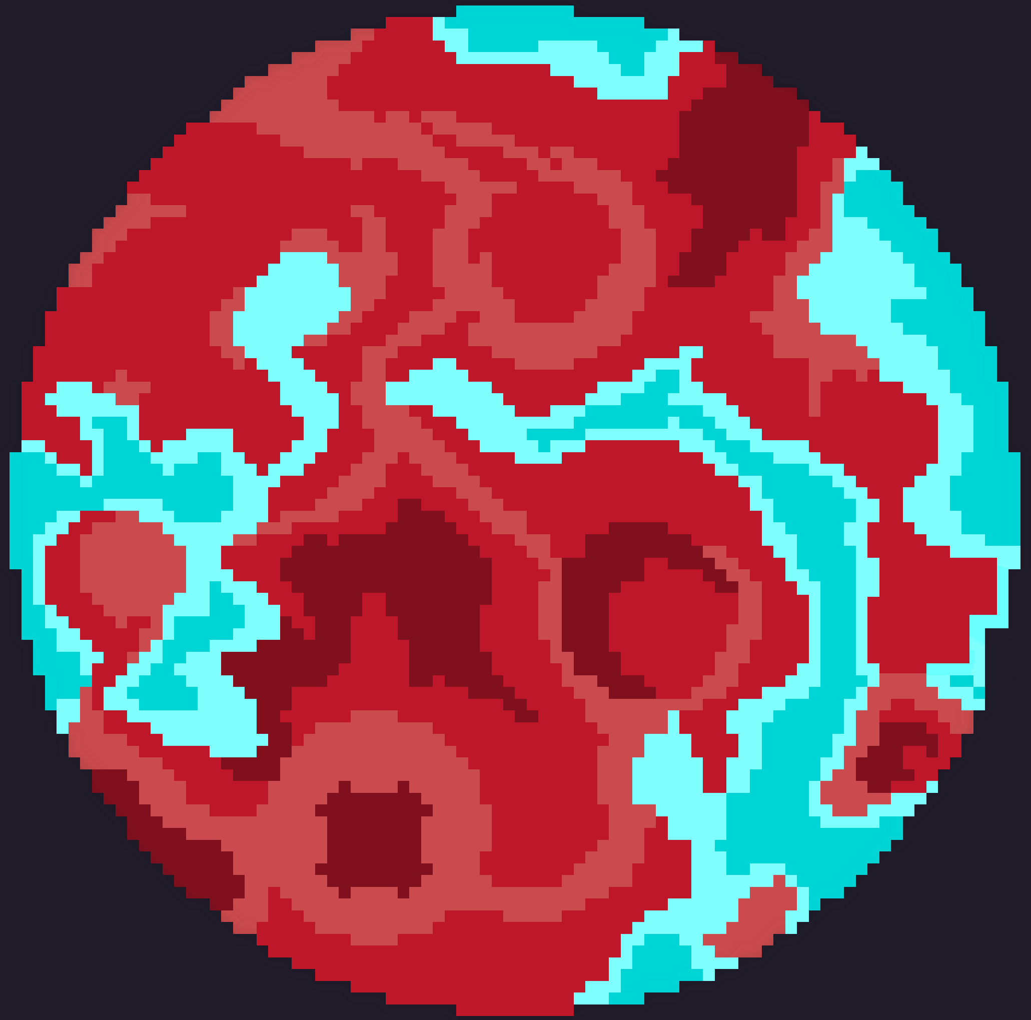
BEHIND THE SCENES

LESSONS LEARNED FROM THE MAKING OF MISSILE CARDS

BY NATHAN MEUNIER

WITH amazon appstore



WHAT AN EXCITING TIME TO BE AN INDIE DEVELOPER

With powerful tools that speed up development and an abundance of diverse frameworks that let you self-publish your games, there's never been an easier, more accessible time to create something unique for the world to play.

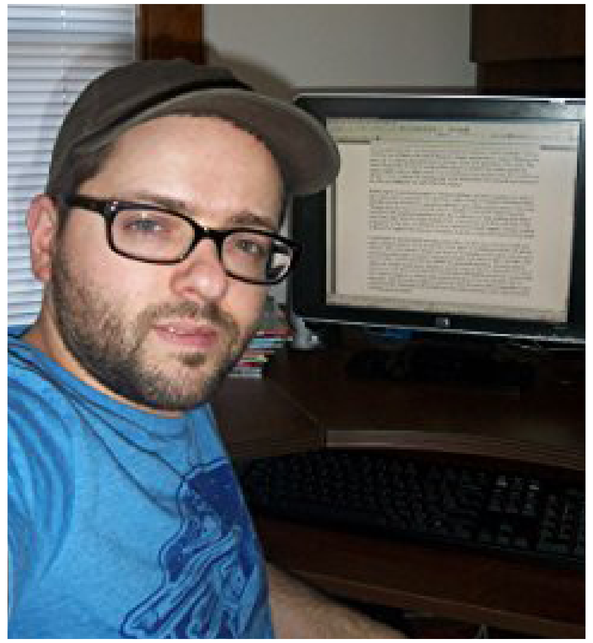
For me, GameMaker: Studio was the catalyst that changed everything. I've spent most of my professional career writing about other people's video games for publications like Nintendo Power, IGN, GameSpot, GamePro, and more. I never expected to eventually make my own games, yet here I am. In just three years of part-time tinkering, I've gone from having zero knowledge about what goes into creating a game to rolling up my sleeves and bringing games from concept to launch.

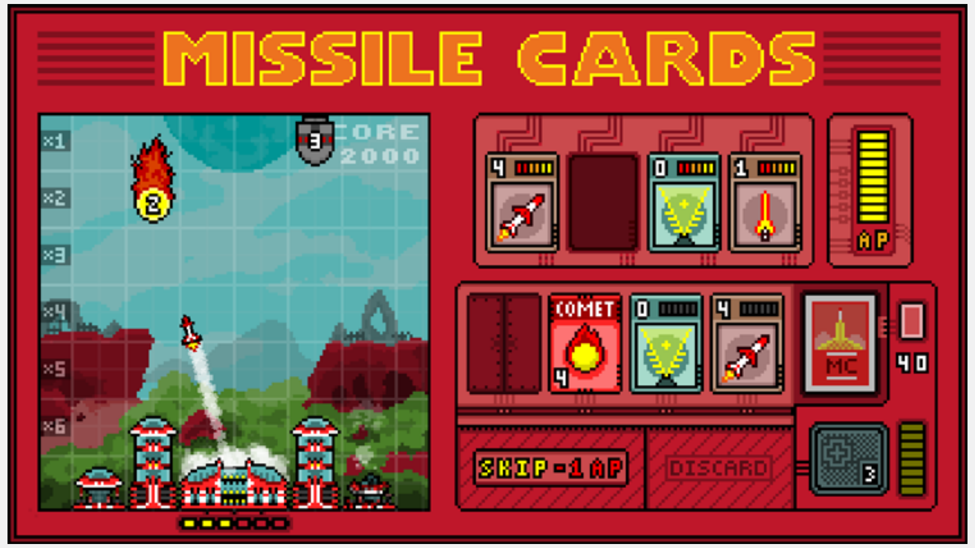
Missile Cards isn't my first commercial project, but it's one of the weirdest, most ambitious, and most successful games I've created to date. For me, learning is an important part of the process and each project brings new challenges, victories, and opportunities to share my experience with others.

In this eBook, I will pull back the curtain on everything from my game design and creative processes to how I tackled tough mechanics. Then, I'll show you how I prepped my game to port it over to Kindle Fire devices. Along the way, I'll offer nuggets of inspiration and insight to help you bring your own games to fruition on Amazon's Appstore and beyond.

Thanks for reading!

Nathan Meunier
Creator of Missile Cards

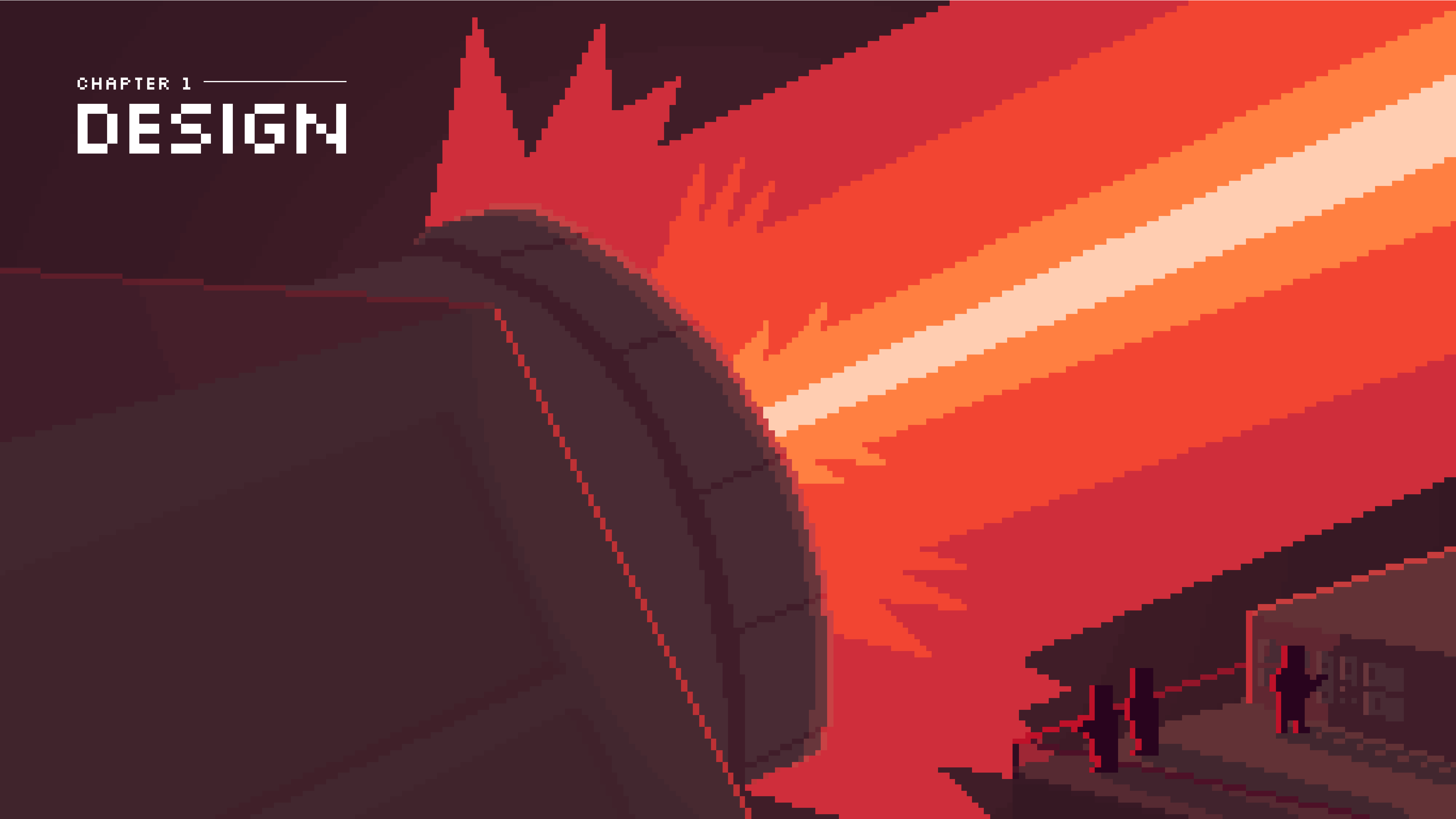




> "I NEVER EXPECTED TO EVENTUALLY BE MAKING MY OWN GAMES, YET HERE I AM."

CHAPTER 1

DESIGN



FINDING THE UNUSUAL CONCEPT FOR MISSILE CARDS

It was late fall, 2016. Exhausted from grinding away on a larger indie console project over the course of two years, I found myself desperate to hit the finish line—aching to feel the cathartic rush of completing a game again. The idea of creating something small, polished, and launchable on a very constrained development cycle was immediately appealing. With no end in sight on the larger project I was working on with another team, I needed a temporary change of pace.

By this point, I would blow off steam in my spare time by prototyping card game concepts.

Making a small card game seemed like a perfect fit for a short-term solo project, given the genre's modular nature and high replayability. I just had to come up with a strong idea I could realistically build out and finish in a matter of months, instead of years.

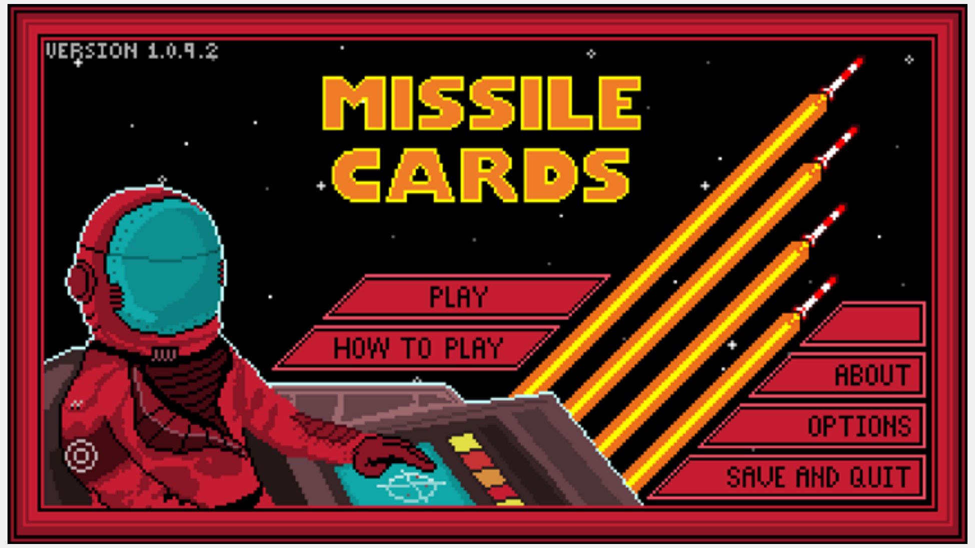
My favorite mobile card games are the ones that mix bold themes with unusual mechanics to create something funky and fresh. Being largely a solo gamer, I also prefer solitaire-style card games that can be played alone. After a lot of brainstorming, I fixed on the idea of a card game where you would defend yourself from giant doom meteors hurtling down from space. You'd lay down defenses that would have to be charged up before blasting the oncoming debris away, and the deck itself would be your adversary, providing a mix of helpful and harmful resources to juggle.

I realized this concept reminded me of a childhood gaming favorite: Missile Command. I decided to roll with the retro vibe this evoked and put a fresh spin on things. The idea of taking the intense arcade defense gameplay of the Atari classic and re-imagining it as a turn-based strategy solo card game was just weird enough to work. Not only that, it had a grabby hook that would potentially appeal both to modern digital card game fans and retro gaming enthusiasts alike.

Missile Cards was born, with the tagline "Missile Command meets Solitaire...only with more doomlasers, death, and explosions."

In reality, it's not quite like either game it pays homage to, but it draws inspiration from both as well as the nostalgia of growing up as a young gamer in the '80s.

Armed with a cool theme and a loose battle plan to make something fun in just a few quick months of coding and design, it was time to get to work.



"MISSILE COMMAND MEETS SOLITAIRE... ONLY WITH MORE DOOMLASERS, DEATH, AND EXPLOSIONS."

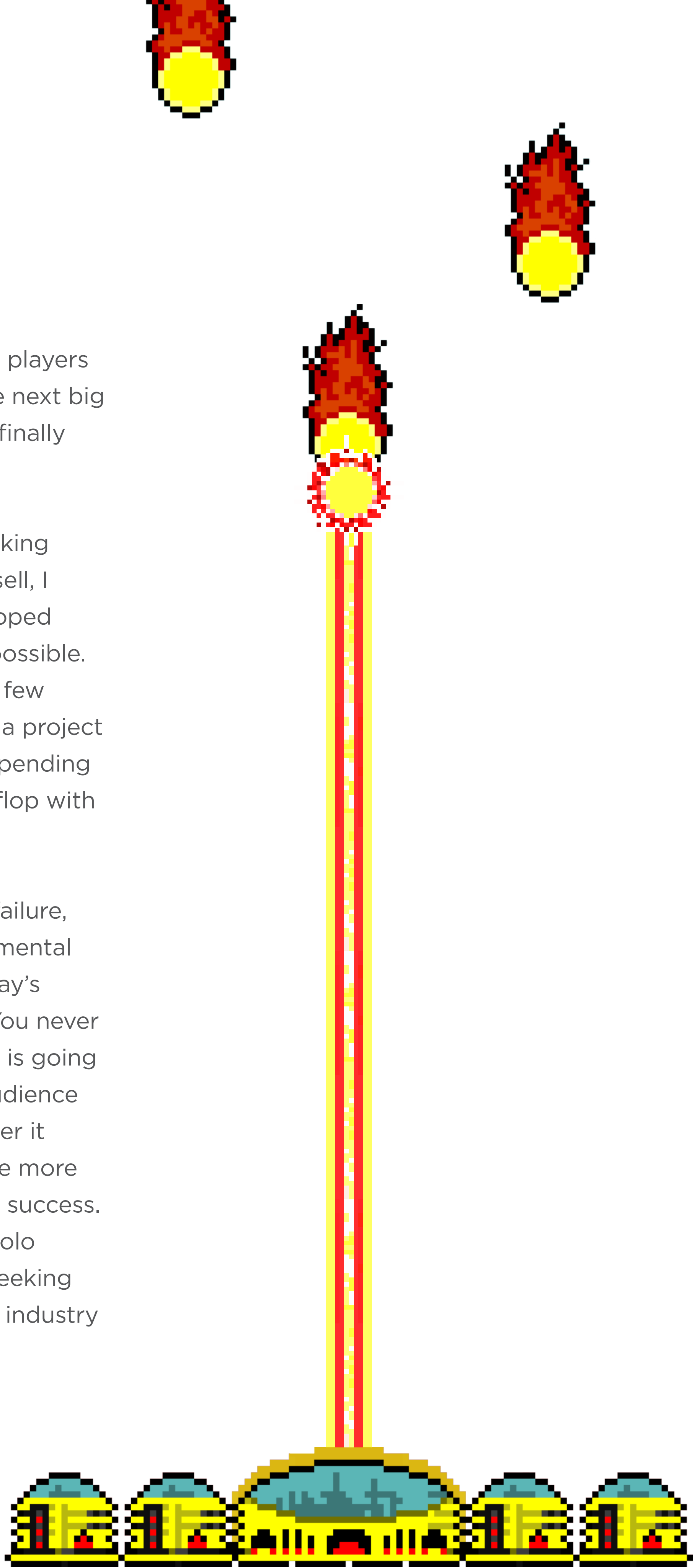
MINIMIZING RISK BY MAKING SMALLER GAMES FASTER

Game development almost always takes much more time and energy than anyone anticipates. With Missile Cards, I set out to create and launch a polished game in just three short months. It took me six. Every facet of the game's design was tuned towards creating a minimum viable product, and even then it still took me much longer to complete than I had planned. This shouldn't be a surprise to anyone who has dabbled in indie game development.

So what does "minimum viable product" mean, exactly? My ultimate goal with Missile Cards was to make a finished game I was proud of, all while minimizing my financial risk on the project. The market segment for game genres and trends shifts so rapidly that many developers start making a game based on what's popular at the moment, only to find that players have already moved on to the next big thing by the time their game finally ships years later.

Instead of spending years making a huge game that might not sell, I wanted to create a tightly-scoped project in as short a time as possible. If it sold, great! If not? Well, a few months of part-time work on a project that fails is a lot better than spending three years on a commercial flop with nothing to show for it.

Planning for the potential of failure, particularly with more experimental projects, is a smart bet in today's hyper-competitive industry. You never truly know when a new game is going to resonate with a broader audience until it launches, and the longer it takes to develop a project, the more pressure there is for it to be a success. This is doubly important for solo developers and newcomers seeking to carve out a foothold in the industry for the first time.



WAYS TO HELP MINIMIZE RISK

- ✓ Keep the development cycle short
- ✓ Create constraints for the project
- ✓ Set internal deadlines and milestones
- ✓ Focus on just the core content (and avoid piling on extras)
- ✓ Be willing to cut features that aren't working out

KEEPING SCOPE CREEP TO A MINIMUM

Scope creep is an omnipresent danger in game development, affecting teams of all sizes across all sectors of the industry. It's all too easy to succumb to the desire to throw every cool idea you come up with into your game. Experimentation is an important part of the design process early on, but there comes a point where continuing to add ideas can easily spiral out of control. Not every neat mechanic or design idea you come up with will make it into your game—and that's ok.

Here are some ways to keep scope creep to a minimum:

Constraints breed creativity

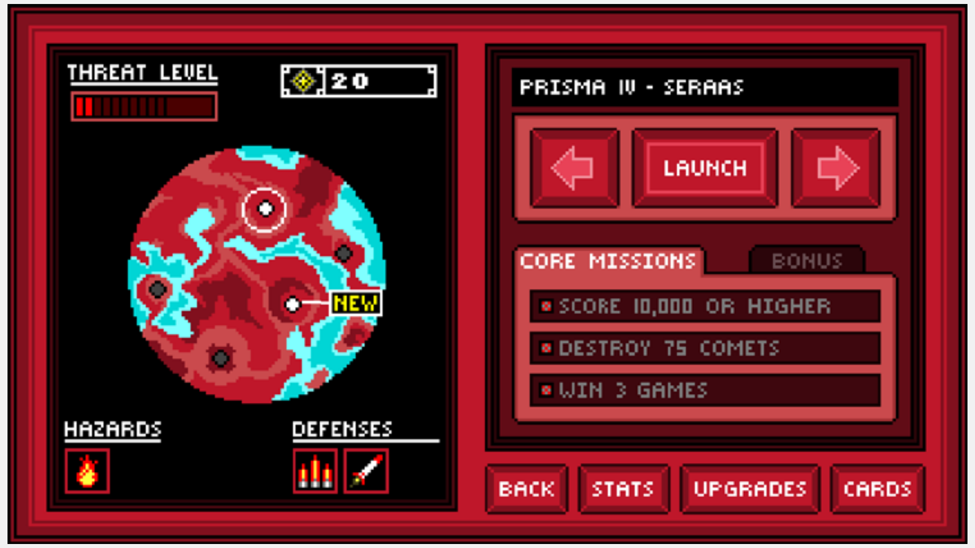
Setting limitations and learning how to identify unnecessary bloat in your game design are valuable skills. When you set constraints for your game project, it makes it a lot easier to identify and trim the fat. Honing this ability can save you a lot of wasted time and heartache.

It's hard to pin down a realistic deadline when planning out your game in its early stages, but setting a target timetable is invaluable in keeping scope creep to a minimum.

Making tough decisions

My original plan for Missile Cards was to create three full planets worth of content. This seemed like a reasonable goal at the time, but once I got deeper into development, I quickly realized that building out that much content would take much, much longer than I anticipated.

It was a difficult decision, but I opted to pare the game down to one single planet with five bases, then double-down on making the game challenging and replayable enough to give players a solid play experience for their money. This was the right call.



TIPS FOR KILLING SCOPE CREEP

- ✓ Impose creative limitations early on
- ✓ More isn't always better
- ✓ Be willing to make hard decisions
- ✓ Try making smaller games, faster

DESIGNING FOR MULTIPLE DEVICES

One of the trickiest parts of game design is taking into account the nuances of each device you intend to launch on. Things like screen orientation and scaling to a variety of different device resolutions, figuring out how to handle UI layout and sizing, and optimizing art and code to run smoothly are all important design decisions that you have to consider.

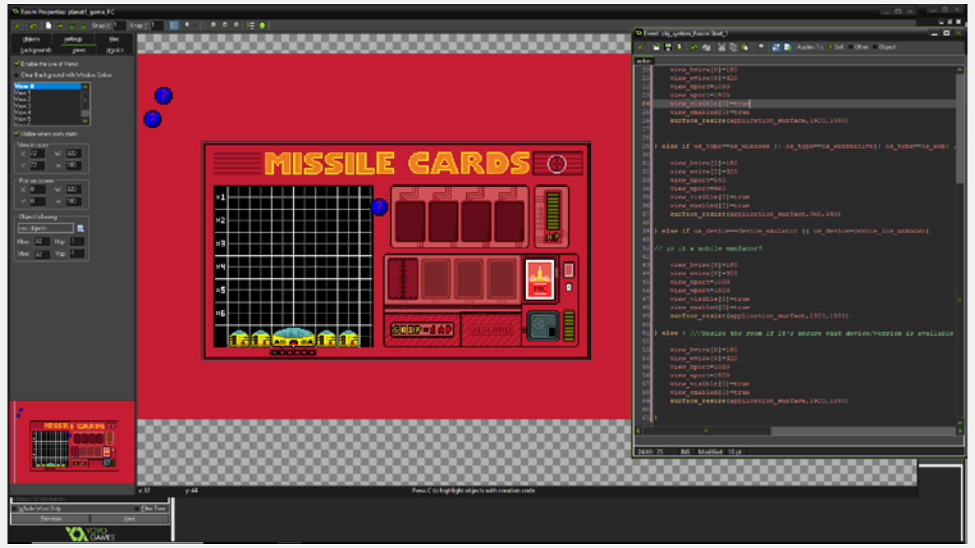
I always intended to eventually bring Missile Cards to mobile devices, but I also wanted to experiment with putting the game on Steam first to test the waters and generate some initial sales momentum. As many developers learn the hard way, launching your game first on mobile, then releasing it later on Steam is almost always a bad idea. Players on Steam are hyper-critical of “mobile ports” and anything that looks too casual. To avoid that backlash this time around, I opted to launch on PC first, then work my way to mobile.

Since PC was my initial launch target, I settled on a widescreen layout that would work well on different devices and upscale cleanly to 1920 x 1080, a fairly standard HD resolution. Because I used pixel art for this project, getting the on-screen view sized just right to avoid pixel distortion was vital. When the game first loads, I have a block of code that detects the device and screen resolution, then adjusts and scales the view to match a best fit. It’s not always pixel-perfect, but it works well and looks good on most devices. Another script toggles different UI and button layouts based on whether you’re playing on a PC, a tablet, or a phone.

With memory limitations being more of a concern on mobile devices, it’s important to optimize everything you can, even if your game runs well on more powerful hardware.

In GameMaker, that often means limiting use of step events, avoiding unnecessary draw calls, being careful with use of particle systems, watching for memory leaks, and making sure to clear audio and textures when they’re not needed.

Mobile optimization can be less of a concern with pixel projects, since you’re working with smaller visual assets and you’re often upscaling the view instead of creating more of a 1:1 layout. But, it’s worth taking the time to optimize everything you can once your game is finished and ready to go. Designing with these concerns in mind from the get-go will make the home stretch of finishing and launching your game a much smoother process.



```
    view_hview[0]=180
    view_wview[0]=320
    view_hport=1080
    view_wport=1920
    view_visible[0]=true
    view_enabled[0]=true
    surface_resize(application_surface,1920,1080)


} else if os_type==os_windows || os_type==os_win8native|| os_type==os_uwp|

    view_hview[1]=180
    view_wview[1]=320
    view_hport=540
    view_wport=960
    view_visible[1]=true
    view_enabled[1]=true
    surface_resize(application_surface,960,540)

} else if os_device==device_emulator || os_device=device_ios_unknown{

// is it a mobile emulator?

    view_hview[0]=180
    view_wview[0]=320
    view_hport=1080
    view_wport=1920
    view_visible[0]=true
    view_enabled[0]=true
    surface_resize(application_surface,1920,1080)

} else { ///Resize the room if it's unsure what device/version is available

    view_hview[0]=180
    view_wview[0]=320
    view_hport=1080
    view_wport=1920
    view_visible[0]=true
    view_enabled[0]=true
    surface_resize(application_surface,1920,1080)

}
```

THINGS TO CONSIDER

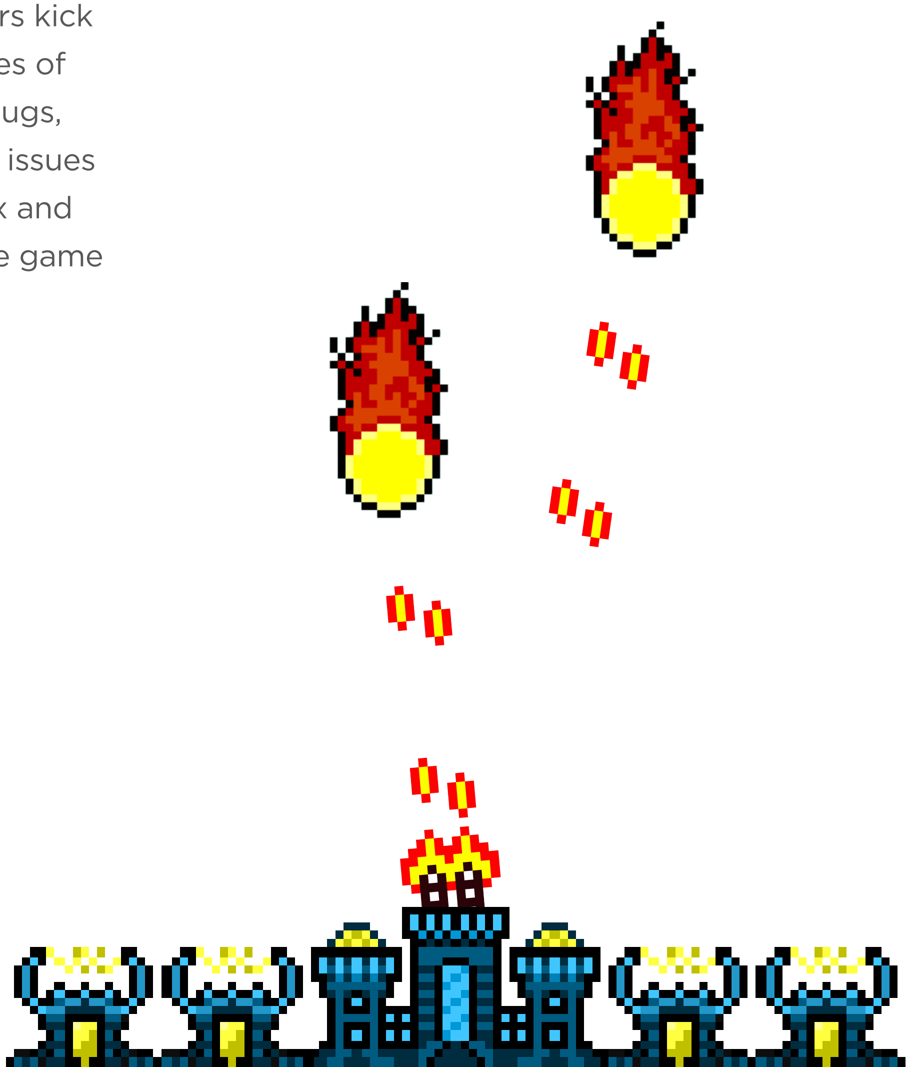
- ✓ Screen orientation and resolution
- ✓ Hardware and memory limitations
- ✓ Device-specific trends and best practices

THE VALUE OF BETA TESTING

Nothing beats getting real-world player feedback to help improve your game BEFORE you release it into the wild. There's a lot you can learn from running beta tests—from how your game works on different device configurations, to how players acclimate to your user experience. Despite this, many new developers overlook this important step because they aren't sure how to find players interested in checking out a beta build of their game.

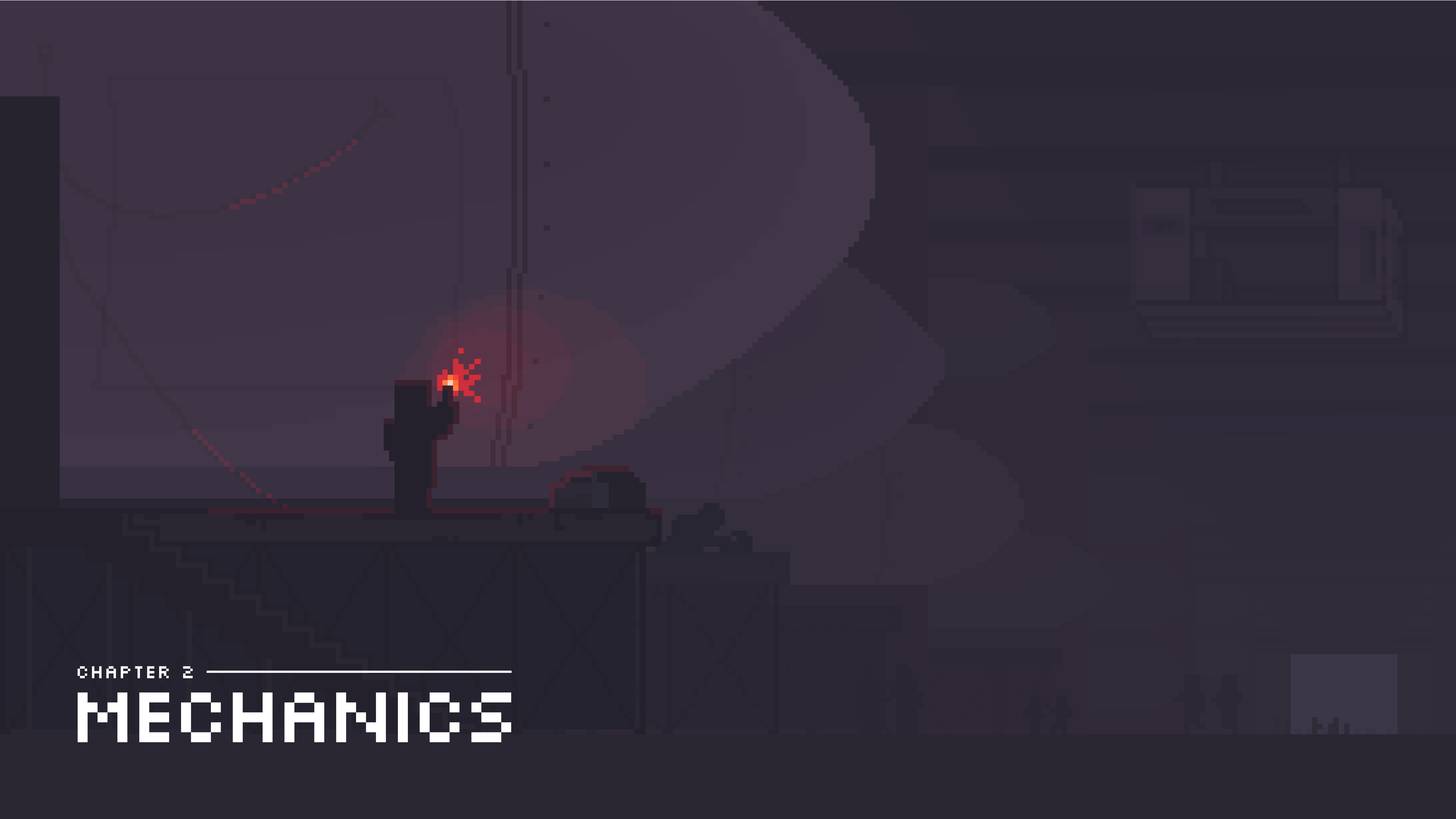
The trick is to cast a wide net, using a mailing list or some other kind of sign-up form. In a lot of cases, it's not that hard to find people willing to play your pre-release game, but it can be a challenge to find people who will follow through and provide the quality feedback you're looking for. That feedback is valuable, though, particularly from strangers who will speak honestly about what works and doesn't work in your game. It also tends to uncover lots of unexpected bugs, since most people will not play your games the way you do or the way you expect them to. Having this intel before you launch gives you more time to polish and tune the experience to ensure you're putting your best foot forward.

Missile Cards' gameplay felt pretty good before I ran my first beta tests, but having hundreds of players kick the tires across several batches of beta test rounds uncovered bugs, some design flaws, and other issues that I had plenty of time to fix and improve before publishing the game to a broader audience.



TIPS FOR FINDING BETA TESTERS

- ✓ Use a mailing list signup to gather interested players' contact information
- ✓ Create a sense of urgency by setting a limited number of signups
- ✓ Include a short questionnaire that focuses on the most important feedback you want
- ✓ Don't forget to follow up several times once testers have had a reasonable time to play
- ✓ Aim to get way more testers than you need, because many will not follow through



CHAPTER 2

MECHANICS

NAVIGATING THE COMPLEXITIES OF DIGITAL CARD GAME DESIGN

Digital card games are a fun challenge to design because they give you a basic and familiar foundation (a deck of cards) that you can adapt to just about any genre, style of play, or conceptual theme. I based a lot of my prototypes, initially, on a single deck of 52 cards, but in the end, precious few of them wound up resembling anything close to a traditional solitaire card game.

Missile Cards may be my first commercially released card game, but I've created dozens of card game prototypes and am working on numerous card-based projects I plan to release in the future. From working on these projects, I've noted some common elements that tend to make or break a digital card game's design.

Here's a look at some of the core design considerations that you need to think about when making digital card games.

A strong theme is everything

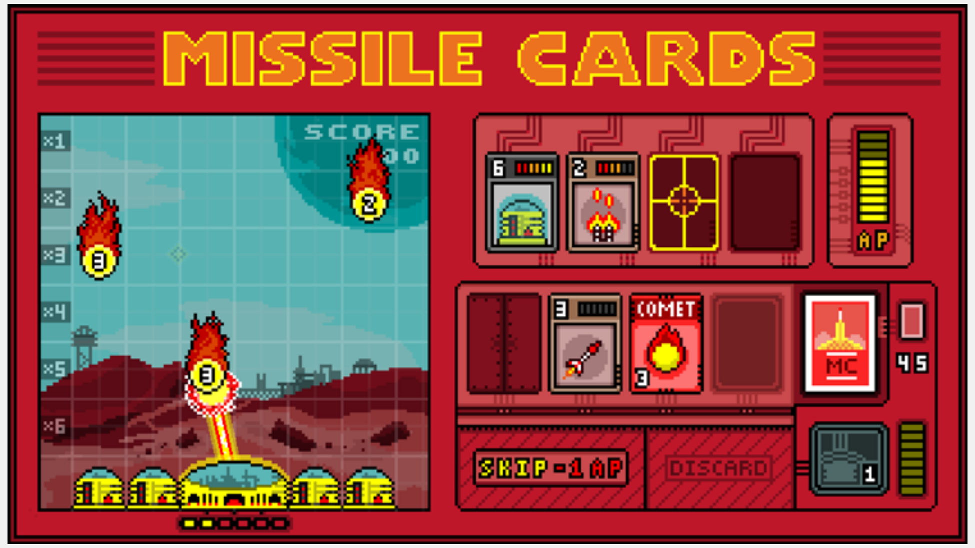
A theme is usually the first place I start when I'm thinking about a new card game project, because it quickly helps me figure out a visual style for my game.

Having a striking visual style is important, but I also find that deciding on a theme helps me come up with interesting ideas for the gameplay itself. Whether you're making a card game about dungeon crawling, fishing, city building, sci-fi combat, or even relationship building, each potential theme holds an exciting range of possibilities for visual style and unique card mechanics.

Figure out your visual layout early-on

When designing card games, I often begin mocking up visual layouts very early-on in the process, before I even start coding up the game itself. This lets me adjust card size, play around with positioning, and make important decisions about how the game might play based on the limitations of resolution and screen space. I often find that doing this helps me identify potential design issues right away, saving me a lot of time and energy.

It's also worth mentioning you shouldn't finalize any of your artwork until you've got your core layout locked down, since making layout decisions can have a huge impact on your games visual direction.



QUICK TIPS FOR DIGITAL CARD GAME DESIGN

- ✓ Pick a strong, distinct theme
- ✓ Make your visual design pop
- ✓ Aim for short, highly replayable gameplay loops
- ✓ Take advantage of the digital format

Balance accessibility, depth, and replay

Many mobile players prefer games they can enjoy in short bursts instead of marathon sessions. This is one of the reasons card games are so popular on mobile, because they often balance short, accessible gameplay loops with high replayability and long-term metagame progression.

When designing card games, it's worth paying close attention to how long it takes to play through a game and tuning that to be a short, highly replayable experience. Making sessions fun and punchy is important, but also explore ways to layer on progression mechanics, unlockables, and other goals that give players a reason to keep coming back for more.

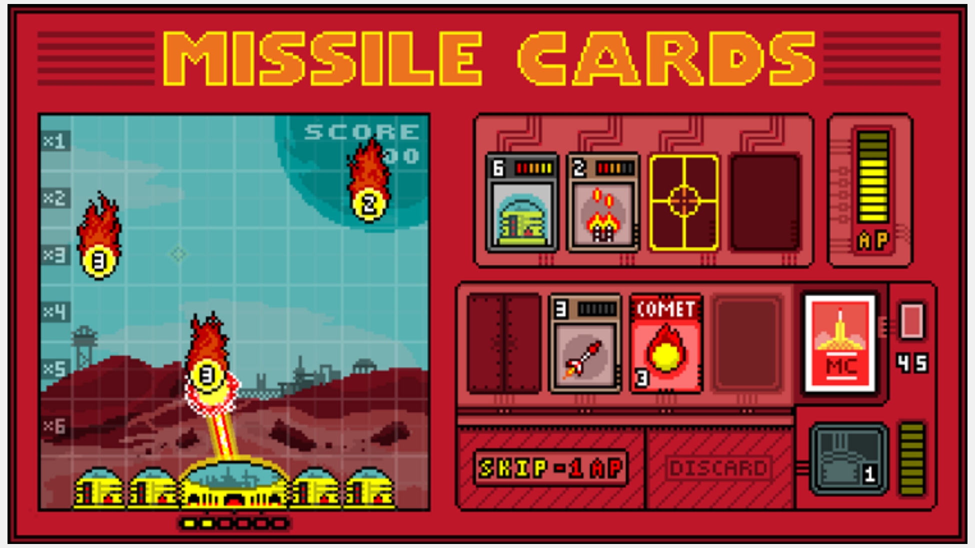
Make the most of the digital format

Physical card game design often tends to be centered around a limited set of simple rules and mechanics—things that can be easily digested without pushing players over the edge of information overload. With digital card games, however, you can get away with much more complexity because you can build it into the behind-the-scenes system that runs the gameplay.

From dice rolls and stat tracking to randomization and special events, a lot of the nitty-gritty can be handled by code, freeing up players to focus on whatever you present to them.

This opens the door to weaving lots of unusual genres into card-based designs. Missile Cards, for example, just wouldn't be possible as a simple card game without forcing players to keep track of an excessive amount of information.

But the digital format let me design systems to automatically handle a lot of the complexity, allowing players to concentrate on the strategic defense gameplay and simply enjoy the experience.



QUICK TIPS FOR DIGITAL CARD GAME DESIGN

- ✓ Pick a strong, distinct theme
- ✓ Make your visual design pop
- ✓ Aim for short, highly replayable gameplay loops
- ✓ Take advantage of the digital format

TURNING ARCADE INSPIRATION INTO A TURN-BASED STRATEGY GAME

The clearest thing I remember about playing Missile Command as a kid is that death is inevitable. As the oncoming barrage of warheads screaming towards your cities grows to a crescendo, it's all but impossible to stave off annihilation long term. The game's fast-paced arcade action and brutal difficulty led to match after match of sweaty-palmed cursing and sporadic hand cramps. I loved it.

In Missile Cards, I wanted to capture that intensity, but channel it into a completely different kind of experience: a turn-based strategy game. That seems a bit counter-intuitive, but oddly, worked out quite well. Here's how I did it:

Subverting expectations

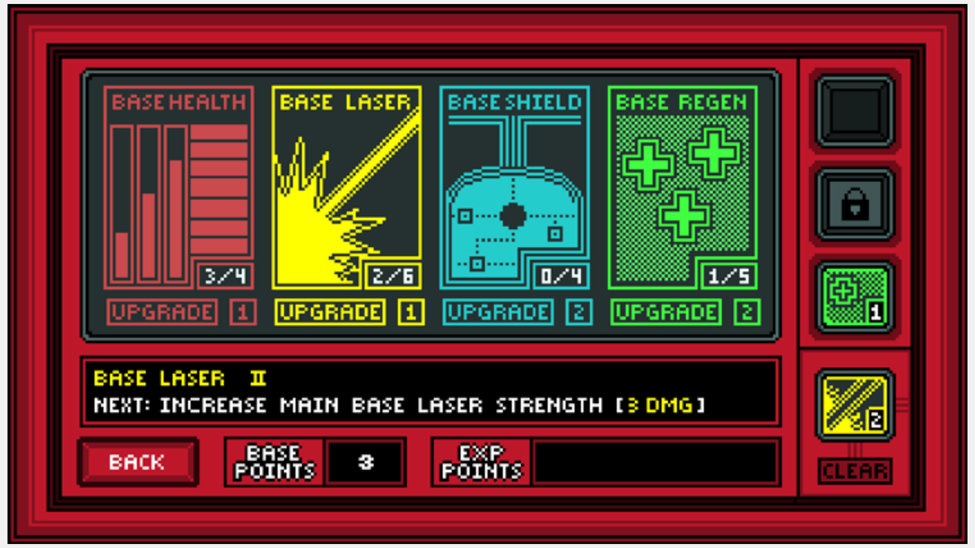
Rather than juggling resources, powering up your defenses, and blasting away hazards hurtling towards your bases in real time, I tapped the brakes and built a system where almost every action you take has a reaction. Whether you're placing a defense card into an available slot for charging or triggering an ability, each move pushes the game forward. Cards advance on the conveyor belt, potentially adding new hazards to the playing field, as existing hazards continue to drop closer to your bases.

The pressure-cooker intensity comes from having to plan out your moves carefully, make tough choices, and figure out the best way to use the resources available to deal with oncoming hazards. Timing is everything.

80s nostalgia for the win

Other areas of Missile Cards' design pay homage to aspects of retro gaming. The pixel art style is a lot more detailed than the super simplistic Atari sprites from the '70s and '80s, but their design and color scheme draw a lot of influence from that era.

The game's interface, menus, and UI are also all designed to look like you're playing one of those old plastic gaming handheld devices. Even the decision to include the name of the game itself on the actual gameplay screen is a subtle reference to the classic days of gaming.



BASE LASER II
NEXT: INCREASE MAIN BASE LASER STRENGTH [3 DMG]

BASE POINTS 3

EXP POINTS

> "MISSILE CARDS IS A TOUGH GAME BY DESIGN, AND IT'S CAREFULLY TUNED TO OFFER A HIGH LEVEL OF CHALLENGE."

CRAFTING CONTROLLED RANDOMNESS IN CARD GAME DESIGN

People who play a lot of card games tend to understand that randomness is a core element. You're shuffling the deck. You're drawing cards. You don't know what you're going to get, and part of the excitement and replay value of card games comes from this unpredictability.

The problem is that sometimes RNG-based systems (random number generators) can slip into off-kilter grooves where the randomness is too lopsided, which can have a huge negative impact on the player experience.

Either they're getting too much good stuff and the game is too easy, or they're getting utterly steamrolled. It's frustrating to lose a match entirely on the whim of a random dice roll, which is why it's important to make sure the RNG systems you use don't make your game unsolvable.

This was an issue I discovered early on with Missile Cards, thanks to feedback from beta testers. While I spent a lot of time fine-tuning the balance between hazards, defenses, ability cards, and the range of numbers across each unlockable "deck," I still found that having a 100% random system was too harsh for players. Matches would begin, and players would get crushed by a deluge of comets right away, or they'd find themselves unable to play any cards for a few turns. Fixing this required a multi-pronged approach.

Designing hidden training wheels

First, I opted to give players a free move on the very first turn of a game. This allows them to play any defense card that starts on the conveyor belt regardless of cost. It gives players a sporting chance to some kind of defense on the board and charged up before they get slammed with hazards. I also added code that ensures that there will always be at least one defense weapon card in the initial three card draw. This made a big difference, but it didn't solve the problem entirely.

Building onto this, I created a sub-system that acts like training wheels when cards are dealt. It uses variables to track how many hazard cards are currently on the conveyor belt when new cards are dealt. If there are more than two, it instantly shuffles the deck and picks a new card value before dealing. I've set it to cycle through this process a few times to ensure that most of the time the hazards don't stack up too much. It's not a perfect system, but it greatly reduced the number of times hazards would stack up.

RNG DESIGN TIPS

✓ Guided randomness is your friend

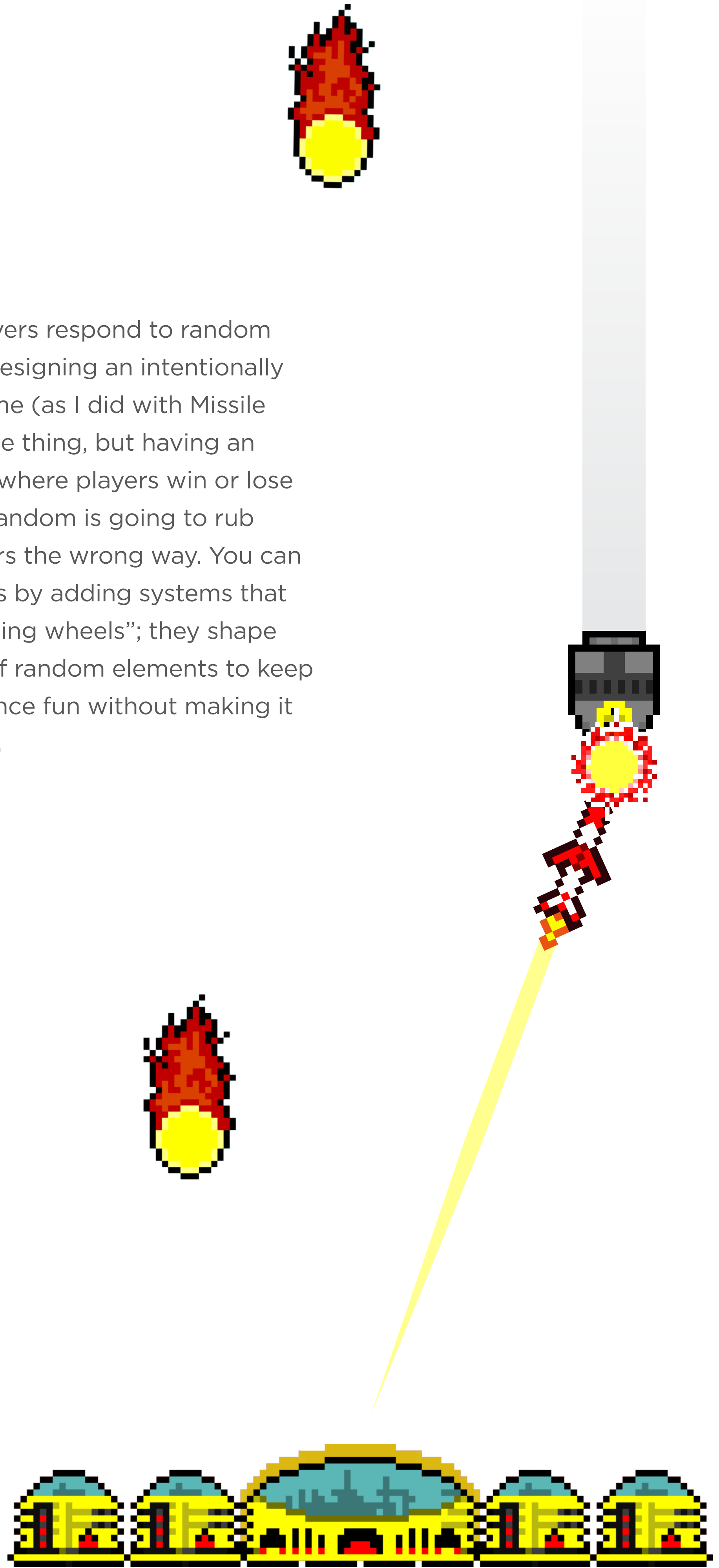
✓ Avoid arbitrary win/lose states caused by randomness

✓ Use pure randomness sparingly

Avoiding pure randomness

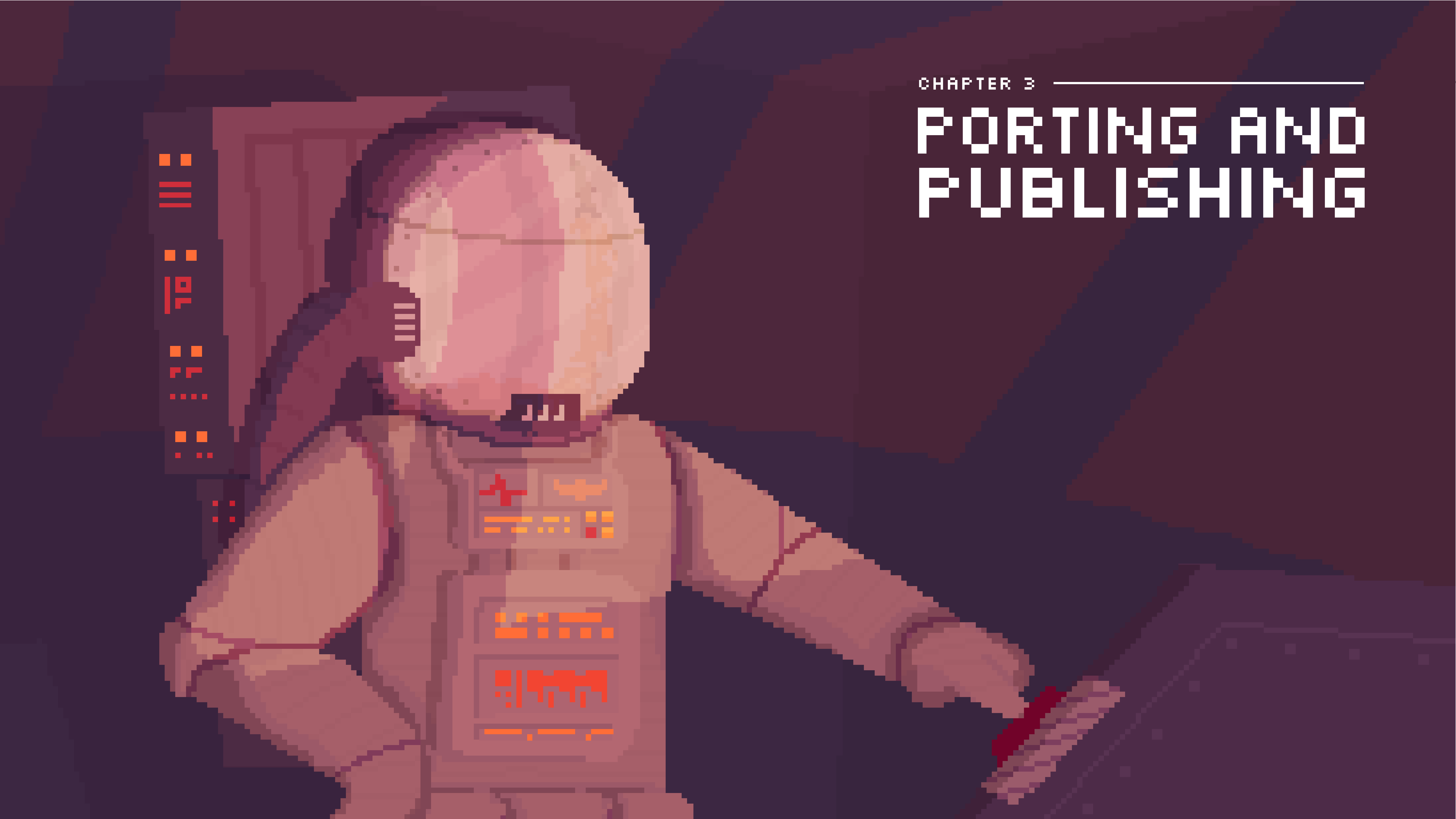
Random number generators play a role in other key Missile Cards system, too, like in choosing which lane comets and other hazards drop into on the battle grid. The card dealing system itself is where RNGs are used most, and where they have the biggest impact on gameplay.

The bottom line: randomness is a great tool for card games (and other genres alike), but it's useful to playtest your games a lot and get feedback on how players respond to random elements. Designing an intentionally difficult game (as I did with Missile Cards) is one thing, but having an experience where players win or lose entirely at random is going to rub many players the wrong way. You can mitigate this by adding systems that act as "training wheels"; they shape the scope of random elements to keep the experience fun without making it predictable**.**



RNG DESIGN TIPS

- ✓ Guided randomness is your friend
- ✓ Avoid arbitrary win/lose states caused by randomness
- ✓ Use pure randomness sparingly



CHAPTER 3

PORTING AND PUBLISHING

WHY PORTING TO ANDROID IS AN IMPORTANT CONSIDERATION FOR MOBILE GAMEDEVS

When you spend the time and energy to make a game, it makes a lot of sense not to limit yourself to just one engine. While it's easier to focus on one engine launch at a time, releasing on other engines can provide extra income (and visibility for your project) if things go well.

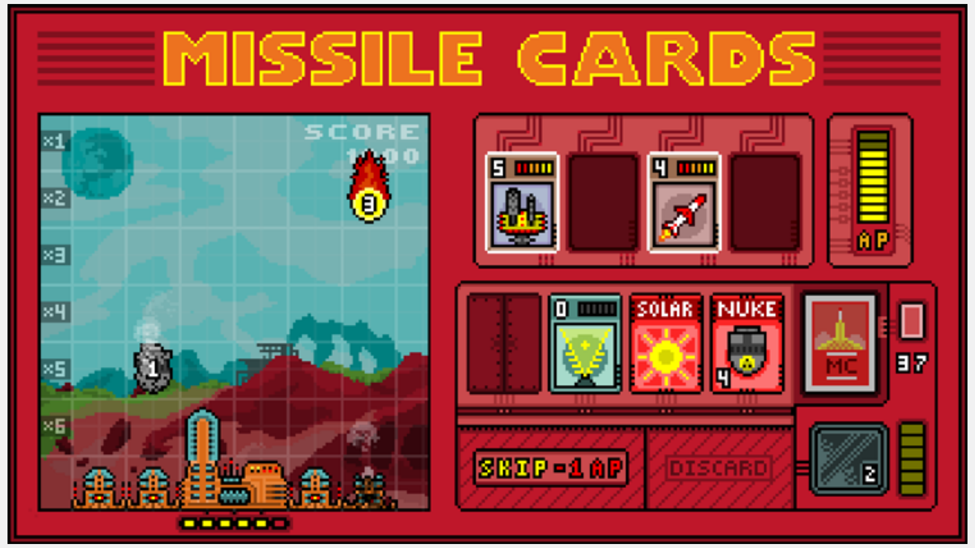
Porting a mobile game to Android doesn't have to be difficult, either. Design your games with multiple devices in mind. That way you have less work to do to get them up and running on other devices. GameMaker can help—GameMaker: Studio and GameMaker 2 both have mobile export modules that, once configured, let you easily export builds to a wide range of mobile devices.

ADAPTING MISSILE CARDS' PREMIUM DESIGN TO F2P

I originally designed Missile Cards to be a premium game: for a one-time purchase, you get a high level of replayability and fun. It takes skilled players between 7 and 10 hours to get through the game's five bases and bonus missions, which feels like a fair amount of content for a few bucks. The problem, however, is that the dominant pricing model across many Android marketplaces is free-to-play (F2P). This spurred me to make some tough decisions about how to handle pricing.

Missile Cards isn't optimized for micro-transactions and in-app purchases, common in F2P games. To use them effectively, they really must be incorporated during the design phase. Still, since most Android games featured on major storefronts are free to download, I didn't want to risk alienating players with an upfront price tag. My compromise was to offer Missile Cards as a free download with a single in-app purchase to unlock the full game.

Basically, players can sink their teeth into the first base (roughly an hour of replayable content) and access a limited selection of unlockable cards and abilities. Once they beat that base, they're asked to pay a one-time fee to access the rest of the game. This isn't the ideal approach for many F2P-modeled games, but it's the choice that most aligned with the way I originally designed Missile Cards.



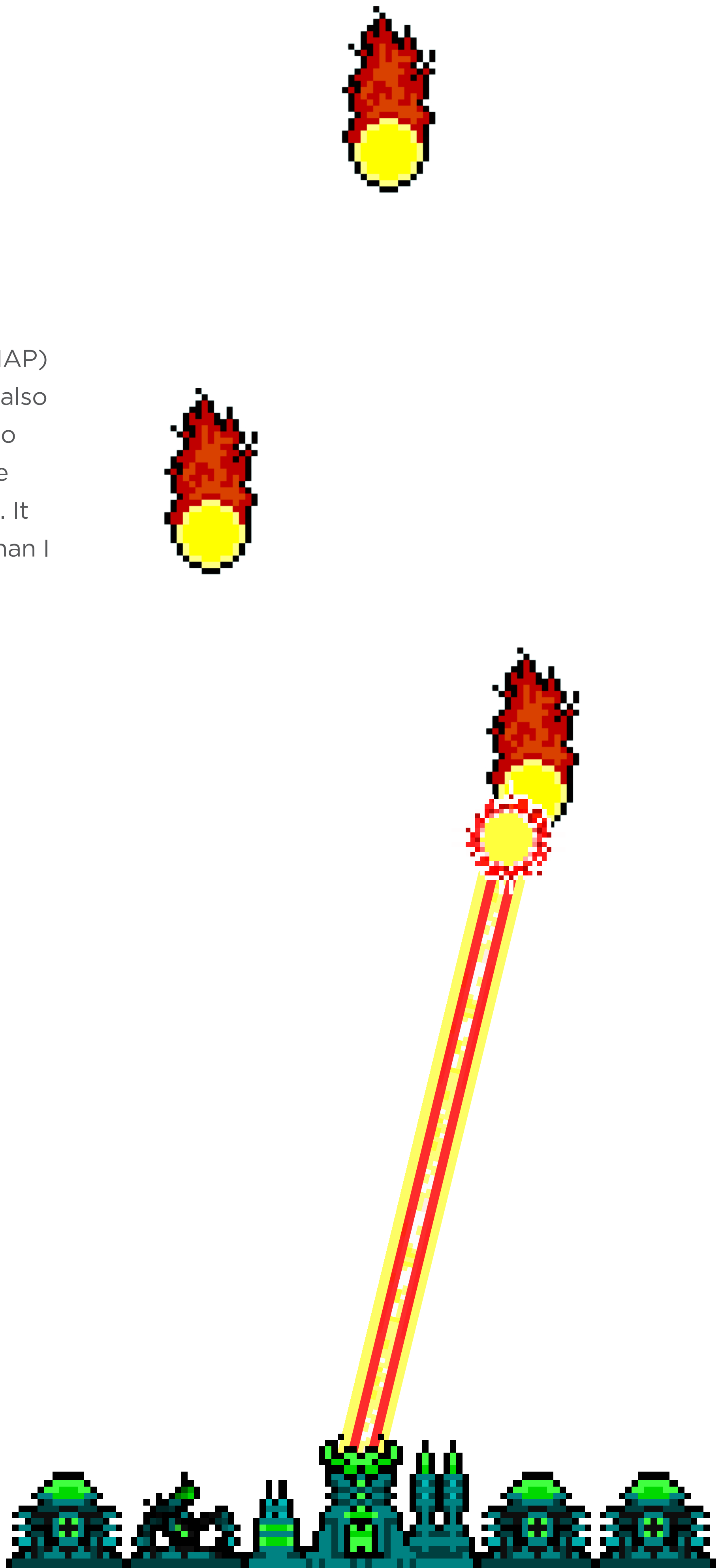
"IT TAKES SKILLED PLAYERS BETWEEN 7 AND 10 HOURS TO GET THROUGH THE GAME'S FIVE BASES AND BONUS MISSIONS."

MY EXPERIENCE PORTING TO AMAZON FIRE DEVICES

The process of porting your game can be a huge headache, but I was truly surprised at how easy it was to get Missile Cards up and running on Amazon Fire devices, which use the Android operating system.

The Kindle Fire is virtually plug-and-play, and it took me less than 30 minutes to get it all set up and start exporting my builds directly from my PC to the hardware for testing. The difference that made in my testing and porting process was night and day from past experiences. Downloading the Android Studio development environment and configuring the right settings was the most time-intensive task, but then I was able to plug in the Kindle Fire and press a single button to get a fresh build running in seconds.

Beyond that, I spent some time working on the in-app purchase (IAP) code to make Missile Cards F2P. I also added code to adjust the screen to properly fit Kindle devices and the most popular Android resolutions. It was a far more intuitive process than I expected.



"IT TOOK ME WEEKS OF FIDDLING WITH SETTINGS AND RUNNING INTO BRICK WALLS BEFORE I FINALLY FIGURED IT OUT."

LESSONS LEARNED WHILE PORTING TO THE AMAZON APP STORE

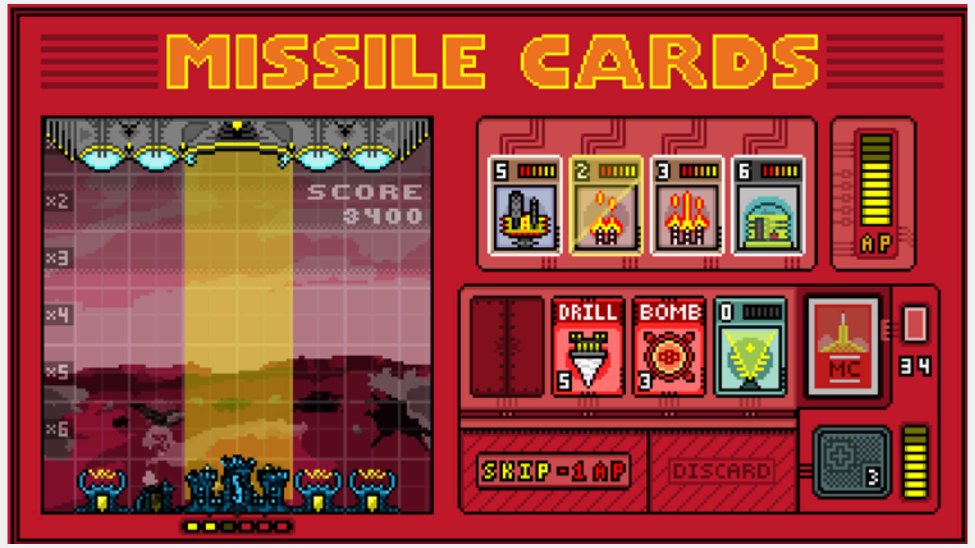
Porting Missile Cards to the Amazon Appstore was a fairly smooth process, but going through these steps again reinforced some core lessons that I've picked up with previous games. Here's a look at a few of the ones that stood out from this experience.

Choose your financial model early on

Designing your game systems so they're optimized for IAPs or ads means you have to structure gameplay in ways to support them. It's a lot easier doing this right from the beginning rather than shoehorning these systems in after the fact, like I had to do with Missile Cards. Even if you plan to have a premium release on one device and an IAP-supported version on another, thinking how you can work those systems into your gameplay early on in the process will save you a lot of hassle.

Your game's presentation on app storefronts is super important

From your game's title and description to choice of storefront imagery and icons, how you present your game to players is often just as important as designing the gameplay itself. It's useful to conduct lots of market research early in the design process by seeing how similar games position themselves and how well received they are by players and press. This can help give you ideas to make your presentation pop while avoiding potential pitfalls, too.



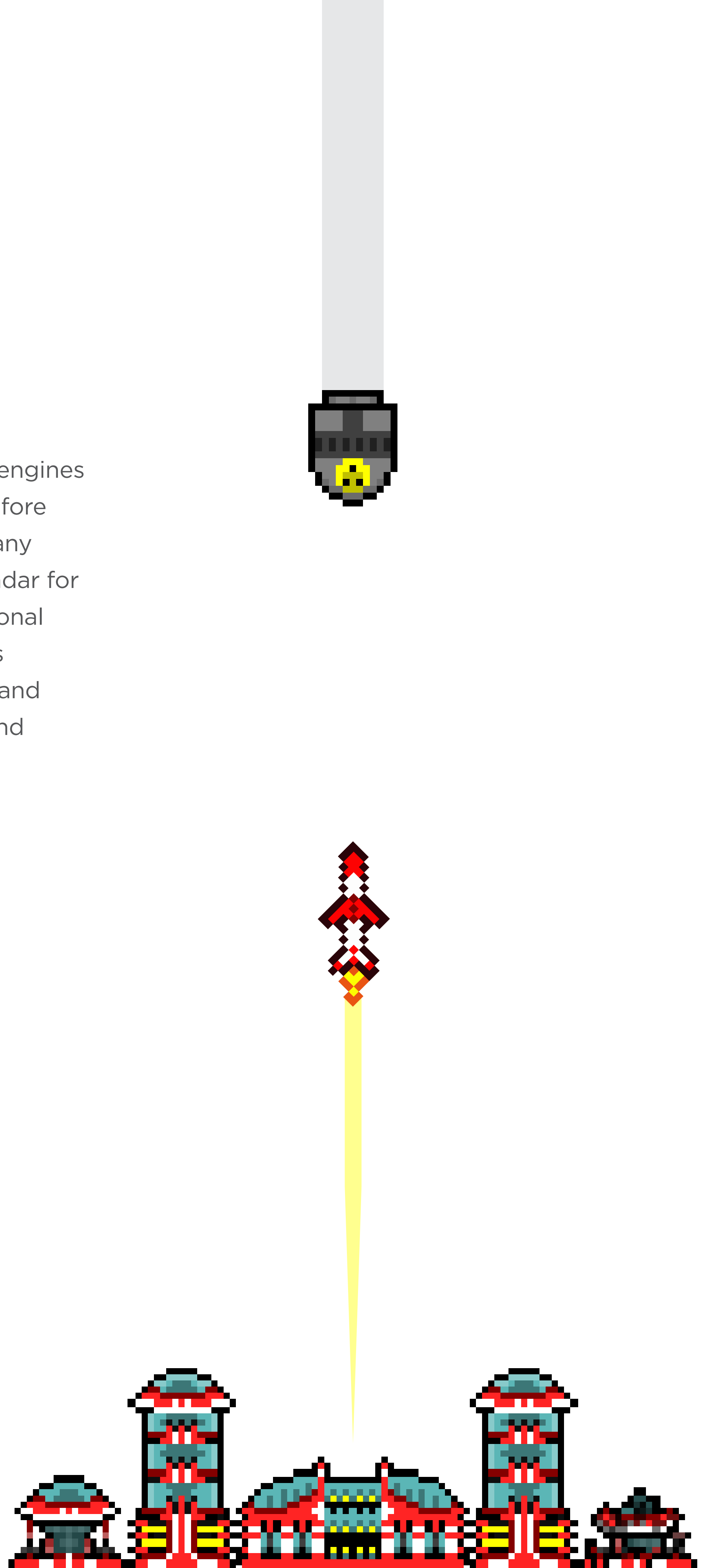
> "HOW YOU PRESENT YOUR GAME TO PLAYERS IS OFTEN JUST AS IMPORTANT AS DESIGNING THE GAMEPLAY ITSELF."

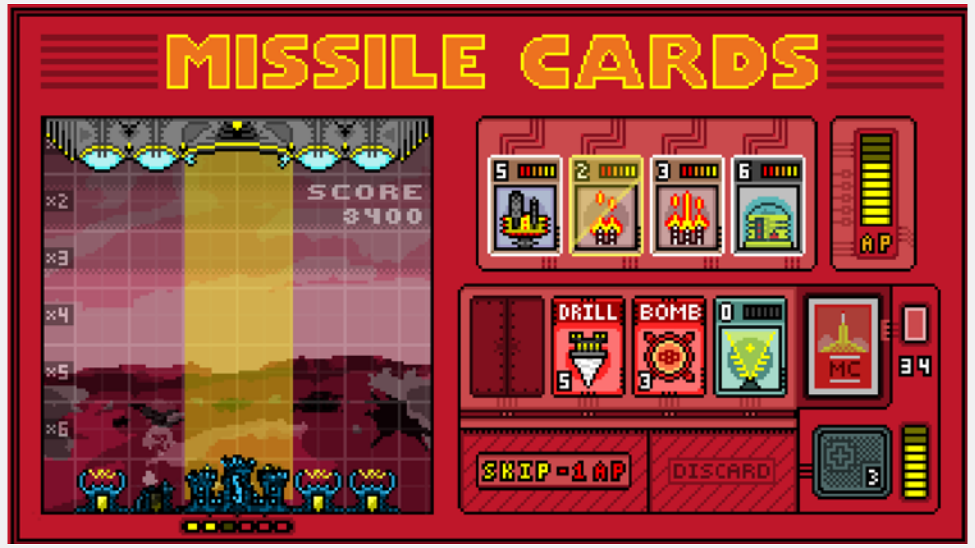
Have a solid launch plan prepped

Executing strong launch sales momentum requires having a plan and following through on it. Between press release and getting review code in the hands of influencers, to boosting your game through social media and creating interesting content to support your launch, this is a vital step that many indies overlook.

Partner with platforms whenever possible

It's always worth letting game engines know about your game well before launch and seeing if there are any opportunities to get on their radar for a feature slot or other promotional activities. Reach out as soon as you have a test build available and something polished to show, and don't forget to follow up!





"HOW YOU PRESENT YOUR GAME TO PLAYERS IS OFTEN JUST AS IMPORTANT AS DESIGNING THE GAMEPLAY ITSELF."

IN THE SIX MONTHS IT TOOK TO ORIGINALLY CREATE AND LAUNCH MISSILE CARDS, I EXPERIMENTED, LEARNED, AND GREW A LOT AS AN INDIE DEVELOPER.

The biggest takeaway I'd like to leave with you is that there can be so much value in narrowing your scope and making smaller games faster, particularly if you're planning to launch on mobile.

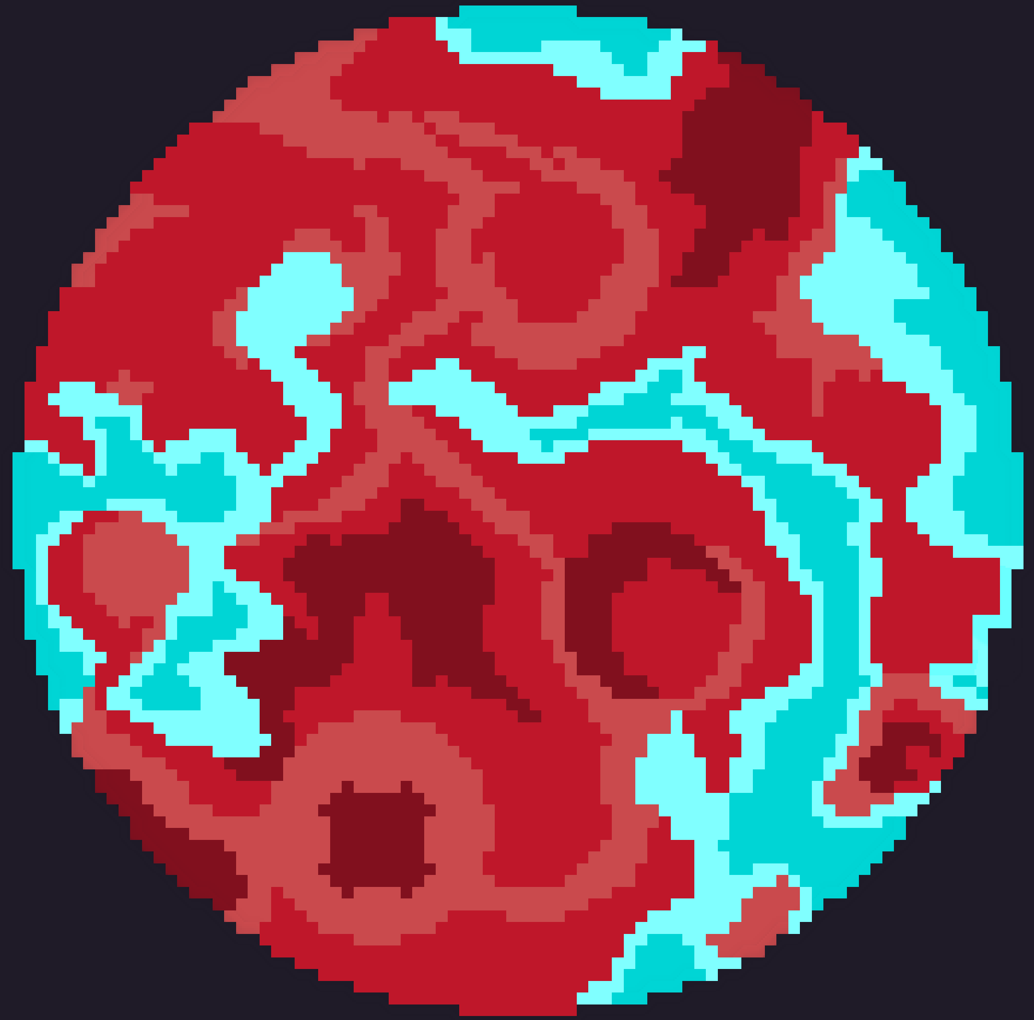
Market patterns shift wildly from year to year, and while there's certainly nothing wrong with taking a few years to craft a really great game, the sheer volume of high-quality games coming out each week means that making a great game alone isn't really good enough these days.

If you can craft something small and unique, and do it it in a fraction of the time it'd normally take, why not go for it? Some of the best gaming experiences I've had over the last few years were smaller mobile games that went out on a limb and dared to be different.

Making a game of any sort is a risky adventure, and there's never any guarantees that your project will gain traction.

But by shortening your development cycle, simplifying your process, and laser focusing on that distinct element that makes your game unique, you open the door to trying weird and unusual ideas while also minimizing risk of failure.

That can be incredibly empowering.



Free content to help intrepid app developers.

To acquire a free digital download of this book, visit the developer portal and search the title, “Behind the Scenes: Lessons Learned from the Making of Missile Cards.”

Content by Amazon.com, Inc.

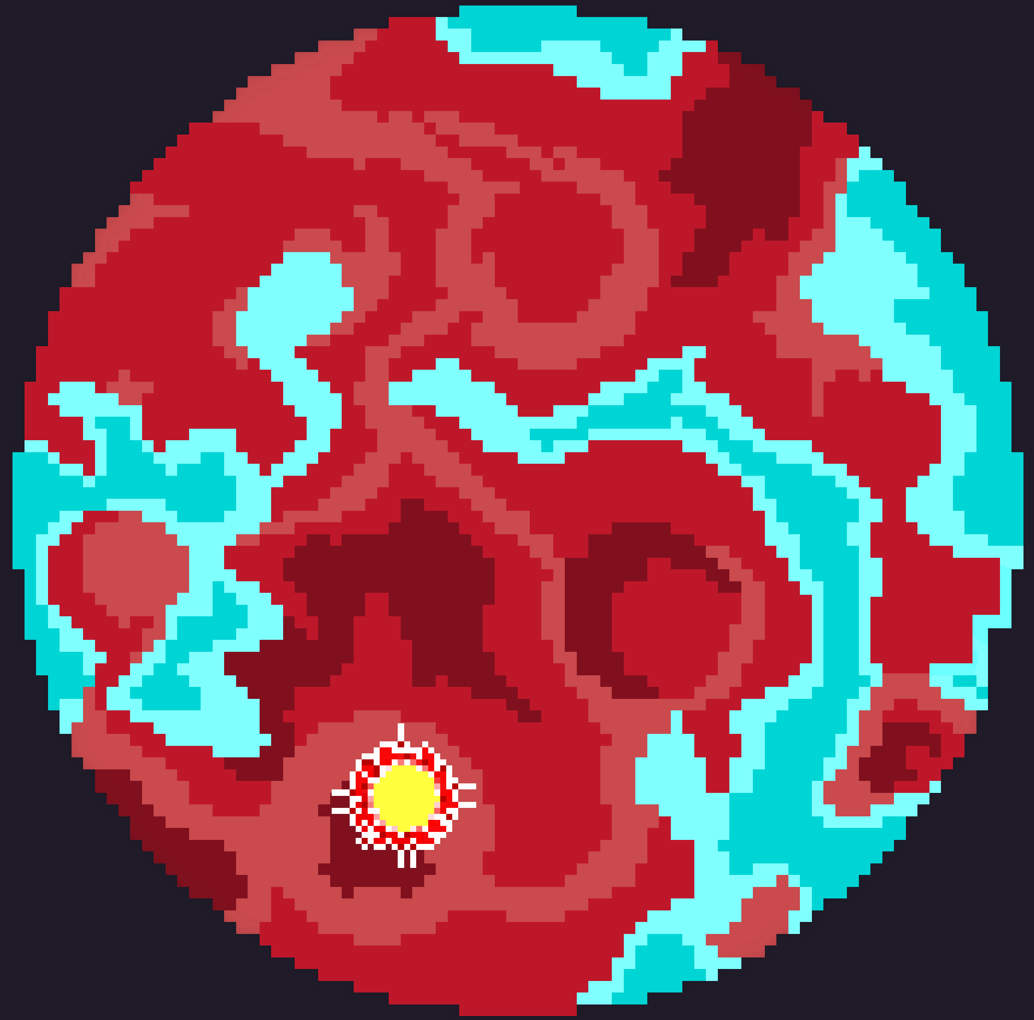
www.developer.amazon.com

Cover design, formatting, and editing by Graphiti.

The content of this book and the website links contained herein are simply for educational purposes, and are not intended to replace professional business advice or guarantee business outcomes. Every effort has been made to ensure that the content provided in this content is accurate and helpful for our readers at the time of publishing. However, this is not an exhaustive treatment of the subjects. No liability is assumed for losses or damages to the information provided. You are responsible for your own choices, actions, and results.

First edition

May 2018



THANK YOU