Game Breaking Tutorial (First Person Shooters)

Written by Judgy

Introduction

Welcome to my introduction of how to break games in the "First Person Shooter" (FPS) genre, this tutorial will cover the basics of how to break games giving information on; what to look for, what to try and what to expect, these will vary from game to game as different games have different mechanics, however I will try to cover pretty much everything as I go along.

As I stated previously many FPS's are different and have tried to renovate the genre by borrowing elements from other genres, such as the inventory and item equipping once only seen in RPG's has now made it way into the FPS genre and usually its in these genre overlaps where the game can be broken.

The mechanics of the game can be abused in numerous ways whether it's manipulating items in a menu, using items in ways not intended by the developer, using regular movements in unusual ways, the list goes on, it could even be that you are simply using everything normally but you find something the developer missed.

Additional

This is not the be-all-and-end-all guide to game breaking and it will be updated with clarifications, ideas and methods when suggested and all credit will be given to those who purvey that information, as will credit for any tutorial videos provided.

Terminology

Term	Meaning	Notes
Auto-Scroller	A section of a game designed so that you have to spend a certain amount of time being moved from one location to another. This is normally used in sections where you travel inside a vehicle which is not in your own control.	
Checkpoint	A marker in the game world which usually corresponds to an objective or event, games will use these to mark progress, load upcoming areas or activate a cutscene / dialogue.	
Clip	To pass through a solid object in the game world.	
Crash / Hardlock	When an error in the games communication with your computers or consoles hardware makes the application close.	This has many causes chief of them being when values in the memory are pushed under or over their allotted memory

		allocation
Cut-scene	A video or in game animation where the player loses control for the duration.	
FPS	abb. "First Person Shooter".	
Invisible Walls	Invisible walls are placed to prevent a player accessing areas of the game, these are usually found "outside" to preserve the idea of the world around the actual playable area.	
OoB / Out of bounds	Any area outside of the intended area of play usually acessed by clipping or scaling invisible walls.	
Save Quit / Save & Quit	This means to save the game, quit, then re-enter the game, this is normally used to respawn something, skip dialogue or events or to return a player to a save point closer to the next objective.	
Sequence Break	To achieve an objective or event out of the expected order.	For example instead of tasks **[1, 2, 3, 4, 5, 6]** in consecutive order you could do [**1, 2, 5, 6]** with **3** being **4** ignored completely. Alternatively you may be required to do all objectives, however, due to travel between them it makes more sense to do them in a different order to what the game makes you.
Skip	To cut out a portion of the game through varying methods.	This includes most of the terms described in this section **Sequence breaks**, **Save Quits**, **Out of bounds**, etc
Softlock	When the game is in a state of error which makes it un-completable but does not "**crash**"	An example of this would be if the game incorrectly moved you to a place where you could not get out of. [Borderlands 2 Softlock]

Glitch - A Glitch is an abused error in the games coding which allows the player to gain some kind of advantage they would not normally be able to have. These were not intended by the developer, however, **they are within the shipped version of the game and therefore are allowed to be used**.

Cheat - A cheat is a code or unlocked effect in the game which the developers placed intentionally to allow the player to have an upper-hand on the AI or to make the game more fun, **these are not allowed to be used**.
Note! : These are not allowed to be used during a run, however, for the purposes of exploration and testing they can be very valuable, with effects such as God-mode, No-clip, Fly (also known as Ghost) which takes away the

risk of death or injury allowing you to concentrate on looking for glitches or skips. [Thanks to Warepire for the tip]

Hack - A hack is an interference of a games memory by means of a 3rd party program. where glitches use objects within the game itself to cause intended effects, hacks allow brute force overwriting of game data by editing specific memory locations manually, **This is not allowed to be used**.

Basic Requirements

This is a simple list of things you need to do before you even start trying to break a game because you will have much more success in doing so once you have done these things;

Play the game;

You might be thinking "well of course I have" but it is important that you know the game, there is no point in trying to find a glitch or skip when you don't know what that glitch / skip has achieved. How do you know your skip is useful if you don't know where the next checkpoint is or what the game was supposed to give you?.

Example: Wolfenstein [2009] Using a method developed by myself and IvyVisual you can skip the entire first mission where you get an amulet giving you access to skills, however, by skipping it you cannot complete the game because the game does not just grant you that skill upon exiting the mission even though it would 'normally' be impossible to complete the mission without it.
[*Credit to IvyVisual for the video footage*]

Also there are situations where KNOWING where to go is required more than LOOKING where to go.

Example: Wolfenstein: The Old Blood There is a sequence break which skips being transported by a Tramcar by using the escape hatch in its roof early, normally the game would play a Cut-scene then you would be in water. Instead because you're in a different location when you used the roof hatch it teleports you into water in the sky, there is nothing around you other than sky and only by knowing what direction you need to swim in the regular circumstance are you able to swim to the next checkpoint and make the game normal again.

Knowledge about WHERE the game progresses (Know your route);

One of the main details you need to understand is where you need to go next. Once you know this you can improve upon it in given situations and question the need to do tasks "as intended" by the developer this could be simple environment skips or avoiding Cut-scenes.

Example: Borderlands 2: You can skip most of Bloodshot Stronghold by going out of bounds and cutting out 80% of the actual map, this was achieved by knowing that the next checkpoint was at a given location and that upon using the skip we had to backtrack a small amount to trigger the continuation of the story.

Counter-Example: Call of Juarez: Gunslinger: You cannot skip any checkpoints in this game because each checkpoint makes the next checkpoint become available as well as auto save locations along the route, therefore the game basically "softlocks" until you go through the required checkpoint.

Knowledge about HOW the game progresses;

I touched on this in the examples above and it's important to know;

- Does the game progress on a checkpoint by checkpoint basis?
- Does the game load everything at the start?
- Do certain objectives need to be done or can some be skipped?
- Are interactable objects already usable?
- Will the mission / story complete or continue if you skip doing objectives?

Once you have knowledge of how the game progresses you might find yourself completely changing how you routed your game previously.

Example: Bioshock 2: Some time back "Blood_Thunder" found a nice OoB trick in Bioshock 2 that would take the player out of bounds at the start of a segment and drop them back in at the very end, skipping everything along the way. Unluckily when you drop back in the item that you use to end that section of the game isn't active, and returning to the previous checkpoint to try activate it doesn't work either.
[It would therefore seem that this part of the game is a checkpoint by checkpoint activation sequence]

Knowledge of WHAT is in the game;

- What do you have at your disposal?
- Can you akimbo weapons?
- Can you use skills?
- Can you drive vehicles?
- Can you gain height from rockets or grenades?
- Can you hide behind cover?
- Can you gain experience points
  and so on... .

Knowledge of what you can and cannot do in a game is fundamental to the "Glitch hunting" or "Skip hunting" you intend on doing.

Take ideas from other games;

This is a pretty big one, a lot of games have tricks which pass over onto other games particularly if two games share the same game engine or are of a similar (or of the same) genre. If two games have an inventory system and one of those games has an item duplication glitch try it on your game and see if anything happens. Sometimes its worth the time to try what we already know and then work from that if nothing comes of it.

Also watching a variety of different speedruns of games from the genre you are working on can often help there are many situations which have similar problems situations which people have overcome in huge amounts of inventive ways, it could be that the solution you are looking for has already been found by someone else in another game entirely.

Environmental Glitches / Skips

Navigation of the environment efficiently is paramount in your effort to complete the game as quickly as possible and doing so can involve various techniques which can vary from very basic to (sometimes) bewilderingly complex. If you are trying to break how the game works the most fundamental method is to exploit the environment to create skips.

CAUTION!!!

When doing this you need to be careful and consider:

- Will creating a skip cause you to miss something that you NEED to have later in the run?
- Will you have to re-route a later section or later sections of the game to incorporate the new skip?

Running through the section using the skip and then without the skip should help clarify any differences between the two, this will mainly affect games where you have an EXP bar or you only have 1 chance to get a particular item or weapon.

Example: Borderlands 2: The Bloodshot Skip causes you to skip the entire map pretty much, however this now means that you have lost a huge chunk of EXP which is required to be the correct level by a certain point, therefore Re-routing to include some farming of high EXP mobs was added in a way that the time gained from the skip was not lost from the time taken to kill different enemies. (which happen to be on-route anyway)

Example: Wolfenstein (2009): In the first mission you are required to get a medallion which allows you to use supernatural powers throughout the game, Myself and IvyVisual found a way to skip the entire first mission which skips getting the medallion which in-turn stops you from being able to use skills you REQUIRE to complete the rest of the game.

Out of Bounds;

Out of Bounds is the term for reaching an area in which you are not supposed to be able to enter in normal playing conditions, the benefit to doing this is that usually you can find shortcuts by cutting across areas you would normally have to navigate around.

The main method of doing this is to either Clip through or to climb over Invisible walls or to pass through fake walls which are only visually present and do not have any collision properties. Clipping through objects can be achieved by doing various combinations of moves to force your hitbox through an object or as simple as standing in the way of something coming at you which will not kill you but will push you through.

Example 1: Borderlands 2 (No collision)
Example 2: Borderlands 2 (Over invisible walls)
Example 3: Wolfenstein: The Old Blood (Over invisible walls + pushed by props)
Example 4: Wolfenstein: The Old Blood (Repeated movements)

Out of Bounds QuickSave/Load;

This is useable in some games which allow you to Quick save and Quick load anywhere in the games environment by doing this when pressed up against an object you can be forced through the object when you reload because collision detection (in some games) does not happen immediately allowing you to clip into objects for a few frames which is sometimes enough to get through them.

Example: Fallout 3: This is used a lot during the speedrun of Fallout 3, by pressing your character against a surface and quick saving you can force yourself through by reloading the save while attempting to move forward.
[*Credit to Ryderc33 for the video footage*]

Water and Out of Bounds;

For the most part if you have a level which has water in it the water will extend out of the confines of the map this means that if you clip outside of the maps by some means instead of falling infinitely or getting stuck you will fall into water instead at which point you can swim and re-enter the map in another location.

Example: Wolfenstein: The Old Blood: at the end of the first 'Real' mission you have to do a long FPS auto-scroller section on a boat which takes you to where the mission ends. By finding a clip prior to this boat ride I found that you can swim to the end of the level and active the trigger by ‘using’ a button.

Grenade boosting or Rocket boosting;

This is a method of gaining additional height on a jump by using your Grenade or Rocket to push you upwards, this is more common in games with a SCI-FI or less "realistic" game-play elements, whereas tactical games which base their mechanicals around the real world this is less likely because in most cases a direct grenade will kill you on the spot.
It’s all about timing normally but if you can get a boost off of an explosive it’s normally a great idea to use this to explore places the developers didn’t think you could get to which are usually high off the ground.

Example: Borderlands 2 - Grenade jumps: We have a very simple method for doing grenade jumps which is to remove all grenade mods, look directly down count to 3 seconds then jump as the grenade explodes this allows us to do a wide range of skips from 3-4 second time savers to 4-5 minute time savers that skip an entire map.

Example: Quake - Rocket Jumps: simple technique, aim away from the direction you want to go then shoot and jump. Quake is also an example of a game where you can easily time the jump by switching to another weapon, shooting a given number of times and knowing how long to wait after taking the last shot.

Grenade boosting with props;

This is a method of gaining height by placing a dynamic object (such as a barrel or box on top of a explosive which would normally kill you or give you no boost, by using the dynamic object the explosive causes it to move and if that object has collision with your character your character moves also.

Example: Deus Ex- Human Revolution: By placing boxes or barrels over explosives then standing on them the explosion will propel the barrel / box and also your character upwards where normally no height would be achieved by the standard explosion without the barrel / box.
[Credit to Apjjm for the video footage]

Use-Through;

This is a method of skipping certain events or avoiding certain routes by using something which is already usable much earlier than it should to be. This can be a number of things such as buttons, levers, NPC's, terminals, Etc. Most have specific method to do so but the premise of the idea is the same by getting close enough to an object regardless of obstacles in direct Line-of-Sight you can still use that item earlier than intended and by doing so open up areas of the game much earlier or advance the storyline much earlier. Sometimes very thorough testing is required to find an angle that allows for this.

Example: Wolfenstein: The Old Blood: there is a Cut-scene which plays out and then triggers a fight in a large room which then has you enter a small control room and press a button to open a second door and exit the level. However, If you avoid the Cut-scene and head straight to the final door (which is still locked) you can slide into it and use the button which moves your character inside the room through the door and you can exit the level saving around 3:30 - 4 minutes.
[*Credit to Blood_Thunder for the video footage*]

Explosive-AoE;

This is a technique which abused the fact that generally in Video games explosives have Area-of-Effect damage (Also known as Splash damage) which you can abuse to trigger events that normally require line of sight to happen earlier for example on older generations of games you might be asked to destroy something on the other side of a wall in some games it's completely possible to use an explosive and damage that object through the wall itself.

This is generally dependent upon the way the game is programmed if it is programmed well it will know the wall is solid and to not allow the blast to pass through, however if it is not programmed as well it might just rely upon an equations which says something similar to:

```
(var Blastradius: if object < 20 units and > 10 units then damage = 50%
                  if object < 10 units and > 5 units then damage = 75%
                  if object < 5 units then damage = 100%
                  else damage = 0%)
[This is pseudo-code]
```

This is also depended upon what you are trying to blast through for example you cannot blast through the wall but that wall also has a locked door, what happens if you use the explosives on the door instead of against the wall will that blast now pass through?.

This is made even more useful by grenades due to the fact that for the most part they bounce around after being thrown (excluding halo style sticky grenades) which makes the line of sight issue even less trivial because you can basically aim round corners by bouncing grenades off of walls or other props.

Example: Wolfenstein:The New Order: Blood_Thunder invented a method to skip a chunk of one of the missions which abuses the fact that you can destroy the brakes on an elevator from a large distance away using a grenade thrown directly up which hits all of the brakes, all of which are out of you characters line of sight.
[*Credit to Blood_Thunder for the video footage*]

Prop Stacking / Prop Moving / Prop Flight;

This method uses moveable object in the games environment to climb over or force you character through objects such as walls or doors. By arranging items in the correct way you can climb over many obstacles which are placed in a way to basically make you "go-around" usually these only have hitboxes equal to their height.
(unless they are used to mark the boundaries of playable space in which case they tend to be much, much taller)

Movable objects often also have different properties such as: some when being moved have no collision with your character yet upon being set down again regain collision, others have collision while being carried and can be used to push you through objects or can be used to make you fly on top of them.

There are many examples of each of these kinds of tricks such as:

- Halflife 2 / Fallout 3 [Item flight]
- Deus Ex:HR / Penumbra: Overture / Skyrim [Item Clipping]
- Bioshock 2 / Return to Castle Wolfenstein / Wolfenstein [2009] [Prop Stacking / Moving]

[*Credit goes to SourceRuns, Error1, TheApjjm, TheKotti, gr3yscale, Pabiloz, TheLDer and IvyVisual for all video footage*]

Props which cause different effects;

This doesn't have a definable method because it can because by various things OR! people just cannot figure out why it happens, but the basis for this category is that objects in the environment can cause your character to be effected in an unexpected but desired way. This could be cause by climbing on objects, or climbing over them, getting beneath objects or just using objects in obscure ways.

Example: In Wolfenstein: The Old Blood: I found out that by climbing onto a barrel that is floating on the surface of water the game loses track of your character's hitbox and collision properties basically throwing your in-game X-Y-Z coordinates slightly off of your actual X-Y-Z coordinates which then makes it possible (by means of continually spamming Crouch and Jump) to clip through solid walls.

When this first happened I had to reverse engineer what I had done to figure out the exact method behind it thus realising that I had used the barrel in a certain way to cause this to happen, Which is why it's to either record your testing (or stream it) or have a high level of recall of what you have been doing. Frequent saving also helps including preserving the save file after reaching the unusual state in case you can’t immediately work out how to reproduce the effect.

[If I stream I always have my Keyboard inputs displayed onscreen so that I can see what buttons I pressed when it happened so I can attempt to do it again later]

Item and Menu Glitches

Items can also be used to glitch games in various ways whether it's by transferring weapon specific attributes to other weapons, combining akimbo (dual wield) weapons into odd pairs or simply abusing the mechanics of normal items to be used more efficiently.

The main questions to ask yourself are:

- Does my game have item slots, Pre-defined item allocations, an inventory system, Primary/Sidearm only.
- What happens when I swap and do something else. (eg. open a menu, use cover, throw a grenade)
- What happens if certain weapons have stats linked to them. (Free shots, Added damage, Bullet effects, etc)
- What happens if I use an item then swap it out while it is in use.
- Can I override long animations by doing other shorter animations.
- Can I keep good items that I shouldn't be able to keep. (and what are those items)

Weapon Slots;

In some games weapons have various slots that they can be put into for a number of reasons chief of them being character personalization so that you can have the weapons you want in the order you want when you switch between them. This can lead to various tricks or glitches which can be abused in various ways.

Merging Weapons;

In Borderlands 2 I invented Weapon merging by noticing that when a weapon is swapped to another equipped weapon and the menu is opened during that swap and the menu closed again your original weapon is brought back out, after noticing this I tried the swap again but when I opened the menu again I swapped the original weapon (in my hand) with another weapon which when I exited the menu caused BOTH weapons to be in my hand.

Note: The original weapon was a Vladof Rocket launcher with the added statistic of - fires a free shot every 3rd shot (basically) and by having it on the free shot and merging it the way I did I passed the infinite ammo onto the second weapon but this can also pass a series of effect from weapon to weapon such as health steal, critical damage%+, etc.

Drop Reloading;

This is a basic method which abuses the fact that you are allowed to drop weapons again and when you pick them up again they have a full clip because the game doesn't track the ammo in the clip of the weapon just the total ammo for that gun type.

Item Duplication;

There are many methods to doing this however I will explain the version I'm most familiar with. In Borderlands 2 you can start a duel with another player for an item you both wanted you put the said item into the window then the winner takes the prize however, If the player with the item drops it on the ground and then loses the game still transfers a copy of the weapon to them with the original still on the ground which can then be picked up and used.

Keep what you shouldn't;

This might sound strange but it's basically a storage glitch by which you keep an item that you would normally drop upon switching to another item or weapon.

Example: Wolfenstein: The Old Blood: you can keep a two handed Minigun which you would normally drop upon equipping another weapon by cancelling the pick up animation, by overriding it with other animations the method is as follows: Aim at the Mingun, swing your melee weapon and pick up the Minigun, Immediately sprint away. what happens here is that the pickup animation is taken over by the two other animations.

The melee attack animation prevents the pickup animation from happening immediately however it is too short for the game to not play the pickup animation so the sprinting animation after overtakes the remaining of the time required to play out the pick up animation and the game doesn't bother trying to play the animation again after that.

In Deus Ex, you get placed in a prison cell and lose all your items automatically. However, there’s a little delay before the latter takes place. During this time you can drop every item you wish to keep and they get preserved.

Other Tricks / Information

There are a few things you will need to know about certain mechanics which can affect your ability to find glitches I will state a few here and explain why they are interesting to know also how they can be exploited.

Dialogue Skips;

Many lengthy dialogues can be skipped by either using items which activate a dialogue, NPC's which do the same thing , by interrupting them with cutscenes, also some games also allow you to skip dialogues by saving and quitting because your progress is saved as "person spoken to / objective completed" and not "dialogue listened to"

Skipping dialogue can also have various effects in games, some merely skip dialogue to save time, however, some events in games are programmed into the dialogue itself therefore if you can skip a dialogue which triggers a door closing ahead of you, you can save major amounts of time.

Example: Borderlands 2: The "Blitz Skip" I found allows you to enter Scooters workshop in Sanctuary faster by skipping Angels dialogue when you run to the main gate of Sanctuary after placing in the power core some lengthy dialogue is playing, by timing it right you can run away from the gate and trigger another dialogue about a Hyperion moonshot blitz this dialogue has more than 1 part to it and this causes dialogues to swap from "blitz" to "story" each cancelling each other out ending each dialogue much sooner.

Save Glitches / Techniques;

Sometimes when you save the game everything gets saved, however, sometimes not everything gets tracked by the save for example movement speed, movement direction etc etc may not be stored what will be stored is your current X,Y,Z coordinates your equipment and you character stats (health and armour etc) this can be used to abuse falling, to clip through objects or in some cases just allow you to warp back to a prefered location.

Example: Red Faction: By jumping into a forcefield and quick saving / loading your character can pass through it while taking a certain amount of damage, normally the game would prevent you from passing through (as you would expect) [Credit to Duane Jones for the video footage]

Example: Thief 2: in this game you can manipulate NPC behaviour by saving/loading, e.g. making them open a door for you.

Crouch Jumping;

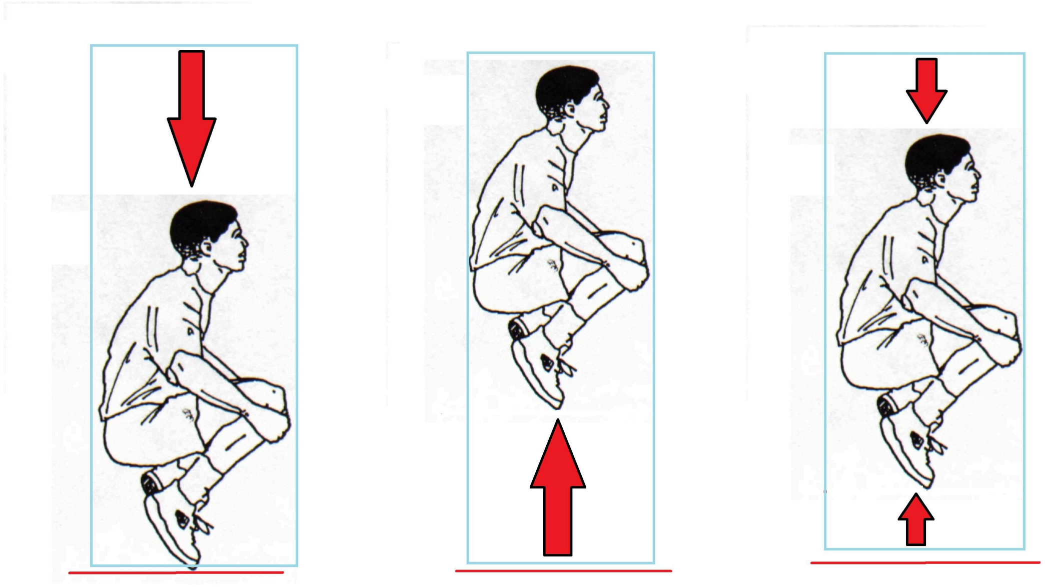
Many games have jumps which can only be done by crouch jumping so its important to know a few things about how jumping works the major point about crouch jumping is the way your hitbox contracts when you do it.

Top Shortening: This is where when you jump and crouch the shortening of your hitbox contracts only from the top making your character gain no height from doing so.

Bottom Shortening: This is where when you jump and crouch the shortening of your hitbox contracts from the bottom only making your character gain the highest jump.

Crunching: This is where your character (when crouch-jumping) shortens from both the top and the bottom gaining a small amount of height compared to a standard jump.

Examples:
[Blue box is regular hitbox, Red line is standard jump clearance WITHOUT a crouch, Arrow is contraction direction.]



Ladders absorb fall damage / death;

This is basically what it says, that generally ladders if jumped onto regardless of the distance you are above them when you jump will absorb your momentum and you can fall from a huge height with little or no damage skipping elevators or skipping taking the long way down.

Example: HalfLife: There is a long elevator sequence which can be skipped by simply falling down the shaft then landing on the ladder at the very bottom which absorbs the momentum and prevents fall damage / death,
[*Credit to SpiderWaffle for the video footage*]

Water absorbs fall damage / death;

Same idea as above, if you need to drop down from somewhere high quickly and there is water then jumping into it is a safe way to get down with little or no damage. (99% of the time)

Using Interactables prevents fall damage / death;

In some games interactable objects will have their own animation when you use them and when they do they will usually move your character to a specific location when that animation happens therefore if you are falling from a high place and you can use an interactable object on the way down the game will begin the animation and cancel your downward momentum.

Example: Call of Juarez: Gunslinger: By using a ladder on the way down a fall that would normally kill me is absorbed by the character animation of grabbing the ladder because my momentum shifts from downward to forwards.

Event triggers and cutscenes can prevent fall damage;

This is similar to the above technique of using interactable objects however, instead you use an event object such as the start of a cutscene or playing an audio dialogue to prevent the damage from happening. usually damage will more likely be prevented if the game takes control of your character during the cutscene or if that cutscene is the end-of-level trigger.

Saving and loading can prevent fall damage;

This method of damage prevention relies upon game saving not tracking your acceleration or starting your acceleration at 0 from reloading a checkpoint this means that you can jump from a height that would normally kill you or damage you in a game quicksave before hitting the ground then quick load that save and your character will drop safely to the ground.

Entering vehicles can skip triggers;

Sometimes in games if you set up a vehicle in the right way when you enter it you can skip over event triggers because your hitbox instantly moves from one position to another regardless of the animations movements

Example: Half-Life 2: You can skip a long section in the 'Water Hazard' section of the game by placing the boat in the correct place (the other side of a trigger) then entering it, by doing this you skip the trigger which would normally close the gate ahead of you.
[*Credit to the Source Runs team for the video footage*]

Exiting vehicles can force you through props;

This technique is based on the fact that most vehicles in FPSs have an exit animation to add a realistic exit rather than just having your character pop out onto the ground around the vehicle, these kind of animations can be used to force your character through objects in the game world such as thin walls, doors, terrain, the standard method for doing this is to place your vehicle in a manner that when you exit it the animation causes you to cross a line that is otherwise un-passable.

Examples of this can be found in a number of games but the premise is the same find out at which angle you exit the vehicle then try placing that direction against an object, if nothing happens try doing this while moving. For the most part games will prevent you from doing this kind of thing by having limits about the distance you are to an object while still being able to exit the vehicle, movement tends to override this limitation because while your character is exiting the vehicle it can still be moving towards the object you want to pass through.