

## The 13 Basic Principles of Gameplay Design

by Matt Allmer

Gameplay design is chaotic and full of frustrations and contradictions. More often than not, the request is to come up with something guaranteed to be successful. This condition steers solutions towards the established -- which means solutions that have been done before.

But in the same breath, the product must separate itself from the competition or stand out in some way. This immediately pulls the designer in conflicting directions.

Then, whatever the solution, it must fit within the confines of the project's resources. Not to mention scheduling pressure and strategy changes coming from executive positions.

Hup hup! No time for analyzing the previous paragraph! We've got a title to ship! Never mind your lack of proper tools! Quit your sniveling! Don't you know?

Game design is like sailing a ship while still building the hull! Jump out of a plane while still sewing your parachute and you'll get a good sense of pace in this business. The horse is never put before the cart. We race them side-by-side to see which one wins!

With so much urgency, conflict and uncertainty, there must be an anchor somewhere. Call me boring, but I'm a fan of preparation and established fundamentals. They give me a better understanding of which rules I can break, and which rules I should think twice about.

I took a traditional animation class in college and on the first day, the professor handed out the "[12 Principles of Animation](#)", introduced by Frank Thomas & Ollie Johnston. If you're not familiar with these two, they were part of the Nine Old Men: The legendary Disney animation crew responsible for the studio's timeless classics, such as, Snow White, 101 Dalmatians, Bambi, Sleeping Beauty, and others.

At first, these 12 principles were difficult to fully grasp. However, by the end of the semester, I noticed the more principles I applied to my work, the better the animation. Remembering that experience, I think to myself, "By George! Game design should have something similar!"

So, George and I scoured the Internet. Unfortunately, I was disappointed after finding so many disjointed theories, strategies, approaches and creeds. There was a lot of broad subject matter like theories on fun, rewarding players' choices, controlling thought activity, mental multi-tasking... and calls to "simplify" (whatever that means. I'm a designer for crying out loud).

I also found principles so apparent, Captain Obvious would roll his eyes: "know your audience", "don't break the player's trust", "give players choice", "know thyself", "one mechanic in the engine is two in the bush". Alright, the last two were made up, but nothing I found really did it for me.

I was perplexed. None of what I found would help a designer on a day-to-day basis. So George, Captain Obvious and I have decided to throw our proverbial hat into the muddled picture. (And quick! For god's sake, before I collect any more metaphorical personalities!)

The 12 Basic Principles of Animation was my starting point. I took the commonalities and added to them based on what I've identified as the different compartments of gameplay design. You'll notice some are described similarly and some even have the same name, but all apply to gameplay.

The purpose of these principles is to cover all your bases before presenting your designs. You might have a principle fully covered in the beginning, but these principles may spark a thought later when circumstances present a new opportunity. Think of this as a reference sheet. And now, without further ado...

## **Direction**

The first three principles have to do with leading and directing the player's experience. Even though this medium is heavily based on personal, interactive discovery, it is still an artistic medium.

Do not underestimate the importance of artistic direction. Just as a painting leads the eye, a book leads the imagery, a film leads the narrative, so too must a game lead the interactivity.

### **1. Focal Point**

Never allow the player to guess what they should focus on. At the same time, always allow secondary subject matter, but it is the designer's job to clearly provide the primary focus at all times. This applies to both visual and visceral aspects of gameplay.

#### *Level design example*

Creating clear, apparent lines of sight.

#### *System design example*

Clearly defined plot points and objectives during game progression/user experience.

### **2. Anticipation**

Time is needed to inform the player that something is about to happen. Always factor in Anticipation when designing and implementing events and behaviors.

#### *Level design example*

A train sound effect occurs before player sees train.

#### *System design example*

An energy charge builds before the lightning attack occurs.

### **3. Announce Change**

Communicate all changes to the player. This short step occurs between Anticipation and the event itself.

The important part to remember is maintaining a hierarchy of notable changes.

A good rule of thumb is degree of rarity. If a change occurs a hundred times in an hour, the announcement may not be required. However, if the change occurs five times throughout the entire game experience, a number of visual cues could be needed.

This principle is so obvious, it can be taken for granted and sometimes overlooked. Be diligent in knowing what changes the player should be aware of at the correct time and on the correct event.

*Level design example*

"Cast-off" animations trigger for NPCs when the player's character boards the ship.

*System design example*

An on-screen notification occurs when quest criteria have been completed (i.e. "Slay 10 goblins for Farmer Bob")

## **Behavior**

These next four principles address the very important aspect of behavior. This tackles the player's expectations, both conscious and unconscious. This is where common design theories are addressed such as player choice, reward and payoff, etc. These principles are also broader, so they can be applied to additional types of design like UI and story...

### **4. Believable Events and Behavior**

Every event or behavior must occur according to the logic and expectations of the player. Every action, reaction, results, emotion and conveyance must satisfy the players' subconscious acceptance test.

*Level design example*

Place destructible objects near an explosive object. This way, the explosion looks more believable.

*System design example*

Weaker enemies run away when the advantage shifts in the player's favor.

*UI example*

HUD elements are affected when player's mech is near death.

*Story example*

Villagers are more upbeat and react positively after the player has slain the dragon.

### **. Overlapping Events and Behavior**

Dynamic is lost if only one change occurs at a time. Discover the right amount of events to occur at any given moment of time.

*Level design example*

Providing the player the ability to build from an appropriate list of structures.

#### *System design example*

The linebacker points to direct fellow players, the defensive end shifts over, the quarterback points and calls out football jargon and the crowd cheers louder because it's third down. All this occurs before the snap.

#### *UI example*

Points accumulate in the score while each kill is individually tallied on screen.

#### *Story Example*

Multiple plot points are at the forefront of the narrative experience. Example: the king is on his deathbed while his war is being waged and he has yet to announce an heir -- all while an unknown saboteur orchestrates a military coup.

### **6. Physics**

The player's primary logic operates within the known possibilities of physics. Keep in mind gravity, weight, mass, density, force, buoyancy, elasticity, etc. Use this as the starting point, but do not be limited by it.

#### *Level design example*

Ensuring a hole in the floor is the correct size for the correct purpose. Whether it is part of the path of level progression, or simply for visual aesthetics.

#### *System design example*

A spark particle effect occurs when the player's vehicle scrapes the side of the concrete wall.

#### *UI example*

The GUI's theme references scrapbook elements. In which case, animated transitions, highlights, etc. follow the physical characteristics of paper.

### **7. Sound**

Ask yourself, "What sound does it make when \_\_\_\_\_ happens?" "Is the sound appropriate?" "Is the sound necessary?" "Does it benefit the experience or hinder it?" If players close their eyes, the sound alone should still achieve the desired affect.

It's debatable whether this principle should be included since Sound Design can be considered separate from Gameplay Design. I've included it because sound is crucial and can easily be neglected. The more it is considered, the better the experience is for the player.

#### *Level design example*

Flies in swamp level make a sound when close to the camera.

#### *System design example*

A proximity system where sound effects volume fluctuates depending on distance of game assets.

*UI example*

Only visually prominent graphics have sound effects attached to them, so as not to muffle the auditory experience.

The next three principles individually touch on other major design components.

## **Progression**

### **8. Pacing**

Keep in mind the desired sense of urgency, the rate in which events occur, the level of concentration required and how often events are being repeated. Spread out the moments of high concentration, mix up the sense of urgency, and change things wherever possible to achieve the proper affect.

*Level design example*

Create areas for the player to admire the expansive view, versus areas where the player feels claustrophobic.

*System design example*

Create long, powerful attacks versus short, light attacks.

## **Environment**

### **9. Spacing**

Understand how much space is available both on-screen and in-world, recognize the spatial relationship between elements and take into account the effects of modifying those spaces.

*Level design example*

Lay out the appropriate amount of space for the appropriate number of enemies to maneuver correctly.

*System design example*

When an AI character moves through a bottleneck area, walk loops switch to standing idle when the AI character is not moving forward, to show that the character is "waiting" to move through the narrowed space.

## **Method**

### **10. Linear Design versus Component Breakdown**

Linear Design involves solving challenges as they come. All solutions and possibilities hold the same institutional value. Focus can be lost with this method, but it provides creative and spontaneous solutions.

Component Breakdown involves systemic categorization and forming a logical hierarchy of all solutions. This method can restrict innovation but preserves clarity of primary design objectives.

This principle does not mean designers must choose one or the other. There are times during development where one method is more appropriate than the other.

For instance, pre-production provides plenty of time for breaking down a sequence of events. However, when the publisher drops a "must have" change after pre-production, linear design can provide an acceptable solution quickly.

#### *Level design example*

Typical blocking of level geometry in an early stage of development, versus adjusting a small area of the same level to implement an idea that wasn't thought of until later.

#### *System design example*

Identifying all major systems (combat, AI, input, etc), and progressively filling in various levels of detail versus conceiving the first couple of levels and extracting possible systems based on a linear player experience.

### **Foundation**

The final three principles mark the foundation of gameplay design, which are listed in reverse order of importance. These should be a surprise to no one.

#### **11. Player**

How does the player factor into this? How does the player interact with everything that has been designed? More than just device input, address how the player contributes to the experience. If it's a good idea and you're able to convey it correctly but the player is not into it, change it or scrap it!

#### *Level design example*

Setting up the player in hopes of making them jump out of their seat.

#### *System design example*

Orchestrating progression so that the player feels empowered, determined, anxious, etc.

#### **12. Communication**

Is the appropriate team member correctly aware of the objective? Are the appropriate developers clear on the solution? If it's a good idea but you can't communicate it correctly, it might as well be a bad idea because it's very likely to be received as such.

#### *Level design example*

Using the elements of the environment so the player is compelled to travel in the correct direction.

#### *System design example*

Using visual cues so the player learns when to punch rather than kick, jump rather than strafe, etc.

#### **13. Appeal**

When addressing anyone, ask yourself, "Does this draw the audience in?" This applies to (but is not limited to) the player, the spectator, your fellow developers, the publisher, and their marketing team. If it's not a good idea, there's no need to continue until it becomes a good idea or is replaced by something better.

*Level design example*

Running down the street is not fun, but running down the street while being pursued by government secret agents is.

*System design example*

Punching can be fun but when the camera shakes on impact, it's even more fun.

**Conclusion**

So, there you have it. These principles have noticeably improved my designs and forced me to think of components from all angles. I thoroughly believe they will give you an edge on all those impatient carts. So, stick that in your horse and race it!

*[If Gamasutra readers have other major principles that you believe are missing from this list, or any suggestion for the current 'Principles' list, please comment below.]*

Fonte: GAMASUTRA

The art & business of making games

<http://www.gamasutra.com/>