Features

The Pac-Man Dossier

by Jamey Pittman [Design]

10 comments Share

February 23, 2009 Article Start Previous Page 7 of 10 Next

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."-Billy Mitchell, 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?

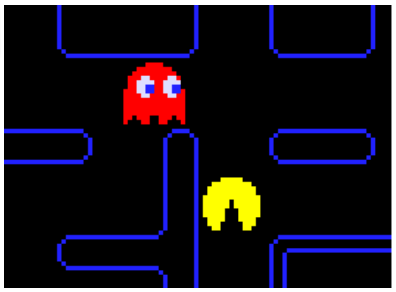
The answer is delightfully simple: *Pac-Man's tile is not always the target*. Every ghost has a distinct method for calculating its target tile in chase mode, resulting in their unique personalities. Some of the ghosts use Pac-Man's actual tile as the target; others only use it as an intermediate step to find another tile.

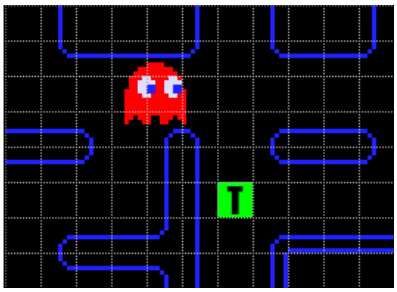
Sometimes a ghost is targeting a tile that has absolutely nothing to do with Pac-Man at all! Regardless of where a ghost's target tile is at the time, Pac-Man will still be killed if he gets in that ghost's way.

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: Itawani's team created the illusion of complex pathfinding by using very simple logic and very little code.

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. 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 to target Pac-Man's tile in lieu of his typical fixed target tile for any remaining scatter periods in the level.

This causes Elroy to chase Pac-Man while the other three ghosts continue to scatter 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 perceptibly faster than Pac-Man.

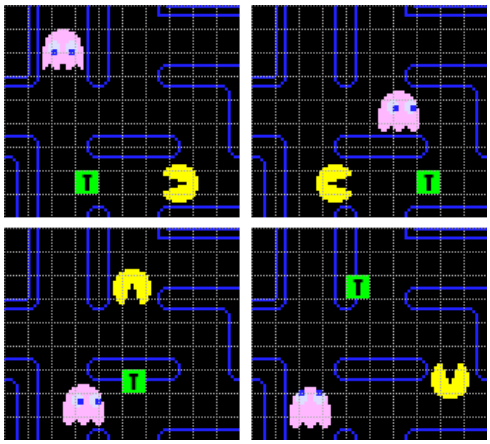
If a life is lost any time after Blinky has become Elroy, he will revert back to his normal behavior and speed when play resumes, heading for his home corner during the initial scatter period. But once the last ghost (Clyde) has left the ghost house in the middle of the board, he will turn back into Elroy again.

Keep in mind: he is *still* in scatter mode the entire time. All that has changed is the target tile-he will still reverse direction when entering and exiting scatter mode as before. 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.

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.

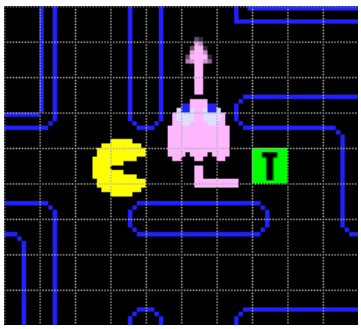
In chase mode, Pinky behaves as he does because he does not target Pac-Man's tile directly. Instead, he selects an offset four tiles away from Pac-Man in the direction Pac-Man is currently moving (with one exception). The pictures below illustrate the four possible offsets Pinky will use to determine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

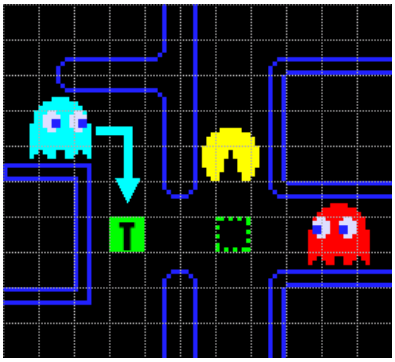
In the picture above, Pinky chooses to turn up at the intersection because moving left would have taken him further away from his target tile. The longest-lived example of this is the technique known as "head faking". This is where the player shakes the joystick to cause Pac-Man to rapidly change direction 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 is all that is needed to get him off your tail.

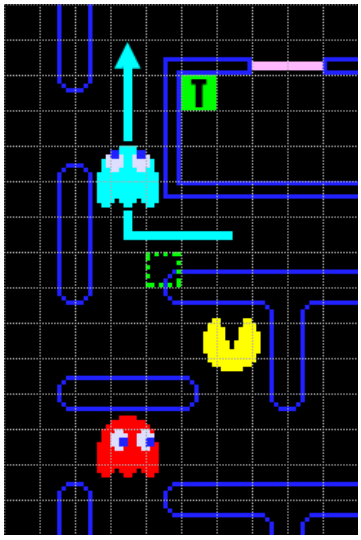
Inky: The light-blue ghost is nicknamed "Inky" and his character is described as one who is *bashful*. In Japan, he is portrayed as *kimagure*, meaning "*a fickle, moody, or uneven temper*". Perhaps not surprisingly, Inky is the least predictable of the ghosts.

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 envision Inky's target, imagine an intermediate offset two tiles away from Pac-Man's tile in the direction Pac-Man is moving (shown as the dashed, green tile in the picture above), then draw a line from Blinky's tile to that offset. Now double the line length by extending the line out just as far again, and you will have Inky's target tile as shown above.



For the same reasons already discussed in Pinky's case, Inky's offset calculation from Pac-Man is two tiles up **and** two tiles left when Pac-Man is moving up (shown above). The other three orientations have the expected offset of two tiles in the direction Pac-Man is moving.

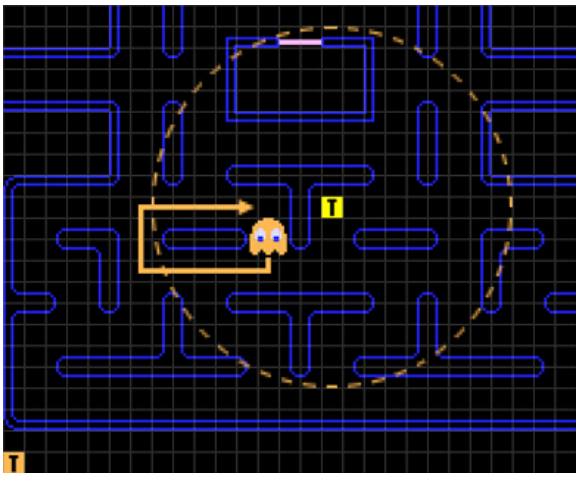
Inky's targeting logic will keep him away from Pac-Man when Blinky is far away from Pac-Man, but as Blinky draws closer, so will Inky's target tile. This explains why Inky's behavior seems more variable as Pac-Man moves away from Blinky. Like Pinky, Inky's course can often be altered by Pac-Man changing 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.

Clyde: The orange ghost is nicknamed "Clyde" and is characterized as one who is *pokey.* 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.



In chase mode, Clyde's target differs based on his proximity to Pac-Man. When more than eight tiles away, he uses Pac-Man's tile as his target (shown as the yellow target above). If Clyde is *closer* than eight tiles away, he switches to his scatter mode target instead, and starts heading for his corner until he is far enough away to target Pac-Man again.

In the picture above, Clyde is stuck in an endless loop thanks to his targeting scheme. Outside of the dashed area, Clyde acts exactly as Blinky would, heading straight for Pac-Man, but upon entering the dashed area, Clyde will change his mind and head for his scatter target instead.

Leaving the eight tile perimeter surrounding Pac-Man causes his target to change *back* to Pac-Man's tile and results in Clyde circling the island indefinitely until Pac-Man moves elsewhere or a mode change occurs.

Clyde's targeting method results in him not being particularly dangerous unless you get in his way as he runs back to his corner or before he can reach an intersection to turn away. Extra care should be taken when Pac-Man is in Clyde's home corner as Clyde is less likely to get out of the way.

Article Start Previous Page 7 of 10 Next

don't know what cloud gaming can offer

Comments

Jake Romigh 23 Feb 2009 at 9:26 am PST

This was a most informative and entertaining read. I'm not sure everyone will agree with me here, but I think these articles have promise. They show the development history, design principals, execution, cultural reaction, and legacy of a game. If the rest of your "Dossiers" are as indepth and quality as this one, I'll be sure to read them.

Login to Reply or Like

Mike Saunders 23 Feb 2009 at 1:59 pm PST

This was an amazing article! I really enjoyed reading it due to the clear descriptions and the great diagrams visually depicting the algorithms/etc. Of course, I'm an arcade junkie from the 80's so the fact that the article focused on one of those games was just icing on the cake.

Please, please do more of these types of in-depth technical analysis articles! I'll read every one.

Login to Reply or Like

Roberto Alfonso 23 Feb 2009 at 4:02 pm PST

Indeed, this is one of the most insightful articles I have ever read. I knew about the chase/scatter modes from an early interview, but didn't know the gameplay was so deep!

Login to Reply or Like

Joshua Dallman 24 Feb 2009 at 12:43 am PST

This is a simply fantastic article and I would love to see more well researched, in-depth game design and theory articles such as these here. There's a lot to sink your teeth into in this article, it isn't just armchair speculation about theory and market trends. Very well done, well researched, and an appreciated read and reference. I hope this type of article can also show newer game developers how much subtle coding behind the curtain occurs to create a well polished and deep gameplay, even if seemingly on the surface it is a "simple game" especially by graphical standards alone. I would also be remiss to not point out the recent "Pac Man Zero G" outlined on your sister site: http://www.indiegames.com/blog/2009/02/freeware_game_pick_pacman_phys.html

Login to Reply or Like

Tom Newman 24 Feb 2009 at 7:05 am PST

The best article on Pac-Man I have ever read. Answered many questions I've had since the 80's.

This proves that you can deliver a deep gameplay experience without lots of CPU, without high end graphics, and with minimal player control (up, down, left, right, no buttons). Peolple will still be playing Pac-Man 50 years from now.

Login to Reply or Like

Christopher Enderle 24 Feb 2009 at 1:33 pm PST

The game's depth is astounding, I'll never look at Pac-Man the same way again. Such in depth analysis as this makes me wish we could see the game's actual GDD, if such official documents exist. I'd imagine they do with the continual remakes that come out, but perhaps that's the same reason they still hold on to those.

Login to Reply or Like

John Leffingwell 24 Feb 2009 at 3:38 pm PST

Absolutely great article. The superlatives do not exist for how I feel about the author's coverage of this subject matter. I didn't know of the author until now, but I've been a follower of Don Hodges work for a while, and I do this sort of thing myself from time to time. I hope we can see similarly gritty technical articles about classic

games in the future. The only thing that could possibly improve this article is if Gamasutra could use some of its clout to get in contact with the original designers for additional insight and background information. I'd love to see something about the work of Vid Kidz. Robotron: 2084 has some nifty bugs, and some of the later technology Jarvis et al. developed was radically ahead of its time. See Halcyon Days.

Login to Reply or Like

Tom Newman 25 Feb 2009 at 7:03 am PST

I would love to see a dossier on Joust. Left, right, and a flap button - the rest of the gameplay built in to the physics of the game itself.

Login to Reply or Like

Matthew Oztalay 27 Feb 2009 at 11:16 pm PST

Very well written, very informative. An excellent window into the level of depth and detail required to design a game.

Login to Reply or Like

Shawn Yates 2 Mar 2009 at 3:04 pm PST

Excellent article. I'm such a sucker for the algorithm diagrams I cannot express how grateful I am. This really helps peel back the layers of complexity of a game that at first glance might seem "simple".

Login to Reply or Like

Login to Comment