

# Insights on Game Mechanics

Chandan RT

---

## Contents

<b>1 Randomness</b>	<b>3</b>
1.1 Incorporation . . . . .	3
1.2 Types . . . . .	4
1.3 Importance . . . . .	4
1.4 Disadvantages . . . . .	5
1.5 Human biases . . . . .	5
<b>2 Information</b>	<b>6</b>
<b>3 Health</b>	<b>7</b>
3.1 Indicators . . . . .	7
3.2 Types . . . . .	7
3.3 Effects on Game-play . . . . .	9
3.4 Nuances in Game Mechanics . . . . .	9
<b>4 Death</b>	<b>9</b>
4.1 Depiction . . . . .	10
4.2 Impact . . . . .	10
4.3 Justification . . . . .	11
<b>5 Pillars of Game-play</b>	<b>12</b>
<b>6 Game Worlds</b>	<b>12</b>
<b>7 Melee Combat System</b>	<b>12</b>
7.1 Basic moves . . . . .	12
7.2 Stages . . . . .	13
7.3 Related Systems . . . . .	13
7.4 Balance . . . . .	14
<b>8 Ranged Combat System</b>	<b>14</b>

<b>9</b>	<b>Detection System</b>	<b>14</b>
<b>10</b>	<b>Pursuit System</b>	<b>14</b>
<b>11</b>	<b>Skill Trees</b>	<b>14</b>
<b>12</b>	<b>Addiction Mechanisms</b>	<b>14</b>

# 1 Randomness

Randomness operates at different levels in games and has a profound effect on experience. It can effectively make or break a game.

## 1.1 Incorporation

- a) **World Generation:** A procedurally generated world allows the player to replay a game many times under differing circumstances. **Minecraft** generates random numbers called seeds and uses them to create worlds (allowing you to recreate a world by knowing its seed). On the other hand, space-themed games like **No Man's Sky** usually generate a universe randomly and let all players experience the same universe.



A randomly generated minecraft world

- b) **Level Generation:** Rogue games like **Spelunky** and **Rouge** (very old) generated entire levels randomly. Such an approach forces players to learn and utilize game mechanics effectively.
- c) **Interactions:** The successful execution of an interaction and/or the effectiveness of an interaction between the player and other entities can be randomly decided. For example: critical hits and shots. In card-based games, randomness plays a central role.
- d) **Future events:** The occurrence and effects of future events can be randomly decided. For example: an opponent's intentions and moves.
- e) **Loots and Rewards:** Loots from in-game chests and kills are randomly decided. Rewards from loot boxes are also random.

## 1.2 Types

1. **Input randomness:** Random elements are introduced before the player executes his move. Card drawn games are perfect examples. Procedurally generated worlds and levels are also considered under the same category.
2. **Output randomness:** Randomness that comes into play after the player makes his move. Examples include: chance and effectiveness of hits, outcome of plans, loots and rewards. This type of randomness inevitably has a profound effect on the player throughout game-play since it affects the outcome of their every move. For example: in **Risk**, the outcome of every move you make is decided by the throw of dice.



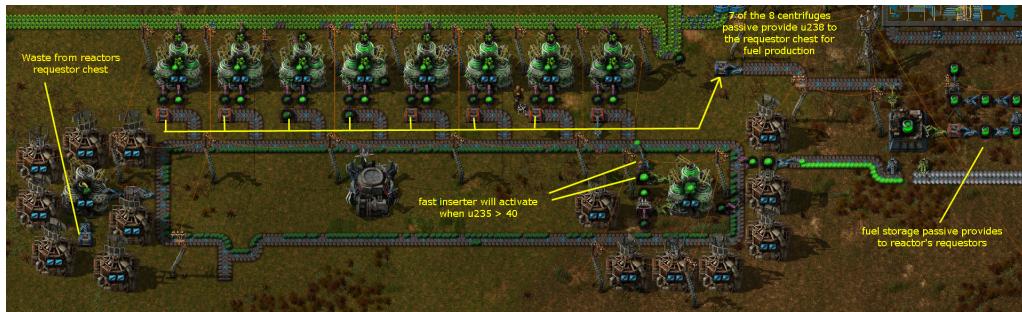
Risk: World Domination

## 1.3 Importance

- When worlds and levels are procedurally generated, players go through a new experience on every play-through. This also has the added benefit of reducing game sizes.
- Incorporating randomness in interactions allows newcomers to obtain some benefits/avoid losses based on chance and not skill, thereby balancing the competition. This approach also allows the system to hedge the player during vulnerable circumstances. Several games actually improve the player's odds when they have low health, whilst nerfing the opponent(s)' abilities.
- The random nature of future events prevents players from considering every outcome of their potential moves and ruining the fun of the game. It also forces players to prepare for the unexpected if/when their plan fails. In games with an abstract combat system, randomness models mistakes.
- When loots and rewards become random, the player is motivated to explore for chests, deal with opponents and play therefore the game significantly more, usually desiring something that is rare.

## 1.4 Disadvantages

- Most game developers are in favour of hand-crafting levels and worlds in order to provide better game experience, at the cost of game size and re-playability. Randomly generated levels may not allow users to experience creatively designed levels and certain game features.
- Certain rare events tend to form a highly non-uniform distribution. In **Factorio**, (during Uranium processing) U-235 is obtained with an odds of 1 against 142. The player naturally expects one U-235 in 143 cycles. In reality, long stretches of U-248 are interrupted by short stretches of U-238. This is due to the pseudo nature of random numbers. Because of this quirk, players are wary of using nuclear power lest their factory lose power due to bad luck.



Uranium Processing in Factorio

- Randomness is responsible for the occurrence of certain impossible events. In earlier versions of **Sid Meier's Civilization**, warriors with sticks sometimes defeated tanks. This quirk is also undesirable since a tank represents more investment.
- Multi-player game often struggle to balance the play-field between newcomers and seasoned players (in order to not demotivate the newcomers). Sometimes, randomness ends up dominating over skills, which frustrates seasoned players.
- In order to increase revenues, game developer companies make certain items/rewards extremely rare or sometimes unattainable, whilst allowing players to purchase the said items/rewards using actual cash. A scenario known as **Play-to-Win** takes place, wherein players who acquired rare but powerful items by buying them end up not-so-unexpectedly dominating the game, thereby frustrating the other players. In such cases, the other players are forced to either invest a lot of time in the game or pay with cash.

## 1.5 Human biases

Humans perceive probabilities differently. An outcome with 99% probability is definitely supposed to occur whereas an outcome with 85% probability is highly likely to occur. Due

to these nuances, games that quote probabilities often roll-them up under the hood. For example: if an outcome shown to have a 70% probability, then it actually has around 90% probability.

Another issue is expectations. If an outcome has a probability of 20%, then players expect it to happen atleast once in five tries. In such cases, games actually keep track of players getting unlucky and definitely reward them on the fifth attempt. Along the same lines, certain games assure that a rare event takes place after a certain (albeit large) number of trials.

## 2 Information

The player is presented with varying degrees of knowledge of the states of in-game entities. However, providing too much information can be overwhelming. Additionally, a player can look-ahead and calculate the best move if they have complete information. Therefore, game developers expose varying amounts of information to the player and attempt to remove determinism through various means. This forces the player to play the game in a relaxed, less-calculative manner (as the game developers intended) and expected the unexpected:

- a) **Fog of War:** An opponent's states and moves are not shown to the player. Classic examples include **Age of Empires** and **StarCraft**.



Fog of War (literally) in Settlers Online

- b) **Exponential Complexity:** In games like **Go** and **Chess**, the number of possible configurations after only 4 or 5 moves is extremely high for a player to consider.

- c) **Outcome of interactions:** As explained in the randomness section. Choosing to hide the odds for/against an outcome represents another level of information hiding.
- 

## 3 Health

A variable associated with the state of in-game, destructible, non-sentient or sentient entities. However, the following paragraphs describe only player health

### 3.1 Indicators

- a) **Heads-On Display:** The player's health may be conveyed via a health bar or a gauge of some sort, usually on the upper left corner of the screen. Some games depict the health as a percentage or in actual numbers
- b) **Screen:** Damage taken is indicated by reddening or blackening of the screen, usually conveying the direction from which damage was inflicted. First adopted by **Call of Duty**.



Call of Duty: Modern Warfare 2

- c) **Player Attributes:** Changes in certain attributes of the player upon taking various amounts of damage can also indicate health. In some **Mario** games, Mario's size served as the only indicator of health

### 3.2 Types

#### Persistent

In games of the 80s era, any threats to the player could be easily avoided. Enemies' attacks were predictable, projectiles moved slowly, and traps could be seen from afar. In such a

system, the health bar provided some leeway by allowing the player to make some mistakes (but not too many). Such games adopted persistent health.

Persistent health usually means that the player's health does not automatically regenerate health in any way. The player has to consume a health utility to reclaim lost health. However, in almost all games, health is replenished at the beginning of levels.

Such a system forces the player to play perfectly, as any mistakes now can affect later prospects. It can make things tricky if the player encounters hard situations when their health is expected to have depleted (boss fights at the end of a level). Games with a persistent health system demand more attention from the gamer whereas low health conditions can make them feel tensed or anxious.

Games with persistent health and health utilities force the player to explore, in search of the latter. In cases where health utilities crafted alongside many other entities, the gamer has to strike a balance. If the character can store health utilities for later use, but the inventory is finite, then the player has to compare the usefulness of other entities. In general, persistent health forces the player to experience many elements of game-play.

### Regenerative

When further elements of the FPS or RPG genre were introduced, the player could no longer predict and avoid threats. Since the player could lose health for no fault of their own, some games decided to allow automatic regeneration of health (without consumption of a health utility) to various extents.

In most cases of regenerative health, regeneration starts only if the player avoids damage for some time. This may require hiding from damaging entities or escaping combat altogether. In some games, there are conditions that are to be satisfied for regeneration. In higher difficulties of **Minecraft**, the player's hunger must be saturated in order to regenerate health.



Minecraft's health bar (left) and hunger bar (right)

Regenerative health shifts the focus from avoiding damage towards limiting exposure to damage. Taking damage does not have any long term consequences. It gives the game a lot of momentum and allows the player to take risks. Players can also face difficult situations with full health.

## Semi-Regenerative

The player's character in **Far Cry**, or ships in **Assassin's Creed** have a segmented health bar. Regeneration takes place only within the current segment. Further restoration demands health utilities.



Health segments in **Far Cry 3**

## One-Hit

In games like **Temple Run** or **Subway Surfers**, a mistake can end the game at once.

### 3.3 Effects on Game-play

As discussed before, persistent health influences game-play much more than regenerative health. However, there are some interesting nuances:

- Glory kills in **Doom: Eternal** rewards the player with orbs of health. This makes players with a low health to take further risks by melee-killing an enemy by closely approaching him. Many games reward health for kills.
- Since Mario's size is dependent on his health, a large sized Mario can stampede obstacles whereas a small sized Mario can explore entire regions of a level.

### 3.4 Nuances in Game Mechanics

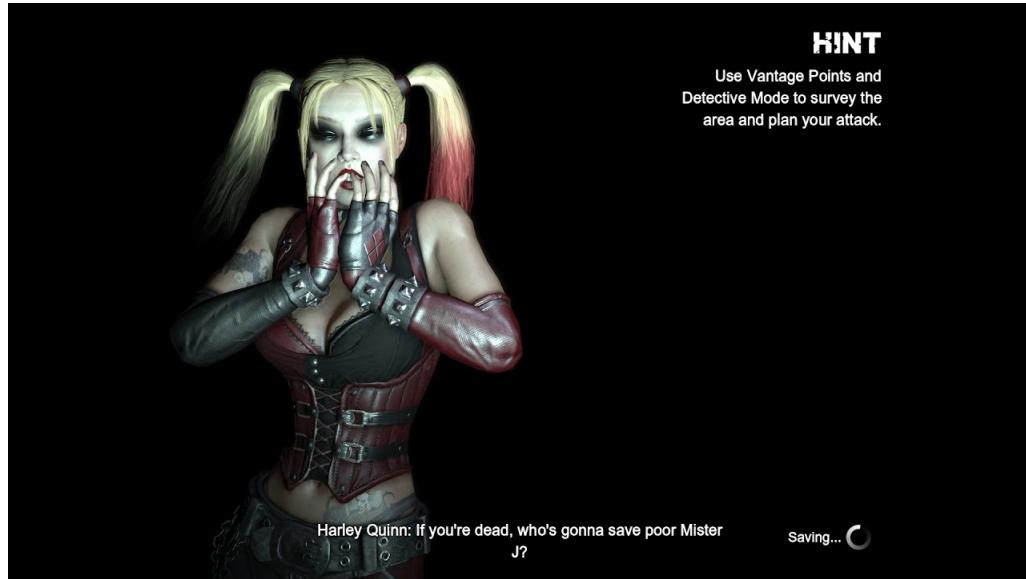
Most modern-era games with a persistent health become forgiving towards the player at lower-health in various ways. Health utilities spawn more often. The enemies' damage is nerfed. The player's chances of landing a critical hit is increased, etc. In several games, the last portion of the health bar can actually account for more damage than usual, in order to provide the gamer with the thrill of playing on the edge.

## 4 Death

A state reached when the player's health reaches zero, or an objective isn't completed within the given constraints.

## 4.1 Depiction

Most games just show the character falling down and dying. In cases where an objective is not completed within the given constraints, the screen just blacks out or fades away (depicting a figurative death). *Advise* is displayed shortly before re-spawning. **Call of Duty: Modern Warfare 1** (in)famously displayed warfare-related quotes from famous men. In the **Batman** games, the adversary taunts you.

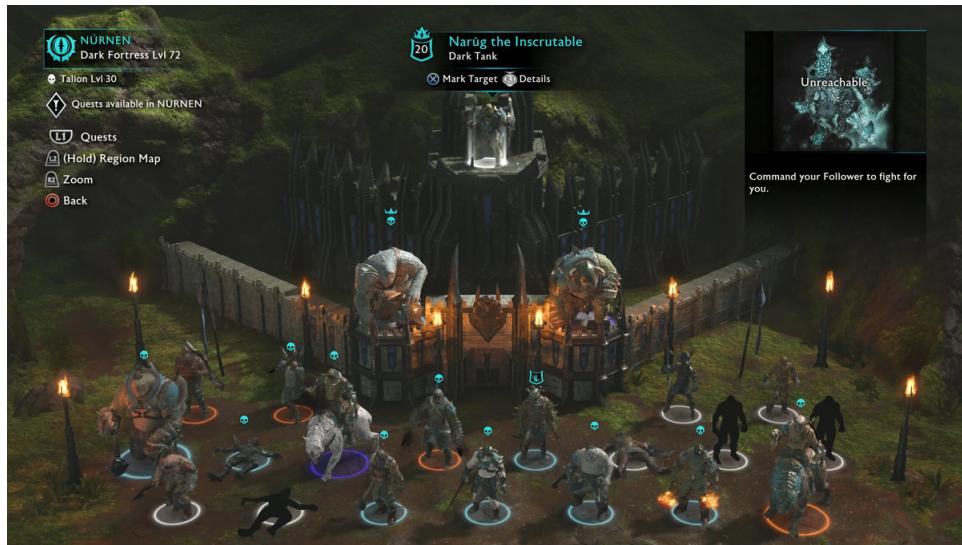


Harley Quinn's taunt in *Batman: Arkham City*

## 4.2 Impact

- a) In most cases, the game just resets to a previous state/checkmate and no evidence of your death is depicted. This is akin to just starting over and having no consequences of failure.
- b) In some games, inventory and experience levels are lost. They may or may not be claimable by going to the location of death after re-spawning. Everything else remains same. Very few games leave the player's dead body and/or blood at the location of death. In **Fable**, the player's character (upon re-spawning after death) showcases scars that were the case of death. Additionally, these scars juxtapose across deaths.
- c) Some games represent the death of the player's allies in the form of gravestones and obituaries. In certain games, a bond is formed between the player's character and other character(s), which fosters emotion upon death of the latter.
- d) In **Shadow of Mordor**, the Orc that is likely to get promoted within the organisation of Sauron's army. This is the famous **Nemesis system**. However, since the player's character is a nazgul, they can resurrect. If the player confronts the same orc after

resurrecting, then the orc makes a mention about killing the player before indulging in a fight again



The Nemesis system in Shadow of Mordor

- e) Among a very small subset of games, the character may lose certain powers or their traits may change. The character may change altogether. In certain anime/manga themed games with multiple protagonists, the story continues after death.
- f) Older games usually had the concept of "lives". Losing all lives led to "Game Over" where one must start afresh. Lives was sometimes a collectible (albeit rare) utility. This concept has been disappearing.
- g) Over the years, the consequences of death are becoming less impactful. Some games let you keep inventory and experience. Some of these don't even bother to reset your kills and loot between the last checkpoint and your death. Checkpoints are themselves becoming more frequent.

### 4.3 Justification

- The overwhelming majority of games don't even remotely try to justify death and re-spawning shortly after. They just shrug it off as mistakes.
- In **Assassin's Creed** games, "death" occurs when the character commits actions vastly different from his ancestor(s). This causes de-synchronisation of the animus' simulation from the player's genetic memories. The animus restores the character to a stable point.



Animus rewinding to a stable point in Assassin's Creed: Rouge

- In **Portal 2**, the player controls robots. Since it is implied that there are an infinite number of these, re-spawning is justified.
- 

## 5 Pillars of Game-play

## 6 Game Worlds

## 7 Melee Combat System

The melee combat system is responsible for contact interactions between the character and their opponents. There are certain base similarities across games:

### 7.1 Basic moves

- a) **Attack:** A move to cause damage. The stages are described with respect to this. The attack move may have several variations depending on the character's and opponent's states. Certain moves may be context-specific, like the finishing-off move when the opponent is at zero health.
- b) **Block:** A move to avoid damage by impeding motion of the opponent's melee weapon. Should be executed before the contact phase of the opponent's attack begins. Certain games have an hierarchy of weapons or skills that dictate whether an attack can be blocked or not. Some games leave this to chance.

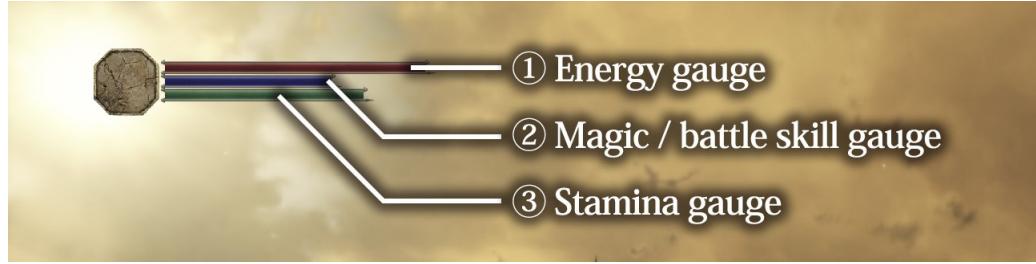
- c) **Dodge:** A move to avoid damage by moving the character's body away from the opponent's melee weapon. The character may either change his poise, or run backwards.
- d) **Counter:** A move to avoid damage and cause damage to the opponent at the same time, by interrupting the their attack and striking them shortly after. Timing is of importance.
- e) **Charge:** A move to stun the opponent by utilizing high momentum before an attack. Drains a lot of stamina/energy. May be interrupted if not timed correctly.

## 7.2 Stages

- 1) **Preparation:** This stage begins shortly after the player presses the attack button. The character begins to build up momentum by swinging their melee weapon. This stage usually lasts several frames. The character is usually vulnerable to attack during this frame. Certain opponent's moves may interrupt the character during this stage and stop the attack. Some games allow the player to switch to a different move during this stage whereas some games don't. High damage moves typically have a longer preparation phase than basic moves.
- 2) **Contact:** This is when the melee weapon comes in contact with the opponent and deals damage. Certain combos may exist which allow the player to quickly return to this phase provided they execute moves at near-perfect timing. The opponent may have moved away, dodged or blocked the move and thereby avoided damage.
- 3) **Recovery:** After successful or failed contact, the character spends some time in regaining posture or preparing/positioning the weapon for the next move. The player is usually vulnerable to attack during this phase. Most games typically don't allow the character to perform a move during this phase, although some games do register the key-press and execute the required move after this phase ends. Some high-energy moves have a longer cool-down.

## 7.3 Related Systems

- i. **Snapping:** Games allow the character to focus their melee moves on an opponent to varying extents. Some games highlight the opponent in focus and allow the player to execute moves that will be carried out on the focused opponent. Few games expect the player to position themselves and direct the attack based on the position of the opponent.
- ii. **Energy/Stamina:** A quantity that dictates the player's abilities to perform certain moves. Typically regenerates over time. Certain games have utilities or skills that can temporarily or permanently increase the regeneration rate or the capacity of energy/stamina. This mechanism forces the player to plan moves carefully and not spam attacks.



Various bars in Dark Souls 3

- iii. **Character profile:** Certain heavy moves attract more attention from the surrounding entities. A character in low profile can carry out stealth kills, provided no one observes the character performing it, or notices the dead body later.

## 7.4 Balance

Games strive to maintain a balance between the player's aggression, defensive and counter stances:

- Over-aggression is prevented with the energy/stamina meters. Interruption of attack moves during the preparation phase and preventing switching of moves during the same phase forces the player to plan attack moves judiciously.
- Defensive stance is discouraged by draining energy/stamina, leaving successful defence to chance, allowing opponents with higher-tier weapons/skills to break the character's defence, and surrounding the character with many opponents.
- Counteracting is usually given a failure rate in order to prevent its overuse. A character in counter stance is also left vulnerable in some cases. Additionally, some opponents or some of their attack moves cannot be countered.

## 8 Ranged Combat System

## 9 Detection System

## 10 Pursuit System

## 11 XP and Skill Trees

## 12 Addiction Mechanisms

- a) Daily Rewards:

b) **Continuous Positive Reinforcement:**

c) **Resource Accumulation:**

d) **Loss Aversion:**

---