

Many moons ago I wrote about a massive misunderstanding in gamification around [game mechanics and what they actually are](#). There were several lists around that said they were key [game mechanics](#), which turned out to be very little to do with actual mechanics. Fast forward almost 2 years and, well it is getting better, but there is still a lot of people getting it confused. That is easily done as even in game design circles there is an argument over what they truly are, but there is a general high level agreement at least. As I summarised in my post:

A distinct set of rules that dictate the outcome of interactions within the system. They have an input, a process and an output.

So if we take shooting in Space Invaders.

- **Input:** User hits fire
- **Process / rules:** Bullet speed, bullet vector, position of enemy
- **Output:** Miss, nothing. Hit: explosion, score increase.

Some started to use the phrase Gamification mechanic, to get around this problem. Sadly this suffers the same issue. Let us look at something that is often referred to as a mechanic in gamification, Epic Meaning. Now, this is something I use in gamification all the time, creating a sense that what you are asking of the user has a purpose or meaning greater than just the activity they are being asked to do. If you consider a car, every part has to work perfectly for the car to perform, from the smallest cog onwards. If we look at the idea that a mechanic is a set of rules that take an input, do something to it and then output something, how does Epic Meaning fit in?

- **Input:** Well, it doesn't really have one. Person is in the system
- **Process / rules:** creating an atmosphere or narrative around meaning. This would break down into many many different elements in the system.
- **Output:** The user feels a sense of purpose and greater meaning

Ok, that was hard! The reality is, you can't break something like Epic Meaning down like this, so it isn't really a mechanic. It is the outcome of many different mechanics and interactions within a well designed system. Let's look at our space invaders example a little more and see what else is going on though.

- "User hits fire". This requires an interface initially, that takes the users inputs and

interprets them. This then triggers the creation of a bullet graphic on screen.

- *“Bullet speed, bullet vector, position of enemy”*. The intrinsic rules of the game dictate how fast the bullet will travel, where the position of the player will determine the exact direction it will take. More intrinsic rules will tell the game where the enemies are and if they have been hit by the bullet.
- *“Miss, nothing. Hit: explosion, score increase”*. If the bullet misses, obviously nothing happens. If it hits an enemy (which will have several rules defining the hit area), there will be some sort of visual feedback. If the enemy only takes one hit to kill, you get an explosion. If it takes more, you will get some sort of other feedback showing you were successful but have more to do. You will also get more feedback showing you an increase in score. If it is the last enemy, you may also move on to the next level (progression).

What would be more help in Gamification?

It may help when looking at gamification to consider the following sets of things it could be. There is cross over in some of them, so this is a general guide, obviously based on the [MDA framework](#).

- **☒ Mechanics:** A set of rules that define what can be done in the system. These are defined by the designer.
- **Schedules:** Rules that define how and when certain things happen in the system, such as how many points lead to the next level, when badges are given, needing to have x & y to get z etc.
- **Dynamics:** How the mechanics and the user act together in real-time. Mostly out of the designers control and can lead to unexpected / emergent outcomes.
- **Feedback:** Representation of the results from actions taken in the system. Points, badges, progress bars, messages.
- **Tokens:** Virtual items. Points, rewards, collectables and even points are all tokens.
- **Interactions:** Points of contact between the user and the system, such as a mouse click.
- **Aesthetics:** The emotional response of the user to the system. Joy, fear, frustration etc.

Taking a common gamification example, how could these be used to describe some of what was happening in a hashtag based competition (Tweet the hashtag the most sort of thing)?

- **Mechanics:** Calculation of user tweet and retweet totals.
- **Schedules:** After 10 retweets of the hashtag, the user gets a badge. Final win condition.
- **Dynamics:** Some users may decide to spam their networks if there are not explicit rules preventing them ie. The “game” allows it.
- **Feedback:** Users are sent an email to thank them. They are also given points, badges and a position on a leaderboard.
- **Tokens:** The user is given redeemable points.
- **Interactions:** User creates a tweet with the correct hashtag and send it.
- **Aesthetics:** Some users will enjoy being on the leaderboard. Others may be frustrated by how other users decide to play.

Obviously there is a lot more going in, but you can see the basics there and why it is so hard to talk about real mechanics in gamification. I have not even spoken about the fact we need to consider [motivation](#) of the user to actually do something. Many of the things that are spoken about as mechanics are really motivations or drives! Last time I wrote about this I asked the question, does it matter? Then my answer was

In the grand scheme of things, probably not. However, as I said at the start, as gamification matures so should the language used to describe it. It is fine to steal ideas and phrases for other disciplines, but as we abuse them they begin to lose their meaning.

I have changed my mind a little. It does matter actually. If we want people to take us seriously, we need to show we understand what we are talking about. Game designers already think we are sheep spouting psychobabble, why give them more ammunition by misusing their terminology.

Please wait...