Guide for Animation Authoring / Sourcing

1 Introduction

1.1 Purpose

This guide is written for anyone authoring animations for a motion matching system. There is some technical nuances to creating animations for motion matching that are often misunderstood. This guide is intended to clarify these nuances. Note that it is not intended to provide step by step instructions for animation authoring, rather just the specifics pertaining to motion matching.

The guide is written with ‘Motion Matching for Unity’ (MxM) in mind. However, most of the principals should be the same for any motion matching system.

1.2 Motion Matching Technology

Motion matching is a relatively new animation technique employed in some AAA games to achieve high fidelity and responsive animation that required no state machine or explicit transitions. Motion Matching is also well known for not requiring mocap takes to be cut into individual clips.

The basic idea is that at any point in time, the animation can jump to any pose within a large database of poses. The choice of pose to jump to is based on the desired trajectory of the character, derived from player input, and the current pose.

The following sections are intended to provide a solid understanding of the problems faced and choices that need to be made in order to create quality animations for motion matching.

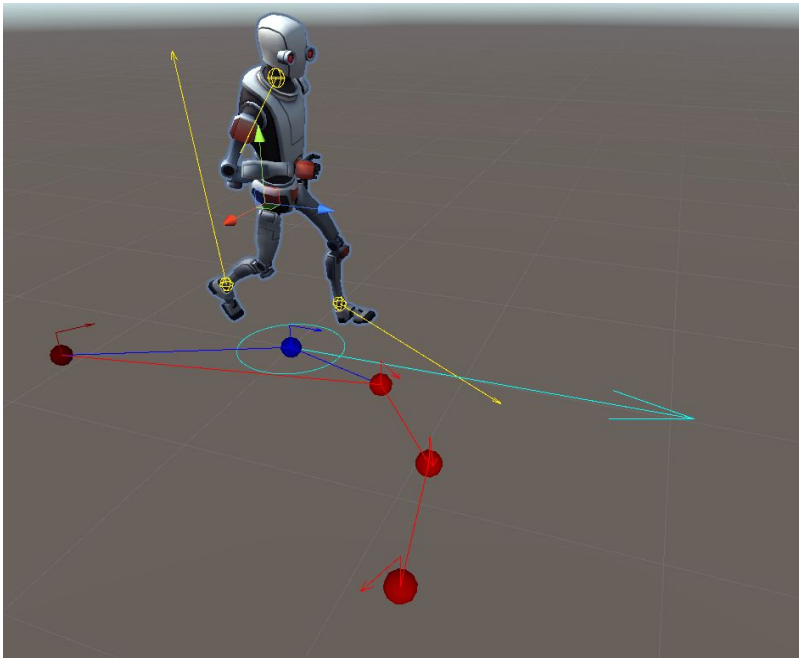


Figure 1 ‘Motion Matching trajectory and pose in a single animation frame*’*

2 Core Concepts

There are several core concepts that need to be kept in mind when sourcing or authoring animations for motion matching. The following sections give a brief explanation of these concepts.

2.1 Trash In - Trash Out

Motion matching has only ever been shown working very well with high quality animations. While motion matching can have some awesome results it cannot output quality animation if the input animations are poor.

It's important to understand that motion matching does not modify the input animations in any way. The animations you input into the system are the animations you will get out of the system. Motion Matching simply facilitates which clips should play at what time and when.

2.2 Coverage

One of the primary factors for achieving quality animation with motion matching is coverage. Coverage basically means, how many possible movements does your animation cover? How many angled turns are there? How many starts, stops, shuffles, banks etc. ? Are there left and right foot variations of each? How many speed variations are there?

As a rule of thumb, the more the better. However, there is a reasonable limit to what is feasible and past a certain point there are diminishing returns. All that matters is that there is enough to make it believable.

This guide contains a list of recommended animations that should be created to make a high quality set for motion matching whether it is for mocap or cut clips. Note that the start and end pose of each of these animations should tie in seamlessly with their logical successor. For example, the pose at the end of a run-start animation should be so that it flows directly into the first frame of the 'run' animation without needing any blending.

2.3 Continuity

Continuity is another vital concept to understand. For motion matching to work, the system needs to know what the future trajectory of that animation is. Typically the future trajectory is measured 1 second into the future and up to 1 second in the past. This means that both before and after any action, there needs to be 1 second of continuous motion or idle.

As a rule of thumb for mocap, everytime, the actor stops, they should stop for a minimum 1 second and every time they move they should continue that movement for a minimum 1 second before slowing again or taking a specific action like a stop or a plant. This 1 second

rule could be broken if the goal is to capture multiple successive plants or shuffles to capture rapid direction changes that simulate player ‘input chaos’.

2.3.1 Continuity for Cut Clips

‘Motion Matching for Unity’ has systems in place to deal with cut clip trajectory continuity issues by allowing the user to either extrapolate animation trajectories or to extract them from other clips. However, there is another problem that requires attention, particularly for ‘start’ type animations which come to an abrupt end. Usually these animations are set up to flow immediately into a moving animation (e.g. RunFwdStart -> RunFwd). However, since there are no explicit transitions in motion matching this doesn’t always work well and can result in some jitter.

Motion Matching thrives on continuous animation because it generates a database of sequential poses. If the next pose being played is the best pose, calculated by the motion matching algorithm, it simply keeps playing it. However, with clips, this continuity is disrupted frequently, and the system is forced to pick a different pose when a clip comes to an end. This can cause some abrupt jitter.

‘Motion Matching for Unity’ solves this by splicing the two clips together at runtime and treating them as one animation, Therefore, eliminating the continuity gap. However, the same effect can also be achieved by extending ‘start’ and ‘plant’ type animations with about half a loop of your ‘move’ animation.

Note: Looping animations (e.g. runs) and ‘stop’ type animations (e.g. run-stop) don’t have this issue. When a looping clip is used, ‘Motion Matching for Unity automatically loops it’s iteration over a section of the database when determining the current pose.

3 Authoring Methods

There are multiple animation authoring methods that can be employed with motion matching and the choice of method should be based upon your budget, style and available resources. Each method has its own benefits and downfalls so it's important to consider all the pros and cons. The primary methods include:

- Motion Capture
- Stitched Cut Clips
- Cut Clips

When selecting a method it is always good to keep in mind the following:

*Quality Mocap** > *Quality Cut Clips* > *Poor Mocap* > *Poor Cut Clips*

** Quality mocap refers to both the capture / cleanup quality, and if it was made correctly for motion matching.*

3.1 Motion Capture

Motion capture is probably the most well known method for use with motion matching. It is also the most expensive and difficult to attain.

The biggest fallacy of using motion capture for Motion Matching is that you can throw raw mocap into a motion matching system without any cleanup and it will be shippable. This is simply not true. The mocap takes need to be cleaned up and modified so that the animations suite your gameplay. This can be extremely tedious, especially with the sheer quantity of animation.

Pros:

- Highest potential for realistic quality animation
- Easy to get good *coverage*
- Continuity is ensured by default
- More variation than other methods

Cons:

- Very expensive
- Requires significant cleanup / modification to suite gameplay
- Difficult to tag
- Requires good actors
- Very difficult to iterate (requires good planning and flexibility)

WARNING: Only consider mocap if you have the resources. IMU motion capture suites are not generally recommended as their ability to track global position is typically not good. Motion matching relies heavily on global root position for pre-processing of trajectories.

3.2 Stitched Clips

Stitching clips together from cut clips to mimic a mocap dance card is a valid method of authoring animation for motion matching. While it does not provide the same amount of variation as mocap, it does ensure continuity of animations making it much easier for the motion matching system to synthesise smooth results.

Pros:

- Relatively accessible
- Continuity is ensured by default
- Can be beneficial to coverage if done correctly

Cons:

- Requires some work to stitch clips
- Can be difficult to iterate
- Difficult to tag
- Some redundant / identical animation data

3.3 Cut Clips

Most motion matching tools do not support cut clips. However, ‘Motion Matching for Unity’ does. Using cut clips like you would for traditional animation systems is the most accessible means of using Motion Matching for Unity but it also has lesser results than the other methods mostly due to lack of continuity and coverage.

It is very important to understand that not just any cut clip animation set will work well with MxM. For an animation set to work it must have all transitions including starts, stops, plants etc. with both left and right feet. If these animations are missing then there is a massive gap in coverage and continuity that is difficult for motion matching to reconcile.

For example, it’s very difficult for motion matching to jump the gap between being idle and running if there is no ‘run-start’ animation. If there are a lot of missing transition animations, then responsiveness will be low or fidelity will have to suffer to improve responsiveness.

Pros:

- Highly accessible
- Can give good results
- Easy to tag
- Less / no redundant animation

Cons:

- Must have as many transition animations as possible
- Lack of continuity can cause problems
- Typically less coverage

Note: while it may seem like cut clips are a bad choice, it is possible to get very good results with cut clips.

4 Motion Capture Guidelines

The purpose of this section is not to explain how to carry out motion capture. Rather it is intended to point out specific issues to consider that are pertinent to motion matching.

Note: I am not a motion capture expert. The information provided below is based upon my experience and my knowledge / understanding of how motion matching works.

4.1 The Space

Is the space big enough to fit planned takes and dance cards while maintaining minimum 1s continuous movement at the chosen speeds? This is very important for reasons explained in the section on 'Continuity'. If not, more planning is required to either increase the capture volume or break the dance cards down into several cards that fit into the space.

At high movement speeds, e.g. sprinting, it can be very difficult to fit dance cards into the space. In such cases it may be necessary to break dance cards down into individual movements. Have the actor sprint into the volume, perform the movement, e.g. Plant135Right, and then run out of the volume. Try to maximise the time moving before and after the plant to ensure continuity isn't a problem.

Note: Not all mocap systems handle running in and out of the volume very well.

4.2 Actor Responsiveness

Motion matching is as responsive as the animations that are fed into the system. If the actor is slow to turn then the character in game will be slow to turn. Remind your actors to keep the following in mind:

- **Don't lean into starts -** When the actor decides to run or move at any speed they must do so quickly. Get up to the speed they want to go as quickly as possible.
- **When stopping or planting, do so as fast as possible -** Try to stop within a single step (may not be possible at sprinting speeds).

In some ways this may seem unrealistic but it is highly beneficial for gameplay. Think of all actions as being explosive with a lot of power because that's how a player generally uses their controls.

Obviously, there are no hard and fast rules, If you want some animations to be more sluggish then the actor should act that way. Instruct your actors to move as you would like the character in the game to behave. The closer the actor gets, the less post processing / cleanup work there will be.

4.3 Uniformity

Diversity is one thing that makes motion matching so realistic. However, to some extent there needs to be uniformity in motion. For this reason it is recommended to use the same actor for each individual animation set created. The actors need to learn the idle pose and hit that pose as close as possible every time they stop. They need to run / walk / move in a relatively consistent manner. Again, the more consistent the actor, the less cleanup is required.

Be careful of tiring actors too much or their movements may suffer.

4.4 Dance Cards

There is nothing particularly special about dance cards and how they are laid out. They could be designed in any way as long as all desired turn movements are being captured and there is enough distance between movement actions (i.e. stops, plants etc). Understanding this, it is relatively simple to decide on your own dance cards for motion matching. Here are the steps to take:

1. **Choose angle increment -** This is the increment in angular turns that your actor will perform. 45 degrees is the minimum recommended. 22.5 degrees is doable and provides very good results but also requires significantly more effort.
2. **Fit Every Increment Into a Dance Card -** This is a process of trial and error. Try to efficiently place every single angle from -180 to 180 based on your angular increments.

Dance cards can become complex and it can be difficult for the actor to remember them. If necessary it can be beneficial to have one or more crew members call to the actor to assist them.

4.4.1 Rectangle

The below dance card shows an example of of this. Note that if your area isn't big enough, the dance card can be broken down into smaller sections.

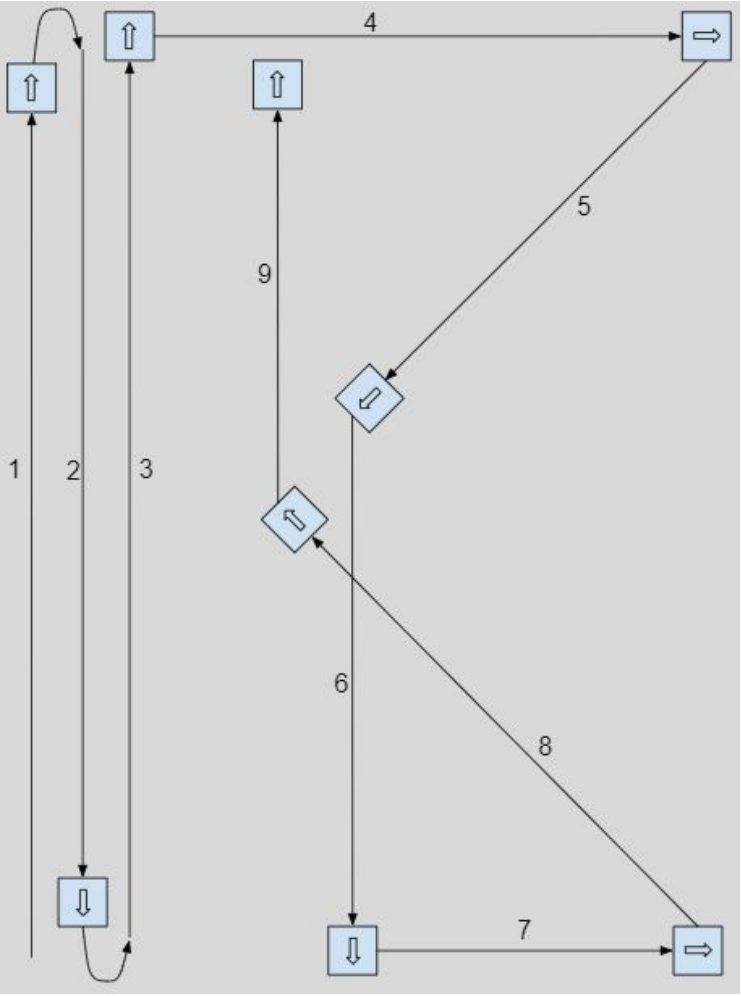


Figure 4.1 Typical RectangularDance Card (credit Kristjan Zadziuk GDC 2016)

Any dance card can be used in different ways. For example, you could perform plants at each target, or stops. You could counter turn (turn the wrong way), strafe, or even try to continuously run and turn without planting. If it's a movement you want in your game, try fit it in into a dance card.

4.4.2 Dial

For slower movements and 'turn in place' movements a dial can be used. If you think of a clock like dial, it contains every possible angle you will ever need. The dial in Figure ## is broken down into 45 degree increments but can be modified for any number of increments.

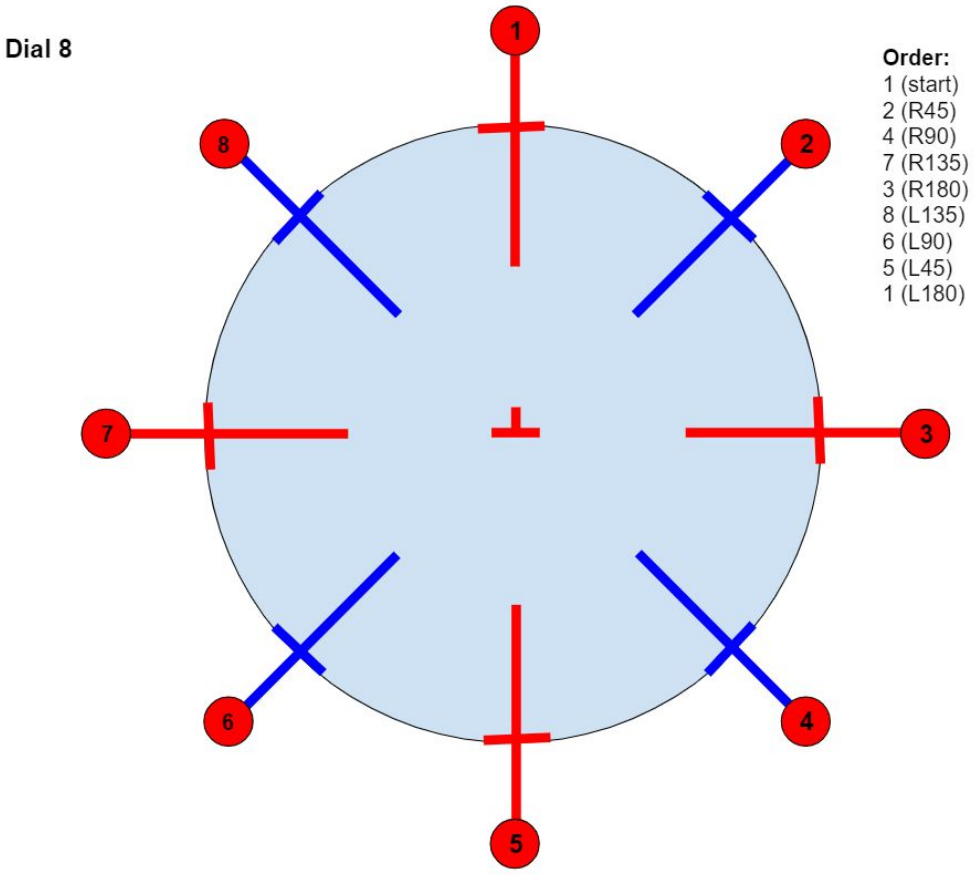


Figure 4.2 Dial Dance Card

Note: The red and blue lines aren't anything special. They simply represent one way you could mark the ground so it is easy for the actor to follow without getting confused.

For each take with this dial, pick a movement type and speed (e.g. walk, plants) and perform the following:

1. Start in the center
2. Move to the next target point (See Order)
3. At the target point - Perform the action (e.g. plant, stop etc.)
4. Move to the center
5. At the center - Perform the action (e.g. plant, stop etc.)
6. Goto step 2 until the entire dial has been complete

The dial is often too small for runs and sprints but can be used for slower movement like walks and jogs, depending on the space you have available. Make sure you can achieve the 1 second timing rule between actions and make sure that when you stop, stop for at least 1 second.

Here is a list of movement actions that could be performed on a dial (at different speeds):

- **Turn in Place** *(follow the order but don't move to the targets just turn to them)*
- **Turn -> Shuffle -> Stop - >Step Back to Center**
- **Turn -> Move -> Stop** *(turn and move towards the target, stop at the target)*

- **Move - > Plant -> Move** *(move towards target, plant at target and turn to the next target)*
- **Move - > BankTurn -> Move** *(move towards target, continue and turn (not plant) towards the next target. Cut corners when going through the centre)*

4.4.3 Strafing Dial

For strafing, the complex ordering of the standard dial is not required and the full diameter of the dial can be used for movement. It is vital that the diameter is large enough to get 2-4 full strafe cycles (not including start and stop acceleration / deceleration).

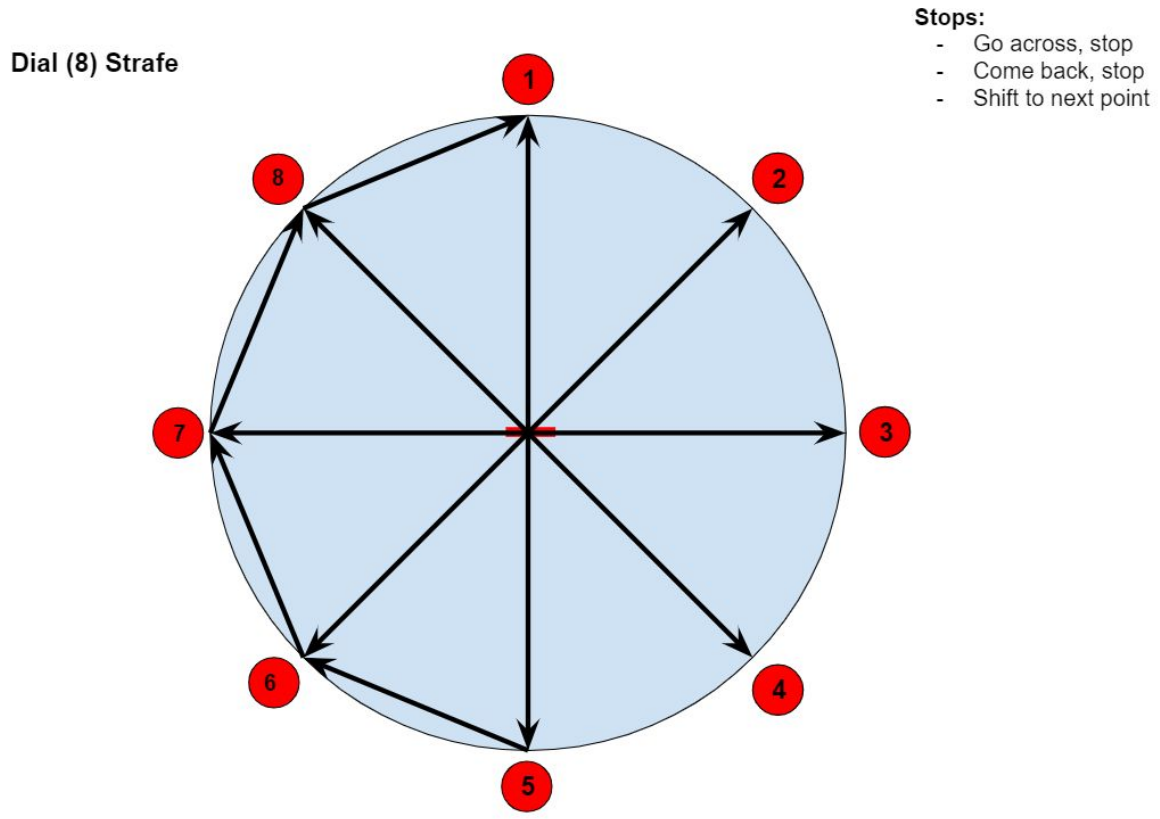


Figure 4.3 Strafing Dial Dance Card

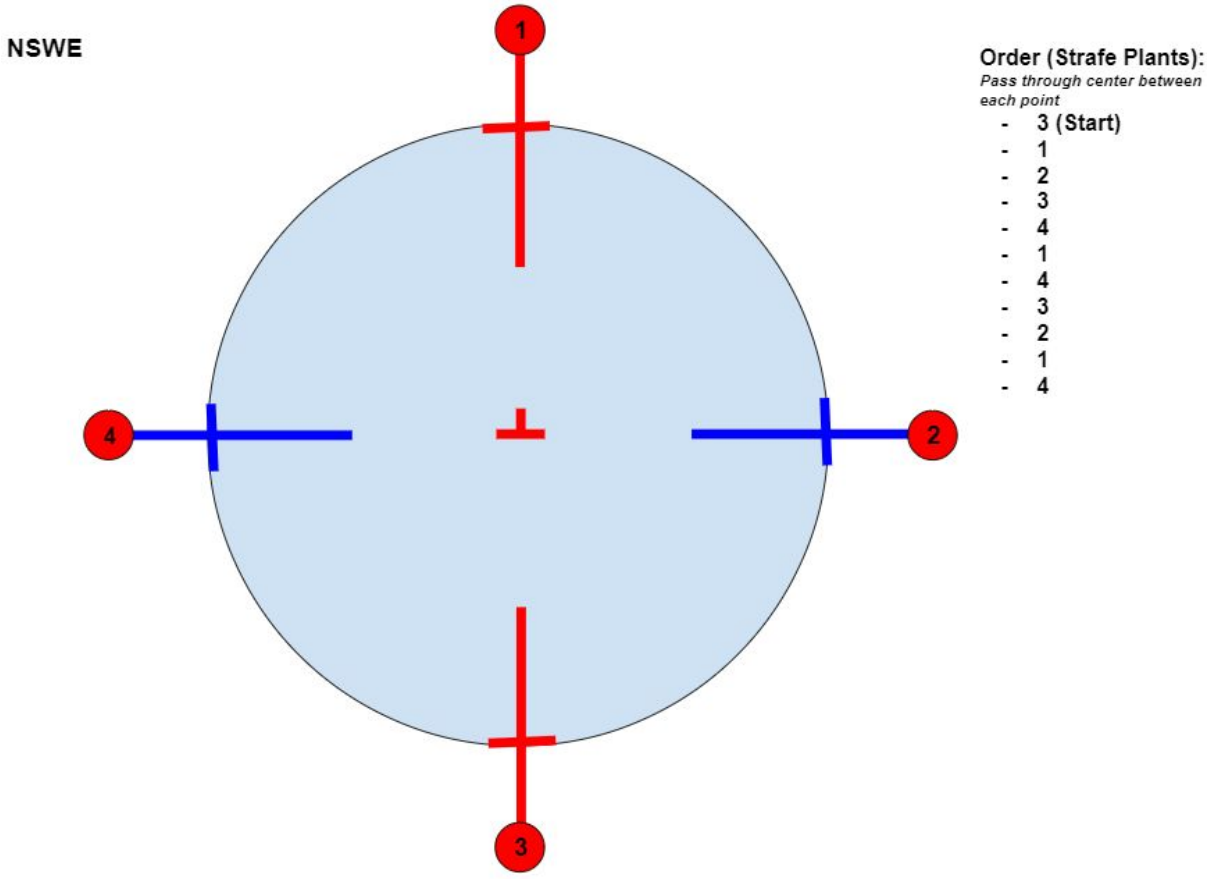
To perform the strafing dial, do the following the following:

1. Start at point 5
2. Strafe move to the opposing point on the dial
3. Stop
4. Strafe move back to the previous point
5. Stop
6. Shift to the next point on your left
7. Goto Step 2

Note: The shift animation between points could be discarded / tagged as 'DoNotUse'
Note: This dance card is not suitable for plants

4.4.4 NSEW Dial (Planting Strafe Dial)

For planting strafes it may be unreasonable to capture every possible combination in 45degree increments or less. This could result in 64 different transitions. Therefore, for strafing plants, it is recommended to use a 4 point dial.



To perform the dial follow these steps:

1. Start at point 3
2. Strafe to center
3. Plant towards next target (if applicable, no plant if going straight through)
4. Plant at target towards center
5. Repeat from step 2 until the dial is complete.

This dial will give 18 different 'from-to' strafing plant transitions. Coupled together with starts and stops from the strafing dial it should be more than enough. While it may technically be possible to capture all 64 possible transitions if using 45 degree increments, it is recommended to only do this if you are a big studio with a large animation budget and using talented actors.

Freeform Strafing - There is some value to free form strafing as it's possible to get many good transitions. Use a director approach by standing in front of the actor and getting the actor to mirror your movements. Strafe in all directions and change direction suddenly and randomly without warning. Just like the game not knowing where the player is going to push the stick next, the actor doesn't know where the director is going to go and this can make for some nice realistic, reactionary transitions.

4.4.5 Straight

The cycle animations captured as part of the previous dance cards are usually not suitable and will likely result in inconsistent loops where the character will slow down and speed up periodically. The 'straight' dance card is intended to rectify this by providing several consistent cycles in a continuous motion.

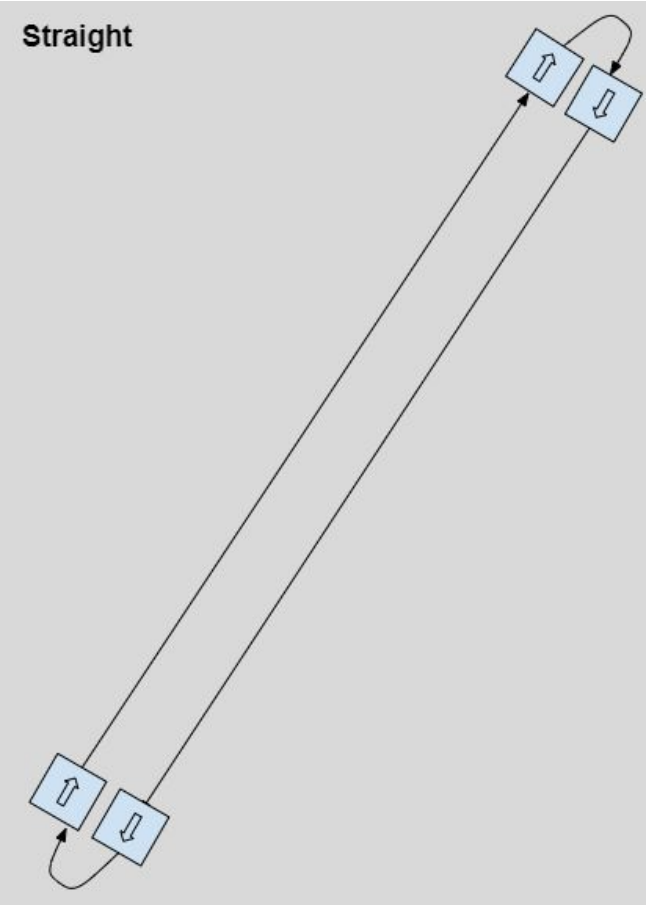


Figure 4.4 'Straight' dance card

Perform this dance card for each movement speed in each stance.Try to get 3-4 consistent cycles within each pass. If your game has significant strafing movements, it may be beneficial to perform all strafe angles with this dance card as well.

Accelerations and decelerations should also be performed with this dance card. Capture these with both left and right foot forward for each combination of speeds. Make sure the change in speed is quick to ensure responsiveness at runtime.

For ‘Motion Matching for Unity’ it is beneficial to cut these cycles into individual looping clip even if using Motion Capture. MxM handles loops very well, ensuring quality playback of loops compared loops embedded in a raw mocap take. For best results aim for 3-4 movement cycles in each loop.

4.4.6 Circles / Spirals

The ‘Circles’ or ‘Spirals’ take is intended to capture arcing turns of different radius. Spirals also help to capture the tightening and loosening of banked turns which will occur as a player eases in and out of banks.

To perform this take, simply have the actor move in a large arcing circle. They should then slowly decrease the radius of that circle (spiral in) until it is no longer possible to move at the desired speed for that radius. Once the middle has been reached, continue moving and spiral outwards.

It may be beneficial to mark a circle on the ground. However, this can sometimes cause the actor to look down. A popular method to overcome this is the ‘Rodeo’ method. Tie a rope around the actor’s waist (as low as possible). Have a helper stand in the middle of the circle holding the other end of the rope. As the actor runs, the helper should keep some tension on the rope to help the actor maintain a consistent arc. Slowly pull on the rope to decrease the radius (spiral in) and slowly release the rope to increase radius (spiral out).

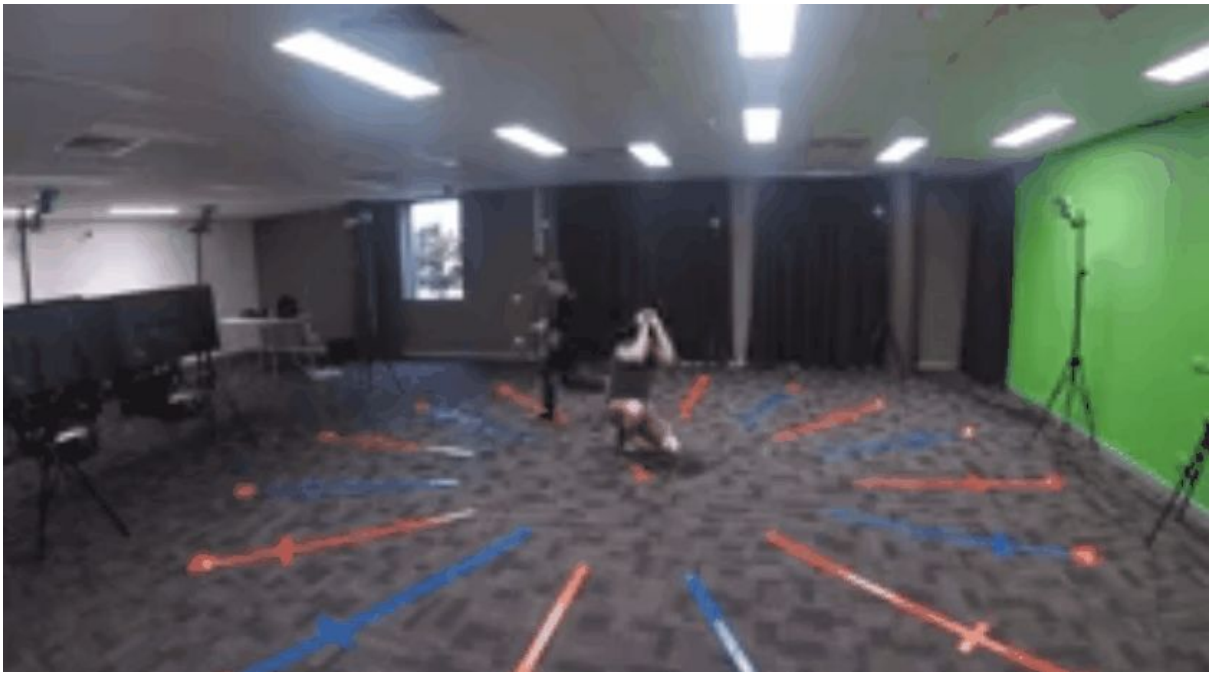


Figure 4.5 The ‘Rodeo’ on a spiral dance card.

When using the Rodeo method, be careful of placing too much tension on the rope as it may impede the actor’s arm movement.

4.4.7 Snakes

Traditional animation systems are unable to animate the weight shifting that occurs when banking left and right in an ‘S’ pattern. However, as motion matching uses both future and past trajectory, this type of movement can be matched and realistic weight shifting achieved. The Snakes dance card is therefore used to capture this animation.

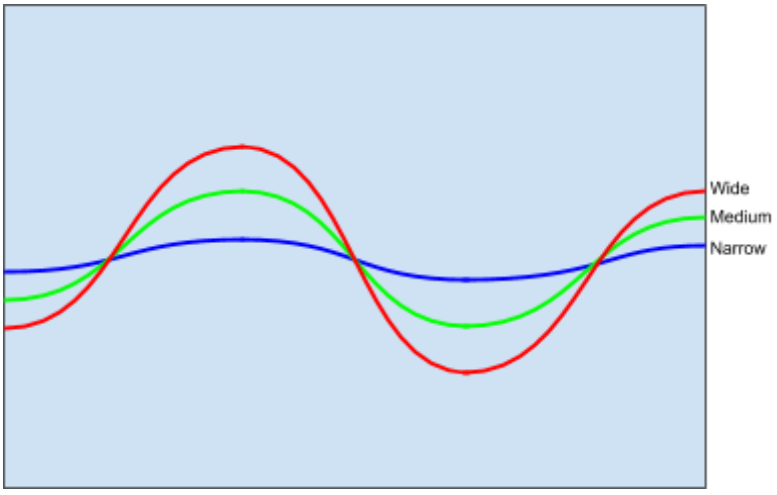


Figure 4.5 ‘Snake’ Dance Card (credit Kristjan Zadziuk GDC 2016)

To perform the snake dance card, start at one end of the capture volume and run to the other end in a winding ‘S’ pattern. Repeat this at narrow, medium and wide arcs for every movement speed present in the game. It may even be beneficial to perform each one twice, starting on different feet.

5 Recommended Animation List

The list below shows what animations are recommended for general locomotion. This list can be extrapolated and applied to different movement speeds or different stances. If the animations are to be created with motion capture these animations can be grouped into logical Dance Cards (See Dance Card Section).

Note that this is just an example animation list. Depending on gameplay needs, it may need less or more animations.

5.1 Character Following Nose

The following animations are for a set where the character always turns to face the direction they are going (i.e. not strafing).

Idle

- Idle (+ left / right foot forward variants if desired)
- Idle emotes (e.g. scuffing feet, scratching head)

Turn in Place

- Left 45
- Left 90
- Left 135
- Left 180
- Right 45
- Right 90
- Right 135
- Right 180

Move Start (Repeat for all speeds)

- MoveStart Forward
- MoveStart 45 Left
- MoveStart 90 Left
- MoveStart 135 Left
- MoveStart 180 Left
- MoveStart
- MoveStart 45 Right
- MoveStart 90 Right
- MoveStart 135 Right
- MoveStart 180 Right

Note: If the character is able to have mirrored stances (i.e. left foot or right foot forward idles) these start animations may need to also be mirrored or recaptured if the pose is not symmetrical.

Move Loops (Repeat for all speeds)

- Move Forward Loop (3 - 4 cycles recommended if using mocap)

Move Stops (Repeat for all speeds)

- MoveStop LeftFoot
- MoveStop RightFoot

Move Plants (Repeat for all speed)

- MovePlant 45 Left
- MovePlant 90 Left
- MovePlant 135 Left
- MovePlant 180 Left
- MovePlant 45 Right
- MovePlant 90 Right
- Move Plant 135 Right
- MovePlant 180 Right

Move Arcs (Repeat for all speeds)

- MoveArc Left Wide
- MoveArc Left Medium
- MoveArc Left Tight
- MoveArc Right Wide
- MoveArc Right Medium
- MoveArc Right Tight

Move Spirals (Repeat for all speeds) - recommended for m
ocap users

Accelerations / Decelerations

- Acceleration from every movements speed to every higher movement speed (e.g. walk -> run)
- Decelerations from every movement speed to lower movement speeds (e.g. run -> walk)

Snakes (Repeat for all speeds) - recommended for mocap users

- Wide snake pattern
- Medium snake pattern
- Tight snake pattern

Shuffles (Repeat for all speeds) - recommended for mocap users

- Shuffle Forward
- TurnShuffle 45 Left
- TurnShuffle 90 Left
- TurnShuffle 135 Left
- TurnShuffle 180 Left
- TurnShuffle 45 Right

- TurnShuffle 90 Right
- TurnShuffle 135 Right
- TurnShuffle 180 Right

Note: Shuffle refers to starting Moving and then stopping immediately within 1 step. This is to accurately animate when the player pushes on the stick and then lets it go immediately.

5.2 Character Strafing

The following animations are for a set where the character always strafes. For strafing motion matching requires a bit more animation than traditional systems. Many state machines reduce the amount of transitions and rely on blending to create transitions. This results in some strafing not looking realistic and it also creates a coverage gap for motion matching.

It is possible to calibrate the motion matching so that it works without the transitions but it's just better if they exist.

Strafe Starts (Repeat for all speeds)

- StrafeForward Start
- StrafeLeft 45 Start
- StrafeLeft 90 Start
- StrafeLeft 135 Start
- StrafeBack
- Strafe Right 45 Start
- Strafe Right 90 Start
- Strafe Right 135 Start

Strafe Loops (Repeat for all speeds) - (3 - 4 cycles per loop if possible)

- StrafeForward Loop
- StrafeLeft 45 Loop
- StrafeLeft 90 Loop
- StrafeLeft 135 Loop
- StrafeBack
- StrafeRight 45 Loop
- StrafeRight 90 Loop
- StrafeRight 135 Loop

Strafe Plants (Repeat for all speeds) (Recommended for mocap users)

- Strafe Forward Plant to Back
- StrafeLeft 45 Plant to 135 Right
- StrafeLeft 90 Plant to 90 Right
- StrafeLeft 135 Plant to 45 Right
- StrafeBack Plant to Forward
- StrafeRight 45 Plant to 135 Left
- StrafeRight 90 Plant to 90 Left

- StrafeRight 135 Plant to 45 Left

Note: It may be unfeasible (and unnecessary) to capture every possible strafe plant animation. This may be dependent on your animation budget. The more the better but will good calibration, it's possible to get reasonable results with fewer transitions.

Strafe Stops (Repeat for all speeds)
- LeftFoot Stop
- RightFoot Stop

Accelerations / Decelerations (Recommended for mocap users)
- Acceleration from every movements speed to every higher movement speed (at every angle) (e.g. walk45L -> run45L)
- Decelerations from every movement speed to lower movement speeds (at every angle) (e.g. run45L -> walk45L)

Shuffles (Repeat for all speeds) - recommended for mocap users
- Shuffle Forward
- StrafeShuffle 45 Left
- StrafeShuffle 90 Left
- StrafeShuffle 135 Left
- StrafeShuffle 180 Left
- StrafeShuffle 45 Right
- StrafeShuffle 90 Right
- StrafeShuffle 135 Right
- StrafeShuffle 180 Right

Note: Shuffle refers to starting Moving and then stopping immediately within 1 step. This is to accurately animate when the player pushes on the stick and then lets it go immediately.