



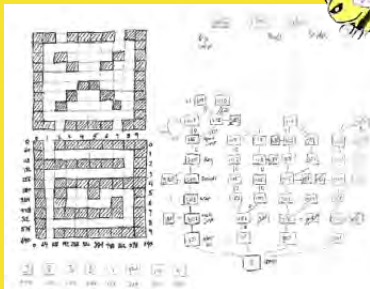
SING-SONG DING-DONG

DEVELOPER: GRAPEFRUITOPIA INDUSTRIES
PUBLISHED - UPDATED: 12-9-25
PRICE: \$6 ON CATALOG
[HTTPS://PLAY.DATE/GAMES/SING-SONG-DING-DONG/](https://play.date/games/sing-song-ding-dong/)
[HTTPS://GRAPEFRUITOPIA.ITCH.IO](https://grapefruitopia.itch.io)

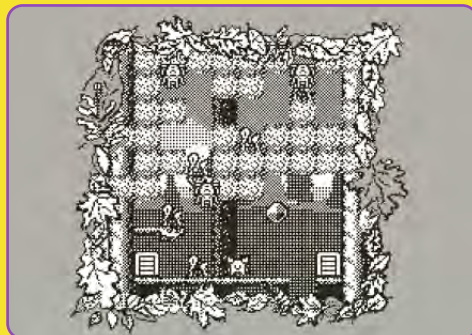


Contrary to popular belief, the artist's life isn't all sex, drugs, and all-night parties. Especially when your dreams of becoming a rock god crash headfirst into vocal cords that weren't exactly built for stadium anthems. Some societies handle this more gently than others (just think of how many times Assurancetourix, aka Cacophonix, the bard from that famously stubborn Gaulish village, was nearly booted into exile). But others? They'll exile their would-be rock star faster than you can say "autotune." That's the sad fate awaiting our protagonist in this adventure: a would-be singer creature now forced to undergo all sorts of trials and tribulations (oh, and fill his saddlebags with glittering loot) just for a shot at being welcomed back into village life with the rest of the civilized folk.

Sing Song Ding Dung is a classic-style arcade platformer that hits all the right notes, instantly bringing to mind titles like Bubble Bobble, The New Zealand Story, and a dozen others you're probably already picturing. If you can overlook the lack of co-op play for obvious reasons, everything else falls perfectly into place to turn your Playdate into a full-blown arcade machine with legit retro vibes. Our hero controls like a dream, movement is fluid and tight and the levels are smartly laid out, offering



GAME'S DESIGN DOCUMENTATION



a well-balanced mix of dexterity and exploration. The reward system delivers where it counts. At its core, it's all about taking down enemies (those who'll be fatally KO'd the moment they hear our sweet, murderous trills), pushing through each screen, and scooping up gems and keys. The latter unlock new areas across a sprawling map containing four trials (worlds) each with ten regular stages, plus a boss fight, then ten extra hidden bonus levels to hunt down once the main game is completed.

That said, we can't just sit here gargling in denial over the fact that the crank doesn't really have any gameplay mechanic to justify its existence, and that the character graphics aren't original, instead

borrowing from some well-known asset packs. Now, that last bit isn't necessarily a bad thing, but I can't help but feel the game truly deserves to match its top-tier technical chops with a set of custom-made, one-of-a-kind sprites. None of the accolades thrown at this game should surprise anyone who's been paying attention to Richard Munn's, the author, frankly impressive résumé: 'I got my first "proper" computer when I was 10 years old (a ZX Spectrum), and learnt to program from the bits of BASIC in the back of the manual, and the "type in" pages in magazines and Usbourne books. Later I did Computing at A level, and a degree in Computer Science at University. Some of my earliest stuff that made it into the wild was being tangentially involved in the Dreamcast VMU hacking scene - you can

still find references to a little 'Etch-a-sketch' toy I made for the VMU: VM-a-Sketch Beta v1.4 - Sega Visual Memory Unit quickloads - mame (0284) software. After University, I was lucky to land my first job in the Games Industry at Codemasters, partly due to my experience with the VMU (even though they dropped DC support soon after) and worked for a few years on the LMA Manager series of games. I later worked at Ubisoft Reflections on the Driver games, and at Sony in the Advanced Technology Group, which meant more working on tools, pipeline and lower level stuff for the Playstation platform. I've always had various game ideas bubbling in my mind, and at various stages of prototyping, but never quite had enough time to get anything properly out there'.



PLAYING WITH THE ZOOM MODE ON

One important thing to note is that we'll be facing death countless times without ever having to start all over again. 'It was mostly just trying to keep the gameplay as simple as possible, and keeping the "game loop" fast. Since the game saves your progress, there seemed to be little point in overly punishing a player and making them start over if they don't want to'.

Finally It's not super secret, as it's already been discovered by the community, but there's a certain button combination that'll



ORIGINAL WEB VERSION

earn you a few seconds of invulnerability when playing, which may help get the player pass some of the more awkward to reach beasts!.

REVIEW BY: PAT

MOUSE GUN



DEVELOPER: SILVERMEDAL GAMES
PRICE: \$5 ON CATALOG

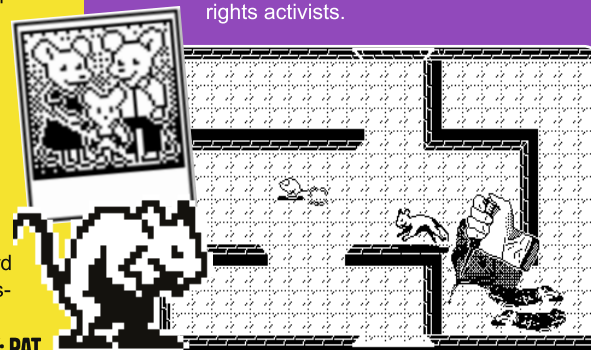
[HTTPS://PLAY.DATE/GAMES/DAYOU-BOATS/](https://play.date/games/dayou-boats/)



Statistics tell us that the average lifespan of a lab rat is around three years, assuming normal stress levels and a decent diet... although we all know those are not exactly the conditions these poor little critters are subjected to in most cases. Still, that description would sound like a five-star resort if we could whisper it to Jonathan, the protagonist of this game, who had the rodent misfortune of falling into the clutches of the merciless Dr. Otto: a ruthless scientist we'll have to confront... or rather, very deliberately avoid confronting, if we want any chance of escaping his sinister complex of mazes.

MouseGun is a frantic top-down shooter where you blast your way through an endless maze of chambers packed with enemies, traps, and pixelated nightmares. Armed with a trusty firearm (which you can upgrade) you'll be munching on cheese now and then just to keep our tiny paws pumping. As if all this weren't enough, we must add the disturbing intrusions of the aforementioned doctor, who, syringe in hand, will try to inject us with a lethal dose of game over.

The game is fun and offers a high degree of replayability, even after we've shone a light into the shadows wrapped around the twisted branches of our family tree. One of its strongest points is how well it plays to the platform's strengths, plus, the mix of humor and sheer bad vibes will probably manage to crack a smile even from the most diehard animal rights activists.



HEXA!
SPECIAL!



ATLANTIC '41

Daily Herald
BRITAIN AND FRANCE
DECLARE WAR

Back in the dying days of World War II, the people of Argentina's Patagonia would watch Nazi submarines pop out of the furious waves battering the coast, only to slip away again without leaving a single clue behind. In that very same ghostly manner, this game rises from the digital deep in the form of a development blog, blindsiding its readers and leaving them gasping for air. Atlantic '41 is shaping up to be the kind of release that'll blitz our Playdates with an almost insane level of craftsmanship. And to launch all the torpedoes behind its creation, we turn the helm over to the captain himself: Cheers Stephan!

Tell me a bit about your background. I'm assuming we're around the same age and that we've gone through pretty much all the classic personal computers (I was more on the Amiga team than in the Atari ST but anyway :), and the great, beautiful experience of learning to program in assembly language, right?

I was born in France in 1970. Like many kids I got a C64, which introduced me to Basic and then Assembly language. The summer of 1983, a friend of mine brought back Exodus Ultima III from the U.S (games didn't cross the Atlantic back then). I'll never forget that strange and beautiful experience, (I could barely read English) which inspired me to making my own games.

A few years of tinkering on many small projects and I eventually completed my first game with a friend, for the Atari ST. Amazingly, the game (Outzone) was

published by Lankhor (Mortville Manor, Vroom), a well regarded publisher and developer. Encouraged, I dropped out of my physics degree in college (to the joy of my parents) to focus on a platformer (Indigo, which you can download for free). Beyond my wildest expectations, Psynosis (Lemmings, Shadow of the Beast, Wipeout) picked up the game, which made wonder if I was gonna make it in games after all. But the dream was short lived. The deal fell through right as I was wrapping up development, for reasons still obscure to me, but likely related to Sony acquiring Psynosis at that time. That put a stop on my indie developer career and I worked in press for a few years. But I couldn't let go of games, so I dropped everything and moved to Canada with my girlfriend (soon to become wife).

This marked the beginning of a career in AAA, first as an environment modeler, then concept artist and art director. I worked for Ubi Soft and Electronic Arts in Montreal for eight years, then joined Blizzard Entertainment, where I Art directed cinematics for various franchises, and did some concept work for Overwatch, a position I still occupy. Then a few years back the Playdate rekindled my passion for indie development, and I've dedicated my spare time to Atlantic '41 since then. Inspired by the journals of Jordan Mechner, I decided to document the journey in a devlog. Partly from luck, partly from having a foot in the door when the industry was less competitive, I've made a living in videogames, and fulfilled my dream of moving to the birthplace of the game that started it all. I had the

privilege to meet a few of my idols, like Dan Malone from the Bitmap Brothers,

who is the sweetest guy and with whom I remained in contact. Eric Chahi told me about Another World when it was still a concept, and we were both part of a group of developers in Paris. I exchanged e-mails with Richard Garriott, aka Lord British, after he heard about a Ultima IV remake I had pitched to EA. I also met with Jorda Mechner a few times at Ubi Soft. I worked with the energetic and passionate Chris Metzen. This is not to name drop, but to share how grateful I am for the opportunities I've been given, and the people I've worked with, many of them a lot more talented that I'll ever be, even if I'd also have less flattering things to say about corporations taking over the video game industry. But overall it's been way better than the French kid who copied Basic programs from magazines could ever dream of.

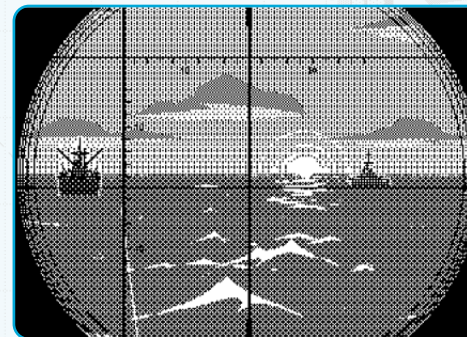
One short passage in one of your texts really caught my attention: you mention that you tend to treat the Playdate as a computer. As a fellow amateur dev, I feel the same way. In fact, I even spent some time looking into whether it was possible to develop in some kind of assembly instead of selling my soul to Lua... ;) and I still haven't ruled it out. Atlantic '41 feels like a manifesto on intentional design under extreme limitations. Do you think that's the magic that has lured so many indie developers, especially from our generation- back into the game?

Absolutely. However, and I can only speak for myself, it's not the sole reason. The Playdate checks all the boxes that made me fall in love with computers in the 80s. Firstly, it's immediate and self-contained. You can write code and run it in minutes. Secondly, Panic (shoutout to them!) doesn't obsessively control what's on the platform or impose reviews on their developers, unlike other console manu-

As a U-boat captain, you have a 25% probability to survive the war. This may sound brutal, and it is, but the game entirely rests on that idea.

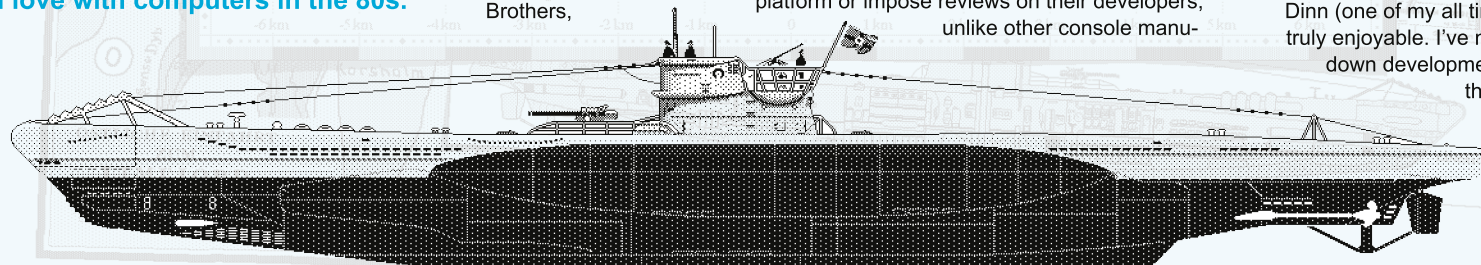
facturers who use them as a form of censorship. Thirdly, unlike PCs, all Playdates are born equal. You don't have to deal with hardware compatibility issues or driver shenanigans. Finally,

coming from early computers where performance was an eternal struggle, I love working with technical constraints, having to work the problem. That being said, the Playdate is a capable machine. In fact, it's a significant upgrade to my old computers!



The sheer volume of reference material you present on the relationship between crew and submersible beast -books, magazines, films- is overwhelming. Did you already have all these books in your collection? If so, did you have to reread them, or did you dive into a whole wave of new reading that, I imagine, massively extended the development time? It's somewhere in between. I already owned WW2 material. Then leading up to Atlantic '41, I read about U-boats and the Battle of the Atlantic. Then when I need to deepen my knowledge of a particular topic for a game feature, I research it. Now I have a pretty well rounded general knowledge on the subject, at least enough for the game. Also, a Discord group of U-boat experts graciously offered advice and insight. They've consulted on the game UBOAT, so they know their stuff. But like Lucas Pope mentioned in one of his devlogs about Return of the Obra Dinn (one of my all time favorites), research is truly enjoyable. I've never thought that it slowed down development; it's an integral part of the job.

Why did you choose a German U-Boat as the main "character" of the game?



The Playdate checks all the boxes that made me fall in love with computers in the 80s.

CHERRY HILL

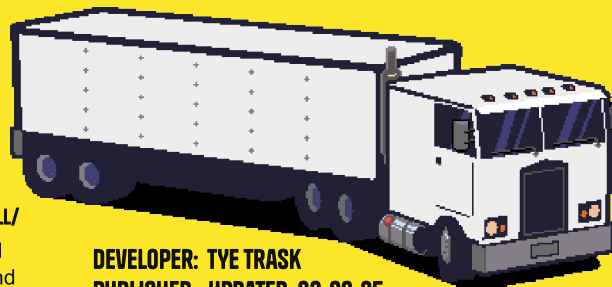


[HTTPS://OATCUP.NET/GAMES/CHERRY-HILL/](https://oatcup.net/games/cherry-hill/)

During my summer breaks from university, I used to spend whole afternoons with a friend who ran a construction supply yard. His dad and uncle were the ones really running the business, while he was in charge of hauling loads from one job site to the next. He drove a heavy-duty truck, and he was damn good at it. It was something else to sit in the passenger seat and watch him pull off the most precise, mind-bending maneuvers behind the wheel of a beast that weighed 12 tons. Each load-in and unload felt like a sucker punch to mother nature and a middle finger to the laws of physics. Like a highway ronin, he'd roll that mechanical fury through back alleys where you wouldn't dare push a bicycle.

This game flung me straight back to those sweat-drenched afternoons in the cab: teenage jokes, burning diesel, wiping off sweat with a rag that had more oil stains than clean spots. But above all, it reminded me of my buddy's love for that truck, a bond that was pure cyberpunk in the most raw, Mad Max sense of the word. In that same spirit, Cherry Hill delivers a payload packed with obsessive love for detail. So much so that if it were ever stopped at a border inspection, they'd have a hell of a time explaining that extra weight in the trailer, because this game is way more than what you'd clock on during a quick flashlight scan. As the developer herself put it: *"My hope was to make something that's more than 'choose destination from menu' and 'drop trailer inside glowing green square'.* Driving a truck is the main task in Cherry Hill, but the world, the people, the little processes like walking inside the depot to pick up your next job - it's all part of what makes it feel fun and alive to me". And yeah, all of that checks out! But it also cleverly sidesteps what's really packed in the trailer: a story rooted in slices of the dev's personal life. Cherry Hill isn't just a truck sim (or a trucker sim), nor is it some mildly open-ended sandbox. This thing's got ambition to burn, both in content and obsessive detail.

Take the tutorial, for example, it sneaks up on you, slipping



DEVELOPER: TYE TRASK
PUBLISHED - UPDATED: 09-26-25
GENRE: RACING - ADVENTURE - LIFE
PRICE: \$7 ON CATALOG & ITCH.IO

into your brain like the lyrics to a Clarence song while you're knee-deep in the first deliveries. By the time you realize you've been taught how to play, you're already in too deep to walk away. Honestly, that onboarding alone could outdo any technical write-up I could cram into this review. So let's switch on the high beams because this deserves more than a passing glance. The highway and city driving sections aren't just visually distinct; they feel very different. The first half is your standard-issue highway stuff: keep a tight schedule, monitor your rig's condition, play nice with traffic laws, and maybe throw in a bit of diesel zen. You'll pick it up quick, steering with the crank, gunning it or braking with the d-pad. The camera angle is that timeless racer's perspective that's been with us since Out Run and refuses to retire. But don't let the familiarity fool you. This is just the cab, the trailer's still missing (pardon the metaphors. Honk! Honk!). The other side of the ride, the cherry on this hill, lets you wander through different roadside stops (on foot or from behind the wheel) to pick up delivery contracts that earn you cash and help upgrade your gear. This all happens through talking with the locals, getting to know their stories... and, along the way, uncovering your own. *"Since I was young, I've always loved all kinds of vehicles: trains, semi-trucks, you name it. I've never driven a semi myself, but my*

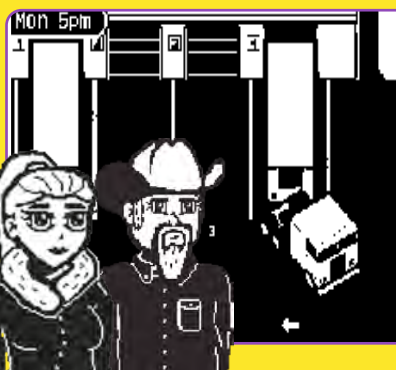


mom did. It was part of her career!" - Says Tye. It's in this part of the game where we start to hit a few potholes, and let's be honest, any road worth traveling has some. Controlling the truck (especially with a trailer hitched up) in these urban areas gets a lot trickier. The camera angle shifts to a top-down view, showing loading zones, gas stations, offices, shops, and even tiny pedestrians eager to make your acquaintance. Among the required tasks is parking your rig in tight warehouse spots. Pulling it off is not easy at all and at times can be quite frustrating. The controls are still straightforward (crank for steering, left arrow to toggle the handbrake, up and down to accelerate or brake) but executing these moves takes finesse, and don't even get me started on nailing a jackknife devilish maneuver (see the loogbook for more details). You won't want to get stuck on side-

walks either. Most of the time, getting out of a jam is such a hassle that it's easier to just use the tow option, which resets you to a safe spot.

But don't take my word as expert advice either, I've always been the guy in the passenger seat for a very good reason!.

REVIEW BY: PAT



OPTIMUS PRIME
VICTORIA TERPSICHORE
(COWBOY BEBOP)
B.J. MACKEY (AND THE BEAR)
LINCOLN HAWK (OVER THE TOP!)

ONE FOR THE ROAD

My name is Tye, I'm a software engineer so I do have programming experience, though I've never done any game development professionally. I started the game in Lua but it took kind of a strange path from there. Around last summer, I started hitting a wall with performance for the visuals I wanted to achieve. I think that's somewhat common for 3D or pseudo-3D games on Playdate. Just then, it was announced that you could target the Playdate with embedded Swift! I think Swift is a great language and thought it would be perfect to squeeze out the extra speed I needed. Unfortunately, while I got the road renderer working, I ran into some snags. I made a bit of headway but ultimately decided to prioritize doing what it took to move development along.

The in-town physics were probably one of the trickier parts of development. For collisions, the Playdate SDK uses axis-aligned bounding boxes, in short: things have to be rectangular. That means the SDK can't help if the trailer is at any sort of angle. I knew this would kill the fun if you couldn't go around a tight corner and squeak by with your trailer. That's part of driving a big truck! I'm sure there's a better way to build it, but the first detail involves... Blender!.

Each truck or trailer model has a separate mesh that's on the "ground" and represents how it would collide with the world. When the sprites are rendered out by the build tooling, it also creates a separate sprite that for just the collision shape. This is the basis for how the collisions can feel almost 3D. That ground collision sprite is added to the world, but it's invisible. From there, when a standard SDK collision is detected, it uses an alpha (or transparency) collision test to detect if the ground sprite is overlapping the object to know whether it should be considered a real collision or not! Over in LDtk, I define each parking space boundary using a custom entity and include the direction it's facing (north/east/south/west). That same ground collision sprite is used to detect how well you're parked inside a spot and provide a score!.

REAL DIESEL HEROES

