ capabilities in the order in which a user will encounter them; start simple and move to more complex
- List physical commands users will need to effect players’ movements
- Describe how players control their avatar
- Describe avatar’s movement model
- If user creates their avatar, describe that process and components to be created
- If there are puzzles in the game, describe their functions and how players manipulate them—but in general, not with implementation details

**Don’t list actual puzzles, but instead include high-level descriptions (e.g., puzzle might involve hitting walls...)**
- Do not list specific configurations
- List the palette of objects and behaviors from which puzzles are created
- Describe how game controls shift from one mode/level to another
- Separate out technical aspects! (these go in technical design doc)
- Describe GUI (graphical user interface)
design document: artificial intelligence

- describe how the world reacts to players' actions
- and also how the world may change on its own, not in response to player (e.g., when player isn't doing anything)

design document: game elements

- palette for level designers
- parts that will be brought together to create compelling experience for user
  - characters
  - items (things they can pick up)
  - objects/mechanisms (things they can do, e.g., open a door)

design document: story overview

- easy-to-read narrative
- not too detailed
- quick way to understand the context and "big picture" of the game

design document: game progression

- may be longest section of document
- go through events players experience, how they change and progress
- guide for artists and level designers
- be descriptive about how level will affect players, not just "it is difficult"
- some games have stages instead of levels. break down stages here.
- some games have neither—in which case this section may not be necessary
design document: system menus

- out-of-game stuff, e.g., main menu, preferences, etc.
- describe interface and how user will interact with it, e.g., mouse clicks

common mistakes

- most important thing is to communicate your vision of the game to the reader
- common mistakes include:
  - too short, brief, high-level
  - too long, overly-detailed story, no game play details
  - too much implementation detail
  - no consideration of technology
- know that document is DYNAMIC and will evolve as game develops

to do

- work on homework assignment for unit III, which is due November 7