

Game Design

Game Mechanics

The game mechanics describe the game play in detailed terms, starting with the vision of the core game play, followed by the game flow, which traces the player activity in a typical game. The rest is all the infinite details.

Core Game Play

In a few paragraphs describe the essence of the game. These few words are the seeds from which the design should grow. Planted in the fertile soil of a known market, they should establish roots that anchor the vision firmly in place and help ensure a successful game. This is similar to the description section in the game concept, except that it's non-narrative and usually expressed clearest in bullet points, though this could vary depending on the type of game.

Game Flow

Trace the typical flow of game play with a detailed description of player activity, paying close attention to the progression of challenge and entertainment. If the core game play is the root of a tree, the game flow is the trunk and the branches. All activity should actualize and extend from the core game play. Be specific about what the player does, though try to use terms like "shoot", "command", "select" and "move" rather than "click", "press" and "drag". This keeps the description distinct from how the actual GUI will work, which is likely to change. Refer readers to specific pages in the User Interface section when you first mention a GUI element such as a screen or window or command bar.

Characters

These are the actors in the game controlled by the players or the AI. This should include a brief description and any applicable statistics. Statistics should be on a rating scale i.e. A to Z or Low to High, so that it's clear where units stand in relation to each other in broad terms. It's a waste of time plugging in the actual numbers until the programmers have written the technical specification and created an environment for you to experiment with the numbers. Special talents or abilities beyond the statistics should be listed and briefly described, but if they are complex, they should be expanded upon in the game play Elements section.

Game Play Elements

This is a functional description of all elements that the player (or characters/units) can engage, acquire or otherwise interactive with. These are such things as weapons, buildings, switches, elevators, traps, items, spells, power-ups, and special talents. Write a paragraph at the start of each category describing how these elements are introduced and interacted with.

Game Physics and Statistics

Break out how the physics of the game should function, i.e. movement, collision, combat etc., separating each into subsections. Describe the look and feel and how they might vary based on statistics assignable to the characters, units and game play elements. Indicate the statistics required to make them work. Get feedback from the programmers as you write this, as how the game handles the physics and the quantity of the statistics will severely impact performance issues.

This can get a little dry, but avoid getting too technical. Avoid using actual numbers or programming terms. These will come later in the technical specification, written by the programmers who will want to do things their way (usually the right way). Just tell them what you want to accomplish. For example: "The units should slow down when going up hill and speed up when going down, unless they are a hover or flying vehicle. How much they are affected should be a factor of their climbing and acceleration statistic as well as the angle of the incline." You would not tell the programmers what math to use to adjust the speed. Assuming you are not a programmer yourself, they're just better at that than you.

Artificial Intelligence

Describe the desired behaviour and accessibility of the AI in the game. This includes movement (path finding), reactions and triggers, target selection and other combat decisions such as range and positioning, and interaction with game play elements. Describe the avenue through which the AI should be controlled by the level designers, i.e. using .INI files, #include files of game stats or C-code, proprietary AI scripts, etc.

Multiplayer

Indicate the methods of multi-player play (i.e. head-to-head, cooperative vs. AI, teams, every man for himself, hot seat) and how many players it will support on the various networking methods. Describe how multi-player differs from solo-play in game flow, characters/units, game play elements and AI.

User Interface

The interface changes so very often that it almost seems pointless to document it; however, it's got to start somewhere. It's structured here to minimize the impact of changes. It starts with a flowchart of the screen and window navigation, and then breaks down the functional requirements of all the screens and windows. That done, the GUI artist is free to do what he or she feels is right as long as it meets the requirements. To get him or her started you should provide mock-ups. This often is to the designer's benefit to think everything through. Then follow up with a description of all the GUI objects that need to be programmed to make all the screens work.

Flowchart

This charts the navigation through the various screens and windows. Use VISIO or similar flowcharting tool to connect labelled and numbered boxes together, representing screens, windows, menus, etc. On the corner of each sheet, put a numbered list of all the items for easy referencing and ease of defining tasks for the programmers.

Functional Requirements

This functional breakdown of every screen, window and menu lists the user actions and the desired results and may include diagrams and mock-ups. While the specific interaction (buttons, hotspots, clicks, drags and resulting animations) can be listed, it's often best to keep this separate from the list of functional requirements as these can evolve during implementation. Of course if it's just easier to think in terms of clicking a button or it's really important that something work a certain way, then by all means get specific about the method of interaction.

Mock-ups

Create a mock-up for all the screens, windows and menus. This may end up getting ignored, but it's a good starting point for the artists if they have no idea what else they may want to do. Don't waste your time creating anything really pretty. Just create simple line drawings with text labels. Color can be very distracting if it's bad, but if it's important, go ahead. Some drawing programs have templates that make creating mock-ups very quick and easy.

GUI Objects

These are the basic building blocks used to create all the screens, windows and menus. This should not include the items seen in the main view portal, as these are covered in the art list in the next section. The GUI objects are primarily listed here for the programmers to know what pieces they'll need to code and have for putting together the screens. You should explain in detail how each is interacted with and how they behave. It may seem a bit obvious and not worth documenting, but it really helps when drafting together the technical spec and schedule to know exactly everything the game will need.

For some games, this can be a very quick list to put together – buttons, icons, pointers, sliders, HUD displays etc. But it's much more complicated in games where the interface is at all different. However, keep in mind that the methods of interaction are not all that different. A button is still a button, even if it's clicking on a gorgon's head instead of a grey rectangle.

Art and Video

This should be the definitive list for all the art and video in the game. We all know how things creep up, though, so add a couple of placeholder references for art to be named later, like mission specific art and art for marketing materials, demos, web pages, manual and packaging.

Overall Goals

This is where you should spell out the motifs, characteristics, style, mood, colors etc. that make up the goals for the art. Gather consensus with the lead artists and art director and make sure they see eye to eye with the project's director or producer. Doing so now will save a lot of time later if they end up redoing everything because the goals were never clearly defined.

2D Art & Animation

This is really just a huge list that can be thrown into the art schedule. It can also include descriptions if needed. Some art isn't self-explanatory, and other may involve specific needs from a design standpoint. Be sure to explain it all. Break your art down into sections. The lead artist may have some particular way he or she would like you to do that. I'll list the typical section and their contents. Read them all to be sure you don't forget anything.

GUI

Screens, windows, pointers, markers, icons, buttons, menus, shell etc.

Marketing and Packaging

You might as well list it here and the schedule, because they'll ask for it. This includes web page art, sell sheet design, demo splash screens, magazine adds, press art, the box and manual.

Terrain

Environment art like tiles, textures, terrain objects, backgrounds.

Game Play Elements

Player and enemy animations (sprites or models), game play structures and interactive objects, weapons, power-ups, etc. Don't forget damage states.

Special Effects

Salvo, explosions, sparks, footprints, blood spots, debris, and wreckage.

3D Art and Animation

This serves the same purpose and has the same requirement of the 2D Art list above. The difference may be in how the work may be divided. Art teams like to divide 3D art task lists into models, textures, animations and special effects, as they usually divide the tasks this way to maximize talent and skill and maintain consistency.

Cinematics

These are the 2D or 3D scenes often shown as an intro, between missions, and at the end of the game. These should be scripted like a film script as separate documents. This, however, is production work. For the purposes of the functional spec, just list them here with the general purpose, content and target length. If any video is involved, list it in the following subsection.

Video

Unless you are doing an FMV (full motion video) game, this subsection is pretty light. If you have any video in your GUI for say pilot messages, break it down here. All video tasks will require scripting, but that is production work. List the general purpose,

expected length, and general content like number of actors and set design, even if it ends up being blue-screened into a 3D rendered background.

Sound and Music

Overall Goals

Stress the aesthetic and technical goals for the sound and music. Describe the themes or moods you want. Name existing games or films as examples to aspire to. Issue technical edicts and editing objectives, such as sampling rates, disk space, music formats, and transition methods.

Sound Effects

List all the sound FX required in the game and where they will be used. Include the intended filenames, but be sure to consult with the sound programmer and sound technician (or composer) on the file naming convention. This makes it easier for people to find the sound FX and fold them into the game.

Don't forget about all the areas that sound FX may be used. You don't want to overlook anything and throw off the schedule. Go through all the game elements and your art lists to see if there should be some sound associated with them. Here are some to consider:

GUI

Button clicks, window opening, command acknowledgments.

Special Effects

Weapons fire, explosions, radar beeping.

Characters

Voice recordings, radio chatter, stomping, collisions.

Game Play Elements

Pick-up jingle, alerts, ambient sounds.

Terrain (Environment)

Birds, jungle sounds, crickets, creaks.

Motion

Wind, footfalls, creaking floors, wading, puddle stepping.

Music

List all the music required in the game and where it will be used. Describe the mood and other subtleties. Music will often reuse the same themes and melodies. Mention where these themes should be reused. Consult the composer on this.

Event Jingles

Success, failure, death, victory, discovery, etc.

Shell Screen

Mood setting for title screens, credits, end game.

Level Themes

Level specific music (designers choose the theme).

Situations

Sets the mood for situations (lurking danger, combat, and discovery).

Cinematic Soundtracks

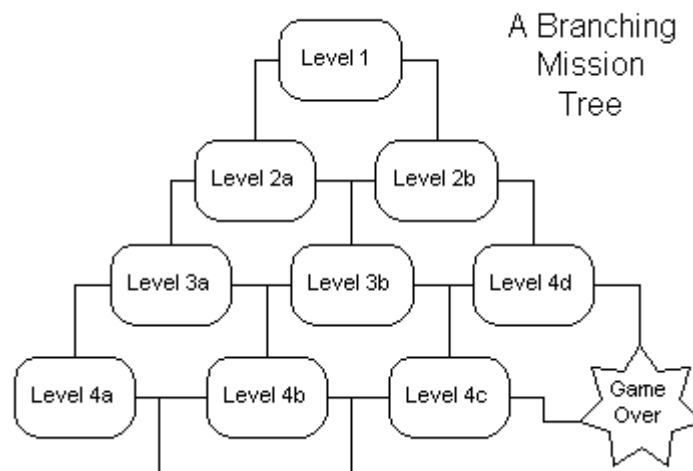
Story

Write the synopsis of the story told by the game. Include the back-story and detailed character descriptions if it helps. Indicate the game text and dialogue requirements so they can be added to the schedule. Some game designs focus so much on this, that they overlook everything else that should be in the spec. Telling a story is not the focus of most games. Of course, if you are doing an adventure game, it is extremely important. Expand and organize this section as is necessary to tell the story.

Level Requirements

Level Diagram

Whether this is a linear campaign, a branching mission tree, or a world-hopping free-for-all, this diagram should be the backbone upon which all the levels are built. A diagram isn't necessary if the structure is so simple that a list would suffice. The following is an example of a typical success/fail branching mission tree. Of course this will vary greatly for each game. The important thing is that it just presents a road map for the level designers and for the readers.



Asset Revelation Schedule

This should be a table or spreadsheet of what level the game's assets are to be revealed to the player for the first time. There should be a row for each level and a column for each general type of asset. Assets include power-ups, weapons, enemy types, tricks, traps, objective types, challenges, buildings and all the other game play elements. The asset revelation schedule ensures that assets, the things that keep the players looking forward to the next level, are properly spaced and not over or under used.

If it's important to the game that certain assets stop being used, then the schedule might be better drawn as a Gantt chart with lines indicating the availability of assets. This gives the level designers a guide to what assets they have to work with so they don't ruin their level or anyone else's.

Level Design Seeds

These are the seeds for the detailed paper designs to follow. Detailed paper designs at this point are less legitimate and unlikely to survive intact. Designs created after the designers have had time to experiment with the tools and develop the first playable level are much more likely to succeed. It's best to just plant the seeds for each level with a description of the goals and game play and where it ties into the story (if applicable). A thumbnail sketch is optional, but very helpful if the designer already has a clear idea of what he or she wants. Be sure to list any specific requirements for the level, such as terrain, objectives, the revelation of new assets, and target difficulty level.