



## Game Design Essentials: 20 Unusual Control Schemes

By John Harris

[Continuing Gamasutra's 'Game Design Essentials' series, which has also included '[20 Difficult Games](#)' and '[20 Open World Games](#)', this fascinating installment examines unconventional control schemes, from *Robotron* through *Crazy Climber* to the *Wiimote* and *Guitar Hero*, with detailed design lessons for each concept.]

Stop for a moment and consider what it is that makes a game control well. It is not as easy a question to answer as you may think.

There is a theory that the controls of a video game should do their best to get out of the player's way. The interaction between the player's mind and the game world should be as simple as possible. The perfect controller, to this thinking, is something that would read minds and eliminate all possibility of confusion, other than that in the player's head. There would be no controls to learn, no buttons to press, and no fumbling with control pads. Input would be completely mental, and output would be a holodeck.

In the absence of such technology, controls should be standardized so a player can move from one game to another easily. They seek to develop a shared control language that applies across games: left stick moves, right controls camera, the major action button shoots, a secondary one jumps, shoulder buttons flip between weapons -- that kind of thing.

Every control style mentioned in this article speaks against this theory. Some present their own standards in its place, due to their being well-suited to their style of game; the dual joystick (shooting) style has been used in a few games itself, from the old classic *Robotron: 2084* to *Geometry Wars*. Others really have no chance of ever becoming a standardized control scheme, but are okay with it. After *Donkey Kong Jungle Beat*, why would anyone else even care to make a platformer controlled entirely with the pressings of two buttons? But then, why would someone have cared to make it to begin with?

For some of these games, special hardware is needed to measure player performance in greater detail, so as to translate it into game terms. In the *Golden Tee* games, rolling the trackball further and faster makes for a stronger shot. Other games use special controllers to accentuate the game experience. *Guitar Hero* naturally does this to help the player feel like a rock star, and the bicycle-powered arcade game *Propcycle* is perhaps the closest we will ever come to experiencing human-powered flight.

The purpose of this article is not to describe the games themselves, unless it is important to do so to explain what makes its control scheme interesting, or if the style is mostly relevant to only one game. Because it's concerned with games that purposely do things in a non-traditional manner, there's an unusually large representation from arcade developer Atari Games on this list.

Finally, again, although the list is numbered, this should not be taken as a list of the "most" unusual control styles ever seen in games. Attempting to make such lists will always make some folk unhappy. I know I'm never happy with them. These games are given as examples. Some are obvious, and some push the limits of the theme. That is the point. The idea is to give you ideas for your own project, and to show some notable successes, and maybe failures too.

## Unusual Interpretations for Standard Controls

*Most of these games are older titles. Although there are a few more recent games that use a strange control method for the sake of challenging the player, these days they tend to be niche titles like *Donkey Kong: King of Swing*, which only uses shoulder buttons.*

### 1. Dual Joystick (Shooting)

Representative games: *Robotron: 2084*, *Inferno*, *Smash TV*, *Total Carnage* (Williams, arcade), *Space Dungeon* (Taito)



America, arcade), *Front Line* (Taito, arcade), *Geometry Wars* and sequels (Bizarre Creations, Xbox 360, DS and Wii)

### Control description:

There are two joysticks on the control panel. The left stick moves the player around, and the right one determines the direction of fire. That means there is no fire *button*; the act of pushing the right stick signifies intent to shoot, if the game doesn't actually have automatic fire.

### Adaptability:

Moderate to high. Most controllers these days have two analog sticks. Best of all, mind, would be two digital sticks, especially for *Robotron*.



### The scheme in use:

In creating the dual joystick control style, Eugene Jarvis solved a problem with 2D run-and-gun shooting games that has since become so ingrained into the genre that it seems strange to think it was fixed so early. Since *Robotron: 2084*, there's been *Commando*, *Ikari Warriors*, *Gauntlet*, *Alien Syndrome*, *Zombies Ate My Neighbors*, *Gain Ground*, *Mercs*, and *Gauntlet Legends*, along with scores more. All of these games support player facing only because of the need to specify what direction shots will go, so in all of them, in order to shoot in a direction the player must first move in that direction, unless (like *Gauntlet*), the player can't shoot while moving at all.

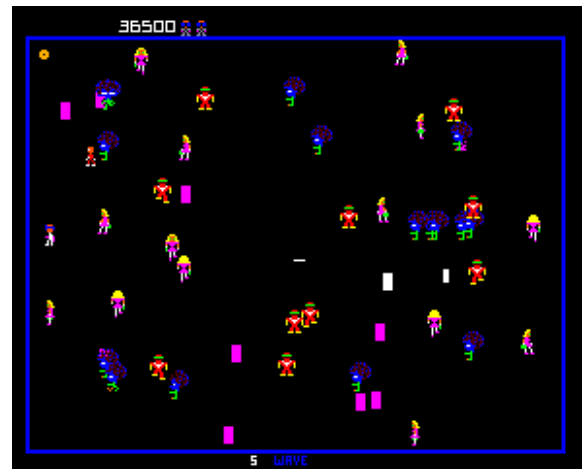
This might seem at first to be a control style for ambidextrous people, and there is a certain head pat/tummy rub aspect to it at first, but it's not hard to get the hang of. Most importantly, it makes the act of shooting at enemies while running from them much easier. All of these games feature hordes of monsters following the player constantly. In a game with the more common scheme, where the player must move the joystick to change facing, fire, then resume running, shooting behind always slows the player down.

### Design lesson:

It is going against the grain of current design trends to recognize that all player ability is ultimately granted. It is not always true that the player should be able to do everything he'd expect he

could in real life. For more modern games this must usually be explained, but for older, more abstract games this is not as important.

This said, I know of no game on the list of button-shooters above that gains much from forcing the player to jerk the joystick backwards to fire at approaching opponents, and some (especially *Commando*) would be greatly improved with *Robotron* controls.



## 2. Joystick and Dial

Representative game: *Aztarac* (Centuri, arcade)

### Control description:

The control panel contains a dial *and* an analog joystick. Moving the joystick controls position. A trigger on the joystick controls firing, a little unintuitively, since it's the dial that aims the gun in any direction.



### Adaptability:

Poor. Attempts to map the dial to an analog stick could work, but would turn the control scheme into *Robotron* redux. It's a little more difficult to aim with a dial, but it could potentially allow for far more precise shooting than a joystick, digital *or* analog, could provide.

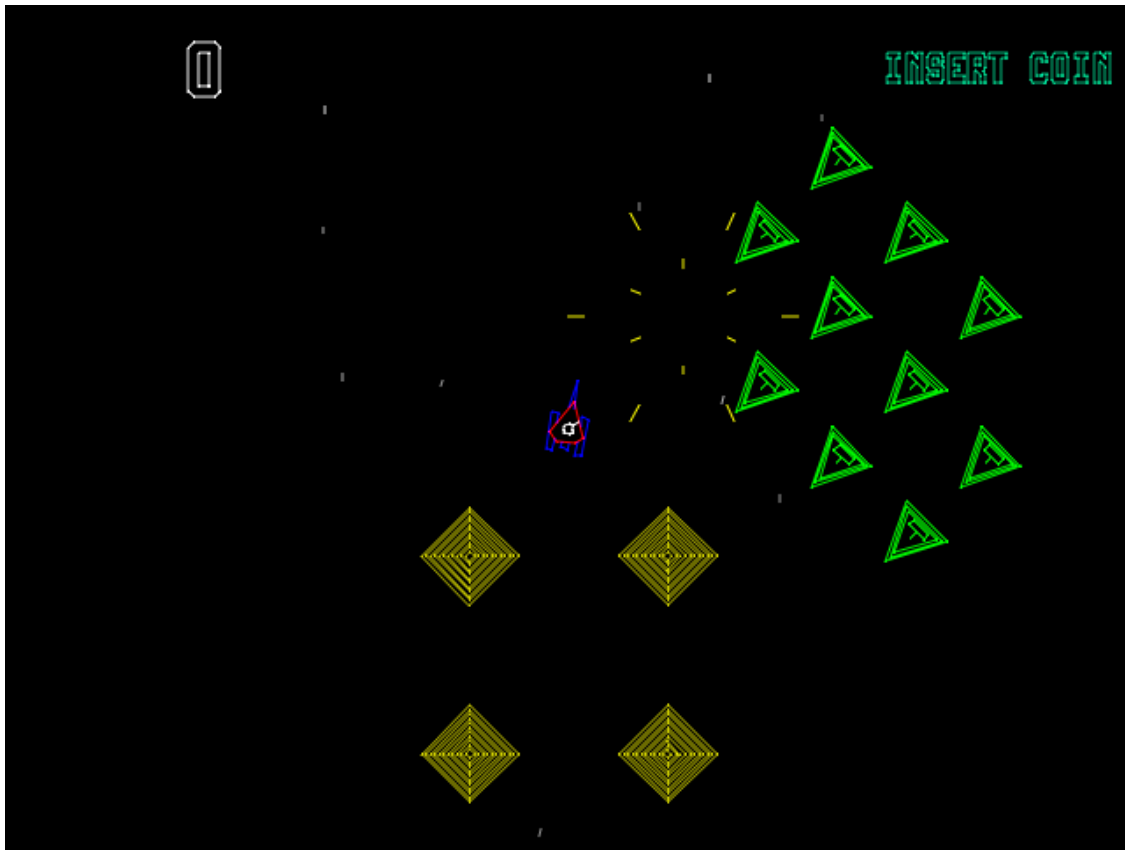
### The scheme in use:

This one's a bit obscure. *Aztarac* is a very rare Centuri vector arcade game that Wikipedia tells us saw production in that fateful year, 1983. It's got extra-sharp graphics for a vector game, and is a

tremendous challenge to play. KLOV states that only 500 machines were made, making playing *Aztarac* as its designer, the late Tim Stryker, intended nearly impossible for most people.

The player's spaceship can fly freely in eight directions with the joystick, and can aim in many more directions using a dial to control aiming. The object of the game is to roam around a huge area of space looking for enemy ships to destroy before they can fly in close enough to destroy a star base, destroying it. Contact is all that's required to

destroy a base, and after the first couple of levels enemy ships begin flying in very rapidly and in ever-wider formations.



Although the player has infinite ships, losing all four bases means the end of the game, and before too long it becomes very challenging to get even one base through the attacks. After three or four levels, enemy formations become so large that they won't entirely fit on the screen at once, forcing the player to both fly around to find the ships and use the dial to shoot them when they show up.

Unlike *Robotron*, which is more-or-less constant shooting, *Aztarac* has long periods of silence while the player either flies out to meet the aliens or waits for them to arrive. When found, the player must take care of large groups of them very rapidly. The act of "hosing them down" is fairly well-suited by the dial aiming control. One thing *Aztarac* -- and for that matter *Geometry Wars*, which uses an analog stick to handle aiming -- have above *Robotron* is that the player can shoot freely in all directions.

#### Design lesson:

*Aztarac* does have a couple of problems with its control. In addition to a stick and a dial there are two buttons: the fire button, which would be impossible to use effectively if it wasn't on the joystick, and a radar button. The game's pace is fast enough later on that taking a hand off the dial and pressing the radar button can be fairly disruptive, especially since it doesn't seem to pause the game while it's held.

---

### 3. Dial Movement

Representative games: *Pong*, *Breakout*, *Warlords*, *Tempest* (Atari Games, arcade), many Atari 2600 games (Atari), *Arkanoid*, *Cameltry* (Taito)

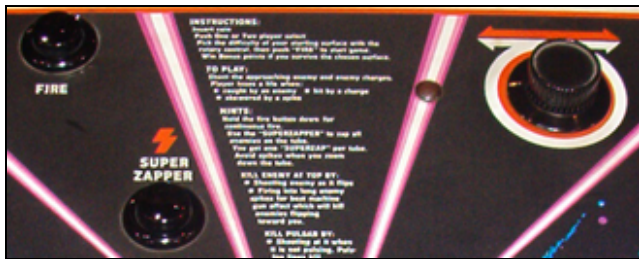
#### Control description:

There's a dial that directly controls player position along a one-dimensional range of movement. The Atari 2600 actually had two controls of this type, the ubiquitous "Paddle" which had a hard stop at the end of its range of movement, and the "Driving" controller, which had no stop and was analogous to a one-dimensional trackball.

#### Adaptability:

Low to Moderate. There are few things sadder in gamedom than a *Breakout*-type game controlled with a joystick. So important is the control to these games that not only did Taito release a special dial controller for the NES so that people could play *Arkanoid* as the arcade developers intended, they're doing the same thing for the upcoming DS version of the game. While *Cameltry* makes due with its button-controlled SNES port, you're not really playing *Tempest*

if you don't have dial control.



### The scheme in use:

Using a rotary control provides the player with the ability to instantly switch his position to any spot along a continuum of possibilities. For games of the *Breakout* style this is essential. *Breakout* is a game of reaction with a component of prediction, but what happens when the ball is sent flying with a horizontal component greater than the player's movement speed? The developer must either increase player speed (making it more difficult to control in normal play) or decrease maximum ball speed (making the game much less difficult). Paddle controls get around the problem easily.

Two of the games in the list are not *Breakout*-style games, and they are both important examples. *Cameltry* focuses on a ball with realistic physics (unlike the *Breakout* ball, which has only simplistic physics) and gives the player the ability to rotate the maze around the ball. The maze is much larger than the ball and new parts must scroll into view, so it is helps to be able to instantly adjust it to allow the ball to fall past obstacles. On the other hand, the SNES adaption of the game (*On The Ball* in Western markets) uses level design that takes the digital control into consideration, with sharp turns that seem perfectly matched to a full swing left or right.

And then we have *Tempest*, one of the greatest of all twitch games. The dial in that game moves the player's ship around the edge of a (usually) continuous web, and the dial control in the arcade version potentially gives the player nearly instant access to any part of it. This is useful because the player wants to keep the Flipper enemies away from the outside edge of the playfield. But the game also requires precision control in order to destroy Flippers that have made it out, by firing on them as they flip into the player's lane. *Tempest* also features enemies that fire up the web at the player, so the dial must be used to dodge shots too. *Tempest* consistently ranks near the top of classic arcade popularity lists, yet compared to other Atari classics it has seen relatively few home ports (Jeff Minter's efforts notwithstanding) because of its special control needs.

### Design lesson:

The trade-off, seen in trying to control NES *Arkanoid* with a control pad, is between long-distance speed and short-distance precision. Among the designers of the dozens of *Breakout* clones created since the 70's, the mark of the clever mind -- as opposed to that who just rolls in special blocks and powerups without thinking -- is in whether situations occur where the ball is impossible to return. Without a paddle it can be difficult to guarantee this.

It is worth noting that it may be possible, depending on the correlation between dial movement and its effects on the screen, for the player's position on-screen to "skip" if he moves the paddle rapidly enough to cause it to move further than the player's width during a single screen refresh. If the game depends on the player not being able to move through barriers, the developer must make sure that he doesn't just flip by them in the middle of a frantic twirl.

## 4. Complicated Control Panels

Representative games: *Asteroids*, *Asteroids Deluxe* (Atari), *Defender*, *Stargate* (Williams)

**Control description:**

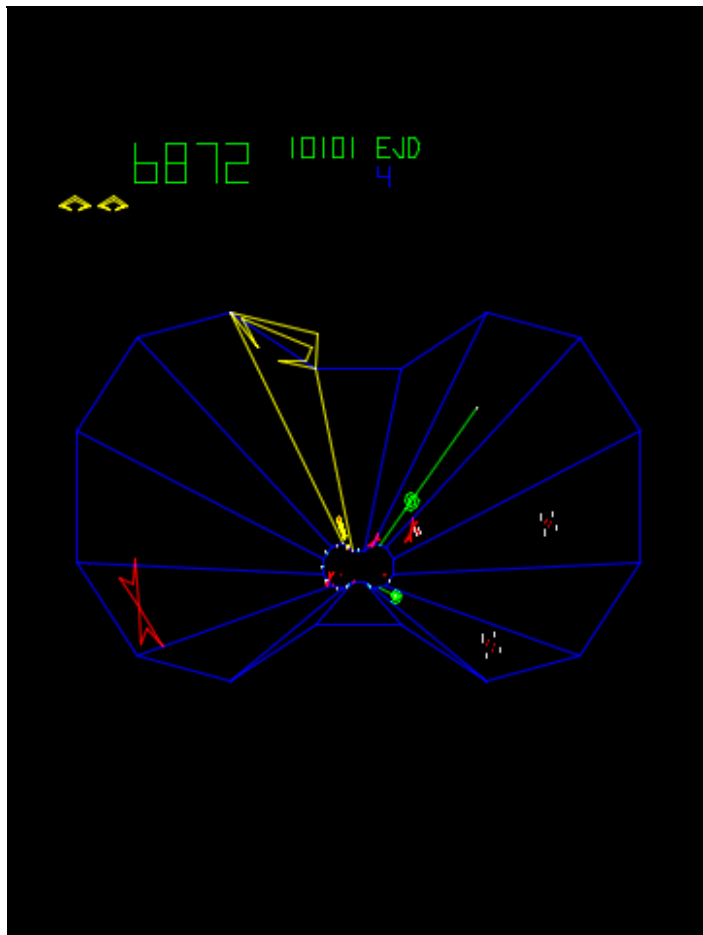
Joysticks existed for video game use when *Asteroids* made it out, but for some reason the game was controlled entirely using buttons marked Rotate Left, Rotate Right, Thrust, Fire and Hyperspace. *Defender* and *Stargate* still hold the record for most controls included in a popular non-fighting game; the former's control panel has a lever and five buttons, and the sequel added yet another button to the controls.

**Adaptability:**

Pretty good, although every known home version of *Asteroids* substitutes Joystick Left and Joystick Right for the rotation buttons. Many also replace Thrust with Joystick Up, which given the potential ease of accidental thrust should be considered a mis-feature. Every consumer version of *Defender* and *Stargate* from the Atari 2600 on to the *Midway Arcade Treasures* emulations of the PS2 generation simplifies their controls, arguably improving them.

**The scheme in use:**

These games often suffer from the complexity of their controls when encountered by current players. In the old days it'd usually take players a few games to get used to *Defender's* formidable control setup, where today most players don't have the patience for that. As the proponents of the Wii would suggest, there's something to be said for not giving the player too many buttons to keep track of at once.



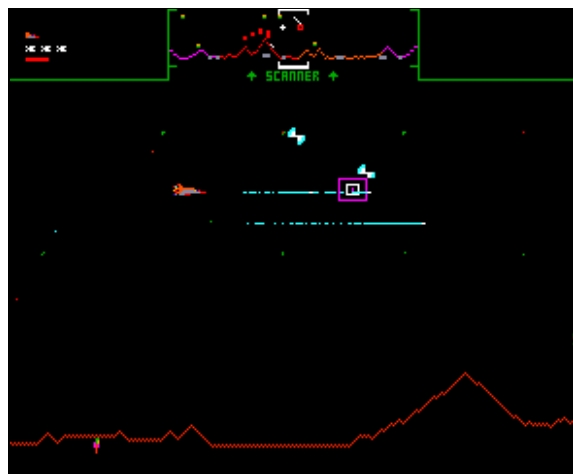
And yet... stop for a moment and consider, is *Defender*, with its vertical lever and buttons for Fire, Thrust, Reverse, Smart Bomb and Hyperspace, really all that complex by our standards? The current most popular game system in the world, the PlayStation 2, has ten buttons, a control pad, and two analog sticks on its controller -- and the sticks can themselves be pressed

to provide two extra buttons. It helps greatly that the Dual Shock is held in the player's hands and not spread out over a flat control panel, but neither has it the luxury of printed instructions lying next to each button.

Back in the days of the Atari 2600 and Intellivision, if a game required the use of a keypad it would nearly always include an overlay that could be slotted into place over the pad to remind the player of button functions. The modern replacement for the overlay, I'm sorry to say, is lengthy obnoxious tutorial sequences that present each control's a small number of times then hopes the player will remember.

**Design lesson:**

As Thoreau urged us to do, *simplify, simplify*. There is something telling about the nagging, suspicious similarity between playing video games and doing work, but that doesn't mean we have the right to force the player to earn an associate's degree in Controlling The Damn Game just to play.



**5. Dual Joystick (movement), a.k.a. Tank Controls**

Representative games: *Battlezone*, *Vindicators* (Atari Games), *Katamari Damacy* (Namco)

**Control description:**

Two levers control treads/legs independently. That is, press up on the left lever or down on the right one to turn to the right, and vice versa. Press up on both to move forward, or down to move backwards. Pushing the levers in opposite directions turns quickly in place.

### Adaptability:

*Katamari Damacy* itself is a console adaptation of this scheme. The PS2's analog sticks work nicely as the necessary levers, and they also provide a bit of extra side-to-side control that Atari's tank games lack.



### The scheme in use:

This style is used precisely because it *doesn't* allow for perfect control. Instead of ease, this scheme gets used for the feeling the player gets when using it. Tanks are heavy equipment with an unusual style of propulsion: independently controlled treads. Back in *Battlezone's* day this provided for an early form of "experience game." (In addition, the game used a periscope-like viewer to the player had to look into to see the action.) *Vindicators*, on the other hand, used an overhead view, but the tank controls lend the player's small vehicle a sense of weight it would otherwise lack.

*Katamari Damacy's* use of the style is more whimsical, but uses it for the same purpose as *Vindicators*, to lend weight to the player's movements. Since the player can ultimately manipulate great balls of stuff over 800 meters in diameter, anything that supports that feeling of weight is useful. The controls also help to remind the player that he's not controlling the ball itself, but a tiny agent who is manipulating it with his hands... an easy thing to forget when the ball towers over mountains but the Prince is still 5 centimeters tall.

### Design lesson:

What lesson? There's little that can be offered through using two sticks for movement that one won't do. Game design is ultimately about figuring out what the player can and cannot do, so nothing has really changed.

But from the point of making the game into an experience as well as a game, driving a tank is an activity that few people will ever get to try, so tank controls are novel and lend interest to the game. For more mundane tasks this might not be such a good idea; no one wants to play a game in which he must manipulate a soldier's legs independently step by step. The unusual operation also makes it a good match for *Katamari Damacy*, a game that relies heavily on the absurdity of its setting.



## 6. Discrete Button Thrust & Rotation

Representative game: *Toobin'* (Atari Games)

### Control description:

A ring of buttons allowing the player to Paddle Left or Right, each Forward or Backward. Similar somewhat to Tank Controls, pressing one button provides rotation as well as thrust, so two buttons must be pressed in concert with each other to provide consistent forward motion. A final button fires projectiles.

### Adaptability:

Moderate. Controllers that use dual joysticks can be configured, in *Midway Arcade Treasures'* emulation of *Toobin'*, to create makeshift tank controls, but most players will probably opt to use the same configuration menu to create one-to-one joystick control instead. Since the controls are a major part of the game it could be argued that this isn't really *Toobin'*, but neither is playing it with two joysticks, really.



### The scheme in use:

Another product of Atari Games' '90s experimentation with control styles, *Toobin'* in the arcade had a unique control style that only used buttons. In the game the player (and maybe a friend, with his own set of buttons on the other side of the control panel) controlled a guy sitting in an innertube that floats down a variety of rivers. The buttons were arranged in a sort of circular pattern, four of them corresponding to each of two feet and two hands that dipped into the water.

Pressing the hand buttons propelled the 'tube forward, and pressing the foot buttons sent it backward. Each button press also rotated the innertube a bit from that limb's side, so creating consistent thrust required hitting the buttons in pairs, either both hands or both feet at once, or alternating in rapid succession. The result was a kind of discrete, impulse-based version of tank controls. The innertube could rotate all the way around and be driven backwards if the user wanted, which sometimes made it easier to aim projectiles at

enemies chasing from behind.

### Design lesson:

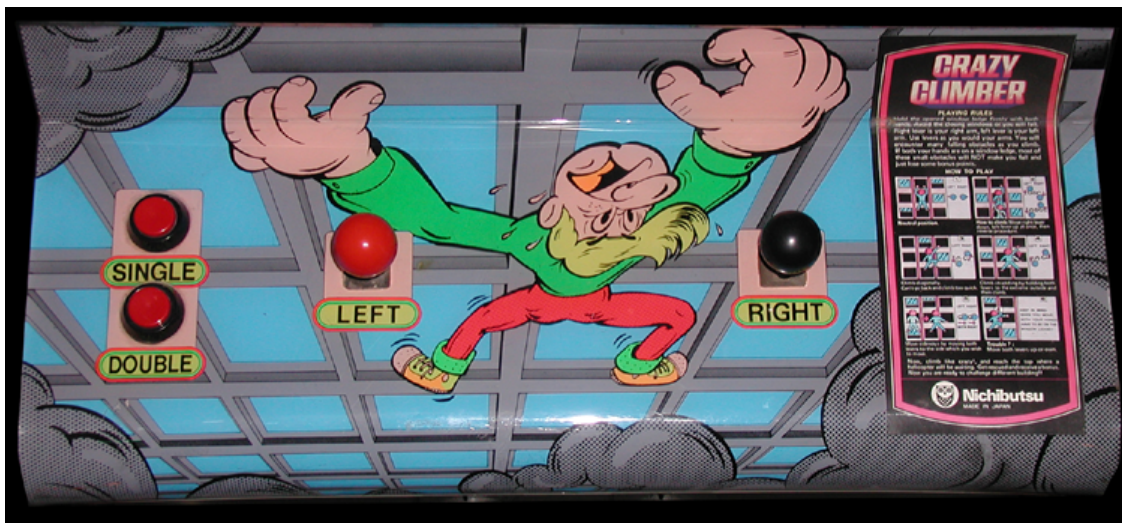
This is a prime example of a game that's purposely made more difficult by its controls. If the player had perfect control over his innertube, the game would be too easy; there really aren't that many enemies on-screen at once, and pinpoint movement would make it much easier to "swish" gates for high scores. Its press-event focused control could be seen as a predecessor of the more recent game *Donkey Kong: Jungle Beat* (see below).

## 7. Dual Joystick (climbing)

Representative game: *Crazy Climber* (Nihon Bussan)

### Control description:

Another dual joystick system, but unlike *Robotron's*, this one really only saw use in this one game. The player's character is trying to scale a tall building. The left joystick controls the placement of his left hand, and the right joystick controls the right hand. Once the player has both joysticks so that they have a solid grip, holding both left or right shift the climber horizontally across the building, or alternating them up/down then down/up causes him to rapidly ascend the building. It's a subtle system that takes some getting used to, but is oddly satisfying once mastered.



### Adaptability:

As with *Robotron*, very good, but *Crazy Climber* has seen little in the way of ports, possibly because of the relative obscurity of its publisher. They exist though. One port is for the Super Famicom, which presumably uses the controller's four buttons as a makeshift second control pad.

### The scheme in use:

The building is a regular grid of open windows, and the player must place his hands skillfully to make upward progress. The game ensures that the player cannot fall off the building just by moving the joysticks; a hand will not move if it's the only one resting on an open windowsill. This sometimes causes frustration, since it may seem like the joysticks don't work if the player moves them without care, but it's for the best.





All over the building, windows are constantly closing and opening. A window that closes on a hand doesn't cause it to visibly move, but it *does* break its grip. Sometimes smiling lunatics appear in open windows and drop things on the player, or other objects may appear from above. If one hits the player he doesn't, typically, die immediately, but one of his hands gets shifted from its grip, and if that unseats the only hand with a grip the climber takes a fatal plunge. If the player is hit once, but can regain his grip before another object hits him, he can continue up the building without penalty. As the game continues larger objects fall from above, which can potentially knock off both hands, and if the player dallies too long in one spot the game will start to close windows around his location to force him to move on.

#### Design lesson:

This is another scheme where the fun comes from learning and mastering a unique set of controls. To proceed up the building the player must shift his grip constantly, but this puts him in danger of getting killed if he is struck at

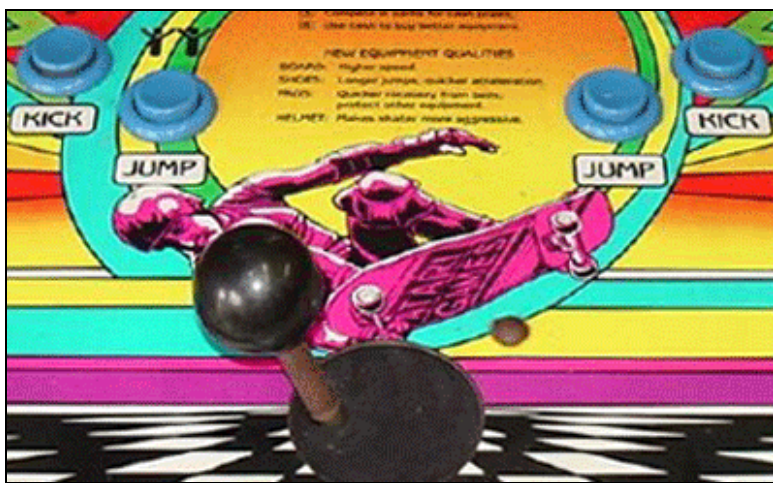
the moment he shifts hands.

### 8. Non-centering Joystick

Representative game: *720 Degrees* (Atari Games)

#### Control description:

A standard digital joystick with one important change: it cannot be centered, effectively creating a simple rotary controller.



#### Adaptability:

Pretty good, considering. *720 Degrees* was included in the *Midway Arcade Treasures* compilation, and while the game is unusually difficult to play with a control pad due to the fact that the player must often twirl his board rapidly (imagine how badly your thumb would be banged up by playing a fighting game with a control pad where the entire game is *one gigantic special move*), playing it with an analog stick makes for a reasonable recreation of the arcade control.

#### The scheme in use:

A non-centering joystick may be considered just broken hardware by some, but in fact it's a brilliant way to turn the physicality of manipulating the controls into an important aspect of game play.

At its core, *720 Degrees* controls like a kind of skate park *Asteroids*. The player can rotate left and right, and the "kick" button is similar to Thrust. But instead of buttons controlling rotation, the game uses its joystick to directly indicate facing. This means that the player's turn speed is limited by the speed by which he can twirl the stick, and it also means that turning in mid-air, in order to perform tricks, carries a hand-smashing physical aspect.

If the player jumps he can spin the joystick to do tricks in the air, but he must return to a proper direction before landing to avoid collapsing and earn points. For the most part, this is all a trick is in *720 Degrees*; there are some other cases, but spins are very important. And the game's unique structure means that performing tricks isn't just way to earn rewards, they are needed to survive. The chain of effect is long but simple: mastering spinning the joystick lets the player earn points quickly; earning points earns the player tickets; earning tickets lets the player enter skate parks; and entering parks *resets the overworld timer*.

If the player is in the overworld hub for too long without entering a park, the game's infamous bees arrive and, before long, end the credit. The only consistent way to earn points, thus tickets, thus additional time, is to perform tricks. In this way, mastering the controls becomes the entire point of the game; the game is exactly as hard as how hard it is to perform tricks.



### Design lesson:

The unusual game structure, which forces the player to constantly be earning tickets --sixteen of them in a full game -- forces the player to adapt to its controls to be successful. The player cannot just muddle through and do well at the game, and since this is an arcade game, it is all about mastery anyway. *720 Degrees* is a game that's received far more than the average number of console ports, yet most of these versions are subtly different from the arcade game because spinning a control pad is not the same as spinning a joystick. The arcade had a large, arcade joystick; home versions have used control pads, digital joysticks and analog thumbsticks. The raw speed with which the player can manipulate the stick directly determines how difficult the game is, so in a sense, it isn't playing the real game if the player isn't doing it on an actual machine.

### Schemes Intended to Test Physical Skill

*The prior control schemes mix up the game's interpretations for player action either to make the game easier or to lend it a kind of realism. The following games, on the other hand, are much more reliant on the human player's physical skills, such as asking him to spin a trackball as fast as he can, press buttons rapidly, or skillfully manipulate a motion-sensing wand. Because of the sensitivity needed to detect a wide range of performance, they all necessarily use analog inputs.*

## 9. Trackball Sports

Representative games: *Golden Tee* series (Incredible Technologies), various other arcade sports games

### Control description:

These are arcade machines that add an aspect of physicality to sports games by forcing the user to give a big ol' spin on a trackball to supply the power to the shot. It makes the most sense with golf, where to some degree hand and arm strength plays a role with the shot as it does with the real game. In the *Golden Tee* games, the force used by the player to spin the ball translates directly into the power of the shot.

### Adaptability:

Poor. Unlike *Marble Madness*, analog sticks do not lend themselves nearly as well to this type of game, meaning we'll probably never see a good home version of one of the greater recent arcade success stories. As to why this is, read on...



### The scheme in use:

Trackball sports games have been around for some time in arcades. Tactical sports like football and basketball tend to use joysticks because of their familiarity to the user, and to allow the spotlight to shine on the athletic abilities of particular simulated team members instead of that of the player. A joystick is a tool for indicating intent to move, and movement is generally key to team sports. Also, joysticks are well-suited because the acts of motion, passing, attempting to make scoring shots, and so on, are done under time pressure, and are usually opposed by AI or other human players. Because the player cannot usually take a long time to act, and because there are plenty of real-time, dynamic pressures on the success chance of the act, simple binary controls, like button presses or joystick movement, are suitable. Signifying intent to act is sufficient, since the precise split-second timing of the button press, compared to the state of the rest of the game, is enough to determine success or failure.

Although it's not unheard-of, team sport games usually don't use trackballs. Bowling and golf are the two most popular subjects for trackball control. They're both slow-paced games where the state of the game doesn't change when players don't make moves. More importantly, because subtleties of motion matter for so much in the real sports, the timing of a binary signal doesn't provide enough variation for the challenge of making a good shot or throw to be significant unless special minigames, usually only related to the physical act of playing in an abstract way, are used. Tellingly, so many console golf and bowling games use "power bars" to determine the fate of each shot or throw that they are expected by players.

This isn't to say that spinning a trackball itself is greatly similar to swinging a golf club, but these acts are more similar to each other than pressing a button when a fluctuating power bar reaches a given point. Both actions translate subtle differences in the movement of each player's muscles so that they have an influence upon the game, and both may be improved through physical practice -- a long but rewarding road.



### Design lesson:

When designing a simulation of a sport like golf, one question to ask yourself is: *Why can't the player always shoot the best possible shot?* This is ultimately why these games mess around with power bars and trackballs, for the games are supposed to be difficult and tricky to control with precision, but still precise enough that the occasional player can shoot a hole-in-one. Even so, power bar games often don't supply enough resolution to the bar to allow shots to be truly varied, a situation remedied, to some degree, by the power bar/motion wand hybrid control in the golf module from *Wii Sports* (below).

### 10. Trackball One-to-One Motion

Representative games: *Centipede*, *Marble Madness*, *Crystal Castles* (Atari Games)

### Control description:

The player uses a trackball to move his protagonist through the game world, and it moves roughly where the player rolls it. When he spins the ball slow, it moves slowly. When he spins the ball fast, it moves quickly.

**Adaptability:**

Not so good, and not for lack of trying. There have been many home adaptations of *Marble Madness*, from the Commodore 64 to PlayStation 2, but since the days of the Atari 5200 trackballs have fallen out of favor as console peripherals. Ever since, we've had to make due with joysticks, either analog or digital. It's just not the same, and usually, it's a great deal easier.

**The scheme in use:**

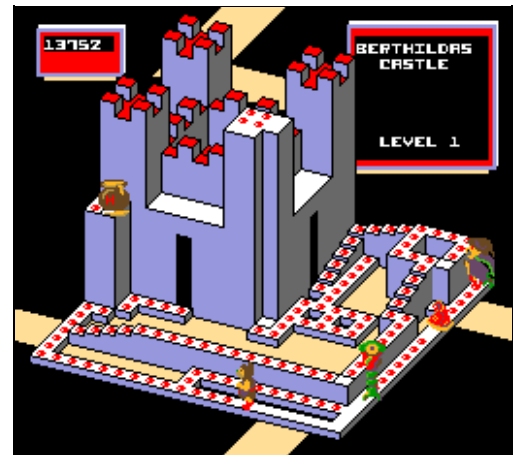
Notice that I lumped *Marble Madness* and *Crystal Castles* together here when in fact their controls are subtly different.

*Marble Madness*'s control has gone on to have the greater influence on game development. In it, the motion of the trackball affects the ball's momentum. When the trackball is rolled down, the game doesn't apply that motion directly to the marble's position, but to its velocity. This means that the player's marble doesn't respond directly to the trackball's movements, but is instead delayed by the physics underlying the game, lending the marble a sense of weight.

*Crystal Castles*' trackball has a more direct effect on the player's motion. This lends Bentley Bear his trademark skittery movement, and pushes the game in the direction of mastering movement more than outsmarting enemies who, to compensate, have simple AI.

**Design lesson:**

One characteristic of nearly all trackball games is that it's very difficult to precisely duplicate one's performance from a prior game, even if the game is otherwise devoid of random elements. The trackball's movement is chaotic enough that it, itself, is a randomizing influence upon the game.

**11. Button Pounding**

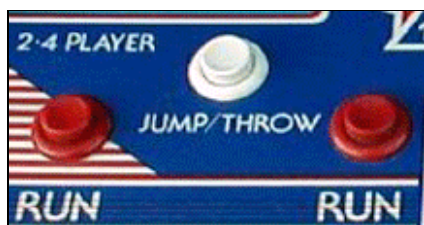
Representative game: *Track & Field* (Konami, arcade)

**Control description:**

*Track and Field*'s control scheme consists buttons, and only buttons. All six of the events in the game consist of building up a power meter with repeated alternating button presses. Some events have a little more to them than that, but in all of them, if you can't hit the buttons fast enough, you're doomed before you start.

**Adaptability:**

Pretty good. The NES port of the game featured new events, but they all stuck with button-operated play, although one, Skeit Shooting, treated the control pad as a button.



### The scheme in use:

*Track & Field*, more than any other game, measures the player's raw muscle-twitch capacity. It is another game where the player's physical limitations also limit his game persona.

Given the straight-forward nature of the play, it's not too surprising that there's very little strategy. What few decisions there are to make have obvious answers. Player technique doesn't involve how to dodge missiles or shoot bullets effectively, but how to arrange one's hands on the controls in order to press them at the

highest rate possible.

The difficulty of the game revolves around the qualifying scores needed to make it to the next level, and many players won't ever be able to hit the buttons fast enough to pass later loops of the game, let alone unseat one of the "World Record" scores. This makes *Track & Field* a rather elitist game when one thinks about it; even with extensive practice, it contains barriers that some players just won't be able to surmount.

### Design lesson:

*Track & Field's* joint-destroying play would be tiresome today. Later sequels and ports tended to de-emphasize events that required raw, button-smashing speed, yet it should be noted that many minigame collections have events that take inspiration from *T&F's* simple sports.

## 12. Motion Wand

Representative game: *Wii Sports* (Nintendo)

### Control description:

Five different games are included in *Wii Sports*, simulating major actions from their respective sports using a motion-sensing wand. Each game duplicates the original motion with varying degrees of fidelity, ranging from surprisingly accurate (Bowling and Tennis) to only vaguely relevant (Boxing). The individual schemes are discussed below.

### Adaptability:

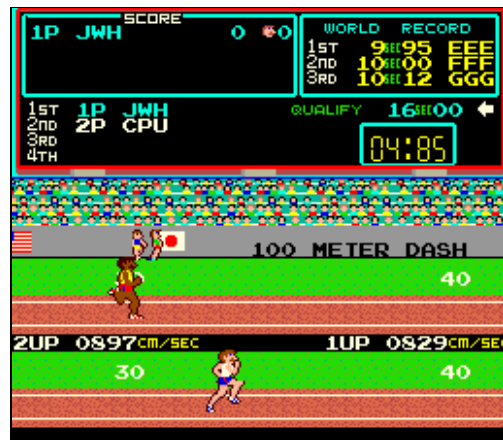
It's already a console game, but the game would be very difficult to port to other machines due to how tightly it's tied to the idiosyncrasies of the Wii Remote. If the Wii's successor offers backwards compatibility, and if it improves the quality of the motion sensors at all, the developers will have to take care that they provide a compatibility mode, or perform lots of testing, for games like *Wii Sports* to remain playable.

### The scheme in use:

*Wii Sports'* five modules all play in tremendously varied ways, and that's not even including the training games included, some of which are even more entertaining than the "full" sports. They are all similar in that timing tends to be a bit more important than the technique in the gestures, and that none of them use the pointer functionality of the remote. Here is a quick description of each:

**Bowling:** The best-realized sport on the disk. Moving the controller like a bowling ball may not be exactly like real bowling, but it's close enough for most people, and it's not too much easier to bowl a perfect game. The game even infuriates sometimes with the degree that unintentional wrist twists can put spin on the ball -- and that *is* like real bowling.

**Tennis:** An interesting demonstration of how to use the motion control to "cheat" at making a sport seem realistic. In the Tennis module, the player runs automatically, taking out a big part of the difficulty in real tennis. The direction the ball is hit is almost entirely determined by the timing with which the ball is hit, but the height and speed of the ball are actually determined by how the controller is moved, which is interesting. And while the game may cheat on its end (as it does, to a degree, with all the sports on the disk), the speed of the game makes it difficult for the player to cheat much himself.





*Golf:* The most interesting thing here is that, while the game does utilize motion control to make the shots, it also relates the strength of the shot to a power bar, and the player must be careful not to exceed maximum power of the shot hooks or slices. Overall it works fairly well in all areas except putting. While the remote's motion sensing is actually capable of being quite precise (as the *Wii Monkey Ball* game illustrates), the way it's implemented here makes putting a frustrating experience.



*Baseball:* The magic formula required for getting pitches to work perfectly every time is elusive, and hitting the ball with the bat seems, again, to rely on timing. Most of the strategy of the game, player movement, etc. has been stripped out, making this the simplest sport on the disk.

*Boxing:* Perhaps the developers were over-reaching a bit with this one, since the remote seems to only really detect forward motion of the remote when it's pointed at the screen. Punches are detected by the rotation of the remote, and are less tiring to perform that way, a significant difference considering how much punching must be done.

Common between all the sports is a mixed emphasis on timing and technique, the moment at which the controller is swung playing as much a role as how it is moved. To control where the ball goes in Tennis, for instance, swinging a split-second early will send the ball "in", towards the swing's direction, and swinging late will send the ball the other way. However, the speed of the swing and how far it "dips" seem to determine the strength of the shot, as does the ball's height on the court.

### Design lesson:

In the individual sports, Bowling and Golf, the controller is basically a more-obfuscated version of the trackball from *Golden Tee*, its movements looked at just hard enough to make the perfect shot challenging to pull off more than once in a while. Even expert players of those two games have difficulty bowling many strikes consecutively, or getting Hole-In-Ones. The other sports make the concept of a "perfect" swing of the remote less relevant due to the situation and actions of the competitor, and thus rely much more on timing in their play, which is easier for the player to get exactly right. These games thus can afford to turn into battles between the players, with the best, most consistent timing winning out.

## 13. Exercise Bike

Representative game: *Propcycle* (Namco, System 22 arcade hardware)

### Control description:

The game's hardware includes nothing less than an exercise bike as its control method. When the player pedals, it provides thrust for an in-game, winged bike, soaring through a 3D world. Steering left and right with the handlebars, *Paperboy*-like, steers the bike similarly, but the player can also pull back on the bars to climb, or push forward to dive.

### Adaptability:

Extremely poor. *Propcycle* is pointless without the bike, and no one has ever successfully marketed an exercise cycle controller for a home game system.

### The scheme in use:

This game is our representative from the large-scale "ride" game trend of arcades of the

'90s. For a while games like Namco's *Alpine Racer* and Sega's *Top Skater* were common fixtures in arcades, huge machines with projection screens and large, seated (or standing) control devices that translated player arm or leg motion into game equivalents. Many of those games would fit in on this list.

This game made it while *Paperboy* did not because there's already plenty of other Atari Games productions on this list. They were constantly throwing down innovative control schemes. Yet, there's also a sense of *Paperboy*'s handlebars being something of a gimmick, while *Propcycle*'s full exercise cycle is... well, it's a *really big* gimmick.

But there's a bit more to it than that, in fact, a big part of *Propcycle*'s coolness is that it's actually physically taxing to play. Almost all games translate the player's agility, dexterity, coordination and intelligence into game-world equivalents. But few games translate player *physical endurance* into the game. An Advanced game of *Propcycle* goes through four levels of increasing difficulty, and since pedaling directly translates into in-game thrust and the game is timed, players are encouraged to pedal very rapidly. If a player fails a level he can continue to try it again with more time, but unfortunately no cash sacrifice or collectable health pack will restore his physical legs to their former state.



The bike adds more to the game than just this, however. In presenting the sensation of flight to the player it is unusually pure; there are no enemies and few moving obstacles. The flight physics are pretty good, to the extent that pedaling too slowly will cause the player's vehicle to lose power and plummet, a trick that's necessary to take advantage of in a few places. The bike on the cabinet could even move around a bit, dipping forward and back in response to how the player pushed the handlebars, and used a fan to blow air in his face when his in-game bike moved quickly.

### Design lesson:

Many games talk about immersion, but really few games are that capable of achieving it. *Propcycle*'s simulation approach, the nifty gizmos on the cabinet dedicated to providing subtle environment cues simulating flight, and of course that bike itself, all speak of designers who wanted to go as far towards presenting the sensation of flight as hardware, and the facts of arcade game

design, would allow. And really, if you could tool around the skies in a winged bicycle contraption, after a few minutes wouldn't you get damn tired too?

### Interesting Mechanics Exposed Through Unique Hardware

*These games all use special input peripherals. Some do so directly, in that the player interacts with a physical object. Others do so indirectly, by sensing the player's motion. They all do their thing in ways that may seem a bit obtuse if their only purpose was to simulate the action in the game. These are games that would not work in a Matrix-style virtual world. The controls matter more than the action on-screen, and mastering them is the same as mastering the game.*



#### 14. Button Platforming

Representative game: *Donkey Kong: Jungle Beat* (Nintendo, Gamecube)

##### Control description:

They're called "bongos," but ultimately they're just oversized, easy-to-press buttons with rubber overlays. The "clap" sensor detects loudness. The way whole thing works here is that the left and right drums are mapped to the left and right directions. The faster one drum is tapped -- and it's easy to tap them fast, easier than *Track & Field* -- the faster the player runs in that direction, though the speed ceiling is fairly low. Hitting both drums together causes the player to jump. There are a few special cases involving object collection, attacks and wall jumps, but the game is ready to remind the player of what they are early on.

##### Adaptability:

It's already a console game, but requires a special controller to really play it. The only reason the drums couldn't be mapped to shoulder buttons is that most people just can't hit a shoulder button as fast as they hit a drum.

##### The scheme in use:

Someone went and took *Track & Field* and, somehow, made it into a platform game. The player doesn't usually hit the buttons as fast here, but then there's a special event or a boss fight that requires righteous button whaling, and the similarities become all too apparent once more.

The real strategy to *Jungle Beat* lies in earning absurdly huge combo scores, and the way to earn combos is very interesting from a design standpoint. You see, a combo in this game is any move performed in mid-air before touching the ground. If the player performs five moves before the jump ends, then all scores collected from that point to the end of the jump are worth 5x. The thing is, how the heck is the player supposed to perform special moves when he only has two relevant "controls" to choose from, and both of them boil down to button presses? How can the player signify to the game he wants to do them?



The answer is that they're all context sensitive. Some of the contexts the player can put himself into at will, like



running and jumping, while others require a wall or other environmental aid to accomplish. There's a special move available when running, there's one available when in the air, there's two that can be done sliding against a wall, there's two that can occur when sliding up the top of a wall, and there's a couple that are performed in proximity to another move.

The result is that high combo scoring, the ultimate focus of the game, depends upon the player understanding which moves get him from one state to another, what moves can be performed from each state, and how to take advantage of the terrain to facilitate moving between them. The designers did their job in creating the moves, and the levels, so well that some levels can be made into one huge combo.

And it's all made possible with just two buttons, neither of which can be held down. The only word for that is *genius*.

#### Design lesson:

Not a lot of games are produced with the idea that the player's control should be *limited*. Probably, *Donkey Kong Jungle Beat* is able to work so well with such restricted controls because it's a 2D platformer. Once a third dimension is added movement and combat becomes much more complicated, and that's not even to speak of the need to add camera controls. But while the game's execution may at first seem to be irrelevant to the facts of 3D game design, the way each level is tightly constructed for scoring opportunities is extremely instructive.

## 15. Sketch Platforming

Representative game: *Kirby: Canvas Curse* (HAL Laboratory, DS)

#### Control description:

The player uses a pointing device, either a mouse or a stylus, to draw platforms directly on the screen, in real time, that the other objects in game treat as physical. In *Kirby: Canvas Curse*, the platform actually works like a conveyor belt.

#### Adaptability:

Even if a system has a pointing device as standard, it doesn't mean it's easy to draw with. Other than the DS, mice present the best opportunity to use this kind of play, as the [Crayon Physics](#) flash games illustrate, so flash development may be the best outlet for this kind of thing.



#### The scheme in use:

The lines the player draws have interesting properties. The ball in the player's care tends to stick to it a bit, and travels along it in the direction it was drawn. If the player draws a loop not only does the ball roll down it with increased speed, but will often intelligently move through the "up" portion of the loop on the way out, *Sonic*-style. Not bad, and the game rewards the player for doing it with a burst of speed.

Drawing lines to maneuver the ball is one thing, but the lines also affect the progress of enemies. Some of the more frantic moments in the game come from trying to draw lines to help the ball without similarly helping the enemies. As the game continues, the level design subtly changes from situations where short lines need to be drawn over minor obstacles, to places where it's obvious where to draw longer lines to get between platforms, to more freeform, improvisational areas where the player is left to his own devices to go where he needs to be.

The shift between the two styles is significant, because this is ultimately a game about creation. Although the player's lines only hang around for a few seconds each, what the player is doing is quickly creating his own portions of the level, on the fly, in order to keep the ball from falling off the map or hitting too many enemies.

#### Design lesson:

Lately, flash videos of hacked *Super Mario World* levels have been making the blog rounds, in which someone creates a layout of obstacles that results in Mario reaching the goal without any controls pressed, like a virtual Rube Goldberg machine. When you come down to it, the same impulse that led to the creation of those levels was the inspiration for *Kirby: Canvas Curse*. It was the first game to show

people what the DS was really capable of, and even now, remains the most compelling use of that system's powers.

## 16. Foot Pad Grid

Representative games: NES Power Pad games (Nintendo/Bandai, NES), *Dance Dance Revolution* series & imitators (Konami and others, arcade and many consoles)

### Control description:

A mat on the floor (or a more durable equivalent for arcade versions) contains pressure-sensitive spots. Think *large, flat buttons*. Instead of using his hands to press them, the player steps on them according to the demanding expectations of scrolling arrows up on the screen.

### Adaptability:

One might think it'd be extremely low, but the game was popular enough to spawn a cottage industry of dance mats.

### The scheme in use:

Like *Track & Field* (also from Konami), there is little strategy here. The arrows scroll up the screen. The player steps on each at the moment it touches a timing bar. It's like *Parappa the Rapper* without improv. And where the music plays okay no matter what the player does.

*Dance Dance Revolution* gets away with it because of its social element. In a way, the symbols and art on the screen are irrelevant to the real game, which takes place between the player himself and the eyes of onlookers, all of whom are thinking either *wow, that's amazing* or *what an incredible geek*. Fortunately, the agility playing the game well requires speaks against the "geek" interpretation.

It also means that the nebulous thing "performance" is essential to enjoying the game. The days where someone could play *Pac-Man* for hours and attract a crowd are long over; merely manipulating a joystick in public won't attract many admirers no matter how dexterously it's handled.



Dancing on a *Dance Dance Revolution* stage may not really have a great deal to do with actual dancing, but it's close enough, which is more than you can say about the similarities between playing a first-person shooter and getting in a real gunfight. And since harder levels mean more impressive performances, difficulty is also a tremendous part of the game. And if there's one thing the *Dance Dance Revolution* series tends to have in spades, it's difficulty.

### Design lesson:

The lessons are two-fold. First, the game proves that millisecond-sensitive twitch games, presented correctly, can still hit it big in the market. Second, that if a game is hot enough, people will buy all kinds of crazy peripherals in order to play it. (See *Guitar Hero*, below.) Whether the market decides a game is really all that great shakes or not can be humbling and risky; Konami had the advantage of the Japanese arcade culture, which is a bit more more open to blatant innovation, to start its ubiquitous franchise.

## 17. Guitar Controller

Representative games: *Guitar Hero* series (Harmonix, PS2, Xbox 360)

### Control description:

The controller is shaped like a miniature guitar, but the inputs themselves are not too special. Five "fret buttons" on the neck play different "notes" when held while a "strum bar" is moved. Notes are only really played when they match up with the song, however. The controller also contains a "whammy bar", rudimentary tilt sensing and a couple of utility buttons, but the main game involves the frets and strum bar.

### Adaptability:

Surprisingly high given that the game relies on a fairly custom piece of hardware. Harmonix have shown themselves willing to make several different, compatible versions of the controller, and customers have shown themselves willing to buy them.



### The scheme in use:

The game may not have originated in arcades, but it's popular for many of the same reasons as *Dance Dance Revolution*. It might be possible to play both games on a controller, but doing so entirely misses why one would ever *want* to play them. The control scheme is the whole point.

*Guitar Hero* is a little friendlier to solo play than *DDR*, because it doesn't take itself as seriously. People have played air guitar in their homes for decades, while energetic, tiring dance steps are not things people usually put themselves through for solitary fun. And for a game that boils down to the same kind of reaction and timing heavy, Simon Says-based play, Harmonix actually managed to put some strategy into the game.

The strategic aspects of the game involve the obtaining and use of Star Power, which can make hard songs easier and greatly increase scores, and making difficult passages easier using hammer-ons and pull-offs. Fret sequences tend to match their songs in timing, but the actual button presses are designed with an eye towards this kind of strategy, which is, fundamentally, a process of interpreting the song. Who do you suppose would do a better job of that -- your average developer, or those who know and love music with their whole being?

**Design lesson:**

Think about it for a moment. There's an entire genre of music games that are fundamentally no different from the *Guitar Hero* games in all ways but one. Yet those games, for some reason, didn't become multi-console megahits strong enough to inspire sales of specialized hardware. In case I haven't reiterated it enough: it's the controller.

Which is not to say that *Tuba Hero* would do anywhere near as well. There is an aspect of culture mining in the success of *Guitar Hero*, of finding some cool aspect of our world previously unexploited by gaming and monetizing it. But, like, hey... if it works, it works.

**18. Video Stream Analysis**

Representative games: *EyeToy: Play* & others (SCEE, etc., PlayStation 2)

**Control description:**

It's really quite brilliant. A small USB camera sits above the player's TV. By using real-time image processing, the game figures out where the player is standing and uses that information to affect the world of the game. While I have no inside information on how it works, but it *looks* like it works by finding the parts of the screen that are "person" and "background," and by examining which parts of the image changes each frame.

**Adaptability:**

One might think pretty high, seeing as how all the current generation systems have USB ports. On the other hand, both Nintendo and Microsoft have shown themselves to be iffy about overly-complex third party peripherals, and have seceded dominance in the game camera arena to Sony, though MS did dabble with its Vision Camera.

**The scheme in use:**

There are arguments to be made that the chasing of processor and video card power in cutting-edge consoles has hit decreasing returns. It helps graphics quality, but so far not much else. Currently, a disproportionately great portion of system time is taken up by visual effects that, strictly speaking, don't really matter to the play. If the player's model has a hundred polygons or 10,000, it can still have the same attacks, spells, health, walking speed, inventory, and etcetera. Better graphics may improve immersion, but the play is rarely affected by it.

This is, to a certain frame of mind, a compelling argument, and Nintendo has bet the farm on it this time around, hoping to rely on their design strengths to make up the difference. Yet, it could be argued that the failure of increased system strength to greatly

change gameplay since the PlayStation days is a failure of imagination, and has nothing to do with the inherent nature of games at all. But there are some games that seek to responsibly use greater power, and perhaps it's fitting that the leader is the console that Sony's own *EyeToy* game series, which have evolved into the games that use the PlayStation Eye for PS3. These games all use real-time image processing and recognition to determine what is happening in the physical world in front of the camera, and apply that to the virtual world behind the screen.

So far, Sony's uses of game cameras have seemed fairly gimmicky. Using a camera to enable a player to smack around tiny virtual ninjas, or manipulate a virtual guitar, or control a virtual hoverboarder... perhaps these are all things that would be better suited to physical controllers. But this need not be the case in the future. Sony has recently released a collectable card game, *Eye of Judgment*, that utilizes the PlayStation Eye.

What could be possible in the future? Allowing players to use their own art tools to draw characters or maps on a piece of paper and importing them into the game, as [was suggested](#) on the PlayStation Blog? Using enhanced image recognition to map a polygonal character's motions to that of the player? Allowing the players to manipulate hallucinatory objects? I know better than to say the possibilities are endless, but in this direction they are really quite wide. And so far, almost totally unexplored.

**Design lesson:**

All the processor power required to analyze a constantly-changing framebuffer comes at the detriment of the rest of the game, which is probably why most EyeToy games are fairly simplistic. The PlayStation 3's tremendous resources may change this decisively.



## 19. Rotary Sensitive

Representative game: *WarioWare Twisted* (Nintendo/Intelligent Systems, GBA)

**Control description:**

The rotation of the portable console itself is detected, using special inertia-sensing hardware included within the game cartridge, and translated into in-game motion. Usage of a button is optionally supported in some minigames.

**Adaptability:**

Extremely poor. Even the Game Boy Player, a special Gamecube attachment that allows for playing GBA games on a TV screen by the inclusion of the full system chipset, disallows playing this game -- unless you want to lift and spin your Gamecube. It simply requires a portable system to play.



### The scheme in use:

Since the original game, Nintendo and Intelligent Systems have wasted little effort in cashing in on the unexpected popularity of the *WarioWare* series. Each uses controls unique to that version: the original a control pad and one button, the DS installment uses that system's touch screen and microphone, and the Wii version makes use of motion control and pointing functionality. By most measures, this is the best of the lot.



The decision to base a game off of a *rotation* sensor is not an obvious choice. A tilt sensor, maybe... that's just a pair of rotation sensors rotated 90 degrees, after all. But *WarioWare Twisted* has just one, and it makes no use of the control pad or any buttons other than 'A' at all. It's a tribute to the ingenuity of the minigame creators that this, the simplest control style of all the *WarioWare* titles, doesn't get old as fast as the styles in *Smooth Moves*.

How does Intelligent Systems overcome the inevitable approach of ennui for so long? First, by including quite an accurate sensor with the game so that it can distinguish between tiny movements as well as broad ones, and utilizing that range by giving most of the minigames their own distinct movement patterns. Second, by mixing up the traditional *WarioWare* game structure a bit with the inclusion of random, sudden "revolution" rounds that are much faster than even the customary five-second time frame. And third, because the act of spinning the controller around -- sometimes fast enough that one feels like tossing it in the air, sometimes slowly and to the extent that the system is often held upside-down -- is just silly fun.

### Design lesson:

And silly fun, in the end, is what the series is about. A rotation sensor would probably not be enough input to sustain a full game, were the game anything besides *WarioWare*. The controls match the game. One can be strange because the other is too. Abstract, comical games are looked down upon by some in the player community, but they serve an important role: it is a lot easier to get away with casual, spurious, joyful innovation in a game where the player is just messing around for a few minutes than if he's trying to save the world from the evil Demogorgon with +5 swords of slaying and AI pathfinding quick event limit break tension gauge active time alternate costume battle combo systems.



## Uncategorized Cases

### 20. vi-like Movement & "Everything has a key" Commands

Representative games: Roguelikes (independent developers, many systems)

**Control description:**

The "hjkl" keys move on the cardinal directions, the same movement keys as in the venerable Unix text editor vi. The "yubn" keys move on the diagonals. Beyond that, there's about a dozen other standardized keys with assigned functions, then depending on the game another dozen keys that do other things.

**Adaptability:**

Extremely low. Most console adaptations of roguelikes get around it with control pads for movement and inventory-based, context-sensitive action menus for object manipulation, in keeping with RPG conventions.

**The scheme in use:**

The roguelike control method seems to be an anachronism. Many recent roguelike releases, both independent (the norm for roguelikes) and commercial shy away from it, opting for more context-driven menus. And there is something to be said, when it comes to it, for simplicity in control schemes, unless for reasons of realism the game actually needs to be complex. A flight simulator game -- now that can be reasonably expected to be complicated, because flying real planes is complicated. Meanwhile there seems to be something fundamentally odd about playing a fantasy game where you can be like Conan the Barbarian, when Robert Howard's musclemans would have taken an axe to anything resembling a keyboard.

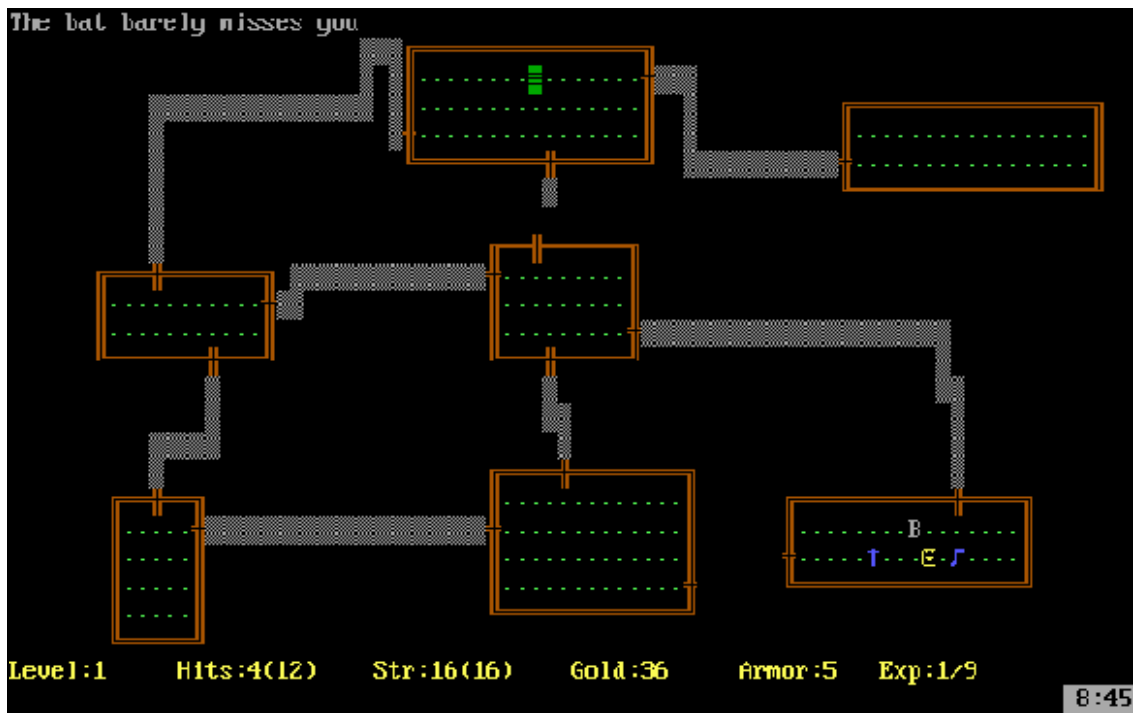
I said the roguelike control system *seems* to be outdated. It should be obvious from my choice of words that I don't think it is. Well, actually, in some ways, it can be overwrought. Control style is the single biggest thing that drives players away from playing roguelikes, more than graphics, more than randomness, more than permanent death -- even more than difficulty.

One might think that there's a bit of elitism in the refusal of these games to adapt to more modern control concepts. Reading the roguelike Usenet groups doesn't help to dismiss this impression. Among some long-time players, there is even a belief that the chance of typographical errors should be part of the game -- and they will cheerfully blame you for your character's death if you hit the wrong key, claiming it's part of the game. Just so we're clear on this: it is an important job of the game designer(s) to make sure controls are easy to understand, difficult to confuse, and appropriate to the game.

But there *are* important advantages granted by the roguelike control style. For example, its similarity to the controls of the venerable Unix text editor vi made the game infinitely easier to pick up back in the days of *Rogue*, when computer games were played on terminals more than anything else, the presence of numeric keypads was not guaranteed, and most players could be expected to know vi. And these controls add an undeniable elegance to the game. In *Nethack*, if you want to perform some action ten times, you actually type the number 10 then the key of the command you want to do, just as in vi!

But the biggest thing that roguelike controls brings does for the genre is that it *institutionalizes obscurity of object function*. Er, sorry. Let me break that down. It makes less obvious the fact that items found in the game have multiple functions. In *Rogue*, the fact that potions can be thrown at point-blank range to affect monsters is important to the game, but this feature is intentionally obscure: it's meant to be a secret to be discovered by attentive players. If the player could bring up an item menu on a potion and see the "throw" option there, it would be a bit of a giveaway.

There are times when Microsoft has been known to make hay over the "discoverability" of their user interface. If the user poked around, so the thinking went, and right-clicked on some icon they wished to do something with, then on the menu that appeared there should be the obvious choices, and maybe some unobvious ones too. In a way, then, the menu itself is documentation. That's fine, yes... unless a feature is supposed to be undocumented. Roguelike games are not as "discoverable" in that they force the player to pick command first then item, and even when the game is asked for a list of relevant items from inventory for the command to apply to, sometimes things are left off the list in purpose.



One of the most complex roguelike games, *Nethack*, takes this idea to extremes. Sometimes it seems like items have more undocumented features than documented ones. It's obvious that the player should be able to drink potions, but throw them? Dip items into them? Mix them? (A)pply a potion of oil and throw it as a bomb? How about breaking wands, eating rings and amulets, writing on scrolls, rubbing lamps, or charging lanterns? And by keeping these options semi-secret, it helps to reduce interface clutter. There are actually few circumstances in the game where the player might want to break a wand, so it's fitting that this feature is not constantly visible.

#### Design lesson:

The trend for managing inventory in game interfaces is object first, action second, with a visible list of objects. Most roguelikes, by contrast, go action first, which helps to keep secret item functions obscure to players who don't know about them, while not restricting their use to players who do. Is throwing a potion actually useful? Could be... why doesn't the player try it?

---

#### Bonus Stage: No Player Interaction

Representative game: *Progress Quest* (Windows)

#### Control description:

The program sits on the system tray and generates random combat messages. After a certain threshold is reached, the character gains a level. He then jumps right in and continues training for the next level. Ad infinitum. The player can call up lists of quests undertaken and completed, monsters killed, and treasure found, if he wants. He can't actually do anything to affect this stuff, though.

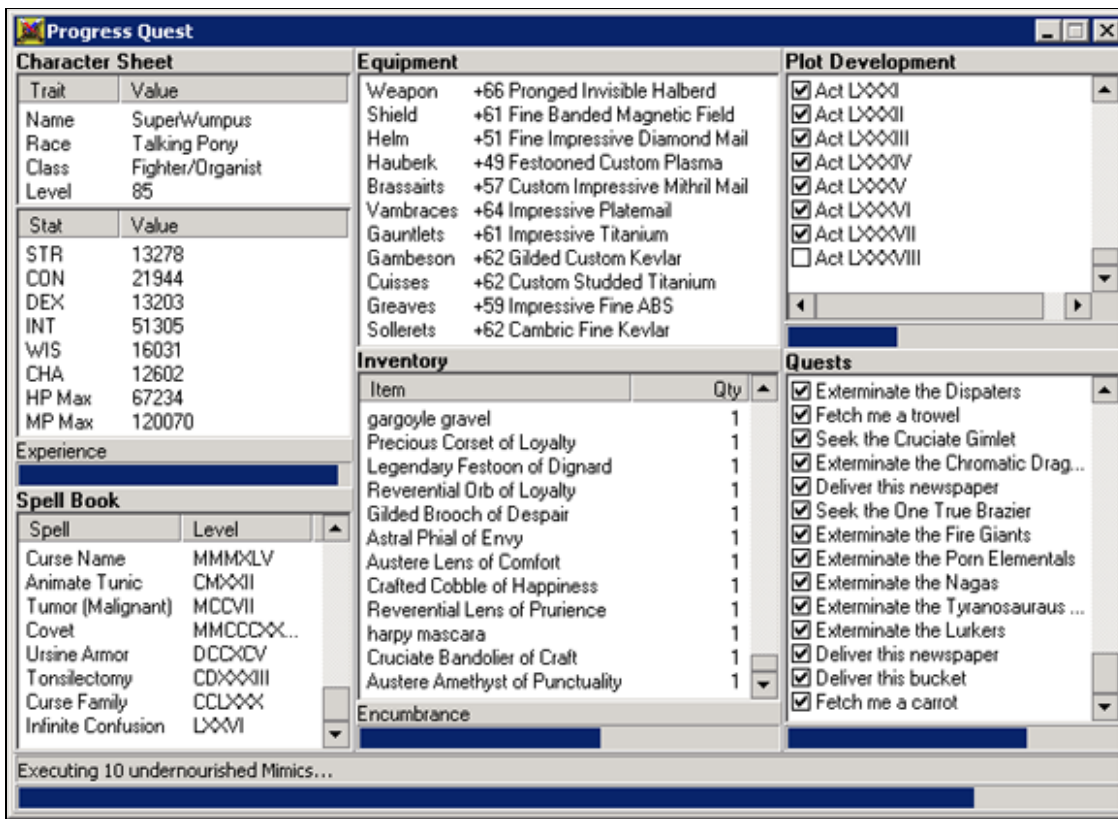
#### Adaptability:

That protein folding applet for the PlayStation 3 is a serious version of this...

#### The scheme in use:

This one's included as something of a joke but it is worth discussing all the same, because *Progress Quest* is a surprisingly addictive little applet. And it *is* the simplest possible control scheme, in that there's actually no control at all!





And there are plenty of games that have aspects of this. It is a common action in god games like *SimCity* for players to "leave their city running" for a while for people to move in, buildings to grow, tax revenue to accumulate, and so on. These games tend to have moments where, once the orders are given, the player's job is basically to watch his hard work pay off, or not. *Progress Quest* is sort of like that, but without the hard work part.

And yet... there is reason to believe this style of play is not just the realm of parody. Take *Dwarf Fortress*, for instance. Before the player can begin a game he has to wait for his computer to generate a randomized fantasy world for him a process that can take over an hour. But once it's done, the resulting playfield is unusually well-developed. It's so well-developed that the game includes a mode for just generating a world without playing, and outputting graphic files of the world's regions.

### Design lesson:

We haven't seen much of it yet because algorithmic content generation is still extremely primitive, but in the future it's possible to see games that, effectively, play themselves, using random processes to engineer worlds that are interesting enough that they can be explored simply through reading or viewing, rather than interaction.

At least they'll give the robots something to watch after they destroy human civilization.

### Glossary:

**Joystick:** Unless otherwise noted this is an "8-way" digital joystick, capable of aiming in cardinal or diagonal directions, or of being in a neutral position. These joysticks are cheap to produce and plentiful. In most cases (but not all) they can be replaced with a control pad. There are specialized joysticks produced for a few arcade machines (specifically, *Sinistar* and *Pigskin 621 A.D.*) that are capable of much more precise, 49-way movement.

**Analog joystick:** These joysticks work on a different principle from digital sticks, and can aim in "any" direction, as well as detecting how far the stick has been pushed away from neutral. It used to be that these joysticks were rather unsuited for general gaming purposes; it wasn't really all that long ago when analog stick centering was considered an optional feature. The use of analog sticks, instead of digital ones, is considered a key aspect of the failure of the Atari 5200 game console, but Apple home computers also utilized them, and they were the default on PCs for a long time. They finally hit the mainstream when a centering analog stick was a prominent feature on the Nintendo 64's innovative controller, and they really took off when the original PlayStation's DualShock controller sported twin analog thumbsticks. Their absence on a console seems unthinkable now.

**Button:** The simplest control possible, and by far the oldest given its use in pinball machines. It can either be pressed, or not. There are games that are controlled only by a single button, but that scheme mostly sees only specialized use. *WarioWare Twisted* has a mode where all the minigames are controlled by single button-presses. Buttons come in many forms: control panel buttons, face buttons, shoulder buttons and triggers. The PlayStation 2 has *thumbstick buttons*, which activate by pressing "in" on a thumbstick, and *analog buttons*, allowing the game to detect how hard they've been pressed. These days, analog buttons are most commonly seen on controller shoulders. The Gamecube's primary

controller famously featured analog shoulder buttons with a hard digital "click" when pressed in far enough.

*Dial*: A circular control that can be turned, with the computer detecting the direction it's pointing and/or how far, or fast, it's been moved. There are two kinds of dials: one where the dial can be turned freely as far as the player likes either clockwise or counter-clockwise, and one where there's a hard stop along the range in both directions. This second control is sometimes referred to as a *paddle*, in reference to the common controller included with Atari VCS/2600 consoles, but note that system also supported full-spin dials in the form of its little-used "driving" controller. While this control is little-seen these days and few current consoles utilize them except in the form of steering wheel controllers, some important games use dials: *Pong*, *Breakout*, *Arkanoid*, and, of course, *Tempest*.

*Trackball*: A ball, often one much like a cue ball, is set into a solid device, but can be freely moved in any direction by the player, and its movements detected and used by the computer. Sometimes considered analogous to an upside-down mouse, but it's not always the case that the two are interchangeable. Still, it sometimes sees use as a pointing device on some systems, and in arcade games, such as in *Missile Command* and *Rampart*. Atari Games made heavy use of trackballs. These days, they mostly see use in the *Golden Tee* line of arcade golf games. Some first-person shooter players swear by them for allowing for greater aiming precision than a mouse.

*Stylus*: Until fairly recently the domain of artists willing to shell out for expensive tablets, touch-screen controls first became popular on PDAs. The Nintendo DS is famous for being the first consumer gaming-focused device to use a touch-screen. While many games use its screen in a gimmicky fashion, a few make great use of it.

*Other controls*: There are many other control systems that have been used in games. This is just a partial list:

- Light guns, essentially screen-based pointer devices, are used in some shooting games. Nintendo's Wii console offers screen-pointing controls that are similar in practice.
- Motion-sensing controls are a major selling point of the Wii, and Sony's PlayStation 3 console can also detect controller movement.
- Sony's EyeToy and PlayStation Eye devices are nothing less than USB cameras, the image stream of which is used by some games to detect the player's (physical) location and movement for game purposes.
- Namco (and to a lesser extent, Sega) became known in the 90s for producing large-cabinet games with unusual control mechanisms, like skis and skateboards.
- Konami popularized the use of foot pad grids in their *Dance Dance Revolution* series.
- Konami also uses body sensors in some of their first-person shooting, dancing, and sword fighting arcade games.
- Nintendo is known for releasing strange controls for most of their systems, ranging from foot pads (Power Pad) to microphones (*Odama*) to bongo drums. The Wii seems designed around the concept of special controller "attachments."
- Sega also made their own fair share of custom controllers, like maracas, microphones and fishing poles.
- Harmonix's Guitar controller, available for PlayStation 2 and Xbox 360, became and continues to be a tremendous hit, riding the continued success of the *Guitar Hero* games.

Sources for images: [KLOV](#), [Mameworld](#), [Travelizmo](#), <http://www.arrozconnori.net/>, [Armchair Empire](#), [Craig Harris' IGN Blog](#), [Eurogamer](#), [Temple of the Roquelike](#), [Coding Horror](#)

[Return to the full version of this article](#)

Copyright © 2014 UBM Tech, All rights reserved