#020 Level Design - Process - Pre-production - Pacing

From *The Level Design Book*.

- ▼ *#020 Level Design - Process - Pre-production - Pacing*
  - *What is pacing?*
  - ▼ *Beats and variation*
    - *Set pieces*
    - *"Pile of beats" approach*
    - *Teach, test, twist*
  - *Critical path*
  - ▼ *Plotting and documentation*
    - *Beat sheet*
    - *Flowchart*
    - *Graph*
  - *Pacing advice*
  - *Multiplayer pacing*
  - *Open world / nonlinear pacing*
  - *Against pacing*
  - *To review...*

What is pacing?

Pacing is the general order and rhythm of activities and events in a level.

Single player levels tend to require strong pacing. If players are confused about what is happening or what they can do in a level, then that may indicate a pacing problem.

An effective pacing plan should address:

- **Scope:** What can the player do in each level?
- **Hierarchy:** Which parts of the level are most important?
- **Causality:** Why does the player do (this activity) before (that activity)?
- **Information:** What do we tell the player and when?
- **Intensity:** When should the player pay more attention, and when do they rest and recover?

Beats and variation

What happens in your levels? What are the various moments and places that define the experience?

A **beat** is a small self-contained chunk of a level. A single area, event, activity, or element.

We can liken these beats to musical beats in a song. When performed together, these beats form a melody and rhythm. They can be understood separately, but also as part of a whole. To make beats more interesting, music composers **arrange** beats in different ways to create **variation:**

- pulse: establish a regular recurring pattern of beats, like a heartbeat; establish a meter
  - example: end every level with a distinctive exit door
- accent / stress: emphasize or intensify certain beats
  - sometimes the exit is difficult to find / reach
- rest: incorporate periods of weaker beats or silence, sensitize audience to accents again
  - sometimes the exit is easy to find / reach
- motif: a short recurring sequence of beats
  - sometimes the player fights a boss before reaching the exit
- variation: repeat a sequence of beats, but with different melody, rhythm, etc.
  - some levels have multiple exit doors
- syncopation: go off-beat; the basis of modern pop music
  - halfway through a boss fight, another boss appears
  - sometimes a false exit door hides a monster
  - final boss destroys the exit door; now there is no escape
  - player gains the ability to create their own exit doors

Set pieces

A **set piece** is an especially elaborate beat with a unique concept or memorable activity.

The practice of set pieces comes from film production, where big budget movie projects often commission big expensive scenes that require unique sets and complex planning -- an intense spectacle that the audience will remember.

For example, Hollywood blockbuster action films are essentially a series of set pieces -- big elaborate fights, chase sequences, or death defying stunts. The rest of the film exists primarily to connect these set pieces together in a semi-coherent way, and provide some rest for the audience between these intense set pieces. Blockbuster action games work much in the same way.

But set pieces don't necessarily have to be big explosive action sequences. Comedies feature hilariously embarrassing situations, romances might emphasize a first date or wedding, dramas may feature tear-jerking confessions or betrayals, murder mysteries end with a detective recounting the true events.

Any scene in a film or game that feels important, memorable, or expensive, is probably a set piece.

In games, a set piece is usually:

- **Expensive,** requiring many unique art and animation assets, and constant iteration. Not easily scoped down nor repurposed, so inherently risky to produce.
- **Unskippable,** tied closely to the project's core pillars and main experience goals. Why spend so much time and money making something if the player might miss it and blame the game?
- **Heavily scripted** and somewhat linear / on-rails, to ensure a reliable experience.
    - If the set piece can play out three different ways, then it is now the equivalent of three set pieces, and thus three times as expensive to make.

The level design for a set piece usually involves:

- **Boss fight,** big **puzzle,** or **choreography** -- something with a lot of level scripting.
- **Arena** typology, a large room to trap the player until they complete the encounter or cutscene.
- **Hero props,** unique environment art assets to make the level feel particularly special.

In any given project, try to design and plan for at least one set piece. These sequences anchor the rest of the project. Ideally, this set piece should be something you are very excited about, something you can't wait for players to experience.

If you're dreading the work of building out a set piece, or worse, you don't want any players to actually play through it, then you probably shouldn't do it.

If you are learning a new toolset, then don't plan a huge epic set piece. If you're new to game development, then gauging the scope of a set piece may be difficult until you have more experience. Remember: the best set piece is the set piece that you actually have a chance of finishing and releasing.

"Pile of beats" approach

Combat-based projects and puzzle games can benefit greatly from designing many beats separately, and arranging them together later. The workflow looks like this:

1. Conceptualize, layout, and blockout one isolated battle / puzzle.
2. Playtest and iterate on the blockout, until it has proven either promising or a bad idea.

3. Repeat steps 1-2 until you have prototyped dozens of battles / puzzles.
4. Arrange the best beats based on common (or contrasting) elements.

The first person puzzle game Portal (2007) was prototyped as a series of disconnected puzzle chambers, and only congealed into a coherent campaign of levels after the designers selected their favorite puzzles. The puzzle element icons displayed on each test chamber's introductory placard are left over from this "pile of beats" process.

Teach, test, twist

Teach, test, twist is a common 3-beat pattern in level design:

- **Teach:** teach the player about a game activity
  - example: in Portal 1 chamber 10, player learns a simple "fling"
- **Test:** test whether the player can repeat and recognize the activity, with prompting
  - later in Portal 1 chamber 10, the player performs a deeper fling
- **Twist:** twist the frame of the activity; with less prompting, can the player recall it?
  - in Portal 1 chamber 12, the player flings from taller heights
  - in Portal 1 chamber 15, the player "double-flings" in mid-air
  - then Portal 1 chamber 18 ends with an "infinite fling" challenge
  - note how these "twist" beats are spaced out, and aren't immediately after the Test

Critical path

Pacing design assumes that we can influence, predict, and constrain what the player can / will do in the game. The **critical path** is the core progression of beats that every player must experience to "complete" the game, whatever that means.

Critical paths can take a variety of forms:

- For encounters, a critical path might be the ideal strategy to defeat an enemy.
- For puzzles, the critical path is the solution to the puzzle, or at least the most satisfying one.
- For movement, the critical path is the ideal route to complete the level.

Plotting and documentation

There are several ways to outline and document the beats of a level:

- **Beat sheet,** simple text list of beats

- **Flowchart,** diagram of how beats connect to each other
- **Intensity plot,** segmented bar graph of beats plotted against a player metric

Beat sheet

In film and TV screenwriting, a **beat sheet** is a list outlining the major scenes and plot points. We can do something similar for plotting beats in level design: write a list of major events, scenes, and levels.

Write the beat sheet on paper, in an online doc / spreadsheet, or as a project board with arrangeable cards. Then write ideas for set pieces, scenes, levels, boss fights, etc. as paragraphs / rows / notecards / post-its, and then rearrange these beats to form a coherent outline.

- In a group, it's important to talk and think out loud, in-person or over a call
- Some free online project board tools: Trello, Notion, Miro, Codecks
- Ideally, use an IRL board / wall, so you won't need to worry about a browser tab

The beat sheet can take many forms, based on what you're designing and what it's for.

For the action RPG God of War Ragnarok (2022), lead combat designer Rob Meyer wrote "boss flow" documents which outline key elements and design rationales for the big boss fight setpieces, which helped Meyer get approval for prototyping a systems-heavy combat concept.

For The Last Of Us (2013), Naughty Dog designers rearranged different levels, story moments, and themes throughout the entire arc of the game. This "beat board" helped them plan out the pacing for the finished game, which actually differs greatly from the early plan detailed here.

Flowchart

A **flowchart** is a visual diagram of a process / program with lines, arrows, and branches. It visualizes the logic of the beats' connections. This is good for puzzle games or non-linear levels with multiple possible strategies and event chains. Use flowcharts when cause and effect matter a lot.

Sketch the flowchart on paper or a whiteboard, or use an online whiteboard / flowchart tool.

- Keep the flowchart as simple as possible. Use few words and omit minor beats. If it feels like spaghetti, then it is too complicated and you should simplify it.
- Try to make the flowchart "flow" in one direction, e.g. from top to bottom, left to right.
- Shade boxes different colors based on the beat type.
- Instead of one big flowchart, break it up into several smaller flowcharts.

Graph

To plan pacing in more detail, sequence the level in terms of a player metric and visualize the plot as a **graph.**

For pacing the levels in Half-Life 2: Episode Two and Left 4 Dead, Valve designers sorted beats into four categories:

- Explore, walk around and navigate the space. Often the start and end of a level.
- Combat, fight enemies.
- Choreo, some dialogue or choreographed scripted sequence. Used before or after combat, to help signal when combat begins or ends.
- Puzzle, find an item stash, unlock a door. Used to space out combat sequences.

Because these are action-based shooters, these designers focused on **intensity.** On the horizontal X axis, they plotted time in minutes. On the vertical Y axis, they plotted intensity with a simple numerical scale from 0-100%, 0-5, or 0-10.

But what does "4 out of 5 intensity" mean? The intensity score is just a gut feeling that emerges from knowing the game and observing playtests. It's not a hard science.

For each project, you must devise your own categories and metrics. How do you define "intensity"?

In Journey, intensity is not necessarily the same thing as game difficulty.

For example, the death of a beloved character might be a low difficulty cutscene with no player input, but it would still have a high emotional intensity. In this sense, intensity is better understood as suspense, the player's stake, or engagement in the game experience.

Pacing advice

Start slow and quiet.

Half-Life (1998) famously began with a mundane 6 minute tram ride. Begin your game or level with something low intensity, like an introductory cutscene or some low stakes exploration. It sets the mood and allows time for worldbuilding exposition. Even high action shooters like Doom or Quake begin with quiet rooms where players can test their controls and "warm up" before launching into combat.

Alternate highs and lows.

Players adjust to prolonged periods of high intensity. Any intense boss fight will feel like a slog after 10+ minutes. To keep it fresh, use occasional downtime as a contrast and palette cleanser, otherwise

the player will simply go numb. Don't try to force the player to be "on" all the time.

After high-intensity boss fights, cutscenes and low-intensity areas feel like rewards. Or your core mechanics can encourage the player to rest and return to town, e.g. to sell loot, repair equipment, turn-in quests, etc.

Avoid maximum intensity final bosses / encounters.

Thatgamecompany read about Joseph Campbell's Hero's Journey monomyth and liked it so much they called their game Journey. Other designers look to classical narrative theories like Three-Act Structure, Gustav Freytag's Five-Act Structure, and Aristotle's Poetics. These traditional narrative theories conclude with a falling action / "denouement" / defeat of the final villain that must feel like the inevitable result of the climax, or else it robs the climax of its impact. Some examples from games:

- the final boss of Dark Souls has lots of health but a relatively simple combat style with obvious tells; it feels challenging, but it is far from the most difficult encounter in the game
- the final puzzles in Portal 1 and 2 feature foreshadowed simplistic solutions, they are definitely not the most difficult puzzles in the game
- the final level of Half-Life 2 features low stress overpowered unbalanced combat encounters while the main villain monologues about their inevitable defeat

Classical narrative theories are useful, but beware:

- "Intensity" in a story might not mean the same thing for a game.
- Most game progressions / learning curves rarely have "falling action."
    - Most games are about avoiding failure. Skillful play means the player will avoid falling actions / committing errors. However, failures and mistakes often make for better stories.
- Outside of games, these are classical narrative theories that many writers seek to avoid. Using these patterns can often feel rote or formulaic.

Multiplayer pacing

Pacing in competitive multiplayer games emerges more from the overall gameplay loop and game mode rules. In battle royale games like Playerunknown's Battlegrounds, Fortnite, or Apex Legends, the match becomes more exciting with the last few remaining players trapped in a very small area. But where exactly will these last few players fight it out, and when? Each match is different, we cannot script or plot how the match will play out.

For team-based games like Counter-Strike, Team Fortress, or Overwatch, level designers must carefully pace the travel times between spawn areas and chokepoints. If a certain team has unfair

access to a chokepoint, then the map can feel unbalanced. Here, pacing is more a direct function of the map flow, layout, and metrics.

Open world / nonlinear pacing

In open world games, the player can often take multiple possible routes toward an objective. Designing a single path is not useful because we are purposely trying to allow the player some freedom in making their own path and progression. But as with multiplayer pacing, open world games minimize scripted pacing, so layout and metrics are our main tools for planning how quickly the player will traverse the level.

For the open world stealth game Assassin's Creed, level designers built blockouts and then planned missions as a series of concentric circle overlays. Each stage / segment gets gradually smaller and denser as the player approaches their objective. As the player gradually solves their way through each layer of the enemy's defenses, each segment escalates the sense of challenge and danger.

Against pacing

Pacing plans and documents are most useful at the early stage of a project.

- docs quickly go out-of-date and require maintenance
- many projects are so big that one document can't capture the complexity anyway
    - big studios and teams maintain internal wikis; dozens / hundreds of design documents
- you can't playtest a document; design docs never survive reality

An initial plan is important because it gets everyone on the same page with shared language. But at some point, you must instead pay attention to the actual game you're making, instead of the imaginary game you planned.

To review...

Pacing is the ideal order and rhythm of activities / events in the level. A beat is an activity / event.

- Single player levels usually have clear critical paths and big **setpieces.**
- For encounter-heavy games with combat and puzzle systems, **a pile of beats** approach works well -- build a bunch of encounters and puzzles first, then arrange them after, maybe into a **teach-test-twist** pattern.
- Multiplayer pacing emerges more from layout rather than authored events.

- Open world / non-linear pacing needs a looser "zone" approach.

Pacing design documents may include:

- **Beat sheet:** written outline, a list (or board) of beats
- **Flowchart:** visual outline with arrows and logic
- **Graph:** graph the beats based on a player metric, like "intensity"

These docs often become obsolete while you work on the game. You can either update these docs, or ignore them.

> ***write by Hellinus - 2023.9.15***