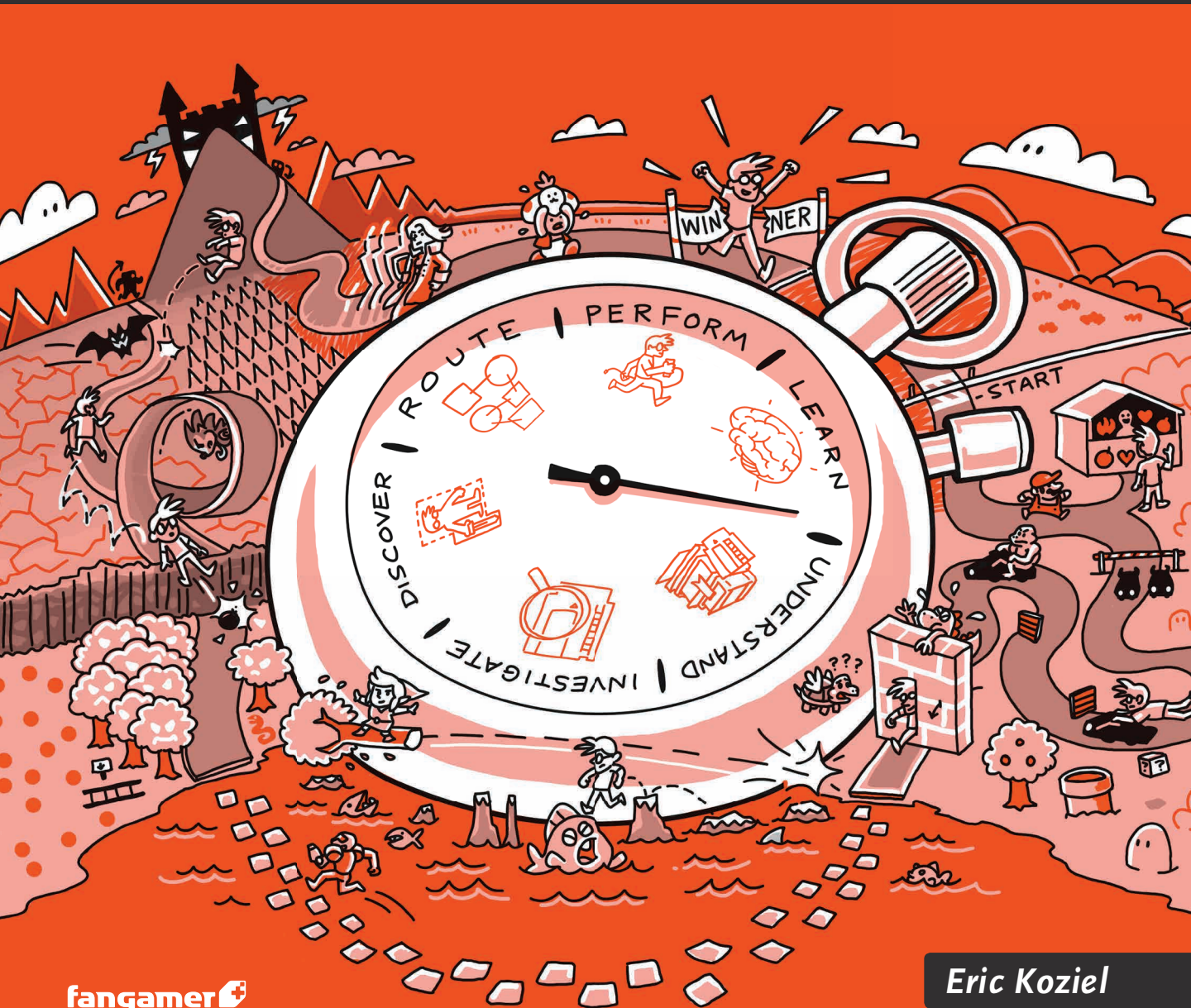


Speedrun Science

— A Long Guide to Short Playthroughs —



CONTENTS

Preface 6

Speedrunning Foundations 9

Speedrun Philosophy	10	Speedrunning History	72
What is a Speedrun?	19	Pre-History: Origins	74
What is a Run?.....	19	Birth of Gaming	76
Speedruns as Challenge Runs.....	20	Competition Arrives	80
Technical Definitions for Speedrunning.....	21	Doom Days	84
Putting all this together:.....	22	Quake Shakes up the Scene	88
Speedruns as Optimization Problems.....	23	Metroid Fever	91
Why Speedrun?.....	24	Morimoto Makes Waves	96
My Take: How to best enjoy speedrunning.....	30	Rise of the TAS	99
Run Timing	32	New Frontier:	
Speedrun Timer Issues.....	32	Expansion and Rule Definitions	105
Timer Characteristics.....	33	The Wonder Years:	
Timing Methodologies.....	34	The Life of an Early Speedrunner	110
Run Artifacts	36	The Streaming Age	114
Categories	39	Speedrun Marathons:	
Category Definitions.....	39	Games Done Quick Arrives	119
Category Creation.....	40	Modern Speedrunning:	
Category Creation Examples.....	43	The Great Decentralization	127
Glitches	45	Speedrun Diversification	130
Version Differences	49	Tournaments.....	130
Version Considerations.....	50	Blind Racing.....	131
Standardizing.....	54	Puzzle Speedruns.....	132
Emulators	55	Multi-Game Speedruns.....	133
Emulator Basics.....	55	Speedrunning Meta-Content.....	135
Emulator Issues.....	55	The Future of Speedrunning	138
Official Emulators.....	57		
Tool Assistance	58		
Cheating	61		
Speedrunning for Viewers	63		
Mirror System and You.....	63		
Mirror System and Speedrunning.....	64		
Speedrunning for Game Developers	67		

Speedrunning Process 145

Getting Started 146

Speedrunning Process Phases 148

The Speedrunning Mentality 151

Speedrunning Phase 0:
Picking Your Game 153

Investigation 156

Developing Understanding 159

Playing The Game 159

The World is Math 161

Understanding Luck 167

Scientific Method: Speedrunning Edition 170

Gathering Data 176

Discovery 185

Discovery 101: How to think 185

Discovery 201: Oversights and Abuses 189

Discovery 301: Using Advanced Tools 206

Discovery 401: Bug Hunting 220

When To Stop 231

Routing 234

Top-Down and Bottom-Up:
Incremental Design Strategy 237

Engineering Design Process: Speedrunning Edition 240

Focus Topics 251

Goal-building 251

Reducing Overhead 254

Resource Management & Rates 259

Planning for Consistency 266

Test Everything 271

Case Study: 999 273

When To Stop 282

Execution 284

Practicing: Insights and Techniques 287

Motivation 287

Chunking 289

Scaffolding 292

Repetition and Reinforcement 294

Frustration Management 296

Execution Optimization 300

Reducing Overhead 301

Developing Cues 304

Response Times 308

Lag Management 314

Racecar Turns 315

Mashing 318

Buffering 323

Camera Control 327

Performing 331

Keeping Cool 331

Target Setting 333

Splits 335

Reset Strategy 338

When To Be Done 341

Closing Remarks 344

Glossary 346

Preface

“How do I get started?” is one of the most common threads in any speedrunning forum. This is as true now as it was when I was just barely cutting my teeth on the hobby. Many new players are inspired to get started after watching a stellar run - the polished product of countless hours investigating, testing, and practicing. Unfortunately, they often don't get to witness the run's full evolution, and thus don't have a reference for what happens before the speedrun attempts even start.

Having experienced the rise from fledgling speedrunner myself, I first thought to make a series of guides to help people out. But what started as a simple list of resources and quick-start guide quickly grew into a huge web of topics, examples, and references. Writing a book had long been one of my life goals (though I assumed I would be writing fiction!), so this gave me the perfect opportunity to give it a try.

Even with the concept set in motion, the core idea behind the book grew and changed over time. I initially intended it to be a massive “How to Speedrun” guide, but I realized that a lot of the “what” of speedrunning was better supported alongside some explanation of “why.” Thus I expanded the book to include details on some of the history and fundamental concepts that form the backbone of the activity. Looking back, I'm glad I took this step; applying some critical thinking to what makes speedrunning worthwhile is crucial to becoming immersed in the hobby. Those sections also make the book more accessible to readers just looking to understand more about speedrunning.

A few years and a frightening quantity of caffeine later, what you now hold in your hands is the finished product. For me, writing it was eye-opening in a number of ways; the more you learn, the more you realize you don't know. I'm proud of the thoughts I was able to put to paper, but all told there's enough out there to double or triple the amount I've written. For every story and example I've included, another handful could have also been added.

I've divided the book's contents into two main pillars:

Speedrunning Foundations: a look at how speedrunning evolved from obscure concept to celebrated activity. This pillar is split into two parts - Philosophy covers the reasoning behind motivations and rules, and History tracks speedrunning's growth through the ages.

Speedrunning Process: a full-featured guide on learning to speedrun your game, from first playthrough to final split. I provide general instructions, examples, and advice for building up a speedrun from scratch.

One of my major goals in writing this was to make it accessible to everybody. Each section should be valuable whether you're an up-and-coming runner, a seasoned veteran, or even a casual viewer. Though I recommend reading through the book in its given order, there should be no problem jumping ahead to whatever chapters fit your current interest. I tried to provide examples from a wide variety of games and game types, but many examples will feature the 16-bit era games I am most familiar with. Even if you have no experience with the games I discuss, the mechanics can be generalized to many other games.

This undertaking has been helped along by far more people than I have the ability to thank, but I will do my best.

- ▶ The staff at Fangamer have been a tremendous help in taking my initial jumble of text and dressing it up into something wonderful. Thanks for believing in this project and helping me to realize it!
- ▶ I have a huge amount of gratitude for everybody who agreed to an interview. I couldn't have filled in all the blanks without your knowledge and experiences.
- ▶ Thanks, as always, to my family for supporting me throughout this and all of my other crazy projects. Love you guys!
- ▶ I could not have written this at all without the countless lessons learned from interacting with many different members of the speedrunning community. For every one thing I've figured out for myself, I've learned a dozen things just from watching other runners. Stay awesome, everybody!
- ▶ Finally, I want to dedicate this book to my father. He taught me that ability comes with understanding, and that creativity is key to accomplishment. But above all, he taught me to find the strength in myself to realize my goals. We love you, Dad.

This book doesn't cover all of speedrunning. Heck, it probably doesn't even touch on a fraction of what makes speedrunning fantastic. But it should have enough information to introduce, expand, and hone your understanding of the hobby. Whether this is a fresh start or a refresher course for you, I hope this book piques your interest and whets your appetite for diving further into this activity that I have so thoroughly appreciated. Enjoy!

~ *Eric 'Omnigamer' Koziel*



Putting all this together:

- ▶ Games are software that take in human inputs and output feedback at fixed, predictable rates.
- ▶ Reaching start and goal conditions can be defined by observing specific changes in the video feedback.
- ▶ Overall time corresponds to the number of video frames produced, which also corresponds to how many inputs a player has used to reach that goal.

VIDEO FRAMES

4 frames = 4 opportunities for player input

$$\frac{X \text{ frames}}{Y \text{ refresh rate}} = Z \text{ seconds of gameplay}$$

From Level 29 onward, the tetrominos in the NES version of Tetris fall one gridcell per frame. You can provide input on each frame, but the game's mechanics have the final say in how to react to those inputs! Just because you hold right doesn't mean the tetromino will move right each frame.



COMMON VIDEO FRAME TIMINGS

Older game consoles have very minor differences in how quickly they produce video frames. Although players won't notice the difference, it does matter for video recordings! Recording or playing back at the wrong rate can lead to discrepancies in the final time.

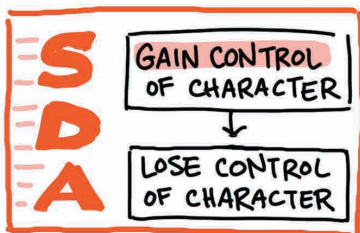
Bottom line: make sure you understand what your recording software is doing!

Note: Interlaced video rates are half of the listed progressive rates.

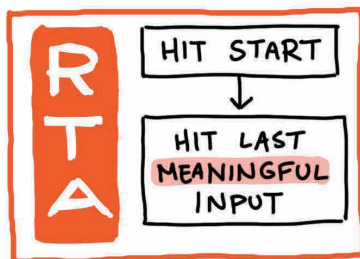
Console	NTSC (Hz)	PAL (Hz)
Standard	59.940	50.000
Atari 2600/7800	59.923	49.861
Commodore 64	59.826	50.125
NES & SNES	60.099	50.007
Gameboy/Color/Advance	59.728	59.728
SMS/Genesis	59.923	49.701
Playstation	59.293	49.765
Modern Consoles	59.940	59.940

Timing Methodologies

There are three major timing methodologies used across the general speedrunning community. Individual game communities make their own variations as needed, but usually they are only slight deviations from these major archetypes.



SDA Timing - The methodology laid out by speedrunning website Speed Demos Archive uses real time and starts when the player first gains control of their character. Timing ends when control of the character is lost at the end of the game, even if that comes well after the last meaningful input. The goal of SDA timing is to capture the time most related to player actions.



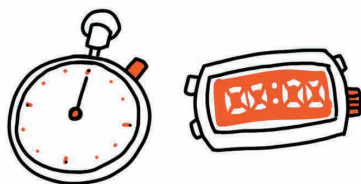
RTA Timing - Real-time Attack (RTA) is a real-time methodology that starts when the player first hits “Start” or an equivalent option on the main menu. Timing usually ends on the last meaningful action. RTA is the most popular timing method, and is used on a majority of games in modern speedrunning.



In-Game Timer (IGT) - Timing is entirely handled by the game’s self-implemented timer. The characteristics of the timer can vary greatly from game to game. This method is the simplest, when available, but usually carries a number of caveats as well.

While SDA and RTA timing are similar, they are adapted for different purposes. SDA’s timing centers around finding consistent starting and stopping points so that true frame counts can be found for comparison; RTA timing developed largely out of convenience for streaming, as the timer and first input could be aligned without additional stress on the player. As a result it is more difficult to compare RTA runs at the frame level, as the exact frame an attack does damage to a boss can be ambiguous when comparing two distinct runs. Even so, RTA timing to a resolution of seconds or centiseconds is sufficient to compare the vast majority of runs.

IGT is a bit tricky, compared to RTA and SDA timing, since it relies on coded game logic. This often means that only certain types of gameplay are counted. For example, an in-game timer may be active during normal gameplay but stopped during loading sequences. Not counting loading can be a great benefit, especially when loading time is dependent on the quality of your platform. In that case, though, it would no longer be necessary to minimize



the number of loading instances. This means that a route built for IGT could be quite slow in real time.

The other factor affecting the usefulness of IGT is lack of resolution. Many older games report times in terms of hours and minutes, which can be unacceptable for a highly competitive game. Super Metroid and older Pokémon games are prominent examples of this limitation.



SUPER METROID



TEENAGE MUTANT NINJA TURTLES IV: TURTLES IN TIME

	Level 26
Date	2013/08/06 00:58:54
Total Play Time	03:15:11
Location	Library
	Level 17
Date	2013/07/20 01:36:38
Total Play Time	02:05:49
Location	Underground Facility: B3
	Level 3
Date	2013/07/10 01:20:18
Total Play Time	00:21:53
Location	Lost Shrine: Roof

NIER



SILENT HILL

THE MANY FLAVORS OF IGT

SUPER METROID - The mission timer is accurate, but the resolution only goes to minutes. Since the timer also does not include pauses and screen transitions, IGT requires different optimization strategies vs real time methods.

TEENAGE MUTANT NINJA TURTLES IV: TURTLES IN TIME - Though the internal game timer is accurate to all sequences of player control, you'd never know it! Due to a bug in the timer display code, the game time readings are unreliable; that 11:21 is in fact 17:33!

NIER - The in-game timer in Nier tracks time according to the system clock, so it includes all loading scenes and cutscenes. This makes it functionally equivalent to real-time methods, but also means it is affected by differences in system loading speed.

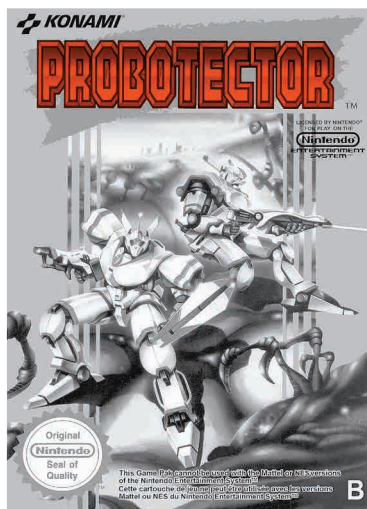
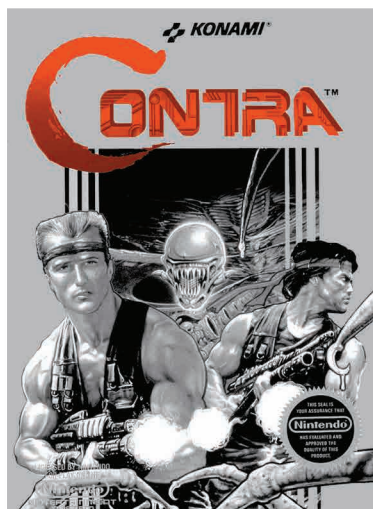
SILENT HILL - This timer excludes loading sequences and cutscenes, meaning only sections where the player is in control are timed. It also isn't affected by differences in system performance as a result. This type of timer is the most reliable for cross-system comparisons and consistency.

Timing methods should always be chosen according to what makes sense for the game at hand. Always remember that the goal of a timer is to measure player skill and provide a metric for two separate runs to be objectively compared. The standard real-time or game-time methods provide a decent starting point, but you shouldn't be afraid to get creative and design your own timing method if that's what makes sense for your game.

Version Differences

In scientific research and academia, experimental results are expected to be listed alongside explicit details about how the experiment was designed and carried out. This serves two major purposes: it allows a separate group to independently verify the results by repeating the entire process, and it allows other observers to identify potential flaws or differences when compared with other existing results. Put another way, it provides a way to ensure independent results can be given a proper apples-to-apples comparison. In the presence of different experimental methodologies, it may also offer a way to translate the results from one methodology to another.

This concept also applies to speedrunning, but with a bit of a twist. If you treat every run as an experiment, and every time as a result, you



LEFT: What the US knows as Contra was released as Probotector in Europe, with completely remade graphics. Beyond the NTSC and PAL frame rate differences, though, the games should be comparable.

BELOW: Mega Man X3 was released on both SNES and Sega Saturn, but the differences in loading make them not directly comparable.



THE IQUE



Way back in 2012, controversy arose around a gaming device known as the iQue. It was an officially-licensed system for playing select Nintendo 64 games, and was released only in China. Among them: Ocarina of Time.

Though largely the same game, the iQue handled lag differently than original Nintendo 64s, leading to a minor time advantage. Thus, for a time, the only way to compete for the Ocarina of Time record was to obtain one of these incredibly obscure systems. Whether runs on iQue should have been directly compared with other consoles is still an open question, but it certainly created a new barrier of entry for the record challengers of the era.

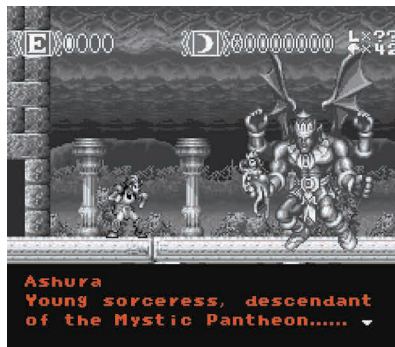
Localization differences aside, some languages just need fewer characters to express the same ideas. In Skyblazer and its Japanese counterpart, Karura Ou, new characters are written once per frame. This same text block is more than half a second slower in the US release compared with the Japanese version!

want to ensure that each separate run is comparable to others. For the most part this is straightforward; run a game on a specific platform, and you should have a time you can compare with anybody else who runs the game on the same platform. But what if the platform is different, such as when the same game is released on both PlayStation 3 and Xbox 360? What if you wanted to compare the Japanese release of a game with the United States release?

This leads us into the murky world of version differences for games. At the end of the day, runners just want to have consistent, comparable times for the purpose of determining which runs are more skillfully executed. If the runs are on different versions, this can be quite challenging to actually determine. Even when they can't be directly compared, it's also important to know specifically what is different between the versions and runs. This covers both trying to know which version is ultimately faster as well as what the optimization limitations are for each.

Version Considerations

For console games, version differences typically comprise different language releases and occasional code updates to the core game. In some cases, the two are one and the same; a game first released in Japan may see some updates or changes to in-game mechanics before its later release in the US or Europe, for example. There are no rules for identifying or generalizing differences; the only way to know is to play the different versions and identify what can and can't be done between them. There are definitely specific things to look for, however; I enumerate some of the most common in the rest of the section, along with a few examples.



SKYBLAZER (USA)
50 individual text characters



KARURA OU (Japan)
11 individual text characters

The most notable difference when trying to compare different versions usually depends on the language itself. Different languages express the same concepts in more or fewer characters. Languages like Japanese, for example, can fit entire syllables or meanings into a single character, which drastically cuts down on the total number of characters that must be displayed during a speedrun. If each character takes a set number of frames to display, this can lead to a significant difference in final times between two versions, even if all of the other gameplay content is the same. This type of difference is usually the simplest to make a conversion for.

Another significant and common difference is that of video refresh rate. Historically, Japan and the US both used the NTSC video standard, which operates at 60 Hz. This meant that most games will show a new video frame every 60th of a second. Europe, Australia, and some other regions of the world conformed to the PAL or SECAM video standards, which operated at 50 Hz. This means that, if nothing was changed about the underlying game, the same game running on PAL consoles will be operating at 5/6th the speed of its NTSC counterpart.

For this reason, most speedrunners prefer NTSC versions of games where possible. PAL times may still be comparable if the video rate is the only difference; simply multiply the PAL time by 6/5 to get an equivalent NTSC time. However, it is very difficult to substantiate the claim that nothing else is different between an NTSC game and its PAL counterpart; developers may have changed movement speed or other mechanics to make the two versions visually equivalent. In other cases, critical game mechanics such as random number generation may also be affected.



TIME CONVERSION EXAMPLE

	Version A	Version B
Characters	1000	1600
Time per Char	0.05	0.05
Final Time	14:40	15:00

Version A of a game displays 1000 characters over the course of a speedrun, and **version B** displays 1600 characters for the same run. If there are no other version differences, and it takes .05 seconds to display each character, what's the better overall run: A 14:40 run on **version A** or a 15:00 run on **version B**? **(ANSWER BELOW)**

ANSWER: the version B run is equivalent to a 14:30 run on version A, thus is better. The text differences correspond to a fixed 30 second difference to convert between versions A and B

VIDEO STANDARDS

Analog video formats around the world vary from country to country, but all fall into categories of NTSC, PAL, and SECAM. NTSC regions operate at 60 Hz, while PAL and SECAM regions typically have 50 Hz analog video. There are plenty of exceptions though - for example, Brazil uses the PAL standard, but operates at 60 Hz. Look up details on your regional video standard if you aren't sure!

Competition Arrives

In March of 1990, Nintendo of America staged an event in Dallas, Texas that they called the “Nintendo World Championships.” While this was mainly a marketing event to capture and further motivate the explosive success of the NES, it grew into a full-on circuit. In the United States, the event toured 29 different cities before culminating in a final competition among all the winners of the city contests. It was everything a kid (or adult) could want: games, a chance to test your mettle against countless others, an opportunity for prizes,

The schedule for the NWC is now being firmed up, and here are the dates:

Date	City	Location
3/19-3/11	Dallas, TX	Fairpark Convention Center
3/15-3/18	Cleveland, OH	Public Hall Convention Center
3/22-3/25	Pittsburgh, PA	D. Lawrence Convention Center
3/29-4/1	Detroit, MI	Pontiac Silverdome
4/5-4/8	Philadelphia, PA	Convention Center Complex
4/12-4/15	Indianapolis, IN	Convention Center
4/19-4/23	Boston, MA	World Trade Convention Center
4/26-4/29	Uniondale, NY	Nassau Convention Center
5/3-5/6	Hartford, CT	Civic Center Exhibition Hall
5/10-5/13	Chicago, IL	Rosemont Exhibition Hall
5/17-5/20	Kansas City, MO	Convention Center

As we mentioned last issue, if you would like additional information on the NWC, call the NWC hotline at 1-900-HOT-4NWC (1-900-468-4692)

ABOVE: The cross-country tour of the NWC featured a new city every week, finishing out at the World Finals in Los Angeles from December 7-9, 1990.

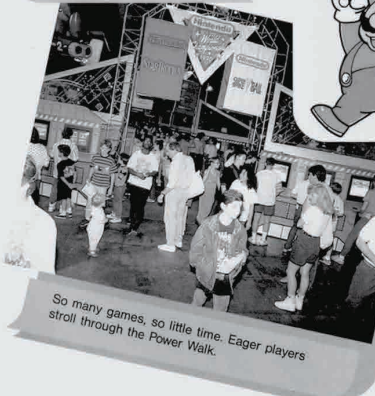
RIGHT: NWC events featured an abundance of game demo stations, a spectacle of inflatable Nintendo characters, a variety of activity booths, and of course the main event.



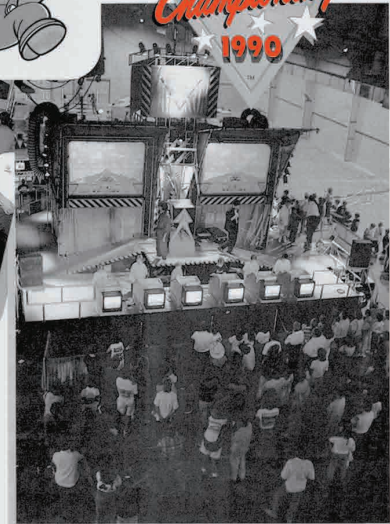
A familiar friend welcomes visitors to the Nintendo PowerFest 1990.



It's a fun time for everyone, playing and competing.



So many games, so little time. Eager players stroll through the Power Walk.



Participants zip through Rad Racer on the Competition Stage.

Some Nintendo PowerFest 1990 attendees participate in the Nintendo World Championships, which are staged at the show. Competitors have six minutes to collect 50 coins in Super Mario Bros., complete the first track in Rad Racer and get as many points playing Tetris as they can in the remaining time. Three finalists from each city are invited to the national Nintendo World Championships in Orlando, Florida which will be held in December. Check your local newspaper to see if the Nintendo PowerFest is coming your way or call 1-900-HOT-4NWC (1-900-468-4692) for information. A charge of 50 cents per minute will be added to your phone bill, so make sure you have permission from whoever's paying the bill.

and even a shot at fame. It also inadvertently became the first major push for speedrun-oriented competition and strategy.

The competition used a specially crafted cartridge that had players competing for highest total score across three games: Super Mario Bros., Rad Racer, and Tetris. But there were additional requirements for each game, a fixed time limit of 6 minutes 21 seconds for the entire attempt, and varying score multipliers as well:

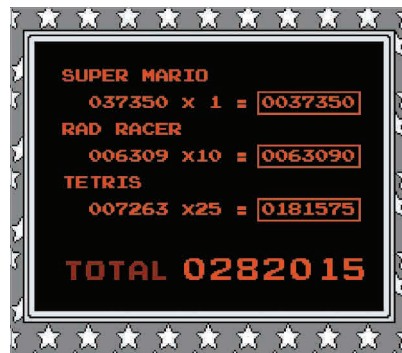
- ▶ Super Mario Bros.: Collect 50 coins. Score x1
- ▶ Rad Racer: Complete the Course! Score x10
- ▶ Tetris: Get High Score! Score x25

In the Nintendo World Championship cartridge, the player automatically advances from one game to the next when the goal is met, and their final score is the sum of what they earned in each individual game after the time limit is hit.

Many competitors quickly noticed a critical facet of the scoring: The disparity between the score multiplier for Tetris and the other two games. Doing even reasonably well in Tetris would dwarf your combined score from the other two games! The strategy became to complete Super Mario Bros. and Rad Racer as quickly as possible, to allow for a much more beneficial scoring opportunity in Tetris. The final time limit also spurred competitors to play for speed in Tetris, as a player's placement rate directly related to their opportunities to score.

The Nintendo World Championships had unintentionally included all the makings of what we would consider a modern speedrun: Learning the fundamentals behind your targeted game(s) and mechanics, exploring strategies for optimizing the time taken, coming up with a best route, and then finally performing that route. The scoring system and time limit devised by Nintendo became the perfect ingredients for spurring players to orient their play toward speed. While the event itself was built around total score, the Nintendo World Championships have a place in history as one of the earliest instances of organized speedrunning.

The other major impact from the Nintendo World Championships was simply the centralization and coordination of rules and times for a challenge race. The extra rules imposed on players were straightforward, but the fact that they were enforced by the game



ABOVE: Scoring screen from the Nintendo World Championships cartridge

BELOW: Though the champions of each age group never directly competed against each other in the main event, they held an informal competition afterward. Thor came out on top, followed by Jeff, then Robert.

The PowerFest concluded in December with the Nintendo World Championship finals held at Universal Studios in Los Angeles. Was the competition stiff? Bet on it! With 90 regional champions from across the country vying for national titles, the PowerFest was the definitive match-up. All the competitors were winners, but only one per age group could be named Nintendo World Champion. The 1990 champs are as follows:

11 & Under

Jeff Hansen Score: 2,009,950
12-17

Thor Aackerlund Score: 2,809,995
18 & Over

Robert Whiteman Score: 1,956,960

Speedrun Marathons: Games Done Quick Arrives

In November 2007, Internet comedy group LoadingReadyRun devised a new kind of event: a continuous gameplay session, streamed via early streaming service UStream, to raise funds for the charity Child's Play. The game of choice was the faux-game Desert Bus, which was a minigame in the unreleased Penn & Teller's Smoke and Mirrors for Sega CD. Desert Bus simulated the real-time experience of driving a bus that makes consecutive 8-hour trips endlessly. The premise was simple enough: donors would watch and interact with the LoadingReadyRun crew as they endured this supremely tedious task, with donations directly leading to extending the play-time. This initial event, "Desert Bus for Hope," raised \$22,805. The style of event was dubbed a "marathon," in reference to its similarity with the grueling 26.2 mile footrace.

The following year, a group calling itself The Speed Gamers (TSG) borrowed the marathon concept and gave it a speedrun-oriented spin. In March 2008, they hosted the first speedrun-oriented marathon in support of charity. Specifically, the event played through every main Zelda title within 72 hours while raising funds for St. Jude's Children's Research Hospital. They would go on to host 7 more marathons with different themes throughout 2008, and kept up the pace in 2009. Though the events were successful, the general audience for streams was fairly small. The concept of streamed games, let alone streamed marathon events of games, was still too new to attract viral attention.

In mid-2009, the marathon phenomenon was gaining notoriety among the users at Speed Demos Archive. In particular, user and site administrator Mike Uyama took to formalizing the myriad ideas of other users and defining the structure and timing of an SDA-oriented marathon.

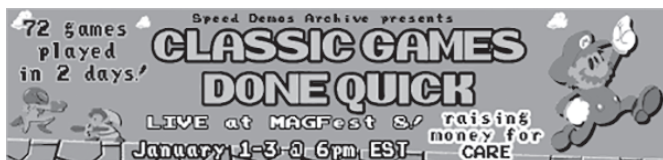
"In Summer of 2009, people on the forums saw these TSG marathons, and they went 'we should do these marathons ourselves!' People kept suggesting games they could run, and all these other ideas, and it just got out of hand really

Marathon

A number of speedruns performed consecutively, usually as one-shot attempts, and often featuring a variety of different games and runners.



Mike Uyama participating at AGDQ '12



fast. [When I stepped in] I said ‘whoa... this is a good idea, but we need to give it some focus.’ And so, I decided two things on the spot. One thing was that it should take place at [gaming convention] MAGFest, because I thought it would be a good idea to have it at an established convention for more structure, and to have other things for us to do like play arcade games and stuff like that. The other thing I decided was that based on all the suggestions from forum members for what games they could do, they were very classic gaming focused. So I decided to make the event 8- and 16-bit gaming focused, hence the name Classic Games Done Quick (CGDQ).”

RIGHT, TOP: The original cast and crew for CGDQ! Most of them, at least. Image courtesy of EmptyEye.

.....

RIGHT, BOTTOM: CGDQ's final destination was Mike's basement. Though a bit cramped for the 20+ participants, the cozy atmosphere definitely added to the marathon's charm. Image courtesy of Mike89.



The <name> Done Quick moniker was carried over from the early Quake Done Quick projects.

“The preparation I did for the event was mainly trying to get the [schedule] full. I was aiming for two days of continuous games, so I had to figure out who was coming to the event, who could play what games, and so on. There was only 20 people or so who were coming to the event[, so there was a lot to coordinate and figure out]. I also talked with the MAGFest admins to make sure we had a space and Internet for it.”

Despite the organization leading up to the event, all manner of things went wrong when MAGFest actually rolled around. The largest immediate issue was a sudden change in Internet availability due to the provider being bought out by a different company. The day before the actual event took place, Mike scrambled to find a workable solution, but even MAGFest staff had not yet identified a workaround. Disheartened, Mike returned home.

“That was probably the most miserable I had been on a New Year’s Eve, because nothing worked, we couldn’t test the Internet, and we had no idea how the event was going to happen. So, I remember going home, and Frezyman, a Swedish runner [who I was hosting at my home], told me, ‘Mike, just have a beer.’ And that was good advice. So I just drank my beer sullenly, and went to bed.

“The next day, since we were at my Mom’s house, we decided to try something. It was 10 AM, and the event was supposed to start at 6 PM, so we decided to try and make sure our capture setup works. We took about 30 minutes to test our capture setup and test the Internet, and it was working. Everything was smooth, not too fancy, but it was working. The webcam was working, and you could see the game capture. So this upped our spirits a bit.

“When we got to MAGFest, the Internet situation hadn’t been resolved. We had to work through the hotel Internet, but because of a contracting agreement, the hotel was restricted to a low-bandwidth connection. We eventually got set up, but between how loud the venue was and the poor connection, we couldn’t make it work out. So, I decided that one place that everything did work was my Mom’s basement. So instead of doing it at MAGFest, we hauled all the equipment back to my

house. We still had to scramble to set up the equipment, and get a spreadsheet to track donations. I had never been so stressed in my life. Miraculously, though, the stream starts, and the rest is history.”

Classic Games Done Quick went on to raise \$10,531 in support of the charity CARE. This initial step, though, formed the basics of what would become the speedrun marathon formula for years to come.



The first Awesome Games Done Quick in 2011 featured a new location and an expanded group of players and volunteers. Image courtesy of trekhaak.

The same group from SDA went on to host Awesome Games Done Quick the following year, in January 2011, in support of the Prevent Cancer Foundation. Mike and event staff learned plenty from their experiences with CGDQ at MAGFest, and organized the event at a separate venue, the 4-H Center in Chevy Chase, Maryland. The expansion of “Classic” to “Awesome” also was intended to be more inclusive of newer games such as Portal, Halo, and Super Mario Galaxy 2. The event was also lengthened to five full days of

gameplay, more than double the length of CGDQ. The extended length and more diverse schedule led to a more successful event, this time raising more than \$50,000. And the staff were hooked.

“Right after the event, there was this realization that ‘Hey, this was really fun. It had a real magic that speedrunning didn’t have before. Let’s do it again!’ I’ve made a lot of mistakes from each marathon, because there’s always a growth and learning process. But I’ve also had a lot of success and grown the community a lot, because each marathon gets bigger and bigger.”

Awesome Games Done Quick became an annual event, occurring every January. It even spawned a sister event, organized by the same general staff, called Summer Games Done Quick. The event growth was explosive, with each consecutive event more than doubling the prior year’s donation total and viewership until 2015. With this massive growth came notoriety and broader exposure, which brought more and more viewers into the speedrunning fold.

Perhaps the most significant impact that marathons had on the actual hobby of speedrunning was giving it a tangible purpose. Runners generally understood that regardless of how much they optimized their runs and games, the result wouldn't have much effect on the world. They would get the satisfaction of completing their project, and the bragging rights that come with any significant achievement, but in the end it was still just an arbitrary challenge in a video game. With charity marathons, however, their efforts could have a real, lasting effect on the world. Not only did that inspire an incredible emotion in many runners, it also became a major motivation for perfecting their runs.

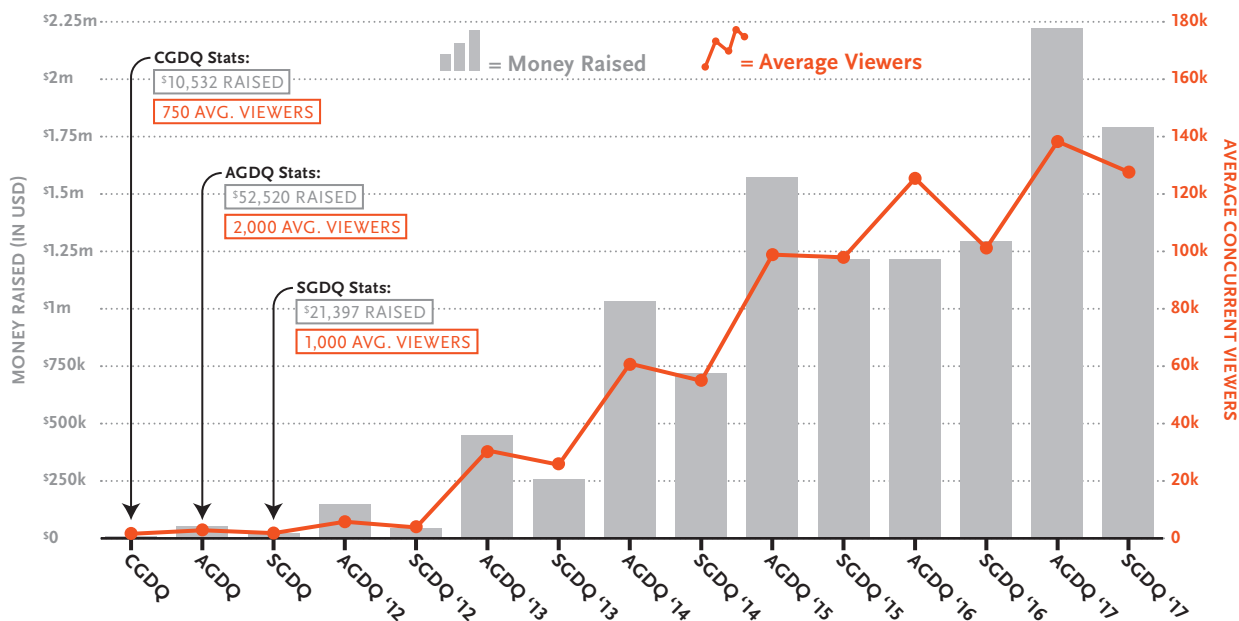


ABOVE: AGDQ returned to the 4-H Center for its 2012 edition, with nearly double the participants.

Of course, with increased growth and focus came growing pains to suit. GDQ organizers quickly became overwhelmed with new submissions for each event. What started as staff struggling to fill in a 2-day schedule had become a very competitive submission process to fill a 150-hour schedule. Technical and organizational needs grew until it reached a tipping point at AGDQ 2014.

BELOW: The growth of GDQ events was explosive. Year-over-year, donation totals and viewership took off.

GDQ GROWTH BY TOTAL DONATIONS & AVERAGE VIEWERS



AUTHOR'S NOTE: MY AGDQ '13 EXPERIENCE

AGDQ 2013 holds a special place in my heart since it was the first speedrunning meetup I attended. I had been active in the community for about a year by this point, and had interacted with many other runners through SDA or their streams. Even so, I remember being nervous about trying to make a good first impression on my peers. Once I arrived, though, it didn't take long to realize we were all just a bunch of goof-balls joined by the common passion for speedrunning.

The marathon itself was wonderful, but was a lot more demanding than I expected. Everybody was called on to help in some way, whether it was operating the stream, setting up equipment, carrying TVs around, or even just filling in seats or adding flavor commentary. Since these events run 24/7, there was always something to do, and a lot of things that needed fixes on the fly. I mainly helped with operating the stream and setting up consoles and computers for runners - including somehow setting up a 4-player co-op run of Left 4 Dead! It was all very draining, but definitely rewarding from how much the event was bringing in for charity.

However, most of the real stories took place outside of the stream room. I was amazed at how creative and enthusiastic many runners were; from hallway push-up races to mystery game routing challenges to "buddy" races in Super Mario World, all manner of silly and exciting activities were always going on. Even normally mundane things like going out to grab food served as a great way to share stories and bolster camaraderie with my fellow speedsters. These things helped me to really feel like I belonged, and fostered my adoration of the hobby further.

In retrospect, I was really fortunate to experience that phase of GDQ events. AGDQ '13 was before the \$1M milestone was breached and speedrunning really opened up to the masses, but it was also among the last events where I could easily meet everybody attending. The event growth that followed was necessary to match the pace of new interest, but the grassroots flavor of '13s organization served as a model for a successful and enjoyable event - both on stream and off.



ABOVE: The stream view of my run of crowd-favorite bad game Timecop. I also ran Skyblazer at this event.

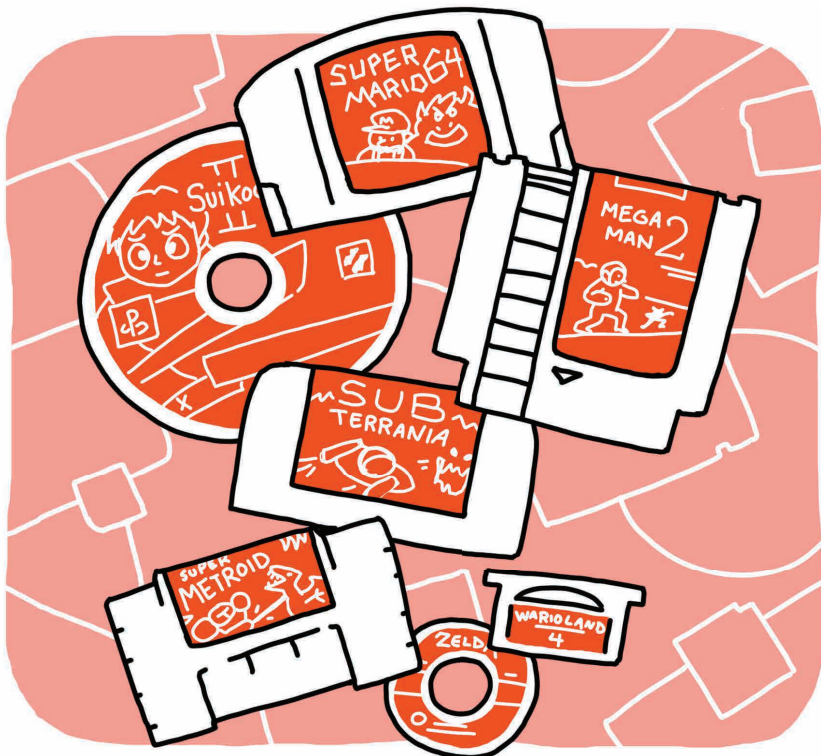
RIGHT: It was easy to get exhausted from the constant activity; attendees try to get in a few moments of shut-eye whenever (and wherever) they can. Some enterprising photographer found me in my secret nap spot under the prize table! Image courtesy of ZachSK.



Speedrunning Phase 0: Picking Your Game

Before you ever get to cranking out runs or trying your hand at glitch-hunting, you need to have a game in mind. “What game should I run?” is the single most common question I see asked in forums or social media, and in my opinion it’s a question that needs to be carefully thought through before you jump in. After all, game choice is the single decision that will ultimately influence every step of the speedrunning process, and consequently your enjoyment of the whole process.

Before doing anything else, I strongly recommend taking a step back and really thinking what it is you want to get from speedrunning. Do you want to see how you stack up against others? Do you want to push the limits for a game? Do you want to breathe new life into a piece of nostalgia? Your motivation for getting into speedrunning at all will have a huge effect on what your game choice should be, so work on understanding yourself before you dive straight into a game.



My choice of games to run comes down to two things: games that I enjoy and that I want to play again.”

NARCISSA WRIGHT - Speedrunner

Your main motivation for giving speedrunning a shot plays a big part in which games you should try. Do you want to breathe new life into an old flame? Are you looking to challenge the best of the best for a shot at the top? Pick a game that best meets your interests!

🎮 I want the games [I speedrun] to be games I enjoy playing [overall]... the controls, music, or anything else shouldn't be unfun or annoying to deal with. After all, it's a video game I want to enjoy playing."

PUWEXIL - Speedrunner



ATTEMPT: 1



ATTEMPT: 300



ATTEMPT: 10,000

Competition: If you're looking for competition, try out an established game that has plenty of resources and active runners. They'll help get you up and running and fill you in on the latest strategies, so you can jump into racing right away. There may not be much room for creativity and new discovery, though.

Nostalgia: If you want to rediscover games from your youth, or just fill the time with something you enjoy, start by listing the games you're particularly fond of. Even if you don't necessarily pick one from that list, you can identify trends in your taste to hone in on other similar titles.

Glory: If your motivation is to grab a few world records, well, my advice is to make sure you pursue something that presents a real challenge. You can meet your goal pretty easily by simply picking a really obscure game and going through with relatively low effort. But for any endeavor the reward is proportional to the difficulty in conquering it. If the notoriety is important for you, I recommend starting with established games; otherwise go with uncharted territory.

Entertainment: If you just want a fun activity to enjoy during downtime, pick a game that you're already fairly familiar with and enjoy a whole lot. Really digging into the fundamentals can take a long time, but if you aren't worried about being the absolute best player of a game, it's simple enough to just jump right in and still enjoy speedrunning.

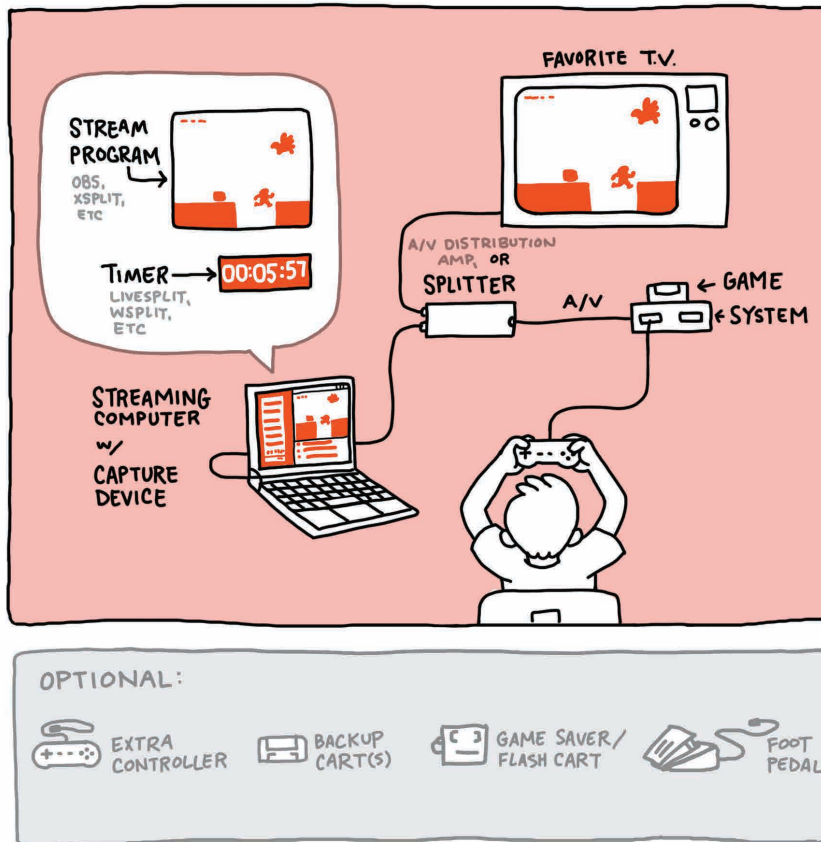
Once you've listed out a few good candidates, go through each one and ask yourself a series of questions:

- ▶ How much is unknown? Would you have to figure everything out by yourself?
- ▶ As far as you know, how much does luck affect playing the game?
- ▶ If you had to play the game many, many times, are there parts that would be particularly tedious?
- ▶ In terms of gameplay, do you have many different options available?
- ▶ Does it have good music? (You'll be hearing it a lot...)
- ▶ How long would one run of the game take? Will you have time to do full attempts?

You won't be able to answer all of these questions right away, but you should be familiar enough with your game to have a reasonable idea of what running it might be like. It all boils down to zeroing in on the things you might enjoy versus the things that might frustrate you. There will always be some of the latter, but it's extremely important to find a good balance so that the fun maintains a (hopefully significant) edge over the possible frustrations.

Look at it this way: choosing to speedrun something is an investment of your free time. If trying to speedrun causes you to be upset more than it excites you, it's not just a bad use of time, it's unhealthy. Your overall enjoyment can sway back and forth as you develop the run, but if you're starting in the red from square one, then it's not going to be a good experience.

EQUIPMENT



TOOLS OF THE TRADE

THE BARE ESSENTIALS:

Game, Console, Controller, and TV

CAPTURE/RECORDING/STREAMING:

VCR/DVD Recorder

PC Capture Device

AV Distribution Amp / Splitter - to split AV signal to TV and capture device

Streaming PC

GOOD TO HAVE FOR PRACTICE:

Backup Controllers - they wear out

Game Saver / Flash Cart - on-console save states!

Backup & Practice Carts - with saves at important points

BONUS TECH:

Foot Pedal - for hands-free splits

Input Viewer - (coming soon...)

Video Scaler - for high-quality capture, etc.

HELPFUL SOFTWARE:

Timer Program - LiveSplit, WSplit, etc.

Streaming Software - OBS, XSplit, etc.

Recording Software - AmaRec, VirtualDub

Emulators - for practice and researching

USEFUL ACCOUNTS:

Twitch - for streaming and interacting

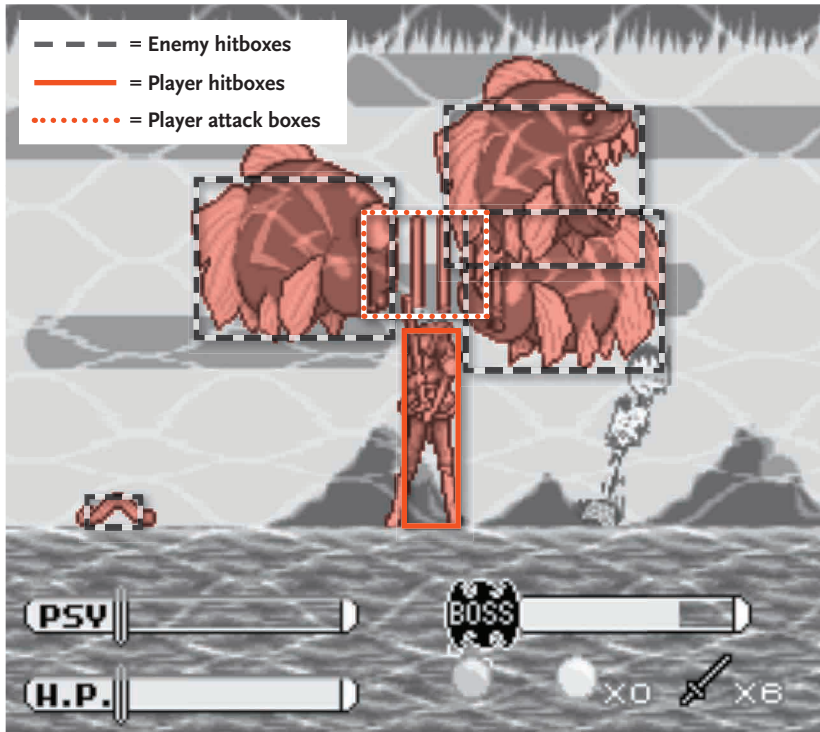
Speedrun.com - for leaderboard tracking

TASVideos - for collaborating with TASers to break your game

Discord - to be part of your game's speedrunning community

SpeedRunsLive - for racing

Speed Demos Archive - the traditional home of speedrunning



MAXIMIZING PARALLELISM

This picture shows part of a strategy that uses many types of parallelism to defeat the 3 fish bosses efficiently. It uses concurrency by letting the PSY gauge charge during attacks that don't drain it (not shown), and then multitasking by using the PSY meter to unleash a powerful attack on all 3 bosses at once while they are close together (and scarily close to hitting me!). It pays off big time, as each boss takes a significant chunk of damage from an attack that would only otherwise affect one of them.

the duration of a super attack. The results were great; definitely a satisfying experience in drawing up the plan as well as executing it.

The principles of removing overhead are generally simple: Make the most of the time that you use. Make do with less, and when possible do multiple things at the same time. These simple concepts will do so much for optimizing speedruns—often even more than you expect. Picking out instances where you can cut out an action or two here or there might have a small impact on its own, but being able to apply that rigor across an entire run will make a huge difference. Going through the motions of removing overhead is often what elevates a route from merely workable to actually good.

Resource Management & Rates

Every game has resources of some variety. HP, magic, items, lives, coins—a resource is anything that can increment or decrement based on your actions. In effect, these are your major tools for clearing the game. From a speedrunning perspective, resource management means deciding when and how to use your resources to maximize the speed. Put another way, you need to make the most of what you have.

Many resources are explicit: You have exactly 6 HP, or you can use a certain item exactly once. Now put those resources in the context of time: how many seconds can that 6 HP save? What's the single best opportunity to use that item? Could it even be faster to use neither of them at all? Thinking about those explicit resources can be straightforward, but there are numerous other resources you can plan your run around. Invulnerability frames from being hit are a resource, and so is the time duration of performing a specific action.

The easiest way to handle resources is to transform them into rates. Imagine a turn-based RPG where you can cast spells with various flashy animations. Each spell takes some amount of real time to go through its animation, but delivers a large amount of damage after the animation finishes. Since you're optimizing for time, you need to be able to assess whether the damage you can deal from casting a spell is worth the time it takes to cast it.

Let's consider three separate spells: A Fire spell, a Wind spell, and a Water spell. The Fire spell does 700 damage and takes 25 seconds of animation time. Water dishes out 375 damage and takes 15 seconds to cast. The Wind spell is quickest, at only 8 seconds, but also the weakest, at 264 damage. Simply take the damage and divide it by the time cost to get the rate for each:



		
FIRE SPELL	WATER SPELL	WIND SPELL
700 DAMAGE	375 DAMAGE	264 DAMAGE
25 SECONDS	15 SECONDS	8 SECONDS
$\frac{28 \text{ DAMAGE}}{\text{SECOND}}$	$\frac{25 \text{ DAMAGE}}{\text{SECOND}}$	$\frac{33 \text{ DAMAGE}}{\text{SECOND}}$

Based on these rates, Wind seems to be the best bang for the buck at 33 damage per second. Great! So we should use Wind everywhere... well, not quite. The rate is just the base cost to perform that particular action. What's missing from the equation so far is the rest of the context: what are you using the spells against? How

many of each will you have access to? What other options (physical attacks, attack items, etc) do you have? What is the impact on other resources, such as your own MP or even the risk to your HP?



Let's say that you're developing a plan for how to deal with a particular boss. How much HP does the boss have? The most effective solution depends on the answer. As an example, let's say that the boss has 600 HP. How would you go about determining what your options are for dispatching it?

The important part of the goal is the value: 600 HP. A single cast of Fire will land 700 damage on the beast, but that means that 100 damage of that cast isn't useful. Your goal isn't to blow everything away with the most powerful spells in your arsenal, but to settle on the most efficient spells for the task at hand. Thus, you should cap the damage dealt for our rate calculations at 600, no matter how much overkill the spells dish out.

With that in mind, we need to assemble all the reasonable options available from the resources at our disposal. I use what I refer to as "Greedy Selection" to systematically build up these options. Simply put, when picking from a variety of resources to accomplish your goal, start with the largest single option you have, and then "add" to it until you breach the goal. Repeat with each next-largest resource until you don't have anything else to check.

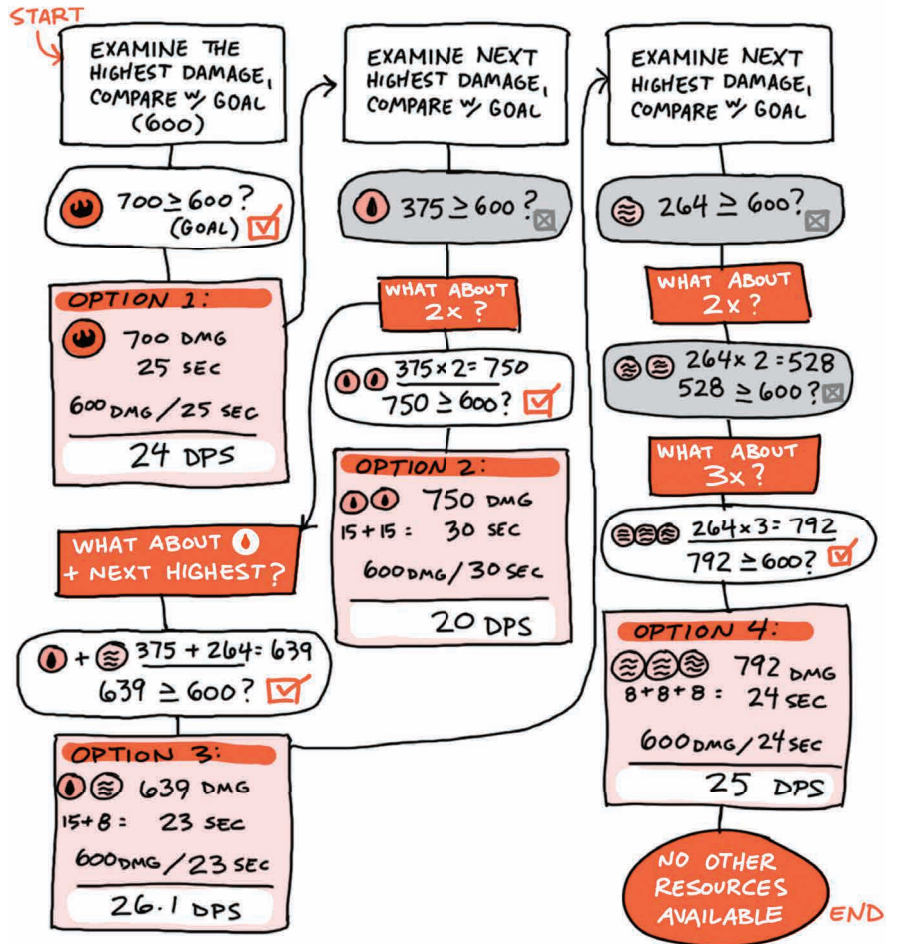
The planning process on the next page is a bit more explicit than you'll typically need, but this general approach is useful for minimizing the number of things you have to consider. Most importantly, it also minimizes the resources needed in a run, which reduces planning complexity and likely also cuts down a bit on overhead from menuing, preparation, or otherwise.

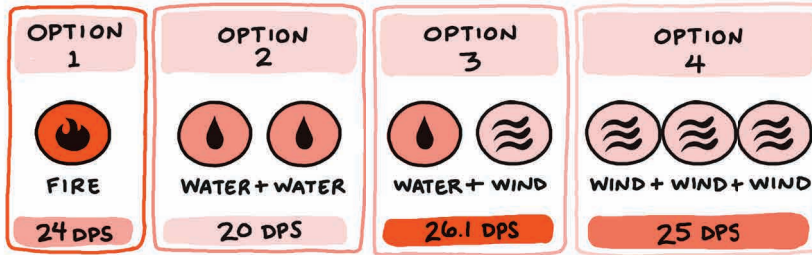
This flow chart is a complete illustration of the thought process when searching through with a greedy strategy. This example is fairly simple compared to many you might encounter in games - finding the best solution becomes much more difficult with each new resource and consideration - but it is a reasonable strategy for finding your best option when no single solution is obviously best.

GOAL: 600 DAMAGE AS QUICKLY AS POSSIBLE

RESOURCES

 700 DMG 25 SEC	 375 DMG 15 SEC	 264 DMG 8 SEC
---	---	--





When your search is done, there are only 4 workable solutions - and with each one, you know the approximate rate they will dish out damage. Sticking with the highest rate will be the fastest... so long as you have those resources available, and don't require additional setup time!

Based on the rates alone, Option 3 seems like the winner, followed by 4, 1, then 2. But this only covers the perspective of the immediate battle; is there setup time associated with getting mages capable of casting Water and Wind? Does using those spells in this battle prevent you from using them in future battles, where they might be more effective? This kind of question drives the motivation behind assessing all options, and not just shooting for the most effective; what's best in the short-term may not be best in the long-term.

Now let's change the circumstances: what's best for an enemy with 500 HP? 1000 HP? What if there are two enemies in the same battle? What if the enemies have weaknesses to specific spells? What works best changes according to the situation, so you'll have to recreate your pool of options for each new scenario.

So far I've only dealt with damage, but this same concept of rates can be applied just about everywhere. You can describe movement in terms of pixels per second. Damage boosts get converted to HP per second, or conversely, seconds saved per HP. In games like Mega Man, you can think of your shots as a resource, in that you can only have 3 on-screen at any one time. You can even set up rates in terms of other resources: gold per potion, frames per damage, HP per pixel; any rate works, so long as you have a reason to tabulate it and it ultimately relates to time.

We've dealt with *expending* resources, but figuring out when and where to *accumulate* resources is often another thing entirely. Many times this will be driven by the game design; you only heal after beating a boss, or there's only 600 gold to collect in the dungeon. Where resource accumulation is optional, consider the costs: how much time must you spend to heal up? Is it worth it to pick up another upgrade before proceeding to the next dungeon?

THANKS FOR READING THIS PREVIEW!

Speedrun Science

A Long Guide to Short Playthroughs

Eric Koziel

Comments or questions?

 @TheOmnigamer

 Omnigamer

 omnigamer@speedrunscience.com

Learn more and preorder this book at:
speedrunscience.com