

Thoughts

[A Single View of Marriage](#) | [Biblical Sexuality](#) | ['Christian' Books](#) | **Gaming** | [Missionary Kids](#) | [Pet Peeves](#)
[Pornography: Confession, Healing, and Contention](#) | [Respecting Women](#) | [Single Notes](#)
[Snippets](#) | [Spiritual Forces](#) | [Tension Points](#) | ["Wild at Heart"?](#)

Gaming

[Why do I game?](#) | [Problems of gaming](#) | [Hacks vs. glitches](#) | [PC vs. consoles](#) | [Downloadable content \(DLC\)](#) | [Open world/sandbox games](#) | [Artificial intelligence](#) | [Best gaming elements](#) | [Related links](#)

Why do I game? I began enjoying them since before I was a teen. They're fun. And they are a temporary—but needed—escape from the caustic, and often painful, real world surrounding us. We don't live in the gaming world, but it is fun to visit. They're also a means of venting frustration that simply cannot be done in real life. They also provide a sense of virtual friendship and comradery with select other gamers.

Problems of gaming: A tendency to lose track of time and addiction can be problems. Being a victim of hackers/cheaters (a [massive problem](#)), and subsequent anger at hacking because you're just not sure whether or not that gamer used an aimbot or wallhack to take you out. Sadly—truth is—a lot of people hack; [PunkBuster](#) and [Valve Anti-Cheat \(VAC\)](#) notwithstanding.

Hacks vs. glitches: This is a legitimately interesting debate; but one in which I will quickly state which side I'm on: glitches. Why? First, some definitions (mine) are in order, as there is a fundamental difference between them:

- A [hack](#) involves *intentional, malicious modification with external tools or code* by an individual or group of individuals to [manipulate software](#) at the programming level.
- A [glitch](#), on the other hand, is an *existing design failure or bug within the developer's own software/code* that does not require or involve any modification to the software. It exists only because it has not yet been identified and corrected by the developer.

Rockstar, developers of the Grand Theft Auto series, are digging a hole for themselves by treating players who [discover glitches](#) (and post them on YouTube) *as hackers/script kiddies* and punishing them. Rockstar, it is *your own faulty code* that needs fixing. Not the gamers who discovered them. Granted, you may not agree with what [gamers are doing](#) with your glitches (and we might not either), but remember that they are, in effect, acting as your biggest—and most thorough—group of belated beta testers. If they are hacking, then by all means ban their accounts permanently or send them directly to the cheater's pool. But if they discover something within the game that's not working right, then *thank them for it* and fix it quickly. We are living in the information age, so of course it's going to be [on YouTube](#) for all to see—including you...so you have zero excuses not to suck it up and fix it.

Your code is *your* problem, and your game becomes more robust when those glitches are discovered, posted in a public place, and subsequently fixed. And perhaps if you worked harder to monitor YouTube for glitches and/or hired people to solely test for glitches, those bugs wouldn't be there very long. Hmm. So get busy. But what's that? Aww, did your execs really lose some Sharkcard sales (read: profit) because some players made some in-game money off your glitches? Let me call a waambulance for you. Man up, Rockstar. Own your software—bugs and all—and stop your whining. You can learn some valuable lessons from how Valve is handling Team Fortress 2. You'll make more money by maintaining goodwill toward your gamers than you'll 'save' by sending them to the "cheater's" pool for punishment (which should be reserved for script kiddies).

PC vs. consoles: The never-ending [debate](#). Interestingly enough, some people own both a PC and one or more consoles. Why?

PC advantages

- Full modularity: any component can be replaced at any time, giving the gamer full control over how much, what, when, and from who their hardware is purchased and upgraded. Games designed for the PC tends to be scalable, since people own different hardware setups with varying levels of performance and capability. Console hardware slowly loses its graphical and general processing edge over time since only its storage device can typically be upgraded.
- Ease of modding: given its makeup and specialized modding tools, the PC has the greatest appeal to the modding community. GTA is a good example of what's possible, and how much value it can generate and retain.

Console advantages

- Price: costs are typically lower during production due to manufacturer buying power and sheer quantities involved.
- Special treatment: both Microsoft and Sony try to outdo each other in terms of providing their gamers with exclusive contracts and content, or exclusive games themselves. But this can get pretty ridiculous with the underhanded deals and secrecy that result from Mafia-like tactics. In other words, it gets really ugly and old really fast.

But whichever platform you choose, let's be fair and objective about it: games should be developed to fully support the maximum power and unique benefits each platform is capable of. No more nerfing for the lowest common denominator. End of story. Game developers need to:



Where it all started; with one of the *original* consoles: the Atari 2600.
(and yes, we're playing [Space Invaders](#))

1. Stop lying to gamers and trying to cover up your under-handed agendas and marketing hype mess-ups (yes, Ubisoft, I'm talking about you). And yes, at times it will hurt, but try honesty and see what happens; you might be surprised. Good things can happen when you work hard and stop lying to people.
2. Never advertise something you can't guarantee you'll ship. Marketing must always take a backseat role to honesty and the team actually doing the real work. When it doesn't, it makes a mess for everyone.
3. Support your game by fixing bugs (yes, Ubisoft, I'm talking about you again). That means frequently, and long enough that it's widely recognized as stable, playable, and fun. Learn from Valve and Rockstar's dedication to better quality gaming; not pathetic excuses. Valve, via Team Fortress 2, is arguably the hands-down winner on this one; bug fixes and new content are still being released on a near-weekly basis more than seven years after its release in October 2007.

Downloadable content (DLC): Something to look forward to for players, it also provides a source of revenue to the game developer, since marketing knows that we gamers are more likely to fork over our hard-earned money if coerced to do so in smaller increments. Sadly however, there's a lot of eye-candy-only DLC (new content that doesn't directly extend or significantly enhance gameplay) being generated and hyped, which is clearly failing to provide any long-term playability to the game itself. Case in point: much of the DLC that Rockstar releases for GTA stirs excitement that quickly vanishes after only a couple of weeks, only to be replaced with excitement for the next upcoming release. Some DLC has the capability to extend playability through new interactivity, but most does not. Again, the majority of gaming companies use it as a means of increasing their post-game profit; not extending gameplay. This needs to change.

Open world/sandbox games: Because they have fewer limits on your movement and actions, there are more opportunities for creativity, interaction with other players, and playability is extended significantly. The fewer reminders there are that you're playing a game, the greater the feel of realism, which translates into playability. Open world games let you ask the perpetual question of "What can I try next?" A good sandbox game should, as much as possible, mimic the real-world through terrain, physics, weather, objects, movement, and interaction between all those elements. There should be only plausible, minimal limits to movement (such as the outermost edges of the map itself; e.g. [The Thirteenth Floor](#)). No unenterable houses, rooms, or fake doors just there for decoration. If the user isn't supposed to be able to go somewhere, it should be an obvious, plausible part of the terrain or structure. Nothing is more annoying than approaching an object only to learn (after several climb/jump attempts) that "sorry; you can't climb on top of this rockface/crate/box/ledge or jump/climb over this fence" fail.

- Include layering variety and flexible tactics: buildings for height, ground level for main travel, and trenches/underground tunnels/caverns for strategic attack/evasion/breathers.
- Include as many plausible means of movement/transport as possible, such as ground vehicles, elevators and stairways inside and outside of buildings (staircases, fire escapes, ladders, climbable objects), zip lines, helicopters, traversable roof tops, catwalks/walkways, hang gliders, parachutes, etc. all of which add great variety to attack/evade gameplay.
- Be completely destructible: objects and terrain. No more fake "pre-fab" animations that ignite by shooting it (yes, Battlefield 3, I'm talking about you) or "only these objects are destructible" fail. Everything takes damage and acts as it would in real life.
- Strike a balance between claustrophobic Call of Duty rats-running-around-in-circles maps, and ridiculously massive maps (ARMA 3; though it is a simulation) that take half an hour of flying to get to, only to get killed and start all over again. Land and air vehicles throughout the map can help with this.
- Incorporate as many real life elements as possible that add unique variety to gameplay, such as attack dogs (introduced in Call of Duty: Black Ops).

Artificial intelligence (AI): Getting better, but still lacking any serious consideration. For examples, take a look at any [Watch Dogs AI glitches](#).

Best gaming elements: Unfortunately, no single game has ever combined the best gaming elements that have proven themselves in the past (and why is that?!), so here's my list of the best elements I've personally encountered over the years:

Realistic character movement (most fluid and balanced): Battlefield 3
Detailed character movement (but slow and very clumsy): ARMA 3

Realistic water effects (beach): Crysis
Realistic water effects (water surface): Grand Theft Auto 5/Watch Dogs
Realistic water effects (underwater): Crysis/ARMA 3

Original elements: Attack dogs: Call of Duty: Black Ops
Sound effect of bullets piercing clothing: Goldeneye (N64)
Grappling hook: Quake
Hang glider: Far Cry
On-the-fly weapon modification: Crysis
Detail of environment: Grand Theft Auto 5/ARMA 3

Realistic sound effects: Grenade launcher ricochet: Quake
Overall weapons: Battlefield 3
Grenade exploding in water: Crysis

Ease of vehicle movement: Crysis Wars

Destructible environment: Battlefield 3/Watch Dogs (though both are still only using outdated pre-fab destructibility)

Overall graphics: Watch Dogs/GTA 5 (PC/PS4/XBox One)

Overall attention to detail: Grand Theft Auto 5

Related links: [Playing Games For All the Wrong Reasons](#) | [Gaming Criticism](#) | [The Zombie Game I'm Still Looking For](#)

[Comments?](#)

