Courier Cat // Paper Cat movement systems

# Document notes

* [Square brackets]: Any value written like this: [1 unit/second]. This is a starting point for the programmer(s) to use. Design will need to test and adjust this value, so please expose these variables in editor.
  + Actually, please expose any variable in editor that you think makes sense to expose.
* We’re using “metre” which is equal to “Unity unit”.

# **Pillars**

The pillars of Courier Cat and how they relate to movement:

#### You Are A Cat

This must feel like controlling a cat. When we move slowly, we’re precise and silent. When we move fast, we’re real fast (and still pretty quiet). Here are some notes that we want to hit for this pillar:

* The camera is low, so we feel small in a world of epic threats and characters
* Jumping up to high ground to survey the terrain is important for navigation
* The cat is nimble and precise. The systems help the player to navigate and jump appropriately even while the action is intense and fast-paced.

#### Messenger of the Gods

* Big, important deliveries to the most important clients in the universe.
* This cat is a trickster – even moreso than the average cat. Maybe there are some cheeky techniques that are very hard to pull off. E.g.: Shine Spark from Metroid or crazy hat jump tricks from Super Mario Odyssey.
* We’re faster than pretty much anything in the universe (although top speed is not always practical).
* Fun-loving. We’re about the journey, not the destination. This cat is a joy to control. We love it, he loves it, and we love that he loves it.

#### ***More*** is Not Better

This game is (by necessity, and by personal philosophy) minimalist. We want the system to be as simple as possible while allowing diverse gameplay. For this reason, “Walk” and “Sneak” are one and the same. Any and all ideas on how to incorporate this pillar are welcome!

# Walk / **Sneak**

Just by using the control stick, we are walking and sneaking. Courier Cat moves in complete silence at [1 metre/second].

* Walk in total silence at 1m/s
* Acceleration is very quick. Within [0.25 seconds] we are at full speed.
* Stopping is almost instantaneous. We are completely stopped after [0.1 seconds].
* Turning is quick, but not instantaneous. The larger the angle, the longer it takes.
  + To turn [30°] or less, the turn is instantaneous.
  + To turn 180° [+ / - 30°], the turn takes [0.5 seconds].
  + Any angle in between these thresholds takes a proportional amount of time.
    - So, a 90° turn would take 2.5 seconds. A 115° turn would take around 3.54 seconds.

# **Run**

Holding the Run button allows us to start running. There are three phases to this movement (which is heavily inspired by Okami – check that out for reference).

1. Take-off. We begin dashing at [3 m/s].
2. Build-up. We build momentum for [5 seconds].
3. Max speed. We are now running at [6 m/s].

## Take-off

To reach baseline dash speed takes [0.25 seconds]. After that, we start accelerating more slowly.

## Build-up

We build velocity over a period of [5 seconds]. The build-up acceleration rate follows [a smoothstep curve]: <https://en.wikipedia.org/wiki/Smoothstep>

In sum: The acceleration is slow at first, picks up around the middle of the build-up period, and slows again when we are close to max speed.

## Max speed

We need to be careful to maintain maximum speed. And to be careful, we need to see where we’re going.

* Stopping for any reason – player input, collisions with a wall or an enemy – will cancel our momentum. We will have to begin the Run process from Take-off again.
* The camera soft-locks to show the road ahead to about [40m].
  + Adjusting the camera with right stick during this phase will work, but we can’t rotate more than [45°] and it will adjust back to centre when the stick is released.
* Jumping at max speed provides the greatest distance and height for our jump possible.

# **Default Movement Toggle**

The player has the option to change “Run” to the default movement mode. In this case, pressing the “Run” button (now the “Sneak” button) lets us walk quietly and carefully.

# Ledges

The avatar will not walk off a ledge without coaxing (unless running!).

* If the avatar walks up to a ledge of more than [0.5m], he’ll stop. (0.5m ledge we ignore)
  + Continuing to hold the control stick toward (perpendicular) the ledge will cause the camera to pan downward. This puts us in “Ledge Drop” mode.
    - “Toward the ledge” here means: the input direction is at 90° with the ledge, give or take [30°]. Beyond this point (so, 120° to 180°) the input is treated as perpendicular.
      * Within this range (120°-180°) the camera will keep pointing downward as if the input is perpendicular with the ledge.
  + In Ledge Drop mode, releasing and re-pressing the stick toward the ledge will allow the cat to drop off the ledge.
  + Jumping from Ledge Drop mode, we can apply our jump force forward using the drop of the ledge. This allows us a bit more forward velocity potential than a normal standing jump.
    - A well-timed jump while running allows the highest possible forward jump force. The cat won’t wait for it, however, so we have to be aware of this interaction to use it.
* If the avatar runs up to a ledge, he’ll ignore it and keep running off.
  + If we are running, even at max speed, and we let go before reaching the ledge, we will stop at the ledge as if we were walking.
    - In this case, we will also enter Ledge Drop mode without needing input.

# Jump

Courier Cat has two types of jump. Tap-jump, and Precision jump. See Jump Metrics at the bottom of this section for numbers relating to jump distance.

## Tap jump

Tap jump is really just a Precision jump without winding up. If the player hits and releases the jump button quickly, the jump is executed after the Wind-Up step of the Precision jump.

The main skill involved in using the Tap jump is in understanding and taking advantage of the system that selects a landing spot for the Wind-Up. Then we can do difficult platforming sections quickly by helping the system select the platform we want to land on.

## **Precision** Jump

It happens in three stages:

1. **Wind-up.** Pressing and holding the jump button puts us into the Wind-up state. We see our jump trajectory and where we’re going to land.
2. **Aim.** Moving the control stick while in the Wind-up State moves our jump target.
3. **Release.** Let go the jump button to execute the jump.

## Wind-up

Entering the wind-up state means:

* Walk/run is disabled. Left control stick is for aiming the jump (more on that below).
* We slow down gradually, [40%] as fast as we would if we just let go of the control stick from running.
* We display the jump arc and the landing reticle. The landing reticle **always aims at the centre of a grid square**.
* Choosing a landing point. The reticle is placed at the chosen landing point.
  + The factors that influence the choice of grid square, in decreasing priority:
    - Our default tap-jump distance
    - Our current momentum
    - Our altitude compared to the terrain ahead
      * We begin by looking for a landing point at our altitude
      * Then we search higher and lower by increments until we find one that fits our other criteria
  + If there is a wall in front of us, the system will see if we can get on top of it. If we can’t, the initial jump arc will display in red, oriented toward the wall.

## Aim

Once the landing spot is chosen and we’re in Wind-up mode, the player can aim their jump using the **left control stick**.

* The reticle only ever sits in the middle of a grid square.
* While moving the reticle, the jump arc itself moves smoothly (ie. it does not snap from grid square to grid square
* Pressing the control stick more than [40%] in any direction moves the reticle instantly by one square in that direction. Should feel like a menu screen.
  + Holding the control stick: after moving one tile, the control becomes a bit more “free”. We will see the jump arc move as if toward the next grid square. We can move the reticle by one square every [1/3 seconds].
  + Releasing the control stick, the jump arc will snap back to the centre of the reticle.
* If the player is running and does not adjust the control stick for the jump, the reticle waits at the initial square for [1 second] before beginning to move forward as it would if the player pressed the control stick again in that direction.
  + Right before it starts moving – at [0.9 seconds] – if the player was running, a Super Jump can be performed by releasing the jump button within a very short window ([0.1 seconds]). See Super Jump below for details.
* The landing reticle can’t move beyond where we can jump. It stops for walls, and at the outside of our maximum jump range.
* If the reticle tries to move to a spot that can’t be reached, it will disappear, and the jump arc turns red.

## Release

Letting go the jump button executes the jump.

* The jump is very quick. It feels fast and heavy. [This means gravity and jump impulse have to be pretty high.]
* Our landing behaviour depends on what we were doing when we started the jump, and what we’re doing with the control stick now.
  + If the control stick is pressed in any direction, we start moving immediately in that direction. Turning rules apply.
  + If we were running, and the control stick is pressed in the same direction, we keep all our previous momentum.
  + If the control stick is not pressed (less than [40%] activation), we stop sharply.
* The case of moving platforms is quite complex... Let’s avoid it entirely for now (:
* If the jump arc is red because the landing target is invalid, releasing the jump button will cancel the jump. The avatar gives a little head shake and a grumpy cat sound.

## Super Jump

While running (not walking!), the avatar can perform a Super Jump. The idea behind this is that the cat is using his momentum and gathering power in jumping limbs at the same time. So, there is only a brief window where there is still momentum and we have had the time to build strength for the jump.

To execute a Super Jump:

1. Start running! The faster you are running, the further the Super Jump will go.
2. Press and hold the jump button. Don’t let go the control stick.
3. After 0.9 seconds, the window begins and lasts for 0.1 seconds. During this time, the aim reticle and jump arc turn yellow and snap to an automatically selected tile further ahead.
4. Release the jump button during the window and the cat will jump to the yellow target.

## Jump Metrics

Standing / walking jump

* Default (tap jump) length: [2m]
* Maximum length: [5m]
* Maximum height: [3.5m] (reaching 3m platforms is the goal)

Running jump (initial run speed: [3m/s])

* Default (tap jump) length: [4m]
* Maximum length: [7m]
* Maximum height: [4.5m] (reaching up to 4m platforms)
* Super Jump length: [8m]
* Super Jump maximum height: [5.5m]

Full speed jump ([6m/s])

***Note! It might make more sense to establish these as a result of the other systems.***

* Default (tap jump) length: [6m]
* Maximum length: [9m]
* Maximum height: [5.5m] (reaching up to 5m platforms)
* Super Jump length: [8m]
* Super Jump maximum height: [5.5m]

# Climb

The avatar can climb most vertical surfaces. In general, if we are holding the **left control stick** toward a wall, we start climbing.

* This only activates if we are falling (i.e. our vertical velocity is at or below 0).
* We climb at a flat rate of [1.5 m/s]. There is no running/walking duality here.
* If we climb up to a ledge, we climb up the ledge nimbly. Once the top of our character collider reaches the ledge, it takes us [0.5 seconds] to climb up.
  + Once we climb up, we can transition smoothly to walking/running.
  + The avatar only stops if we release the control stick while climbing up to standing.
* If we impact a wall during a jump:
  + If we are near enough a ledge to climb up to the top, we will do a faster ledge climb rather than going into a climbing state.
    - Jumping ledge climb takes [0.15 seconds] and we keep all our momentum from when we began the jump.
* We can climb directly up, or at an angle of up to [40°] by holding the control stick diagonally.
* If the control stick is released, we stop instantly.
* Pressing the control stick away from the wall causes us to drop down.
* Jumping while climbing is possible and functions similarly to a normal jump.
  + The reticle begins on the wall, 1m up
  + Tap-jump sends us straight up by 1m. This ends up being faster than climbing with correct timing.
  + We can move the target:
    - up by [1 metre]
    - behind us, up to [0.5m] high (reaching platforms at our own altitude) and [3m] long (or longer if the target is below us)