The Celeste Frame Data Catalog

DeathKontrol

Contents

Chapter 1

Introduction

This document is an attempt to catalog ALL relevant information for how game mechanics work in Celeste. Most of the research has been done by myself using the TAS tools to execute moves frame by frame and examine position and speed information. Some of the information here may be credited to other players, possibly due to their own testing or experience, be it with TAS or code modding. I make every attempt to have the information be as accurate as possible and test it myself. I currently cannot offer any experimental evidence to prove these findings, but in the future I may try to compile a library of TAS scripts with a test map to demonstrate these concepts, at least the more interesting ones.

The document you are reading is currently in beta. It is highly likely that there are inaccuracies and missing data. Please feel free to reach out to me on Discord at DeathKontrol#7106 for corrections, or to request or submit additional information. I'm also happy to try to answer frame-data related questions, or learn about new tech that I haven't seen before. Celeste is full of hidden gems and there's always something new that I haven't seen yet.

1.1 Units

The fundamental unit of distance in Celeste is a pixel. The basic ground is made up of foreground tiles, which are 8×8 pixels (these cannot be subdivided). Speed is measured in pixels per second (px/s), this unit will be denoted by a bold face **s**. Acceleration is measured in speed per frame (**s**/f), which is pixels per second per frame (px/sf). There are actually two different speed measurements: *speed* measures how fast Madeline "should" move on the current frame, while *velocity* measures how fast Madeline actually travels given instantaneous environmental factors. The Y-axis is reversed, so negative Y speed means traveling up.

One frame is exactly 17ms, and therefore the game runs 2% faster than 60fps. Since the in-game timer is programmed at 60 frames per second, that means the timer is 2% faster than real time.

Thirds are rounded off, so 0.33*6=2.00. Two-thirds are truncated in this

> this probably needs to be understood and explained better

> verify this

3

document instead of rounded, so it will be written as 0.66 instead of 0.67.

1.2 Physical Constants

The following table is fundamentally what "defines" Madeline:

Hurtbox	8×9 pixels, 2 pixels above ground
Hurtbox (Crouching)	8×4 pixels
Hitbox	8×2 pixels

Table 1.1: Madeline data

The following basic numbers always apply to Madeline's horizontal motion, except under specific conditions such as being underwater, in space (end of Core), or using [extended] variants to change physics:

Walking Speed	90.00 **s**
Ground Acceleration	16.66 **s**/f
Ground Friction	-6.66 **s**/f
Air Acceleration	10.83 **s**/f
Air Friction	-4.33 **s**/f

Table 1.2: Horizontal speed

The maximum walking speed of 90 applies both on the ground and in the air. If you are going slower than this speed, you will accelerate at the ground or air acceleration rate until you reach it.

If you are traveling faster than walking speed and holding in the same direction you are traveling, you will decelerate at the rate of friction.

If you are holding opposite to the direction you are traveling, or not pressing any direction, you will decelerate at the acceleration rate. In the former scenario, your speed can rollover 0 on the frame you reverse directions.

The next table shows information for how Madeline's vertical speed changes

Gravity	+15.00 **s**/f
Half Gravity	+7.50 **s**/f
Gravity Halving Between	[-40.00,+40.00] **s**
Terminal Fall Speed	+160.00 **s**
Fast Fall Acceleration	5.00 **s**/f
Fast Fall Speed	+240.00 **s**

Table 1.3: Vertical speed

If Madeline is falling slower than +160, she will usually accelerate at the rate of gravity. However, if Madeline's vertical speed is between -40 and +40,

holding the jump button will cause the acceleration of gravity to be cut in half so that she only accelerates at 7.50 **s**/f. This makes Madeline feel slightly more "floaty" at the top of her jump. This affect is applied whenever Madeline is airborne regardless of the cause.

Holding down does not affect Madeline's fall until she reaches +160 speed. At this point, holding down causes her to accelerate +5 **s**/f until she reaches the fast fall speed of +240. Subsequently releasing down will cause her to decelerate at -5 **s**/f until she returns to the terminal fall speed. Madeline's speed must land on 160 for at least one frame before entering fast fall, it will not roll over. If she is falling at +150 and down is pressed, she will be going +160 on the next frame, and then +165 on the next.

Chapter 2

Basic Mechanics

2.1 Jump

Jumping from the ground is one of Madeline's three basic moves.

Initial Jump	-105.00 **s**
Rise Duration	12 frames
Speed Boost	40.00 **s**
Buffer Window	5 frames

Table 2.1: Basic Jump Data

Madeline's initial vertical speed is -105. So long as the jump button is held, Madeline will not be affected by gravity during the rise duration and her upward speed will not change. She will begin to accelerate down at the rate of gravity on frame 13, but if the jump button is still held, gravity halving will apply at the top of the jump arc (frames 18-27). If the jump button is released any time during the jump, Madeline will immediately begin to accelerate downwards at the rate of gravity ($+15.00$**s**/f). If left or right is held during the jump, Madeline will get a boost of 40 speed in that direction.

Madeline can jump up to a height of slightly higher than 3 tiles. The maximum height can be calculated by summation. The jump can be buffered up to 5 frames early and Madeline will jump on the first possible frame.

2.1.1 Jumping from Walls

While climbing or next to a wall in the air, Madeline can execute 3 different types of wall jumps. A *Climb Jump* is performed by holding grab and jumping without pressing away from the wall. A *Wall Jump* is performed by pressing away from the wall and jumping (with or without holding grab), or by pressing towards the wall and jumping with grab released. Finally, a *Neutral Jump*

is performed by releasing the grab button and left and right buttons before jumping.

Fortunately for the sake of simplicity, all three types of wall jumps have the exact same vertical behavior as a standard grounded jump. However, their horizontal behavior changes.

Madeline can jump from walls even from 2 pixels away.

2.1.2   Climb Jump

Climb Jumping produces no horizontal speed; Madeline simply jumps straight up. She can climb to the top of ledges with spikes or spinners on top and jump over without dying. If you press away from the wall after jumping within 11 frames, the climb jump will become a wall jump, see *Wall Boost*. Climb jumps are the only type of jumps that consume stamina: They cost exactly 27.5 stamina, see *Climbing*

link

2.1.3   Wall Jump

Wall Jumping sends Madeline away from the wall at a speed of 130. For the next 11 frames, she will decelerate at the rate of air friction until reaching the walking speed of 90, regardless of right/left inputs. Madeline also immediately turns around to face away from the wall.

Frame	Speed (away from wall)	Description
1	130.00 **s**	Fixed
2-10	-4.33 **s**/f (to 91.00)	Cannot be affected
11	90.00 **s**	Control of horizontal motion is regained on the next frame

Table 2.2: Wall Jump Horizontal Speed

2.1.4   Neutral Jump

Neutral Jumping also sends Madeline away from the wall at a speed of 130, however, the subsequent trajectory can be controlled immediately. With neutral jumps, it is possible to jump away from a wall and move back to the wall to arrive higher than the starting point. This enables the player to scale vertical faces or stall in place on walls without consuming stamina.

Grab must be released for 1 frame before beginning the jump, or else it will be a climb jump. Right and left must be released on the frame the jump is started, but may be held on any other frames.

A neutral jump starts Madeline off with 130 speed away from the wall, but the player regains control immediately on the next frame. Therefore if the player holds away from the wall, Madeline will lose speed at the rate of air friction and the trajectory will match that of a wall jump, but if the player holds towards

the wall or doesn't hold either direction, Madeline will lose speed at the rate of air acceleration, allowing her to move back towards the wall faster. This is why it is possible to get back to the wall higher up with a neutral jump than with a wall jump.

When a neutral jump is performed, Madeline does not turn away from the wall, but instead remains facing it unless the direction away from the wall is pressed.

2.1.5 Crouch Jump

If Madeline crouches and jumps, she will remain crouched until her vertical speed hits 0. For a normal full-height jump, this occurs on frame 21. The crouch state cannot be affected by basic movement; holding down does not matter. This rule applies to other sources that cause Madeline to become airborne while crouched, and the crouch state can be preserved by climb jumping before reaching 0 speed.

2.2 Dashing

The base dash speed is 240. Diagonal dashes go at the same speed, so the x and y coordinates are $\frac{\sqrt{2}}{2} \cdot 240 = 169.71$

Horizontal/Vertical Speed	240.00 **s**
Diagonal Speed (x and y)	169.71 **s**

Table 2.3: Base Dash Speeds

Dashing lasts for 15 frames. The following table shows the different stages of a dash.

Frame	Event
-4-0	Buffer Window
1	Startup
2-4	Freeze Frames
5-9	Dash Cooldown
10-14	Dash Recharge
15	End Frame

Table 2.4: Dash Breakdown

Madeline stops moving the same frame that the button is pressed. On frames 2-4, the entire game including the speedrun clock is frozen. During frames 5-15 Madeline is in the dash state and maintains a fixed trajectory. If Madeline leaves the ground before frame 10 her dash will not recharge, but if she leaves the ground on or before frame 10 the dash will be recharged. Madeline's dash speed ends on frame 15, although other dash mechanics still apply on this frame:

for example, she can still hyper or super, and she cannot start a new dash until the next frame. Dashes can also be buffered up to 5 frames early, like jumps.

2.2.1 Horizontal Speed

The x speed for horizontal and diagonal dashes is actually only a minimum: If Madeline's speed is less than this, she will be brought up to this speed when the dash starts. If however her speed is greater than this number when the dash starts, the x value of the dash will not be changed.

For example, if Madeline starts a horizontal dash while traveling at 200 speed, she will immediately be set to 240, the minimum. However, if she dashes diagonally, the dash speed will be (200,169.71). Therefore, she will not dash at a 45 degree angle. The horizontal speed will not change for the duration of the dash.

Furthermore, Madeline's horizontal speed is immediately set to 160 on the end frame, regardless of how fast she was going during the dash. When an up-diagonal dash ends, her horizontal speed is set to $\frac{\sqrt{2}}{2} \cdot 160.00 = 113.14$ and her vertical speed is set to -84.85. When a down-diagonal dash ends, Madeline's horizontal speed is not reset so long as she is in the air (If she is grounded, it is set to 160). This is a critical component of Ultras, see *Ultra (Disambiguation)*.

2.2.2 Dash Redirect

The first 4 frames of a dash are stationary; the dash does not start until the 5th frame. The game actually does not determine what direction the dash will go in until the 5th frame, therefore it is possible to switch directions in the time between pressing dash and this first active frame. This is called *Dash Redirecting,* it is a forgiveness mechanic designed so that if the player accidentally presses dash slightly before pressing the direction they intend to go, the dash will go in the intended direction.

2.3 Climbing

Climbing refers to anything having to do with holding grab on a wall. Various aspects of climbing consume stamina. This section will cover climbing speeds and stamina usage. Madeline can grab walls even from 1 pixel away.

Climbing Up	-45.00 **s**
Climbing Down	+80.00 **s**
Climbing Acceleration	15.00 **s/f**
Grind Acceleration	+1.95 **s/f**
"Catch" Speed	+32.00 **s**
Grab Startup	6f

Table 2.5: Climbing Speed

Madeline accelerates up and down the wall at the same rate at which she accelerates horizontally while walking on the ground. Grinding is when Madeline is pressing towards the wall without grabbing, or trying to grab without having stamina. Grind acceleration is the rate at which Madeline accelerates downwards if traveling less than 160 while grinding. Madeline's speed is set to the "Catch" speed if she is traveling faster than 32 downwards and grabs a wall. For the first 6 frames of a grab, no stamina is consumed.

The next table shows stamina usage in various situations. Certain climbing maneuvers consume stamina, which does not refill until Madeline is standing or interacts with an entity which refills stamina. In vanilla Celeste, all entities which refill dashes also refill stamina, such as dash refills, feathers and dream blocks. Stamina units shall be denoted **u**

Maximum Stamina	110 **u**
Stationary Grab	10 **u**/second
Climbing Up	1 **u**/px
Climbing Down	0 **u**
Climb Jump	27.5 **u**
Critical Stamina	20 **u**

Table 2.6: Stamina Usage

The maximum amount of time Madeline can grab a wall before sliding down is 11 seconds. Attempting to grab while out of stamina causes Madeline to grind. Climb jumps are possible so long as Madeline's stamina is greater than zero; if she has less than 27.5 stamina, it will all be consumed for the final climb jump. Attempting to climb jump while out of stamina results in a wall jump.

Based on the maximum jump height, climb jumping up a wall uses only slightly more stamina per pixel as climbing up, assuming full-height jumps. However, climb jumping up is faster.

When the Critical Stamina is reached, Madeline will begin flashing red. She can continue to hold or climb on the wall she is currently grabbing, but cannot initiate a new grab. She is also unable to grab entities such as Theo or Jellyfish while below the Critical Stamina threshold.

Chapter 3

Advanced Mechanics

The distinction between which mechanics are basic, advanced, and technical is somewhat arbitrary. My goal here is to separate these mechanics into categories mostly based on how well they are understood by the community, and to a degree how much they are used in the game. The "Basic" mechanics in Chapter 2 are essentially fundamental mechanics which are required for basic completion of the game, and everyone who has beaten Forsaken City should be able to execute these moves easily. The exception here is neutral jumps, but these fit into this section with the rest of the wall jump tech better than anywhere else.

The first three "Advanced" mechanics, being Wall Bounce, Hyper Dash, and Wave Dash, are taught at the end of the game and required for the final few levels. Super Dashes are not explicitly taught, but are simple enough that most players usually figure them out. Ultras and Demo Dashes are not taught.

The unifying theme between these mechanics and what separates them from the "Technical" mechanics in Chapter 5 are that these are all used extensively in speedrunning and custom maps, whereas the technical mechanics are more obscure and specific. While most players who have finished the game 100% are at least familiar with the advanced mechanics, the technical mechanics are lesser known or understood and have more niche applications.

3.1   Wall Bounce

A wall bounce is executed by dashing straight up while next to a wall, and jumping off the wall while in the dash state. Wall bounces are fast and give a great amount of height, therefore they can be required in maps. They are taught on flag 8 in 7B and also required for flag 7, some C-sides and Farewell. Wall bouncing creates "speed rings" behind Madeline.

Wall bouncing gives an initial vertical speed of -160.00 which can be maintained for the first 15 frames by holding jump. Then normal gravity kicks in, with gravity halving applying to the top of the jump as usual. Releasing jump early causes gravity to start immediately. The table shows vertical speeds as-

Initial Speed	170.00 **s**
Initial Jump	-160.00 **s**
Rise Duration	15 frames

Table 3.1: Wall Bounce Speeds

suming the jump button is held for the entire sequence.

Wall bouncing also sends Madeline away from the wall at a horizontal speed of 170, which can be controlled immediately.

Factor	Horizontal Pixels
Vertical Corner Correction	4 px
"Perfect" Alignment	1 px
Wall Bounce Forgiveness	5 px
Total	10 px

Table 3.2: Wall Bounce Positioning

Horizontal alignment for wall bounces is incredibly forgiving: The player has a 10 pixel margin (larger than a tile) to line up a wall bounce. If the player is further than 4 pixels "under" the wall, Madeline will "bonk" her head, and more than 5 pixels away will result in no jump. If there are spikes or spinners under the wall, corner correction is not available, and the wall bounce margin is reduced to 6 pixels. If Madeline is dashing up towards a zig-zag of one tile wide ledges, the possibility to wall bounce is guaranteed.

Wall bounces can also be started up to 5 frames after the dash ends, therefore up to 20 frames after the dash is started. This is a forgiveness mechanic designed to help them feel more fair.

3.2 Hyper Dash

A hyper dash, often simply called a "hyper," is executed by dashing down-diagonal along the ground, and then jumping while in the dash state. This sends Madeline forward at a fixed speed with reduced jump height. A hyper also creates speed rings behind Madeline. This mechanic is taught in 8C and not explicitly required elsewhere in the base game besides Farewell (which teaches it's variant the "wavedash"), although it's incredibly useful for multiple reasons.

Initial Speed	325.00 **s**
Initial Jump	-52.50 **s**
Rise Duration	12 frames
Max Height	-12.38 px
Max Distance	$\approx$ 17 tiles

Table 3.3: Hyper Speeds

A hyper gives a fixed speed of 325 horizontally and can be controlled immediately. The initial jump speed of a hyper is half that of a normal jump and the rise duration is equally as long, which means a hyper should go slightly less than half the height of a normal jump. The maximum height turns out to be a little bit over 1.5 tiles. The maximum distance is only an estimate assuming Madeline is on flat ground - She lands at approximately 131 pixels (between 16 and 17 tiles). This distance is slightly more than can be covered by jumping and dashing up-diagonal. Maximum height and distance were determined experimentally but could be calculated by summation.

3.2.1 Extending

One of the most important factors for making hypers viable is the concept of extending. This means jumping from the ground after the dash cooldown has ended, so that the dash recharges before leaving the ground. This grants Madeline another dash in the air.

Frame	Event
1-4	Startup and Freeze
5-9	Not Extended
10-15	Extended Hyper

Table 3.4: Extended Timing

Jumping on frames 10-15 extends the hyper and Madeline will be able to dash again, while jumping on frames 5-9 causes Madeline to have no dash in the air. In order to extend, Madeline must also have ground under her some time between frame 10 and the jump frame (inclusive); see *Coyote* for more information. It is possible to do an extended hyper from a 2 tile wide platform, but Madeline must stand on the 3 edge pixels. Extending allows Madeline to cover more distance; with no environmental factors, an extended hyper followed by an up-diagonal dash is the most distance Madeline can cover in a single jump. or to execute another move after hypering, such as a wall bounce or another hyper. Doing an extended hyper involves a short "slide" across the ground. It is something that most players can execute consistently from muscle memory with practice.

3.2.2 Wavedash

Wavedashing is taught at the beginning of the Event Horizon checkpoint in Farewell. Although the setup is different than a hyper, it's essentially the same thing. A wavedash starts by dashing down-diagonal into the ground, then landing and jumping before the dash ends. The trajectory is identical to a hyper.

Wavedashing provides two practical advantages to doing an extended hyper: The first is that an extended hyper requires some ground to slide across while a wavedash does not; it is possible to wavedash from a 1 tile platform by dashing

down-diagonal onto it, and much easier to wavedash from a 2 tile platform than to try to stand on the edge and extended hyper. The other advantage to wavedashing is that the jump can be buffered, therefore so long as the dash is initiated from the correct height, the window to do a wavedash is larger than with an extended hyper. For example: say the dash is positioned so that Madeline lands on frame 11. Then jump can be pressed anywhere between frames 7-15 This can be contrased against doing an extended hyper, where the hyper will start if jump is pressed on frames 7-9, but won't extend.

It is possible to start another wavedash from the air from a previous wavedash. Chaining wavedashes together is often significantly faster than simply dashing forward, and is the fastest way to travel on flat ground, neglecting environmental factors, besides using *Grounded Ultras*.

3.3   Super Dash

Super Dashing is never explicitly taught in the base game, but many people discover it through messing around. A super dash, or just "super," is simply a dash followed by a jump, essentially a hyper without holding down. In contrast to a hyper, a super goes slower and higher, making it another option for different applications. The trajectory is different, but the frame data is essentially the same. Supers can be extended in the same way as hypers. Supers also produce speed rings behind Madeline.

The term "Super Dash" may cause a little confusion since there is an option in the Variants menu called "Super Dashing" which causes Madeline's dash to become longer and controllable. However, this variant is generally less relevant and so in this document it will be referred to as *Super Dashing Variant* and treated in a later chapter.

Initial Speed	260.00 **s**
Initial Jump	-105.00 **s**
Rise Duration	12 frames
Max Height	-26.75 px
Max Distance	$\approx$ 15 tiles

Table 3.5: Super Speeds

The speed from a super is slightly higher than that of a regular horizontal dash. The jump height is identical to the height of a normal jump, and therefore the trajectory is the same, including the maximum height. The distance obtainable from a super is slightly less than a hyper, and about the same as jumping and dashing up-diagonal, but by extending the super Madeline can go further.

3.3.1 Midair Super

Similar to a wavedash, a super can be started from the air. However since Madeline is dashing horizontally, this means she must be lined up with the ground, so that she reaches the ground and is able to jump off it. It is more precise than a wavedash, but *Corner Correction* makes it easier. This makes it possible to do extended supers from platforms which are too narrow to do a standing extended super from. Midair supers have some applications in speedrunning and custom maps.

3.4 Reverse

Hypers, Wavedashes and Supers all have an interesting property that they can be *Reversed*. This means dashing in one direction, then doing the hyper/wavedash/super in the opposite direction. When Madeline jumps, the game only checks to see if she is dashing (and whether it is a horizontal or down-diagonal dash) and which direction is being held, the direction of the dash doesn't matter.

To execute a reverse, simply dash in one direction, then press in the opposite direction and jump. The new direction must be pressed before jumping, or on the same frame.

Reversing makes it possible to get an extended hyper or super from a platform which is too narrow but has a wall on one side; by dashing into the wall, it is possible to run out the cooldown timer before jumping in the other direction.

Reversing has some applications in speedrunning. It is also a very popular mechanic to include in custom maps.

3.5 Ultra (Disambiguation)

Ultras are a mechanic that were originally not intended, but after they were discovered they became an accepted part of the game by the developers and the community. This is probably because ultras are fun, skill-based, and allow Madeline to go very fast.

Unfortunately, the term "Ultra" is somewhat ambiguous. Unlike a hyper, there are several factors that generally constitute an ultra, and it is not always agreed upon which factors are necessary for a move to be an ultra. So in this entry, I will try to list the factors that usually contribute to a move being called an "ultra" and break down how each specific element works.

The typical ultra sequence is as follows:

1. Start with speed. The easiest way to get some kind of speed is to simply do an extended hyper, but another way to go fast is to utilize *Block Boosts*, or have speed from a previous ultra.

2. Do a down-diagonal dash from the air. Due to *Horizontal Speed Preservation*, the dash will be stretched, at an angle between 0 and 45 degrees

(closer to horizontal the faster Madeline is going). Occasionally people refer to simply this stretched dash as an "ultra" without requiring the following steps.

3. The dash ends before landing. Again due to *Horizontal Speed Preservation*, Madeline's speed is not lost, since the dash was down-diagonal.

4. Madeline lands and the *Ultra Boost* is applied, multiplying Madeline's speed by 1.2.

5. Madeline jumps and gains an additional 40 speed. This is a normal jump, therefore full height with no speed rings. Jumping is not actually necessary; Madeline can instead slide along the ground. However due to the 40 speed boost and since friction is lower in the air, jumping makes Madeline go faster.

Here are some facts about ultras:

1. The ultra boost is multiplicative, therefore by chaining ultras it is possible to increase speed without limit. In contrast, the hyper boost is constant, therefore chained hypers/wavedashes cannot go faster than a fixed speed.

2. If the dash *does not end* before landing and Madeline jumps *during* the dash, it will be a wavedash. If the dash ends *while* landed, Madeline will be slowed to 160 speed and be unable to wavedash or ultra. Only when the dash ends *before* landing will it be an ultra.

3. Due to the previous fact, ultras must be performed from raised ground. The minimum height is about 28 pixels (3.5 tiles). Typically 3 tile height is enough to do a hyper into an ultra. When doing an ultra into an ultra, 1 tile height is the minimum, but it is easier to do with at least 2.

4. One of the clearest ways to differentiate a wavedash from an ultra is that a wavedash has speed rings behind it while an ultra does not.

3.5.1 Ultra Boost

Whenever Madeline dashes down-diagonal and then touches the ground, her speed gets multiplied by 1.2. This occurs whether or not Madeline is still in the dash state, or if the dash has finished or was *Cancelled*. If Madeline is going slow this boost is not noticeable, but if she is going fast it is easy to see.

There is a story that the 1.2 factor was programmed into the game so that dashing down-diagonal into the ground causes Madeline to slide a similar distance to a horizontal dash. However, so long as Madeline's *most recent dash* was down-diagonal, the ultra boost will be applied when she lands.

3.5.2 Boost Bunny Hop

Ultras are not originally intended tech, and they don't work perfectly. One specific nuance of ultras is that it is possible in some scenarios to bunny hop upon landing (jump immediately or buffer a jump) and not get the ultra boost.

This can happen if Madeline is 1 pixel above the ground on a specific frame. It is possible to jump on this frame, but the ultra boost will not be applied until the next frame when she actually lands, therefore if she jumps on this frame, she will not get the ultra boost.

Once the dash ends, Madeline's downward speed is set to +160.00 (terminal speed), which is 2.66 pixels per frame. So for example, if Madeline is 2.40 pixels above the ground on a frame, then on the next frame she will become grounded and be able to jump and the ultra boost will be applied immediately. If however she is 3.20 pixels above the ground, then on the next frame she will be 0.54 pixels above the ground, and be able to bunny hop before getting the ultra boost on the next frame.

It is very difficult to control Madeline's precise pixel position, and nearly impossible to control her subpixel position RTA. Assuming a random starting height, Madeline has a $\frac{1}{2.66} = \frac{3}{8}$ chance of arriving between 0 and 1 pixels above the ground. Therefore, if the player bunny hops or buffers the jump input, there is about a 3/8 chance to "bunny hop over" the ultra boost and miss it.

3.6 Demo Dash

Demo Dashing is another unintended mechanic which has come to be an accepted part of the game by the Celeste developers and the community. The name "Demo" Dash is attributed to Chinese player DemoJameson who is credited for discovering it. A demo dash is a horizontal, up-diagonal or straight up dash where Madeline is crouched for the duration of the dash. This works due to two factors:

1. If down is held when the dash is initiated, Madeline becomes crouched during the startup and freeze frames.

2. If the dash is then *redirected* to a non-down direction, Madeline remains crouched until the end of the dash.

The crouch affects Madeline's hitbox, making her smaller. If the demo dash is performed horizontally, there is a visual indication: the white dash line that normally appears behind Madeline at approximately her waist-level is lower, closer to her knee-level.

Demo dashing has two major applications: The first is that by making Madeline's hurtbox smaller vertically, it is possible to dash between narrower gaps. The most well-known application of this is dashing between gaps in spinners, which are frequently 4 pixels high. The other application is performing a *Demo Hyper,* a hyper from a demo dash, since Madeline is considered to be crouched

for the duration. Due to how the game handles directional inputs, it is possible to bind a *Demo Button* which uses *Input Priority* to make is possible to do horizontal demo dashes without redirecting.

3.6.1 Demo Hyper

By using a demo dash, it is possible to do a horizontal dash into a hyper, rather than a super. This is called a *Demo Hyper.* The properties of the hyper are identical to an ordinary hyper, but the dash leading up to it is slightly different. Since a demo dash is just a horizontal dash, it will travel along the ground at 240 speed, whereas a down-diagonal dash from standing will travel at $203.65 = 1.2 \times 169.71$. Also, it is possible to do a midair demo hyper in the same way one would do a midair super. This may be more useful than a wavedash in certain situations.

3.6.2 Demo Button

It is possible to use *Input Priority* to bind a "demo dash" button which makes horizontal demos possible without manually redirecting. To do this, bind a button to press both dash and down with down being lower priority than the main control scheme. Here is the order of input priority:

$$\text{Controller D-Pad} > \text{Controller Analog Stick}$$
$$> \text{Keyboard Arrows} > \text{Keyboard Rebinds}$$

For a keyboard player using arrow keys, the demo button can be bound in settings.celeste by binding any button to both down and dash. This rebind isn't possible in the settings menu of vanilla Celeste, but can be done with the latest version of Everest. For a controller player, the demo button can be bound by using a controller rebinder (such as JoyToKey) to bind any button to keyboard down and a keyboard dash button. Alternatively, the button can be bound to an arbitrary keyboard button, and that can be bound to both down and dash like before. For a keyboard player using a control scheme other than arrows (such as WASD), it is necessary to rebind the movement keys to arrows externally using hardware or a program like AutoHotkey.

By holding left or right and pressing the demo button, it is possible to execute a horizontal demo dash without redirecting. This is because the down input from the demo button will be sent to put Madeline into crouch when starting the dash, but it will be overridden when the dash starts, resulting in a demo dash. Alternatively, the lower-priority down input could be bound to a separate key, which can be held while holding left or right and using the ordinary dash button.

Chapter 4

Spike mechanics

4.0.1 basically Kii's gif

Chapter 5

Technical Mechanics

5.1 Coyote

5.2 Crouch Technicalities

5.3 Corner Correction

5.4 Snap to Ground

5.5 Grounded Ultra

5.6 Dash Cancel

5.7 Delayed Ultra

5.7.1 Theo/Jelly Ultra

5.7.2 Cutscene Ultra

5.8 Corner Boost

5.8.1 Reverse Corner Boost

5.9 Block Boost

5.10 Wall Boost

5.11 Transition Wall Bounce

5.12 Transition Drop

5.13 Water Boost

5.14 Ceiling Pop

5.14.1 Input Priority

When Celeste receives directional input from two different sources, one gets prioritized and overrides the other. The primary application of this fact is to

make demo dashing easier, but it could also be used for other situations where one would want to make dash redirecting easier.

The following is the basic order of input priority:

Controller D-Pad > Controller Analog Stick > Keyboard Arrows > Keyboard Rebinds

If directions from two or more sources are input simultaneously, the one with highest priority will override the rest.

Chapter 6

Entities

github test

Chapter 7

Notes

Coyote: For the first 5 frames after walking or dashing off a platform, and the first 9 frames after exiting a dream block (in any direction except vertical), the player enters the coyote state. The player may jump or dash during coyote, as well as super or hyper dash. See Coyote super/hyper.

Fast fall: (remember to include jumping will holding down) Acceleration from falling: Jumping while holding down: Terminal velocity: 240.00

Climbing: Stamina: 110. Neutral grab consumes 10 stamina every second (1 every 6 frames). Grab animation activates on frame 2. 5 frame grace period before stamina consumption, so stamina is consumed on frame 11, 17, 23, etc. Up-climb stamina consumption: () Down-climb stamina consumption: (0?) Critical stamina threshold: (27.5?) Determine critical stamina behavior (theory: can continue holding current wall, can't grab new wall, can stamina jump if stamina ¿ 0) See Jump off wall, Climb jump, Neutral jump, Wall bounce

Jump off wall: Velocity: (-105.00 Y?) Check differences between inputs (pressing in, pressing out, holding or not holding grab)

Climb jump / Stamina jump: Velocity: (-105.00 Y, 0.00 X?) Stamina consumption: (110/4=27.5?)

Neutral jump: Velocity (seems to be -105.00 Y, check X)

Coyote super/hyper: If the player dashes during the coyote state, the state is maintained during the 4 freeze frames of the dash. After the freeze frames end, the player will still be in coyote for the remainder of the coyote time, minus one frame. For example, if the player walks off a platform and dashes on the first coyote frame, then they will have the coyote state during the next 7 frames, 4 for the dash freeze frames and 3 for the rest of the normal duration of the platform coyote (the first was used), minus 1. If the player dashed horizontal during the coyote frames, the player can perform a super dash by jumping during the remainder of these coyote frames. However, if the player dashed down-diagonal, the jump button must be pressed the first active frame of the dash. The jump can be buffered on the same frame that the dash is started.

Demo dash: Demo dash is dashing horizontal, up, or up-diagonal with a crouched hurtbox, which is 4 pixels tall instead of 8. To perform this move, the

player must be holding down during the first frame of the dash, and then be holding the direction the player wishes to dash in (not down) during the fifth frame of the dash. Speed is the same as a regular dash. This can be used to go through gaps in spinners than cannot otherwise be cleared. See Demo hyper.

Entering dream blocks in crouched state: There are two ways to enter a dream block in a crouched state: Either by standing on top of it and dashing down or down diagonal, or by demo dashing into it.

Snapping to ground from dash: If you jump for 1 frame, you will be -105/60 = -1.75 pixels off the ground, but if you dash, you will be snapped back to the ground immediately (your Y-position will be returned to ground position and your state send to Ground the frame you press dash).

Super dash:

Hyper dash:

Wave dash:

Reversal:

Ultra: The term Ultra is somewhat ambiguous in the Celeste community. For the purposes of this document, ultra will refer to the simple fact that if you dash down-diagonal, you get a speed boost when you land, so that's it: 1. Dash down-diagonal 2. Land on the ground. Your X-speed will be multiplied by 1.2 The boost can be applied even after the dash ends. This is what players normally do to get an ultra dash. The boost If you are within 1 pixel of the ground, it is possible to bunny hop before the ultra boost triggers.

Demo hyper:

Wall bounce: Margin of error: 4 pixels too far away, or 5 pixels too close, so including the 1 pixel of perfect alignment, there is a 10 pixel window to perform a wall bounce.

Corner boost:

Water boost:

Spike jump:

Block boost:

Map mechanics (?):

Spike Trigger spike Spinner

Spring Crumble block Falling block Refill Zip mover Dream block: You get 3 freeze frames upon exiting a dream block, just like when starting a dash.

Mr. Oshiro Green bubble Move block Cloud Disappearing cloud Wind Snowball Red bubble Swap block Seeker Theo Kevin block Bumper Feather Badeline boss Badeline booster Core block: Fire mode: If the block disappears under you, you get 5 coyote frames If you are holding the side of the wall when it disappears, you get 2 coyote + wall frames, then 4 more coyote frames Ice mode: If the block disappears under you, you get 5 coyote frames If you are holding the side of the block when it disappears, you get no coyote frames. Fireball Ice ball Wall booster Wall booster (ice) Core switch Rising/falling lava Double refill Puffer Jelly Bird Power box