Games Designed for Flow

In chapter five of the book I talk a bit about flow states and give some examples of games with flow. This 'be-the-game' state is not easy to design for. The problem for game designers comes in creating a game that has the right balance between flow states and human levels of concentration and motivation. Knowing no player can remain totally focused for hours at a time the game needs to have 'rest spots' or changes in tone (audio, visual, competition level) so that a player can move in and out of the flow state. A flow state can be achieved in games of varying complexity, Space Invaders (Taito, 1978), Pac-Man (Namco, 1980) or Tetris (Pajitnov, 1984) players can get into 'the zone' in the same way that players of first person shooters or high octane action games can. It's not about the complexity of the game, it's about how engaging and, importantly, how fun it is for the player.

Some games go after the ecstatic nature of play very deliberately; these games are often shooters with fairly simple controls but with hidden depth that are complex to master. Games such as *Everyday Shooter* (Queasy Games, 2007), *N_2O: Nitrous Oxide* (Gremlin Interactive, 2008) and *Wipeout* (Psygnosis, 1995) rely on quick reactions ('twitch controls') and familiarity with the game space to achieve level ups or high scores. Players of these games need flow or zone states to be able to play the games to a high level.

Tuning a Game for Flow

Jenova Chen, creator of the PlayStation 3 game *Flow* (Thatgamecompany, 2006) centered his MFA thesis on how flow works in games. He breaks down designing a system that adjusts player difficulty based on performance and this system is the Dynamic Difficulty Adjustment Loop (DDA).

The Dynamic Difficulty Adjustment (DDA) loop

- Player: As the player goes through the game, performance data is gathered.
- Monitor System: Looks for critical data (what choices has the player made, which is their preferred spell/weapon? How many times has the player died?). Pass this data for analysis.
- Analysis System: Are elements of the game appearing too hard or too easy for the player? Did the player spend a few minutes or a few hours on this level? Are they dying too much or not enough?
- Game System: Adjust the game according to the Analysis System.

The issue (as Chen points out) is that while this is a solid enough system on paper it does not take account of the intricacies of the player's personality. These are analyses based on assumptions. Not every player plays to beat the game, some are there to explore, some are there to get every gem and some players just want to test the physics of the game. If a game adjusted the difficulty level because a player kept blowing themselves up, it's assuming they're a 'bad' player – but they might just be having fun with seeing what it looks like to blow up their character in the game. To design with flow in mind a game has to adjust to the whims of the player to an extent.

Engagement Entropy.

Play testing is crucial for designers when creating any game (I look at more on testing and iterations in Chapter 6 of the book) and it's important to get feedback from testers to evaluate levels of engagement. Game designer are looking for *engagement entropy*, the point in the game where the player stops having fun and begins instead to resent the experience. This form of entropy can come from many areas; unskippable cut-scenes, narratives that make no sense, the level being too difficult, players getting lost and not knowing where to go, and so on. The problem with testing for these factors is that every player is different; no two players are going to enjoy the exact same thing in the exact same way. All of these factors can break the flow of the game in general and would certainly break the flow mindset for the player.

To tune for flow there has to be a wider adjustment of the *arousal* and *skill* function balances (see the illustration in chapter 5). Capcom's *Resident Evil 5* (2009) used a system they dubbed the 'difficulty scale' which ranked a player's performance from one to ten. The game engine would adjust the enemy tactics, damage and attacks based on the level the game's player analysis engine felt the player was achieving. There was an explicit player selected difficulty level for the game (Amateur, Normal and Veteran) but this form of a DDA nuanced the gameplay, so that the player would ideally never become so frustrated with an element of the game or feel too overwhelmed by difficult enemies.

Valve's *Left for Dead* (2008) approached the issue of flow entropy by not having a linear narrative but instead making the game a series of combat vignettes. *Left for Dead* is a game where you, as a solo player or part of a team, take on a zombie horde. The problem was that, as with previous games of this genre, the developers recognized that once the maps have been studied and the spawn areas for the zombies established, the game could become repetitive and boring. Players could avoid the horde or just camp out near a spawn point and get high scores. In answer to this Valve created an 'AI Director' to procedurally generate experiences for the player every time they entered a chapter/environment. The AI Director code would analyze the individual player and the team dynamic and would then generate enemies based on this information. The result was a game that was never the same twice, the experience was always novel and one that adapted to the player's skill levels. Flow was achieved by balancing skill against anxiety. The player didn't have to think about the map or environment, but instead just react to what the game was throwing at them in unexpected ways. This was a relatively new form of game mechanic, a game that was always a different experience consistent with the world it was situated in.

Flow as a game mechanic.

One solution to the problem of flow entropy and player disengagement from flow that Jenova Chen puts forward is the inclusion of player choice. Embedding choices in the gameplay, and allowing the player to look at those choices as part of the game, enables a player (and the game tracking them) to get a better sense of the flow state. Choices enable the player to be more intuitive, and a conscious choice is a much better reflection of the player's state of mind in the game. For example, if the player wants to only use the sniper rifle in a level, adjust the game for that. If they only ever play as a barbarian or 'tank' adjust the game's parameters for that, or conversely build in missions or enemies that encourage experimentation with different player styles (but do not force the player to adopt them). A designer should never 'punish' a player for making the 'wrong' choice in a game if they're given an option. It may be a lot harder to snipe through a level, but adjusting the game dynamically for that allows for a deeper sense of reward for the player and being comfortable with that choice enables better flow states.

> In 2006 Chen nuanced his proposed DDA system to make it more relevant to contemporary games and their complexity.
>
> - Include a wide spectrum of gameplay with varying difficulties and options for the player.
> - The DDA system has to be player-oriented and allow for variance is player styles and pacing.
> - Embed DDA choices into the core gameplay mechanic, let players make meaningful choices.

The study of flow in games has increased in the past decade, and taking flow to another level are the rhythm games such as Rockstar's, *Dance Dance Revolution* (DDR) and other music or rhythm-based games. The only way for a player to do well in these games is to push them towards a state where they're not conscious of their body movements. This is the same if it's the movement of their feet on a DDR mat or their hands on the colored keys of a *Rock Band* guitar (Harmonix, 2007). The player is focused on timing and rhythm, without being conscious of their movements. Rhythm games, perhaps more than any other game, create the feeling of being 'in the zone'. People cannot maintain the state for long periods of time; it's rare for people to be swept away for hours, so a three of four minute music track is a close-to-perfect model for zone engagement. The zone state can then be re-entered time and again once the player has had time to mentally 'reset'. As with the updated Chen DDA model, players in Rockband can select different tracks to match their moods as well as difficulty level that provides feedback on mastery of the game.

Conclusion.

Not every game has to have flow elements, some will occur in a game by chance rather than by design. It is something worth considering when designing a game, but is harder to create the 'in the zone' state in sprawling narrative games over repetitive rhythm-based games. However, a version of flow is certainly 'getting lost' in the world the developer has created. As a player becomes less and less conscious of the room they are sat in, as they wander around the game's landscape and play for hours at a time; this is a valid form of flow. The psychology of the open-world game comes directly from the concepts of designing for flow and is one reason it's hard to put down games like *Grand Theft Auto 5* (Rockstar North, 2013) and Fallout 4 (Bethesda, 2015), because they are built around difficulty adjustment and flow concepts.