

The
PAC-MAN
Dossier



Jamey Pittman



version 1.0.26 June 16, 2011

 **Foreword** 

Welcome to *The Pac-Man Dossier*! This web page is dedicated to providing *Pac-Man* players of all skill levels with the most complete and detailed study of the game possible. New discoveries found during the research for this page have allowed for the clearest view yet of the actual ghost behavior and pathfinding logic used by the game. Laid out in hyperlinked chapters and sections, the dossier is easy to navigate using the [Table of Contents](#) below, or you can read it in linear fashion from top-to-bottom. [Chapter 1](#) is purely the backstory of Namco and *Pac-Man's* designer, Toru Iwatani, chronicling the development cycle and release of the arcade classic. If you want to get right to the technical portions of the document, however, feel free to skip ahead to [Chapter 2](#) and start reading there. [Chapter 3](#) and [Chapter 4](#) are dedicated to explaining pathfinding logic and discussions of unique ghost behavior. [Chapter 5](#) is dedicated to the "split screen" level, and several Appendices follow, offering [reference tables](#), an [easter egg](#), [vintage guides](#), a [glossary](#), and more. Lastly, if you are unable to find what you're looking for or something seems unclear in the text, please feel free to contact me (jamey.pittman@yahoo.com) and ask!

If you enjoy the information presented on this website, please consider contributing a small donation to support it and defray the time/maintenance costs associated with keeping it online and updated. Donations can be made **safely and securely** via PayPal by clicking the "Donate" button below. You can also make a small contribution (at no cost to you whatsoever) by simply clicking on one or more of the advertisement banners. Your support is greatly appreciated!

 **Table of Contents** 

[Introduction](#)

[Chapter 1 – Welcome To The Machine](#)

[Chapter 2 – Gameplay Details](#)

[The Basics](#)

[Modus Operandi](#)

[Reversal Of Fortune](#)

[Scatter, Chase, Repeat...](#)

[Frightening Behavior](#)

[Speed](#)

[Cornering](#)

[Home Sweet Home](#)

[Areas To Exploit](#)

[Chapter 3 – Maze Logic 101](#)

[What Tile Am I In?](#)

[Just Passing Through](#)

[Target Tiles](#)

[Looking Ahead](#)

[Intersections](#)

[Fixed Target Tiles](#)

[Chapter 4 – Meet The Ghosts](#)

[Blinky](#)

[Pinky](#)

[Inky](#)

[Clyde](#)

[Chapter 5 – Deciphering The Split Screen](#)

[On The Edge Of Forever](#)

[Playing The Level](#)

[Believe It Or Not](#)

[References And Further Reading](#)

[Appendix A — Reference Tables](#)

[Table A.1 — Level Specifications](#)

[Table A.2 — Difficulty Specifications](#)

[Appendix B — Easter Eggs & Tricks](#)

[NAMCO Easter Egg](#)

[Ghost Bug](#)

[Appendix C — Hardware Information](#)

[Specifications](#)

[Midway Operator's Reference Books](#)

[Appendix D — Vintage Pac-Man Guides](#)

[Glossary](#)

[Frequently Asked Questions](#)

[About the Author](#)



In 1999, Billy Mitchell of Hollywood, Florida became the first person to obtain a perfect score of 3,333,360 at *Pac-Man*, eating every possible dot, energizer, ghost, and bonus on every level without losing a single life in the process. But perhaps what is most amazing is the fact he can play without using any memorized routines widely known as “patterns”. Instead, he relies on his familiarity with how each ghost behaves as it moves through the maze, using that knowledge to keep Pac-Man one step ahead of his enemies at all times.

Unlike Mitchell, most players are only able to rack up high scores with the aid of multiple patterns that take advantage of the game's deterministic nature. These patterns require perfect memorization and recall to be of any real use—a single hesitation or wrong turn during execution can make the remainder of a pattern useless. Not surprisingly, an over-reliance on these routines leaves many a player clueless to how to effectively avoid the ghosts and finish off the remaining dots in the higher levels once a mistake occurs.

Most *Pac-Man* strategy guides available online today are very similar in content to the books that were sold back in the early 80s: a summary of gameplay and scoring is provided first, followed by a list of patterns to be memorized by the reader, but very little insight is offered on how the game works or how the ghosts make decisions. Therefore, the purpose of this guide is to give the player a better understanding of *Pac-Man* without the use of patterns by taking a closer look at gameplay, maze logic, ghost personalities, and the mysterious “split screen” level. I can't promise that you'll play as well as Billy Mitchell after reading this guide, but you will definitely feel much better about recovering from a mistake in your ninth key pattern once you have a good understanding of how the ghosts “think”.

All information provided has been extracted from or verified with disassembly output from the original *Pac-Man* code ROMs in conjunction with extensive testing and observations made during gameplay. As such, I am highly confident in the accuracy of this document. That being said, if you notice an error or omission, please contact me at jamey.pittman@yahoo.com so it can be corrected as soon as possible. Hopefully you will find this material as interesting and useful as I did for gaining a better understanding of this classic arcade title.

Special thanks to Don Hodges (www.donhodes.com) whose invaluable contributions to this guide can be found in every chapter.

[Back](#)



CHAPTER 1:

Welcome To The Machine



“I don't have any particular interest in [computers]. I'm interested in creating images that communicate with people. A computer is not the only medium that uses images; I could use the movies or television or any other visual medium. It just so happens I use the computer.”—Toru Iwatani

It was 1977 when a self-taught, capable young man named Toru Iwatani came to work for Namco Limited, a Tokyo-based amusement manufacturer whose main product lines at the time were projection-based amusement rides and light gun shooting galleries. He was just 22 years old with no formal training in computers, visual arts, or graphic design, but his creativity and aptitude for game design were obvious to the Namco executives that met with Iwatani. They offered to hire him—with assurances they would find a place for him in the company—and he accepted.

Iwatani eventually found his place designing titles for Namco's new video games division. His limited computer skills necessitated his being paired with a programmer who would write the actual code while Iwatani took on the role of game designer for the project. This was a new job for the game industry in 1977 when most games were designed by the programmers who coded them. In addition to a programmer, Iwatani's team would usually include a hardware engineer to develop the various devices and components, a graphic artist to realize his visual ideas, and a music composer for any music and sound effects needed in the game.

Iwatani had initially wanted to work on pinball machines, but Namco had no interest in the pinball business. Perhaps as a concession, his first game design, called *Gee Bee*, was a paddle game similar to Atari's *Breakout* but with a decidedly pinball-inspired slant to the gameplay. Released in 1978, it was Namco's first original video game—they had only ported existing Atari games to the Japanese market up to this point—and it enjoyed moderate success in the arcades.



But the paddle games were losing ground fast to a new genre. The unprecedented success of Taito's *Space Invaders* in 1978 caused an industry-wide shift toward space-themed, shoot-'em-up games (as well as a national coin shortage in Japan). Game manufacturers scrambled to match Taito's success with space shooters of their own. Namco was quick to follow suit, assigning a team to start work on a *Space Invaders* clone at once. It was around this time that Toru Iwatani began thinking about designing a different kind of game. He felt the shoot-'em-up craze was destined to fade away like the paddle games before them. Rather than make another space shooter, he wanted to take his game design in a completely new direction that did not focus on violence or conflict, and would appeal to both male *and* female audiences.

He took inspiration from a children's story about a creature that protected children from monsters by eating them. One of Iwatani's design methods included taking key words associated with a story to aid developing his ideas. The kanji word *taberu* ("to eat"), became the premise for the game. The word *kuchi* ("mouth") has a square shape for its kanji symbol and provided the inspiration for the game's main character—the better-known legend of Iwatani receiving his inspiration from a pizza pie with a slice missing was, by his own admission, not entirely correct:

"Well, it's half true. In Japanese the character for mouth (kuchi) is a square shape. It's not circular like the pizza, but I decided to round it out. There was the temptation to make the Pac-Man shape less simple. While I was designing this game, someone suggested we add eyes. But we eventually discarded that idea because once we added eyes, we would want to add glasses and maybe a moustache. There would just be no end to it. Food is the other part of the basic concept. In my initial design, I had put the player in the midst of food all over the screen. As I thought about it, I realized the player wouldn't know exactly what to do: the purpose of the game would be obscure. So I created a maze and put the food in it. Then whoever played the game would have some structure by moving through the maze. The Japanese have a slang word—paku paku—they use to describe the motion of the mouth opening and closing while one eats. The name Puck-Man came from that word."



—Toru Iwatani



The monsters from the children's story were included as four ghosts that chase the player through the maze, providing an element of tension. Attacks on the player were designed to come in waves (similar to *Space Invaders*) as opposed to an endless assault, and each ghost was given a unique personality and character. The children's story also included the concept of *kokoro* ("spirit") or a life force used by the creature that allowed him to eat the monsters. Toru incorporated this aspect of the story as four edible power pellets in the maze that turn the tables on the ghosts, making them vulnerable to being eaten by the player.



With a name and a basic design in place, Iwatani was ready to begin work. The team Namco assigned Iwatani to bring *Puck-Man* to life included a programmer (Shigeo Funaki), a hardware engineer, a cabinet designer, and a music composer (Toshio Kai). Development got underway in early 1979. In the course of that year, two new pinball-themed designs from Iwatani—*Bomb Bee* and *Cutie Q*—were both released during *Puck-Man*'s development cycle. Both games were similar to *Gee Bee* but with stronger gameplay and improved visuals.

The Namco team working on the *Space Invaders* clone for the past several months had just achieved a technological coup for Namco: the first game to use a true, multi-colored, RGB display instead of the monochrome monitors with colored cellophane tape so prevalent at the time. Thanks to the breakthrough of the other team, Iwatani now had the new promise of color to enhance his design. Mindful that he wanted the game to appeal to women, he immediately decided to use it on the ghosts, choosing pastel shades for the bodies and adding expressive, blue eyes. Dark blue was used for the maze itself, while Puck-Man was drenched in a brilliant yellow.

The look and feel of *Puck-Man* continued to evolve for over a year. A large amount of time and effort was put into developing the ghosts' unique movement patterns through the maze and tweaking the game difficulty variables as boards were cleared. Bonus symbols (including the *Galaxian* flagship) were added into the mix at some point, and the ghosts were finally given names: Akabei, Pinky, Aosuke, and Guzuta. Sound effects and music were some of the final touches added as development neared an end along with constant tweaking of the ghosts' behavior.

Puck-Man's creation was a year and five months in the making—the longest ever for a video game to that point. Finally, on May 22nd, 1980, it was released to arcades in Japan. Initially, the game did moderately well, but was no overnight sensation. In fact, Namco's multi-colored *Space Invaders* clone, called *Galaxian*, was much more popular with the gaming public—the predominately male, game-playing audience in Japan was unsure what to make of *Puck-Man* with its cartoon-like characters, maze, and pastel colors, whereas *Galaxian* was more immediately familiar to them with its shoot-'em-up space theme.



Midway was a distributor of coin-operated video games in the U.S. that was always looking for the next big hit from Japan to license and bring to America. They opted for both *Puck-Man* and *Galaxian*, modifying the cabinets and artwork to make them easier to manufacture as well as providing a more American look and feel.



Puck-Man went through the majority of the changes: the cabinet was modified slightly, changing the color from white to a bright yellow to make it stand out in the arcade. The detailed, multi-colored cabinet artwork was replaced with



CHAPTER 2:

Gameplay Details



“As Pac-Man was originally conceived to appeal to women players, it is a very easy and approachable game. I believe that is an ingredient in the longevity of the game.”—Toru Iwatani, creator of Pac-Man

The Basics

The premise of *Pac-Man* is delightfully simple: using a four-way joystick, the player guides Pac-Man—up, down, left, and right—through a maze filled with dots for him to gobble up. Four ghost monsters are also in the maze and chase after our hero, trying to capture and kill him. The goal is to clear the maze of dots while avoiding the deadly ghosts. Each round starts with the ghosts in the “monster pen” at the center of the maze, emerging from it to join in the chase. If Pac-Man is captured by a ghost, a life is lost, the ghosts are returned to their pen, and a new Pac-Man is placed at the starting position before play continues. When the maze is cleared of all dots, the board is reset, and a new round begins. If Pac-Man gets caught by a ghost when he has no extra lives, the game is over.

There are 244 dots in the maze, and Pac-Man must eat them all in order to proceed to the next round. The 240 small dots are worth ten points each, and the four large, flashing dots—best known as *energizers*—are worth 50 points each. This yields a total of 2,600 points for clearing the maze of dots each round. Players have two ways to increase their score beyond what is earned from eating dots:



The first way to increase your score each round is by turning the tables on your enemies by making them your prey. Whenever Pac-Man eats one of the four energizer dots located in the corners of the maze, the ghosts reverse their direction and, in early levels, turn the same shade of blue for a short period of time before returning to normal. While blue, they are vulnerable to Pac-Man and can be gobbled up for extra points providing they are caught before the time expires. After being eaten, a ghost's eyes will return to the monster pen where it is resurrected, exiting to chase Pac-Man once again. The first ghost captured after an energizer has been eaten is always worth 200 points. Each additional ghost captured from the same energizer will then be worth twice as many points as the one before it—400, 800, and 1,600 points respectively. If all four ghosts are captured at all four energizers, an additional 12,000 points can be earned on these earlier levels. This should not prove too terribly difficult to achieve for the first few rounds as the ghosts initially remain blue for several seconds. Soon after, however, the ghosts' “blue time” will get reduced to one or two seconds at the most, making it much more problematic to capture four before time runs out on these boards. By level 19, the ghosts stop turning blue altogether and can no longer be eaten for additional points.

The second way to increase your score each round is by eating the *bonus symbols* (commonly known as *fruit*) that appear directly below the monster pen twice each round for additional points. The first bonus fruit appears after 70 dots have been cleared from the maze; the second one appears after 170 dots are cleared. Each fruit is worth anywhere from 100 to 5,000 points, depending on what level the player is currently on. Whenever a fruit appears, the amount of time it stays on the screen before disappearing is always between nine and ten seconds. The *exact* duration (i.e., 9.3333 seconds, 10.0 seconds, 9.75 seconds, etc.) is variable and does not become predictable with the use of patterns. In other words, executing the *same* pattern on the *same* level twice is no guarantee for how long the bonus fruit will stay onscreen each time. This usually goes unnoticed given that the majority of patterns are designed to eat the bonus fruit as quickly as possible after it has been triggered to appear. The symbols used for the last six rounds completed, plus the current round are also shown along the bottom edge of the screen (often called the fruit counter or level counter). See [Table A.1](#) in the appendices for all bonus fruit and scoring values, per level.

[Back](#)

Modus Operandi

Ghosts have three mutually-exclusive modes of behavior they can be in during play: *chase*, *scatter*, and *frightened*. Each mode has a different objective/goal to be carried out:

1. **CHASE**—A ghost's objective in *chase* mode is to find and capture Pac-Man by hunting him down through the maze. Each ghost exhibits unique behavior when chasing Pac-Man, giving them their different personalities: Blinky (red) is very aggressive and hard to shake once he gets behind you, Pinky (pink) tends to get in front of you and cut you off, Inky (light blue) is the least predictable of the bunch, and Clyde (orange) seems to do his own thing and stay out of the way.
2. **SCATTER**—In *scatter* mode, the ghosts give up the chase for a few seconds and head for their respective home corners. It is a welcome but brief rest—soon enough, they will revert to chase mode and be after Pac-Man again.
3. **FRIGHTENED**—Ghosts enter *frightened* mode whenever Pac-Man eats one of the four energizers located in the far corners of the maze. During the early levels, the ghosts will all turn dark blue (meaning they are vulnerable) and aimlessly wander the maze for a few seconds. They will flash moments before returning to their previous mode of behavior.

[Back](#)

Reversal Of Fortune



In all three modes of behavior, the ghosts are prohibited from reversing their direction of travel. As such, they can only choose between continuing on their current course or turning off to one side or the other at the next intersection. Thus, once a ghost chooses which way to go at a maze intersection, it has no option but to continue forward on that path until the next intersection is reached. Of course, if you've spent any time playing *Pac-Man*, you already know the ghosts *will* reverse direction at certain times. But how can this be if they are expressly prohibited from doing so on their own? The answer is: when changing modes, the system can override the ghosts' normal behavior, forcing them to go the opposite way. Whenever this happens, it is a visual indicator of their behavior changing from one mode to another. Ghosts are forced to reverse direction by the system anytime the mode changes from: chase-to-scatter, chase-to-frightened, scatter-to-chase, a scatter-to-frightened. Ghosts do not reverse direction when changing back from frightened to chase or scatter modes.

When the system forces the ghosts to reverse course, they do not necessarily change direction simultaneously; some ghosts may continue forward for a fraction of a second before turning around. The delay between when the system signals a reversal and when a ghost actually responds depends on how long it takes the ghost to enter the next game tile along its present course after the reversal signal is given (more on tiles in Chapter 3). Once the ghost enters a new tile, it will obey the reversal signal and turn around.

[Back](#)

Scatter, Chase, Repeat...

Ghosts alternate between scatter and chase modes during gameplay at predetermined intervals. These mode changes are easy to spot as the ghosts reverse direction when they occur. Scatter modes happen four times per level before the ghosts stay in chase mode indefinitely. Good players will take full advantage of the scatter periods by using the brief moment when the ghosts are not chasing Pac-Man to clear dots from the more dangerous areas of the maze. The scatter/chase timer gets reset whenever a life is lost or a level is completed. At the start of a level or after losing a life, ghosts emerge from the ghost pen already in the first of the four scatter modes.

For the first four levels, the first two scatter periods last for seven seconds each. They change to five seconds each for level five and beyond. The third scatter mode is always set to five seconds. The fourth scatter period lasts for five seconds on level one, but then is only $1/60^{\text{th}}$ of a second for the rest of play. When this occurs, it appears as a simple reversal of direction by the ghosts. The first and second chase periods last for 20 seconds each. The third chase period is 20 seconds on level one but then balloons to 1,033 seconds for levels two through four, and 1,037 seconds for all levels beyond—lasting over 17 minutes! If the ghosts enter frightened mode, the scatter/chase timer is paused. When time runs out, they return to the mode they were in before being frightened and the scatter/chase timer resumes. This information is summarized in the following table (all values are in seconds):



Mode	Level 1	Levels 2–4	Levels 5+
Scatter	7	7	5
Chase	20	20	20
Scatter	7	7	5
Chase	20	20	20
Scatter	5	5	5
Chase	20	1033	1037
Scatter	5	1/60	1/60
Chase	indefinite	indefinite	indefinite

[Back](#)

Frightening Behavior

Whenever Pac-Man eats one of the four energizer dots located near the corners of the board, the ghost reverse direction and, on earlier levels, go into *frightened* mode for a short period of time. When frightened, the ghosts all turn the same shade of dark blue and move more slowly than normal. They wander aimlessly through the maze and flash white briefly as a warning before returning to their previous mode of behavior. Ghosts use a pseudo-random number generator (PRNG) to pick a way to turn at each intersection when frightened. The PRNG generates a pseudo-random memory address to read the last few bits from. These bits are translated into the direction a frightened ghost must first try. If the selected direction is not blocked by a wall or opposite the ghost's current direction of travel, it is accepted. Otherwise, the code proceeds in a clockwise fashion to the next possible direction and tries again, repeating this test until an acceptable direction is found. The PRNG gets reset with the same initial seed value at the start of each new level and whenever a life is lost. This results in frightened ghosts always choosing the same paths when executing patterns during play. As the levels progress, the time ghosts spend in frightened mode grows shorter until eventually they no longer turn blue at all (they still reverse direction, however). Refer to [Table A.1](#) in the appendices for the frightened time in seconds and number of flashes, per level.

[Back](#)

Speed

The game starts with Pac-Man at 80% of his maximum speed. By the fifth level, Pac-Man is moving at full speed and will continue to do so until the 21st level. At that point, he slows back down to 90% and holds this speed for the remainder of the game. Every time Pac-Man eats a regular dot, he stops moving for one frame (1/60th of a second), slowing his progress by roughly ten percent—just enough for a following ghost to overtake him. Eating an energizer dot causes Pac-Man to stop moving for three frames. The normal speed maintained by the ghosts is a little slower than Pac-Man's until the 21st level when they start moving faster than he does. If a ghost enters a side tunnel, however, its speed is cut nearly in half. When frightened, ghosts move at a much slower rate of speed than normal and, for level one through four, Pac-Man also speeds up. The table below summarizes the speed data for both Pac-Man and the ghosts, per level. This information is also contained in [Table A.1](#) in the appendices.

LEVEL	PAC-MAN SPEED				GHOST SPEED		
	NORM	NORM DOTS	FRIGHT	FRIGHT DOTS	NORM	FRIGHT	TUNNEL
1	80%	~71%	90%	~79%	75%	50%	40%
2 – 4	90%	~79%	95%	~83%	85%	55%	45%
5 – 20	100%	~87%	100%	~87%	95%	60%	50%
21+	90%	~79%	—	—	95%	—	50%

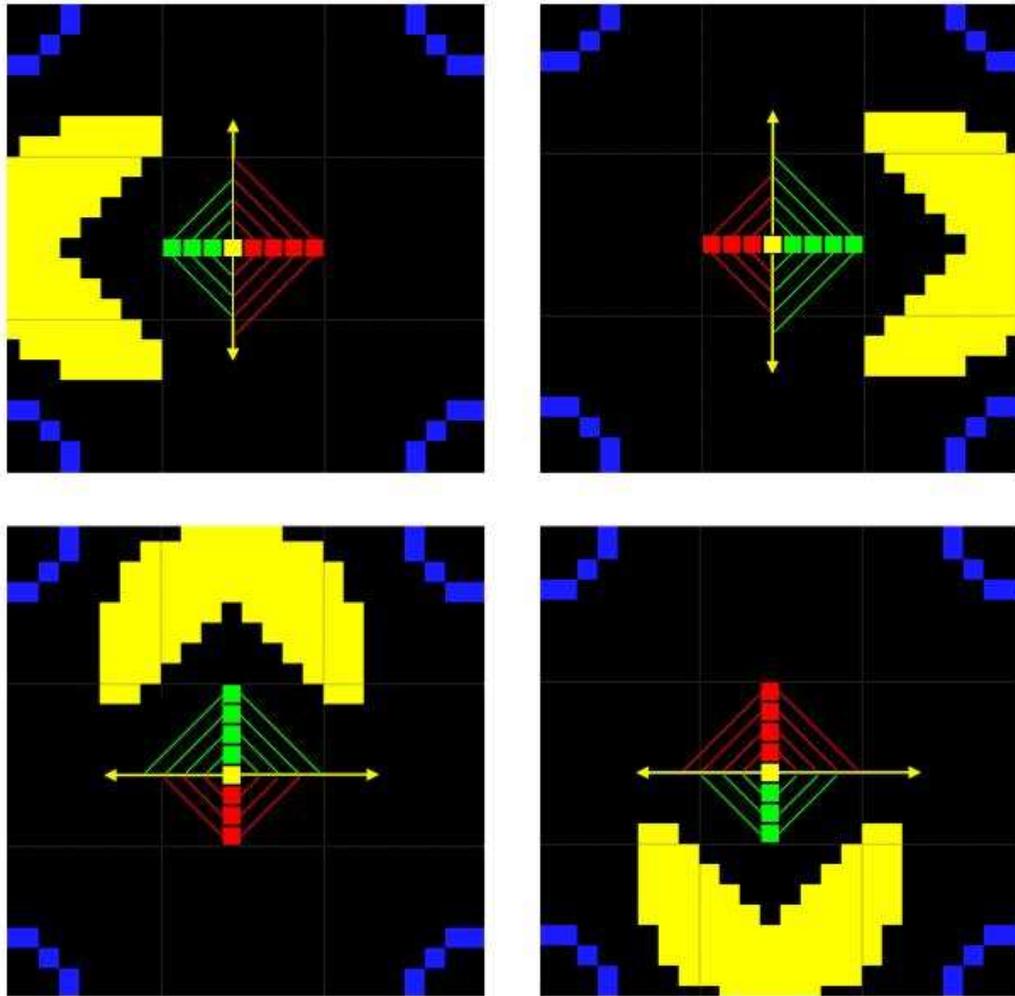
[Back](#)

Cornering

Pac-Man is able to navigate the turns in the maze faster than his enemies. He does not have to wait until he reaches the middle of a turn to change direction as the ghosts do (see picture below). Instead, he may start turning several pixels before he reaches the center of a turn and for several pixels after passing it. Turns taken one or more pixels before reaching the center are "pre-turns"; turns taken one or more pixels after are "post-turns". Players learn to consistently move the joystick in the direction Pac-Man should go well before arriving at the center of a turn, ensuring each pre-turn is started as many pixels away from center as possible. This technique is known as *cornering* and is one of the first skills a new *Pac-Man* player should master. For every successful pre-turn maneuver, Pac-Man puts a little more distance between himself and any ghosts following close behind. Such a small gain in distance may not seem terribly significant at first, but cornering through a quick series of turns will shake off even the most determined pursuer. It is a vital tool for survival in the higher levels of the game.



Whenever Pac-Man makes a pre-turn or post-turn, his orientation changes, and he starts to move one pixel in his new direction for every pixel traveled in his old direction, effectively doubling his speed as he moves at a 45 degree angle. Once he reaches the centerline of the new direction's path, he starts moving purely in that direction and his speed returns to normal. The greatest distance advantage is thereby gained by making the earliest pre-turn possible. The illustration below shows the layout of pre-turn pixels (shown in green), center point pixels (shown in yellow), and post-turn pixels (shown in red) for each of the four possible directions a turn can be approached. Each example shows Pac-Man entering the same four-way intersection from a different direction. When entering from the left, there are three pre-turn pixels before the center of the turn, and four post-turn pixels. Conversely, entering the same intersection from the right yields four pre-turn pixels and three post-turn ones. Entering from the top as opposed to the bottom exhibits the same property. For any turn that is made later than the earliest possible pre-turn, Pac-Man will be one frame behind where he would be for every pixel of "lateness" in the turn. Basically, it pays to move the joystick well before reaching a turn to maximize your speed.



Turning at the earliest pre-turns possible is also required to successfully execute most any pattern. Patterns are meant to be played with perfect cornering because it removes the human element of uncertainty as to when Pac-Man will turn. Without cornering, it would be nigh-impossible to reproduce the exact timing of every turn as made by the pattern's author, thereby increasing the possibility of unpredictable ghost behavior due to Pac-Man not being in the exact same tile at the exact same time anymore. Typically, the most popular patterns have been those that tend to "hold together" well when small input timing flaws occur (turning three pixels away from center instead of four when approaching turn from the right is a timing flaw, for example). Other patterns—especially those that bring Pac-Man very close to the ghosts late in the sequence—tend to "fall apart" unless every turn is perfectly cornered. During a long *Pac-Man* session, even the best players will make occasional timing mistakes during a fast series of turns and have to deal with the possible consequences. As such, one should aim for perfect cornering at all times but remain alert for unexpected ghost behavior from subtle input timing flaws creeping into the pattern.

[Back](#)

Home Sweet Home



Commonly referred to as the *ghost house* or *monster pen*, this cordoned-off area in the center of the maze is the domain of the four ghosts and off-limits to Pac-Man. Whenever a level is completed or a life is lost, the ghosts are returned to their starting positions in and around the ghost house before play continues—Blinky is always located just above and outside, while the other three are placed inside: Inky on the left, Pinky in the middle, and Clyde on the right. The pink door on top is used by the ghosts to enter or exit the house. Once a ghost leaves, however, it cannot reenter unless it is first captured by Pac-Man—then the disembodied eyes can return home to be revived. Since Blinky is already outside after a level is completed or a life is lost, the only time he can get inside the ghost house is after Pac-Man captures him, and he immediately turns around to leave once revived. That's about all there is to know about Blinky's behavior in terms of the ghost house, but determining when the other three ghosts leave home is an involved process based on several variables and conditions. The rest of this section will deal with them exclusively. Accordingly, any mention of "the ghosts" below refers to Pinky, Inky, and Clyde, but not Blinky.

The first control used to evaluate when the ghosts leave home is a personal counter each ghost retains for tracking the number of dots Pac-Man eats. Each ghost's "dot counter" is reset to zero when a level begins and can only be active when inside the ghost house, but only one ghost's counter can be active at any given time regardless of how many ghosts are inside. The order of preference for choosing which ghost's counter to activate is: Pinky, then Inky, and then Clyde. For every dot Pac-Man eats, the preferred ghost in the house (if any) gets its dot counter increased by one. Each ghost also has a "dot limit" associated with his counter, per level. If the preferred ghost reaches or exceeds his dot limit, it immediately exits the house and its dot counter is deactivated (but not reset). The most-preferred ghost still waiting inside the house (if any) activates its timer at this point and begins counting dots.

Pinky's dot limit is always set to zero, causing him to leave home immediately when every level begins. For the first level, Inky has a limit of 30 dots, and Clyde has a limit of 60. This results in Pinky exiting immediately which, in turn, activates Inky's dot counter. His counter must then reach or exceed 30 dots before he can leave the house. Once Inky starts to leave, Clyde's counter (which is still at zero) is activated and starts counting dots. When his counter reaches or exceeds 60, he may exit. On the second level, Inky's dot limit is changed from 30 to zero, while Clyde's is changed from 60 to 50. Inky will exit the house as soon as the level begins from now on. Starting at level three, all the ghosts have a dot limit of zero for the remainder of the game and will leave the ghost house immediately at the start of every level.

Whenever a life is lost, the system disables (but does not reset) the ghosts' individual dot counters and uses a global dot counter instead. This counter is enabled and reset to zero after a life is lost, counting the number of dots eaten from that point forward. The three ghosts inside the house must wait for this special counter to tell them when to leave. Pinky is released when the counter value is equal to 7 and Inky is released when it equals 17. The only way to deactivate the counter is for Clyde to be inside the house when the counter equals 32; otherwise, it will keep counting dots even after the ghost house is empty. If Clyde *is* present at the appropriate time, the global counter is reset to zero and deactivated, and the ghosts' personal dot limits are re-enabled and used as before for determining when to leave the house (including Clyde who is still in the house at this time).

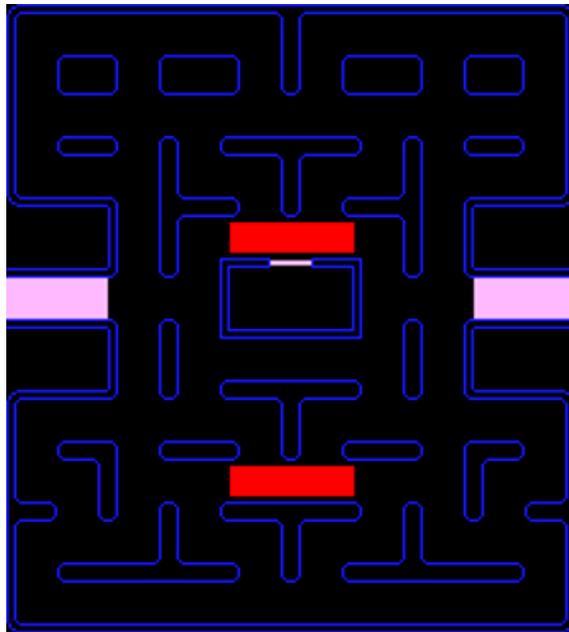
If dot counters were the only control, Pac-Man could simply stop eating dots early on and keep the ghosts trapped inside the house forever. Consequently, a separate timer control was implemented to handle this case by tracking the amount of time elapsed since Pac-Man has last eaten a dot. This timer is always running but gets reset to zero each time a dot is eaten. Anytime Pac-Man avoids eating dots long enough for the timer to reach its limit, the most-preferred ghost waiting in the ghost house (if any) is forced to leave immediately, and the timer is reset to zero. The same order of preference described above is used by this control as well. The game begins with an initial timer limit of four seconds, but lowers to it to three seconds starting with level five.

The more astute reader may have already noticed there is a subtle flaw in this system resulting in a way to keep Pinky, Inky, and Clyde inside the ghost house for a very long time after eating them. The trick involves having to sacrifice a life in order to reset and enable the global dot counter, and then making sure Clyde exits the house *before* that counter is equal to 32. This is accomplished by avoiding eating dots and waiting for the timer limit to force Clyde out. Once Clyde is moving for the exit, start eating dots again until at least 32 dots have been consumed since the life was lost. Now head for an energizer and gobble up some ghosts. Blinky will leave the house immediately as usual, but the other three ghosts will remain "stuck" inside as long as Pac-Man continues eating dots with sufficient frequency as not to trigger the control timer. Why does this happen? The key lies in how the global dot counter works—it cannot be deactivated if Clyde is outside the house when the counter has a value of 32. By letting the timer force Clyde out before 32 dots are eaten, the global dot counter will keep counting dots instead of deactivating when it reaches 32. Now when the ghosts are eaten by Pac-Man and return home, they will still be using the global dot counter to determine when to leave. As previously described, however, this counter's logic only checks for three values: 7, 17, and 32, and once those numbers are exceeded, the counter has no way to release the ghosts associated with them. The only control left to release the ghosts is the timer, which can be easily avoided by eating a dot every so often to reset it. Click on the YouTube video below to see a demonstration of this curious behavior:

The last thing to mention about the ghost house is how to determine whether a ghost will move right or left after exiting the house. Ghosts typically move to the left once they get outside, but if the system changes modes one or more times when a ghost is inside, that ghost will move to the right instead of the left upon leaving the house.

[Back](#)

Areas To Exploit



The illustration above highlights four special “zones” in the maze where ghost behavior is limited by certain conditions which can be exploited to the player's advantage. The two red zones represent the areas where ghosts are forbidden to make upward turns. Once a ghost enters either of these two zones it may only travel from right-to-left or left-to-right until exiting the area. Thus, only Pac-Man has access to these four, upward-facing tunnel entrances. It will serve the player well to remember the ghosts can still access these tunnels from the other end! The red zone restrictions are enforced during both scatter and chase modes, but in frightened mode the red zones are ignored temporarily, allowing the ghosts to turn upwards if they so choose. The pink zones are in the two halves of the connecting side-tunnel. As mentioned previously, any ghost that enters the tunnel will suffer an immediate speed penalty until leaving the zone. This slow-down rule is always enforced and applies to ghosts only—Pac-Man is immun

[Back](#)

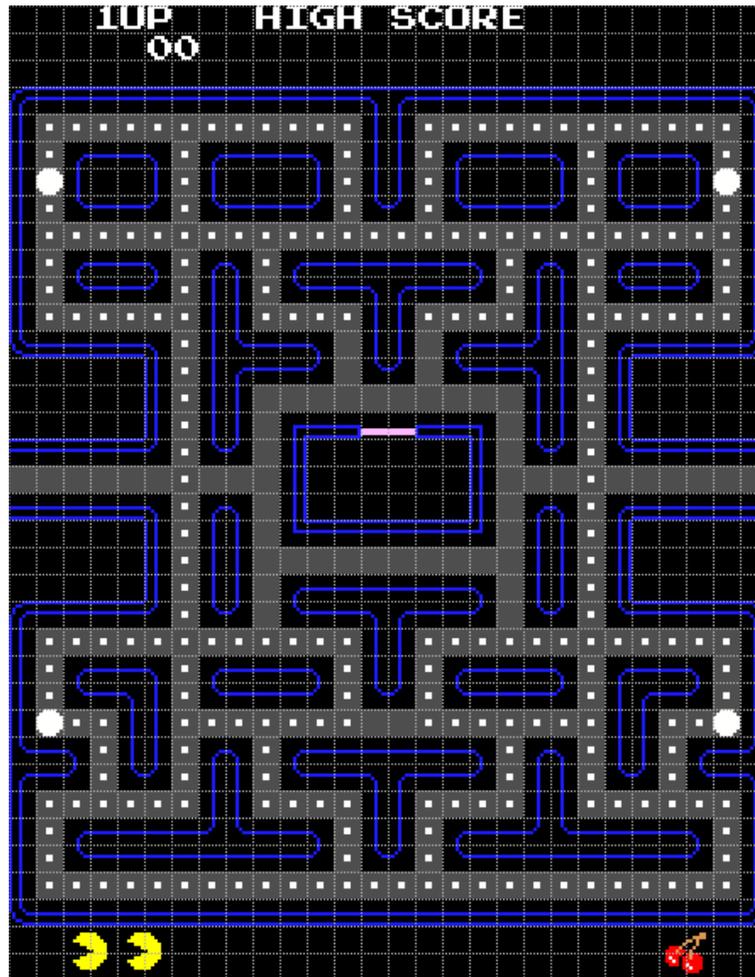


CHAPTER 3:

🍌 Maze Logic 101 🍌



We need to take a look at how ghosts are able to move through the maze in pursuit of a goal. All pathfinding logic described in this chapter is shared by the four ghosts—it is important to understand what they have in common before we get into what makes them different. Before we proceed, let's see how the game tracks the location of Pac-Man and the four ghosts (herein referred to as *actors* for brevity's sake). The visible game screen should be thought of as a regular grid of tiles, each eight pixel square. The actual pixel dimensions of the screen are 224 x 288, so dividing each value by eight yields grid that is 28 x 36 tiles in size:

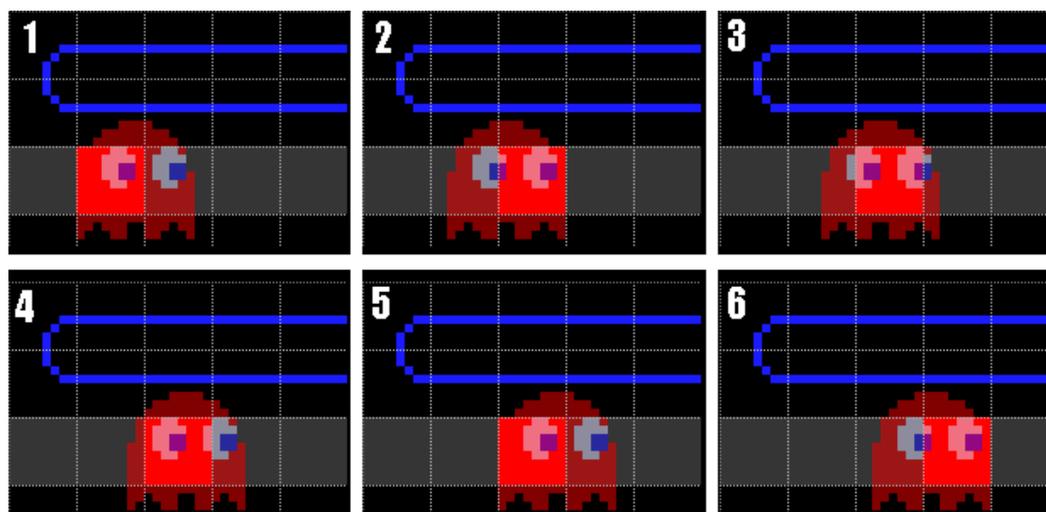


Each tile is either in legal space or dead space. In the picture above, legal space is shown as the gray-colored tiles; all other tiles are considered dead space. Actors only travel between the tiles in legal space. Each dot sits in the center of a tile, meaning they are exactly eight pixels (one tile) apart—this is useful for estimating distances during gameplay.

[Back](#)

What Tile Am I In?

As the actors move through the maze, the game keeps track of the tile each one occupies. An actor is only associated with a single tile at a time, although its graphic will overlap into the surrounding tiles. The location of the actor's center point is what determines the tile it occupies at any given time. As the actors can move at pixel-level precision, they are often not centered directly on top of the tile they are in. Consider the following example:



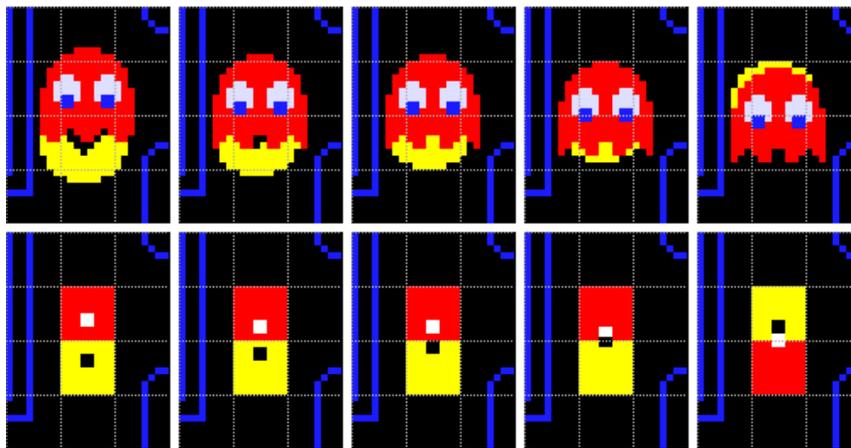
The transparent red ghost is moving left-to-right across a row of tiles in legal space. In frame one, its occupied tile (shown in bright red) is near the left side of the picture. It does not matter that some of the ghost's graphic is not in the tile—what matters is that the ghost's center point *is* in the tile. By frame two, it has moved far enough for its center point to be in the adjacent tile to the right and its occupied tile is updated accordingly. The ghost continues to be associated with the same tile until frame six when its center point has now crossed over into the next one.

The underlying concept of tiles is essential for understanding the ghosts' pathfinding logic as it only cares about the tile an actor occupies—not its per-pixel location within that tile. To the logic routines, the five actors look very much like the picture below. Each actor is defined by the tile it presently occupies along with its current direction of travel. Distances between actors are also measured in tiles (the pink ghost five tiles away from Pac-Man horizontally and one tile away vertically, for example).

Just Passing Through

It wasn't too long after the release of *Pac-Man* when word began to spread of players occasionally passing straight through a ghost unharmed, seemingly at random. This rumor turned out to be completely true as most die-hard *Pac-Man* players can attest. If you play the game long enough, you will eventually see Pac-Man run into one of the ghosts and come out unscathed on the other side—it doesn't happen very often so enjoy it when it does! Some players have even gone so far as to incorporate this mysterious pass-through oddity into their patterns.

The root cause of this elusive peculiarity lies in the way the game detects collisions between Pac-Man and the four ghosts. Any time Pac-Man occupies the same tile as a ghost, he is considered to have collided with that ghost and a life is lost. It is irrelevant whether the ghost moved into Pac-Man's tile or Pac-Man into the ghost's—the result is the same either way. This logic proves sufficient for handling collisions more than 99% of the time during gameplay, but does not account for one very special case:



The above picture illustrates the conditions necessary to produce this curious behavior. There are five consecutive frames showing Blinky and Pac-Man passing through each other. Below each frame is the same scene represented by the tiles they currently occupy and the per-pixel location of their center points. Pac-Man and Blinky are at just the right position and speed relative to one another to cause them to swap tiles with each other simultaneously. In other words, Pac-Man's center point moves upwards into Blinky's tile in the same $1/60^{\text{th}}$ of a second that Blinky's center point moves downwards into Pac-Man's tile, resulting in them moving past each other without colliding. Note that Pac-Man's origin point is centered on the top edge of his tile in frame four; this is still considered to be inside the bottom tile, but moving up one more pixel will push him over the edge into the next one. Pac-Man and Blinky have now swapped tiles with each other in frame five, and Pac-Man can go on his merry way because he never "collided" (i.e., shared the same tile) with Blinky at all! Click on the YouTube video below to see an example of the pass-through bug (it happens 40 seconds after playback begins):

[Back](#)

Target Tiles

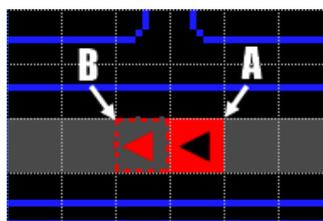
Whenever a ghost is in chase or scatter mode, it is trying to reach a *target tile* somewhere on (or off) the screen. A target tile is merely a way to describe the tile a ghost would like to occupy at any given moment. This tile can be fixed in place or change location frequently. Whenever the ghosts scatter to the corners of the maze, for example, each ghost is striving to reach a fixed target tile located somewhere near its home corner. In chase mode, the target tile is usually (but not always) related to Pac-Man's current tile which changes often. Although it may not be obvious at first, the only difference between chase and scatter mode to a ghost is where its target tile is located. The same pathfinding logic applies in either case.

[Back](#)

Looking Ahead

Ghosts are always thinking one step into the future as they move through the maze. Whenever a ghost enters a new tile, it looks ahead to the next tile along its current direction of travel and decides which way it will go when it gets there. When it eventually reaches that tile, it will change its direction of travel to whatever it had decided on a tile beforehand. The process is then repeated, looking ahead into the next tile along its new direction of travel and making its next decision on which way to go.

When a ghost looks ahead into the upcoming tile, it must examine the possible exits from that tile to determine a way to proceed. In the picture below, the red ghost has just arrived at tile A and is moving right-to-left. It immediately looks ahead to tile B (the next tile along its direction of travel). Each tile has four potential exits to be considered: right, left, up, and down. In the case of tile B, the up and down exits are blocked by walls and must be discarded as potential candidates. The right exit is also discounted because it would only take the ghost back to tile A again, and ghosts never voluntarily reverse direction. With three of the four possible exits eliminated from tile B, moving left is the only remaining choice.

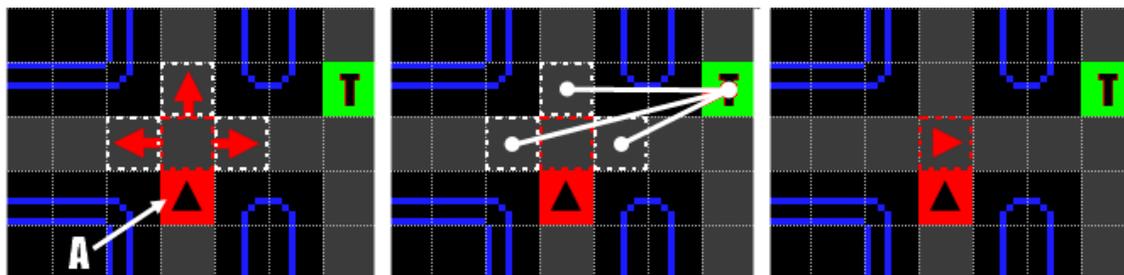


This example is the most simple to explain as the ghost has but one way it can legally move. As such, you did not have to worry about where its target tile was located. The majority of game tiles in legal space are similar to this one, but things get more interesting when a ghost approaches a tile with more potential exits to choose from.

[Back](#)

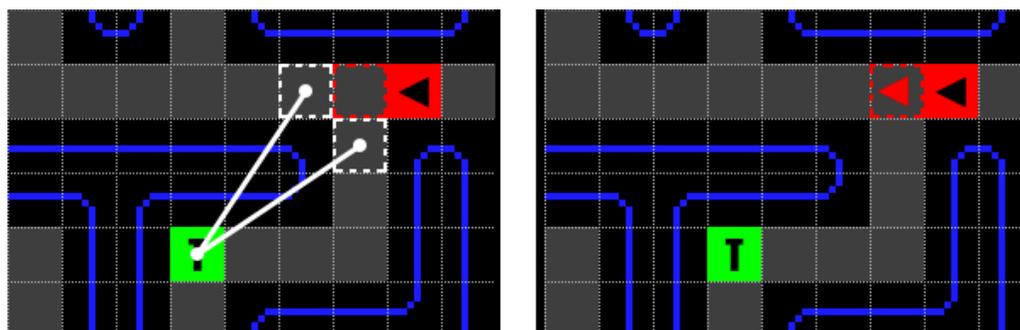
Intersections

When a ghost arrives one tile away from an upcoming intersection, it must choose between several possible directions in which to proceed. Consider the following example:



In the first picture, the red ghost has just reached tile A and is seeking its target (shown as the green tile). It immediately looks ahead to the subsequent tile along its present direction of travel (up). In this case, that tile is a four-way intersection. As this intersection tile has no walls blocking off any of the exits the ghost can only discard his reverse direction (down), leaving three exits open for travel. It looks one tile beyond the intersection in each of the three remaining directions, collecting "test tiles" (shown as tiles with dashed, white lines). In the middle picture, the ghost triangulates the distance from each of these test tiles to its target tile. Whichever direction's test tile has the shortest distance to the target becomes the direction the ghost will take upon reaching the intersection tile. In this case, the right test tile has the shortest distance to the target, and the ghost updates its chosen direction for the intersection tile accordingly.

Sometimes a ghost is presented with two or more test tiles that have the same distance to the target tile. In the example below, the red ghost must choose between moving down or left at the upcoming intersection tile. Unfortunately, both test tiles have the same distance to the target (bottom left). To break the tie, the ghost prefers directions in this order: *up, left, down, right*. Up is the most preferred direction; right is the least. Therefore, the ghost chooses to go left at the intersection because left precedes down in the preference list. Although it may seem obvious to a person that going down was the better choice to reach the target, ghosts are not that smart. They cannot see more than a few tiles ahead and, as a consequence, cannot recognize the disparity between these two options.

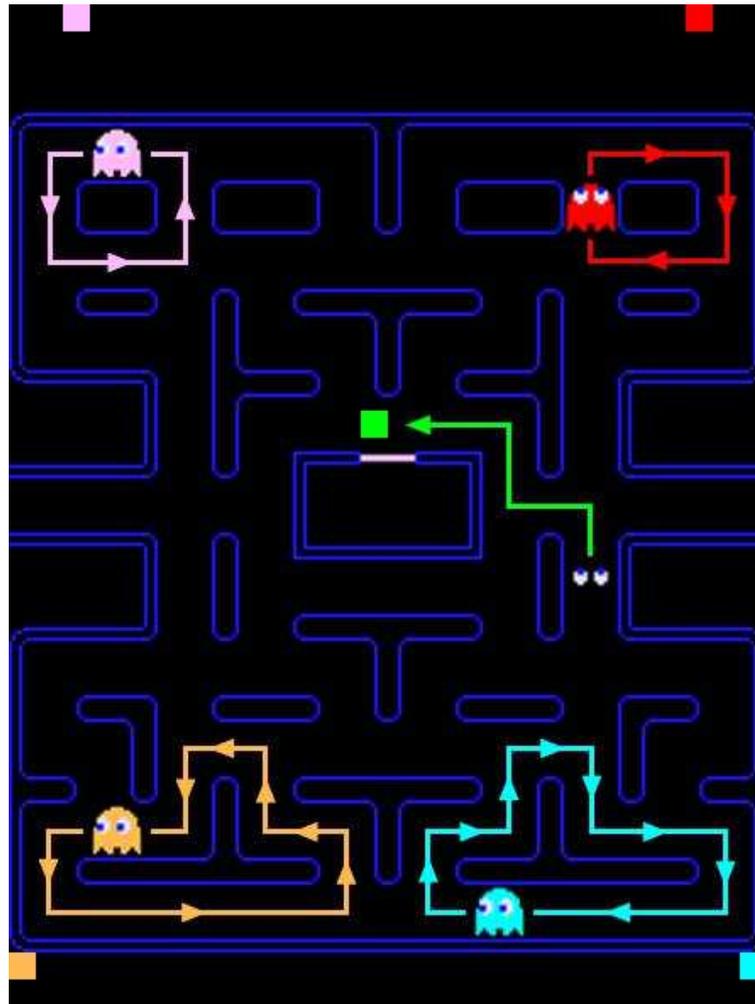


[Back](#)

Fixed Target Tiles

Each ghost has a fixed target tile it tries to reach while in scatter mode. The picture below shows the physical location of the scatter mode targets used by each ghost (matched to each ghost's color scheme). Notice each target tile is in dead space above or below the actual maze making them impossible for the ghosts to reach. This results in each ghost heading toward the corner of the maze nearest its respective scatter target and then making circles around this area until another mode change occurs. That's all scatter mode really is. The only reason a ghost has a "favorite corner" of the maze at all is due to the location of a fixed target tile it will never reach.

An additional fixed target tile is employed whenever a ghost is eaten by Pac-Man and its disembodied eyes need to return to the ghost house in the center of the maze. This target is located directly above the left side of the "door" to the ghost house and is shown in the picture below as the green tile.



[Back](#)

CHAPTER 4:

 **Meet The Ghosts** 

“First, you’ve got to learn how to control the monsters. See how the red, pink and blue are grouped together? It’s easier to control two monsters than four monsters.”—Chris Ayra, champion Pac-Man player

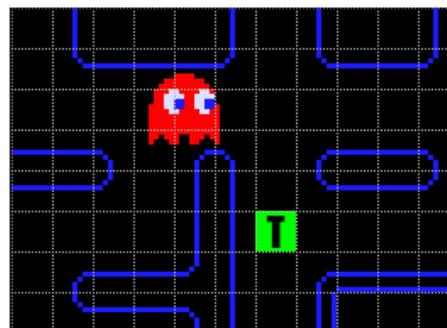
In the last chapter, we learned how a ghost follows a target tile through the maze. Now we will take a closer look at Blinky, Pinky, Inky, and Clyde to better understand why they behave so differently when in chase mode. They all share the same pathfinding logic for chasing a target tile, so how is it each one behaves differently when following Pac-Man? The answer is delightfully simple: *Pac-Man’s tile is not always the target*. Every ghost has a distinct method for calculating its target tile in chase mode, resulting in their unique personalities. Some of the ghosts use Pac-Man’s actual tile as the target; other only use it as an intermediate step to find another tile. Sometimes a ghost is targeting a tile that has absolutely nothing to do with Pac-Man at all! Regardless of where a ghost’s target tile is at the time, Pac-Man will still be killed if he gets in that ghost’s way.

Rumor has it Toru Iwatani and his team spent months doing nothing but tweaking and refining the ghost A.I. routines before releasing Pac-Man to the world. Their efforts show in the final product: Iwatani’s team created the illusion of complex pathfinding by using very simple logic and very little code.

[Back](#)



Blinky: The red ghost’s character is aptly described as that of a *shadow* and is best-known as “Blinky”. In Japan, his character is represented by the word *oikake*, which means “to run down or pursue”. Blinky seems to always be the first of the ghosts to track Pac-Man down in the maze. He is by far the most aggressive of the four and will doggedly pursue Pac-Man once behind him.



Of all the ghosts' targeting schemes for chase mode, Blinky's is the most simple and direct, using Pac-Man's current tile as his target. In the pictures above, we can see Blinky's target tile is the same as Pac-Man's currently occupied tile. Targeting Pac-Man directly in this way results in a very determined and tenacious ghost who is tough to shake when he's right behind you.

All ghosts move at the same rate of speed when a level begins, but Blinky will increase his rate of speed twice each round based on the number of dots remaining in the maze (if Pac-Man dies this is not necessarily true - more on this in a moment). While in this accelerated state, Blinky is commonly called "*Cruise Elroy*", yet no one seems to know where this custom was originated or what it means. On the first level, for example, Blinky becomes Elroy when there are 20 dots remaining in the maze, accelerating to be at least as fast as Pac-Man. More importantly, his scatter mode behavior is also modified at this time to keep targeting Pac-Man's current tile in lieu of his typical fixed target tile for any remaining scatter periods in the level (he will still reverse direction when entering/exiting a scatter period). This results in Elroy continuing to chase Pac-Man while the other three ghosts head for their corners as normal. As if that weren't bad enough, when only 10 dots remain, Elroy speeds up *again* to the point where he is now moving faster than Pac-Man. As the levels progress, Blinky will turn into Elroy with more dots remaining in the maze than in previous rounds. Refer to [Table A.1](#) in the appendices for dot counts and speeds for both Elroy changes, per level.

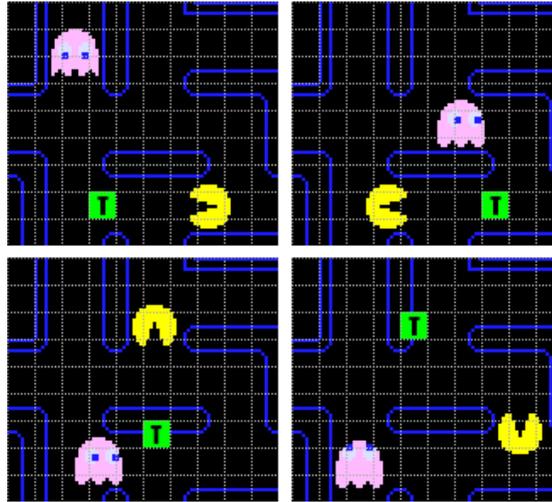
Determining when Blinky turns into Elroy can become more complicated if Pac-Man is killed. The ghosts and Pac-Man are reset to their starting positions whenever a life is lost and, when play continues, Blinky's "*Cruise Elroy*" abilities are temporarily suspended until the orange ghost (Clyde) stops bouncing up and down inside the ghost house and moves toward the door to exit. Until this happens, Blinky's speed and scatter behavior will remain normal regardless of the number of dots remaining in the maze. Once this temporary restriction is lifted, however, Blinky will resume changing his behavior based on the dot count.

[Back](#)

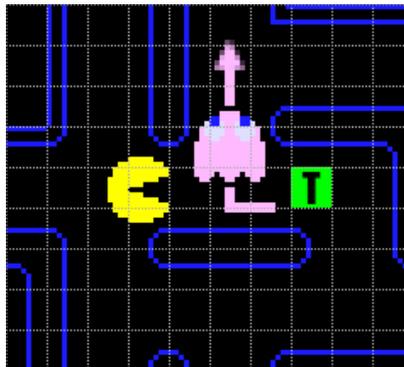


Pinky: Nicknamed "Pinky", the pink ghost's character is described as one who is *speedy*. In Japan, he is characterized as *machibuse*, meaning "*to perform an ambush*", perhaps because Pinky always seems to be able to get ahead of you and cut you off when you least expect it. He always moves at the same speed as Inky and Clyde, however, which suggests *speedy* is a poor translation of the more appropriate *machibuse*. Pinky and Blinky often seem to be working in concert to box Pac-Man in, leaving him with nowhere to run.

In chase mode, Pinky behaves as he does because he does not target Pac-Man's tile directly. Instead, he selects an offset four tiles away from Pac-Man in the direction Pac-Man is currently moving (with one exception). The pictures below illustrate the four possible offsets Pinky will use to determine his target tile based on Pac-Man's orientation:



If Pac-Man is moving left, Pinky's target tile will be four game tiles to the left of Pac-Man's current tile. If Pac-Man is moving right, Pinky's tile will be four tiles to the right. If Pac-Man is moving down, Pinky's target is four tiles below. Finally, if Pac-Man is moving up, Pinky's target tile will be four tiles up **and** four tiles to the left. This interesting outcome is due to a subtle error in the logic code that calculates Pinky's offset from Pac-Man. This piece of code works properly for the other three cases but, when Pac-Man is moving upwards, triggers an overflow bug that mistakenly includes a left offset equal in distance to the expected up offset (we will see this same issue in Inky's logic later). Don Hodges' website has an excellent article giving a thorough, code-level analysis of this bug, including the source code and a proposed fix—click [here](#) to go there now.

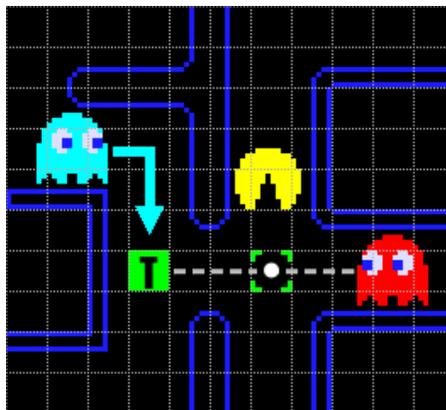


Pinky is the easiest ghost to exert control over thanks to his targeting scheme. By changing direction, you can dictate where Pinky will turn next when he is nearby (see above picture). If you are facing off closely with Pinky, he will turn before he reaches you if he can. This happens due to the fact Pac-Man has come close enough to Pinky for Pinky's target tile to now be behind him. In the picture above, Pinky chooses to turn up at the intersection because moving left would have taken him further away from his target tile. The longest-lived example of this is the technique known as "head faking". This is where the player shakes the joystick to cause Pac-Man to rapidly change direction back and forth, hopefully causing a ghost to change course in the process. As it turns out, the shaking is not necessary—one well-timed, quick reversal of direction towards Pinky just before he decides what to do at an upcoming intersection all that is needed to get him off your tail.

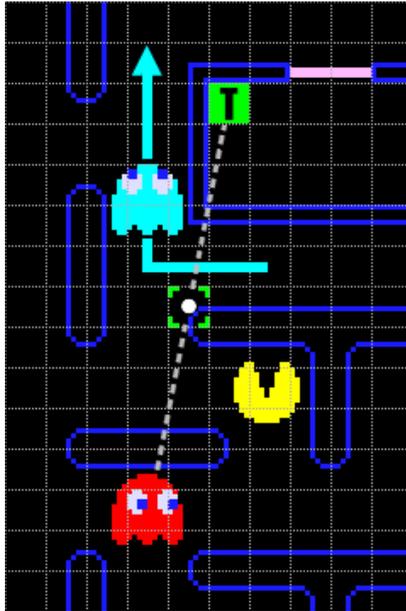
[Back](#)



Inky: The light-blue ghost is nicknamed "Inky" and his character is described as one who is *bashful*. In Japan, he is portrayed as *kimagure*, meaning "a fickle, moody, or uneven temper". Perhaps not surprisingly, Inky is the least predictable of the ghosts. Sometimes he chases Pac-Man aggressively like Blinky; other times he jumps ahead of Pac-Man as Pinky would. He might even wander off like Clyde on occasion! In fact, Inky may be the most dangerous ghost of all due to his erratic behavior. *Bashful* is not a very good translation of *kimagure*, and misleads the player to assume Inky will shy away from Pac-Man when he gets close which is not always the case.



Inky uses the most complex targeting scheme of the four ghosts in chase mode. He needs Pac-Man's current tile/orientation **and** Blinky's current tile to calculate his final target. To determine Inky's target, we must first establish an intermediate offset two tiles in front of Pac-Man in the direction he is moving (represented by the tile bracketed in green above). Now imagine drawing a vector from the center of the red ghost's current tile to the center of the offset tile, then double the vector length by extending it out just as far again beyond the offset tile. The tile this new, extended vector points to is Inky's actual target as shown above.



For the same reasons already discussed in Pinky's case, an overflow error occurs with the intermediate offset tile generated for Inky's calculation when Pac-Man is moving up resulting in an offset tile that is two tiles above **and** two tiles to the left (see above picture). The other three orientations (left, right, down) produce the expected result of an offset two tiles in front of Pac-Man in the direction he is moving.

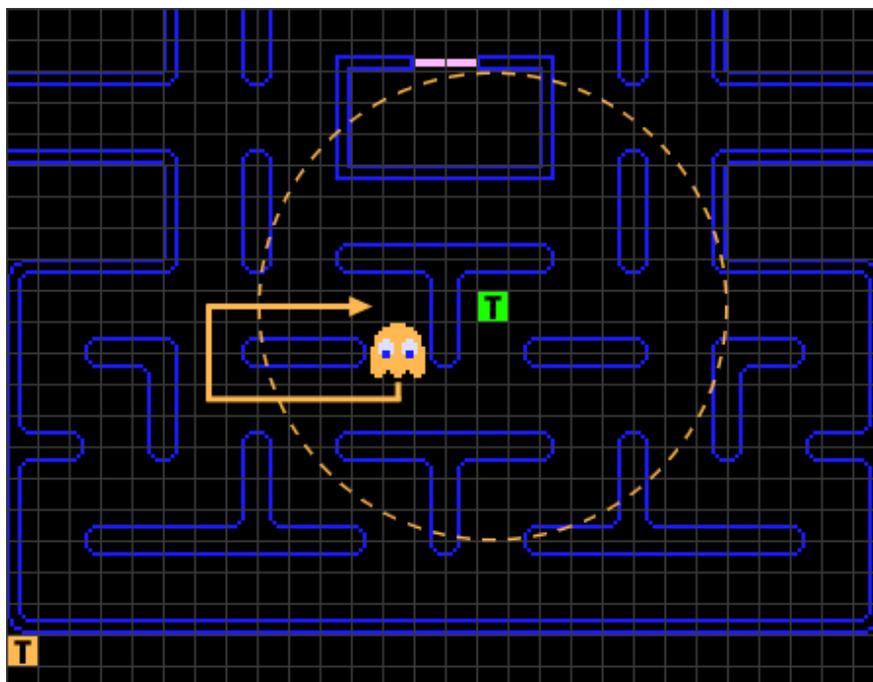
Inky's targeting logic will keep him away from Pac-Man when Blinky is far away from Pac-Man, but as Blinky draws closer, so will Inky's target tile. This explains why Inky's behavior seems more variable as Pac-Man moves away from Blinky. Like Pinky, Inky's course can often be altered by Pac-Man changing direction or "head-faking". How much or how little effect this will have on Inky's decisions is directly related to where Blinky is at the time.

[Back](#)



Clyde: The orange ghost is nicknamed "Clyde" and is characterized as one who is *poke*. In Japan, his character is described as *otoboke*, meaning "pretending ignorance", and his nickname is "Guzuta", meaning "one who lags behind". In reality, Clyde moves at the same speed as Inky and Pinky so his character description is a bit misleading. Clyde is the last ghost to leave the pen and tends to separate himself from the other ghosts by shying away from Pac-Man and doing his own thing when he isn't patrolling his corner of the maze.

Although not nearly as dangerous as the other three ghosts, his behavior can seem unpredictable at times and should still be considered a threat.



During chase mode, Clyde's targeting logic changes based on his proximity to Pac-Man (represented by the green target tile above). He first calculates the Euclidean distance between his tile and Pac-Man's tile. If the distance between them is eight tiles or more, Clyde targets Pac-Man directly just as Blinky does. If the distance between them is less than eight tiles, however, Clyde switches his target to the tile he normally uses during scatter mode and heads for his corner until he gets far enough away to start targeting Pac-Man again. In the picture above, Clyde is stuck in an endless loop (as long as Pac-Man stays where he is) thanks to this scheme. While occupying any tile completely outside the dashed perimeter, Clyde's target is Pac-Man. Upon entering the area, Clyde changes his mind and heads for his scatter target instead. Once he exits the perimeter, his target will change *back* to Pac-Man's current tile again. The end result is Clyde circling around and around until Pac-Man moves elsewhere or a mode change occurs. Clyde is fairly easy to avoid once you understand his targeting scheme. Just remember: he is still dangerous if you manage to get in his way as he runs back to his corner or before he can reach an intersection to turn away from you.

[Back](#)



CHAPTER 5:

Deciphering The Split Screen



“This is the way the world ends

This is the way the world ends

This is the way the world ends

Not with a bang but with a whimper.”

—T. S. Eliot

On The Edge Of Forever

Pac-Man was always meant to be a game with no ending. The developers at Namco mistakenly assume the game's increasing difficulty was sufficient to prevent anyone from playing indefinitely. Of course, within a few years of Pac-Man's release, players had discovered that every level beyond the 21st was identical. Patterns were quickly created to exploit this fact and, for any player able to get past the first levels, the game now became a test of endurance to see how many points you could rack up before losing focus and making a mistake. High scores soared into the millions and most players agreed the game simply went on forever. Eventually, a few highly-skilled players were able to complete 255 consecutive levels of play (scoring over three million points and taking several hours to accomplish) and found a surprise waiting for them on level 256. It was a surprise no one knew about—not even the developers at Namco.



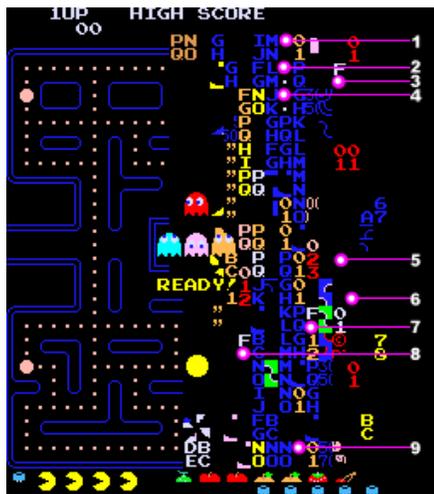
The 256th level displays the left half of the maze correctly, but the right half is a jumbled mess of randomly colored letters, numbers, and symbol. Notice the bonus counter in the lower-right of the screen is also malfunctioning. The left side of the maze plays normally, but the right side is a different story. Although both the player and the ghosts can navigate through the right half of the screen, the original maze walls no longer apply. Instead, Pac-Man must be guided through a confusing series of open areas, tunnels, one-way intersections, lone walls, and pass-throughs—all *invisible* to the player—while four ghosts are in hot pursuit.

Why does this broken level happen in the first place? The culprit is the routine responsible for drawing the bonus symbols along the bottom edge of the screen. Here's what happens: when level 256 is reached, the internal level counter is incremented to 255 (the level counter starts at zero – not one) and the routine for drawing the bonus symbols is called.

The routine loads the current level counter value (255) into a CPU register and increments that register by one. Unfortunately, 255 is the largest number that can fit in a single byte which is the size of the Z-80 CPU registers, so when the value is incremented the overflow is discarded leaving a zero in the register instead of the expected value of 256. This zero value leads the routine to believe this is an early level since its value is less than seven. The routine starts drawing bonus symbols using the confused register as a counter. At the end of every drawing loop, the register is decreased by one and *then* checked to see if it is zero (the signal for the routine to stop drawing symbols). Since the register already has a zero in to start, the first decrement will roll the value back to 255. It will keep decrementing the register and drawing symbols until the register is reduced to zero again, causing the loop to run a total of 256 times. This means that memory locations outside the bounds of the bonus symbol table are drawn to the screen at increasing locations in video memory. This half-broken level was named the "split screen" by players and developers refer to it as a "kill screen". Don Hodges' website has an in-depth article analyzing this bug that includes source code and a proposed fix—click [here](#) to go there now.

[Back](#)

Playing The Level

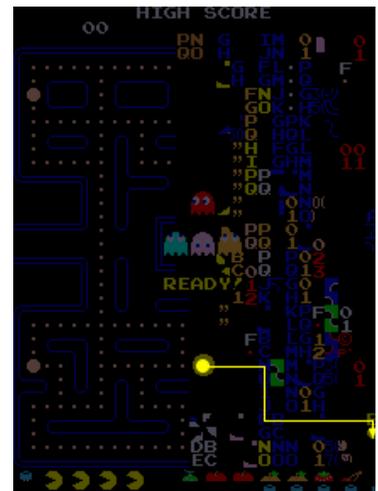


There are 114 dots on the left half of the screen, nine dots on the right, and one bonus key, totaling 6,310 points. When all of the dots have been cleared, nothing happens. The game does not consider a level to be completed until 244 dots have been eaten, so there is nothing left to do but sacrifice Pac-Man to a hungry ghost. Interestingly, every time a life is lost, the nine dots on the right half of the screen get reset and can be eaten again, resulting in an additional 90 points per extra man. In the best-case scenario (five extra men), 6,760 points is the maximum score possible, but only 168 dots can be harvested—not enough to change levels—so we are stuck. There are more dots to gobble or energizers to eat. There is no final victory waiting for Pac-Man, only an empty half-maze full of ghosts. The game has an ending after all—just not a very happy or exciting one.

Four of the nine dots on the right half of the screen are invisible, but can be heard when eaten. The picture on the left shows all nine dot locations. Dots 1, 5, 6, and 9 are invisible; the rest can be seen but some are a different color than normal.

Anyone reaching this level quickly realized: to safely map out the right side of the screen something had to be done about the ghosts. After much tinkering, it was discovered that a ghost would get “trapped” at the right edge of the screen if he got too close to it. Once trapped, a ghost can only move up or down but never right or left again. By leading ghosts near the edge of the screen, a skilled player could eventually get the ghosts out of the way and concentrate on exploring the right half of the maze and collecting the dots.

There are many methods for trapping the ghosts. One of the easiest ways to trap the three important ghosts is shown in the picture to the right. The yellow line shows Pac-Man's path from the start of the level to a spot near the bottom-right. The exact instructions are as follows: begin by going right until you reach a blue letter 'N', then go down. Keep going down until you reach a blue letter 'F', then go right. Keep going right until you reach a yellow 'B', then go down again. When executed properly, Pac-Man will hit an invisible wall almost immediately after the last turn is made. Now we wait. The red ghost will get stuck first. The pink ghost follows a few seconds later. The blue ghost will continue to move freely for several moments until the next scatter mode occurs. At that point, it will try to reach some location near the right edge of the screen and get stuck with the pink and red ghost instead. Now the orange ghost is the only one still on the loose (bottom-right). Clyde is no real threat, however, since he runs to his corner whenever Pac-Man gets close (see Chapter 4), making it relatively easy to clean up all the dots. Be sure to take care around the lower-left corner of the maze—the orange ghost will have nowhere left to run to and will be much more aggressive.



Click on the YouTube video below to watch this ghost-trapping method in action:

[Back](#)

Believe It Or Not

Some versions of the *Pac-Man* ROMs have a “rack test” feature, allowing the cabinet owner to skip ahead to the next level of play whenever they want. To date, the only known way to legitimately get past level 256 is by using the rack test switch inside these machines. The result is that the game loops back around to the first board, but with the score intact and the ghosts still behaving as though it were level 21 or above. Many of the *Pac-Man* ROMs available for use with the MAME emulator also have this rack test feature, making it relatively easy for those without an arcade version handy to quickly get to the split-screen and beyond.

For decades, *Pac-Man* enthusiasts worldwide have heard the whispers about a “secret trick” allowing a player to get past level 256 and continue playing without using the aforementioned rack test. Several players have boasted having acquired this holy grail of *Pac-Man* knowledge over the years, but no one has been able to make good on their claims by actually proving it. This topic became so hotly debated in the upper echelons of the arcade gaming community that Billy Mitchell—who was convinced it was impossible—offered a \$100,000 cash prize to the first player to prove they could legitimately get past level 256, leaving the challenge open for a full year. The prize money went unclaimed.

In spite of the evidence against there being a way to get past level 256, rumors still persist and can occasionally be found in classic gaming forums online, yet no one has been able to back up their words with indisputable proof. It's hard to imagine why anyone who could legitimately get past the level did not collect Mr. Mitchell's prize money to be sure. Still the occasional whispers can be heard. Perhaps it is simply natural for people to want to believe in the possibility as opposed to not—like Santa Claus or the Easter Bunny. Then again, maybe there is some middle-aged *Pac-Man* junkie out there who is withholding secrets to a 30 year-old amusement device for his or her own unfathomable reasons. Stranger things have happened. You be the judge.

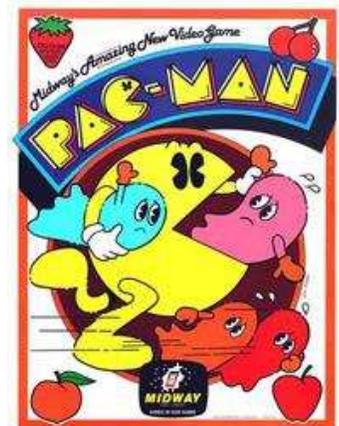
[Back](#)



References and Further Reading



- [Splitting Apart The Split Screen](#), Don Hodges
- [Pac-Man's Ghost Behavior Analyzed And Fixed](#), Don Hodges
- [Pac-Man Hardware And Memory Layout Info](#), Alessandro Scotti
- [Disassembly And Analysis Of Pac-Man Sound Code](#), Frederic Vecoven
- [The Virtual Pac-Man Museum](#), Chuck and Vicki Gill
- [Pacman: The Phenomenon – Part 1](#), Marty Goldberg
- [The History of Pac-Man](#), Doug Trueman
- [Pac-Man Entry At Arcade History](#), Alexis Bousiges
- [Google's Pac-Man 30th Anniversary Doodle](#), Ryan Germick and Marcin Wichary
- [Understanding Pac-Man Ghost Behavior](#), Chad Birch



- [Reawakening The Sleeping Giant: The Pac-Man CE Interview](#), Christian Nutt

[Back](#)



APPENDIX A:

Reference Tables



Table A.1 — Level Specifications

Level	Bonus Symbol	Bonus Points	Pac-Man Speed	Pac-Man Dots Speed	Ghost Speed	Ghost Tunnel Speed	Elroy 1 Dots Left	Elroy 1 Speed	Elroy 2 Dots Left	Elroy 2 Speed	Fright. Pac-Man Speed	Fright Pac-Man Dots Speed	Fright Ghost Speed
1	Cherries	100	80%	~71%	75%	40%	20	80%	10	85%	90%	~79%	50%
2	Strawberry	300	90%	~79%	85%	45%	30	90%	15	95%	95%	~83%	55%
3	Peach	500	90%	~79%	85%	45%	40	90%	20	95%	95%	~83%	55%
4	Peach	500	90%	~79%	85%	45%	40	90%	20	95%	95%	~83%	55%
5	Apple	700	100%	~87%	95%	50%	40	100%	20	105%	100%	~87%	60%
6	Apple	700	100%	~87%	95%	50%	50	100%	25	105%	100%	~87%	60%
7	Grapes	1000	100%	~87%	95%	50%	50	100%	25	105%	100%	~87%	60%
8	Grapes	1000	100%	~87%	95%	50%	50	100%	25	105%	100%	~87%	60%
9	Galaxian	2000	100%	~87%	95%	50%	60	100%	30	105%	100%	~87%	60%
10	Galaxian	2000	100%	~87%	95%	50%	60	100%	30	105%	100%	~87%	60%
11	Bell	3000	100%	~87%	95%	50%	60	100%	30	105%	100%	~87%	60%
12	Bell	3000	100%	~87%	95%	50%	80	100%	40	105%	100%	~87%	60%
13	Key	5000	100%	~87%	95%	50%	80	100%	40	105%	100%	~87%	60%
14	Key	5000	100%	~87%	95%	50%	80	100%	40	105%	100%	~87%	60%
15	Key	5000	100%	~87%	95%	50%	100	100%	50	105%	100%	~87%	60%
16	Key	5000	100%	~87%	95%	50%	100	100%	50	105%	100%	~87%	60%
17	Key	5000	100%	~87%	95%	50%	100	100%	50	105%	—	—	—
18	Key	5000	100%	~87%	95%	50%	100	100%	50	105%	100%	~87%	60%
19	Key	5000	100%	~87%	95%	50%	120	100%	60	105%	—	—	—
20	Key	5000	100%	~87%	95%	50%	120	100%	60	105%	—	—	—
21+	Key	5000	90%	~79%	95%	50%	120	100%	60	105%	—	—	—

[Back](#)



Table A.2 — Difficulty Specifications

There is a small spot on the Pac-Man PCB where you can solder two pads together to set the game to "hard" difficulty. The only difference in hard difficulty is that five of the levels (1, 3, 6, 19, and 20) are eliminated from play. The system does *not* eliminate any of the bonus symbols, however, causing much confusion as to what level you're really playing. Level two is the first board in hard difficulty for example but the cherry symbol is used instead of the strawberry. It's still level two in terms of gameplay, but with cherry symbols in place of the usual strawberries. Also, the bonus point values are changed to match the current symbol being used. You can determine the difficulty setting of a machine by observing which ghost kills Pac-Man during the attract mode demo game. In normal difficulty, Pac-Man gets captured by Inky in the lower-left area of the maze. If the difficulty jumper has been connected, however, he is captured by Clyde near the same location.

Normal	Normal Bonus	Hard	Hard Bonus
1	Cherries	—	—
2	Strawberry	2	Cherries
3	Peach 1	—	—
4	Peach 2	4	Strawberry
5	Apple 1	5	Peach 1
6	Apple 2	—	—
7	Grapes 1	7	Peach 2
8	Grapes 2	8	Apple 1
9	Galaxian 1	9	Apple 2
10	Galaxian 2	10	Grapes 1
11	Bell 1	11	Grapes 2
12	Bell 2	12	Galaxian 1
13	Key 1	13	Galaxian 2
14	Key 2	14	Bell 1
15	Key 3	15	Bell 2
16	Key 4	16	Key 1
17	Key 5	17	Key 2
18	Key 6	18	Key 3
19	Key 7	—	—
20	Key 8	—	—
21+	Key 9	21+	Key 4

[Back](#)





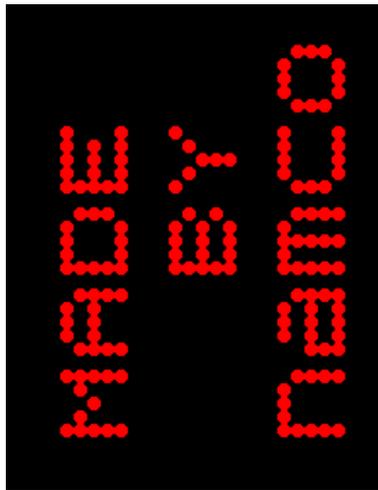
APPENDIX B:

Easter Eggs & Tricks



NAMCO Easter Egg

There is a secret message hidden in *Pac-Man* by the developers at Namco. To see it, put the machine in service mode and wait for the settings screen to appear. Now quickly toggle service mode off and on (a alignment grid will appear on the screen). While holding down the player 1 and player 2 buttons, toggle service mode off and on again very quickly (if done properly, the grid will stay on the screen), and then push the joystick in the following directions: UP x 4, LEFT x 4, RIGHT x 4, DOWN x 4. The message "MADE BY NAMCO" will appear sideways on the screen, spelled out using energizers.



Ghost Bug

A curious bug can be triggered when Pac-Man eats a ghost just as it is leaving the monster pen (the orange ghost is the easiest one to catch this way). This may take a few attempts as the timing is crucial—try to capture it just as the lower edge of the ghost is passing through the door on top of the pen—but when done correctly, the eyes will return directly to the pen and the regenerated ghost will immediately start moving straight up and **through the walls of the maze!** As the ghost nears the top of the screen, it wraps around reappearing at the bottom edge of the board and continues upwards ignoring the maze walls. The ghost eventually makes it back into the monster pen at which point it resumes normal behavior. If Pac-Man eats another energizer and manages to capture the ghost before this happens, however, the ghost's eyes will move around but slightly out-of-sync with the maze and have difficulty returning to the pen. Sometimes the eyes will eventually correct themselves and find the way back home (restoring the ghost back to normal) but sometimes they remain lost until the level is completed.

Click on the YouTube video below for a demonstration of this strange behavior:

[Back](#)



APPENDIX C:



Hardware Information



Specifications:

- Platform — *NAMCO 8-bit PCB*
- CPU — *Z80A at 3.072 MHz*
- ROM — *16K in four, 4K chips*

- RAM — *Almost 2K*
- Display — *Raster*
- Orientation — *Vertical*
- Resolution — *224x288*
- Colors — *16*
- Attributes — *Six 16x16 hardware sprites*
- Refresh rate — *60.606061 Hz*
- Sound — *Custom monophonic 3-voice waveform sound generator chip*
- Controls — *One 4-way leaf joystick, 1P/2P buttons*
- Models — *Upright, Mini, and Cocktail*

Midway Operator's Reference Books:

- [Pac-Man Operator's Manual](#)
- [Troubleshooting Logic Board Part I for Pac-Man and Ms. Pac-Man](#)

[Back](#)



APPENDIX D:

Vintage Pac-Man Guides



- [Break A Million At Pac-Man](#) — Ernest Zavisca, Ph.D. and Gary Beltowski.
- [How To Win At Pac-Man](#) — Penguin Books.

[Back](#)



Glossary



- apple:** The bonus symbol for levels five and six. Worth 700 points.
- bell:** The bonus symbol for levels eleven and twelve. Worth 3,000 points.
- Blinky:** The red ghost. Also known as "Akabei" or "Macky" in *Puck-Man*.
- bonus symbol:** The often fruit-related symbol that appears twice per level below the ghost house and can be eaten for additional scoring. The point-value depends on the specific symbol and can range anywhere from 100 to 5,000 points each. Also known as *fruit*.
- cherries:** The bonus symbol for the first round of play. Worth 100 points.
- Clyde:** The orange ghost. Also known as "Guzuta" or "Mocky" in *Puck-Man*.
- cornering:** The technique of moving the joystick in the direction one wishes to go well before reaching the center of a turn, ensuring Pac-Man will take the turn as quickly as possible.
- Cruise Elroy:** When a certain number of dots are all that remain in a level, Blinky (red ghost) will change "gears", speeding up as well as chasing Pac-Man even in scatter mode. He speeds up yet again when half the dots remain from the first change,
- dots:** The 244 objects in the maze Pac-Man must eat to move on to the next round. There are 240 small dots worth 10 points each, and 4 energizer dots worth 50 points each. Also known as *pills*.
- energizer:** One of four, large, flashing dots located near the corners of the maze worth 50 points each. When Pac-Man eats an energizer, the ghosts simultaneously reverse direction and on earlier levels, turn dark blue. Pac-Man can then eat the blue ghosts for additional points, scoring more for each consecutive ghost eaten from one energizer: 200, 400, 800, and 1,600 points respectively. Also known as a *power pills*, *fuel tanks*, and *vitamins*.
- flipping:** "Flipping the machine" refers to when a player earns one million points. The game is unable to display a score larger than 999,999, so the score readout "flips" over to zero and keeps counting. Also known as rolling the machine.
- fruit:** See [bonus symbol](#).
- galaxian:** The bonus symbol for levels nine and ten. Also known as a *tulip* or a *thunderbird*. Worth 2,000 points.
- ghost house:** The rectangular area in the middle of the maze where the ghosts start each new level and new life, returning to the house whenever they are captured by Pac-Man. Also known as the *monster pen*.
- ghosts:** Pac-Man's four enemies in the maze are typically referred to as *ghosts* or *monsters*.
- grapes:** The bonus symbol for levels seven and eight. Also known as a *melon* or a *grenade*. Worth 1,000 points.
- grenade:** See [grapes](#).
- head faking:** Changing Pac-Man's direction back and forth in quick succession in an attempt to affect the turning logic of one or more ghosts in play. Blinky and Clyde do not use Pac-Man's current direction in their chase logic, so they are unaffected by head faking.

Inky: The blue ghost. Also known as "Aosuke" or "Mucky" in *Puck-Man*.

intersection: Anywhere pathways in the maze intersect, yielding more than one option on which way to proceed.

key: The bonus symbol for levels 13 and above. Worth 5,000 points.

melon: See [grapes](#).

monster pen: See [ghost house](#).

monsters: See [ghosts](#).

orange: See [peach](#).

pattern: A memorized series of turns associated with a particular level or levels that, when repeated, clears the maze of dots without getting Pac-Man captured by any of the ghosts. Also known as a *routine*.

peach: The bonus symbol for levels three and four. Also known as an *orange*. Worth 500 points.

pills: See [dots](#).

Pinky: The pink ghost. Also known as "Micky" in *Puck-Man*.

power pill: See [energizer](#).

routine: See [pattern](#).

side tunnel: The connecting tunnel between the right and left edges of the screen. Entering this tunnel will "wrap" the player around to the other side of the screen. The monsters always suffer a speed penalty while in the tunnel while Pac-Man does not. Also known as *The Tube*, *The Time Warp*, and *The Scoot*.

split screen: The 256th level of the game, where the right half of the screen is filled with garbage instead of the usual maze.

strawberry: The bonus symbol for level two. Worth 300 points.

thunderbird: See [galaxian](#).

tulip: See [galaxian](#).

[Back](#)



Frequently Asked Questions



Q: How do I get past the split screen?

A: The only known way to get past the split screen is via the "rack test" feature available on some Pac-Man ROMs (see [Chapter 5](#)).

Q: What is a "perfect score"?

A: A perfect score in *Pac-Man* is 3,333,360 points, which can only be attained by playing a perfect game. This requires catching all four ghosts at every energizer, gobbling down every bonus fruit, a never once losing a life for 256 consecutive levels of play. All extra lives are needed once the split screen is reached to eat the nine dots hidden on the right side of the screen the maximum number of times—they respawn every time a life is lost. This was first achieved by Billy Mitchell of Hollywood, Florida in 1999.

Q: Is it true that some of the ghost A.I. routines examine the joystick directly to make decisions?

A: This is false. The memory-mapped IN0 joystick port is completely removed from the pathfinding and logic routines in the code.

Q: What other games from the *Pac-Man* family will this ghost logic work with?

A: *Pac-Man Plus* and *Ms. Pac-Man* both use the same basic pathfinding/targeting logic as the original *Pac-Man*. Many popular bootleg ROMs like the Atlantic City Chip and Hanglyman also use this logic.

Q: Where can I download a copy of the *Pac-Man* ROMs?

A: The *Pac-Man* ROMs are under copyright to NAMCO Bandai, LTD and are their intellectual property. I cannot legally provide you with them, and I certainly cannot recommend that you go to [Google](#) right away and search for "pacman puckman ROMs MAME32" because you would probably find them after a little searching.

Q: Why are all of the bonus symbols food-related except for the galaxian, the bell, and the key?

A: The galaxian was added as a nod to the Namco space-shooter title, *Galaxian*, which was under development at the same time as *Pac-Man*. No one knows why Toru Iwatani chose a bell and a key for the final two bonus symbols. It has been theorized the bell may actually be some sort of food like an Asian cashew or even a blancmange dessert, which would make it consistent with the food theme. The bell at least has the *possibility* of being something else—no one has any theories on the key being anything but ... well ... a key.

Q: Why are some members of the NAMCO development team listed by name but not others [Chapter 1](#)?

A: I have not been able to find out these persons' names—I'm sure they are documented somewhere but I have yet to find that information. If you know of any reliable sources for the names of the full NAMCO development team, please let me know (jamey.pittman@yahoo.com).

Q: Where can I download the MAME emulator?

A: Click [here](#) for the MAME website.

[Back](#)



About the Author



Jamey Pittman (jamey.pittman@yahoo.com) has been fascinated with video games ever since he saw his first *Pac-Man* machine in 1981. Eventually, his interests in computers and video games led to a successful career programming various PC game titles through the 1990s including *Cybermage*, *Nox*, and *Wizards Warriors*, developing the 3D engine technology for the latter. In his spare time, he enjoys playing *Pac-Man*, *Ms. Pac-Man*, *Donkey Kong*, and *Galaga* until his wrists hurt and his eyes are sore.

[Back](#)



If you enjoyed the information presented on this website, please consider contributing a small donation to support it and defray the time/maintenance costs associated with keeping it online and updated. Donations can be made **safely and securely** via PayPal by clicking the "Donate" button below. You can also make a small contribution (at no cost to you whatsoever) by simply clicking on one or more of the advertisement banners. Your support is greatly appreciated!



[Search Engine Optimization and SEO Tools](#)

 [link to](#)

search engine submission