

Motion Matching

Instrukcja obsługi dla wersji Unity 2019.4.8f1

1. Instalacja:

Należy wkleić folder z systemem do katalogu Assets projektu a następnie zainstalować za pomocą package manager pluginy:

1. Burst
2. Jobs
3. Mathematics
4. Collections

2. Ogólne omówienie systemu.

MotionMatching zaimplementowany jest w sposób, aby praca z nim była podobna do pracy z wbudowanym systemem animacji Mecanim.

Praca z systemem polega na:

1. Przygotowaniu AnimationData – są to assety odpowiadające animacjom
2. Przygotowanie MotionMatchingAnimatorController – jest to asset odpowiadający standardowemu AnimatorController

Następnym etapem jest dodanie komponentu **MotionMatching** w którym umieszczamy wcześniej przygotowany kontroller, oraz komponentu **TrajectoryMaker** który na podstawie danych pobranych od użytkownika tworzy trajektorię potrzebną dla poprzedniego komponentu.

MotionMatchingAnimatorController – posiada stany w których umieszczane są AnimationData. Pomiędzy poszczególnymi możemy tworzyć przejścia które następują na podstawie określonych w nich warunków.

3. AnimationData jako odpowiednik animacji.

AnimationData jest kontenerem na animacje z których została utworzona oraz na dane klatek animacji (obliczane co ustalony interwał) takie jak:

1. Trajektoria – wskazuje gdzie animacji przeniesie animowany obiekt po określonym czasie
2. Poza – przechowuje informacji o wybranych kościach takie jak ich lokalna pozycja oraz lokalna prędkość jako Vector3
3. Przynależność do sekcji animacji

Dane te są obliczane podczas tworzenia AnimationData i nie mogą być później zmieniane (chyba że animacje zostaną przeliczone ponownie).



Inspector AnimationData

AnimationData jest tworzona za pomocą specjalnego edytora. Po jej utworzeniu możliwa jest zmiana niektórych z jej opcji.

Opis Inspektora AnimationData:

Basic Options:

Data Type – typ Animation data. Aktualnie istnieją trzy typy:

1. SingleAimation

2. BlendTree
3. AnimationSequence

Jest on wykorzystywany przez Komponent MotionMatching w celu poprawnego tworzenia animacji w AnimationPlayableGraph.

Length – długość animacji z jakich jest stworzona AnimationData. W przypadku BlendTree jest to długość najdłuższej animacji. W przypadku AnimationSequence jest to długość obliczona na podstawie odpowiednich opcji (więcej o tym w opisie **Motion Matching Data Creator**).

Motion Matching Data Frame Time – czas co jaki była próbkowana animacja w celu obliczenia danych wykorzystywanych przez MotionMatching.

Motion Matching Data Frame Count – Ilość obliczonych klatek animacji posiadających dane potrzebne do działania Motion Matchingu.

Blend To Yourself – MotionMatching polega na ciągłym wyszukiwaniu najlepszego miejsca w danym zbiorze animacji. CheckBox ten odpowiada za zachowanie w następującym momencie: Aktualnie odtwarzana jest AnimationData „A” jeżeli w następnym szukaniu znowu wylosuję tę samą animację „A” i **BlendToYourself** będzie wynosiło **true** nastąpi blendowanie do animacji, w przeciwnym przypadku blendowanie nie wystąpi. Opcja ta powinna być odznaczona w przypadku krótkich zapętlonych animacji takich jak chodzenie lub bieganie.

Find in Yourself – jeżeli opcja ta jest odznaczona w następnym szukaniu najlepszej animacji mamy pewność że nie trafimy do żadnej animacji znajdującej się w AnimationData. Opcja ta może być przydatna w niektórych przypadkach lecz głównie powinna wynosić **true**.

Sections

Każda AnimationData posiada Sekcje. Sekcje są to przedziały czasu którymi można oznaczać poszczególne animacje. Ich zastosowanie jest podwójne:

1. Posiadamy długą animację trwającą kilka minut. Część tej animacji stanowi chodzenie, natomiast część bieganie. Tworząc odpowiednie sekcje w AnimationData możemy później z poziomu kodu wybierać z których sekcji mają być wybierane animacje.
2. Sekcje mogą zostać wykorzystane w odpowiednim stanie w MotionMatchingAnimatorController do określenia zależności (Więcej o tym przy omówieniu **SectionDependencies**).

AnimationData posiada 3 podstawowe sekcje które nie mogą zostać usunięte, mogą natomiast być edytowane. Sekcje te to:

Never checking – sekcja ta określa klatki które nie będą nigdy przeszukiwane przez system MotionMatching, nawet jeżeli klatki znajdujące się w tej sekcji będą znajdować się również w innej sekcji.

Not looking for new pose – jeżeli aktualny czas aktualnie odtwarzanej animacji zawiera się w przedziałach oznaczonych przez tę sekcję nie wykona się kolejne szukanie animacji. Może ona zostać wykorzystana np. do oznaczenia animacji szybkiego zwrotu podczas biegu. Jeżeli oznaczymy w niej czas w którym wykonuje się ten zwrot i zostanie wybrana ta animacja w tym czasie, następne wyszukiwanie nastąpi dopiero w czasie animacji nie znajdującym się w tej sekcji.

Always – jest to podstawowa sekcja w której jest wykonywane wyszukiwanie po wejściu do danego stanu.

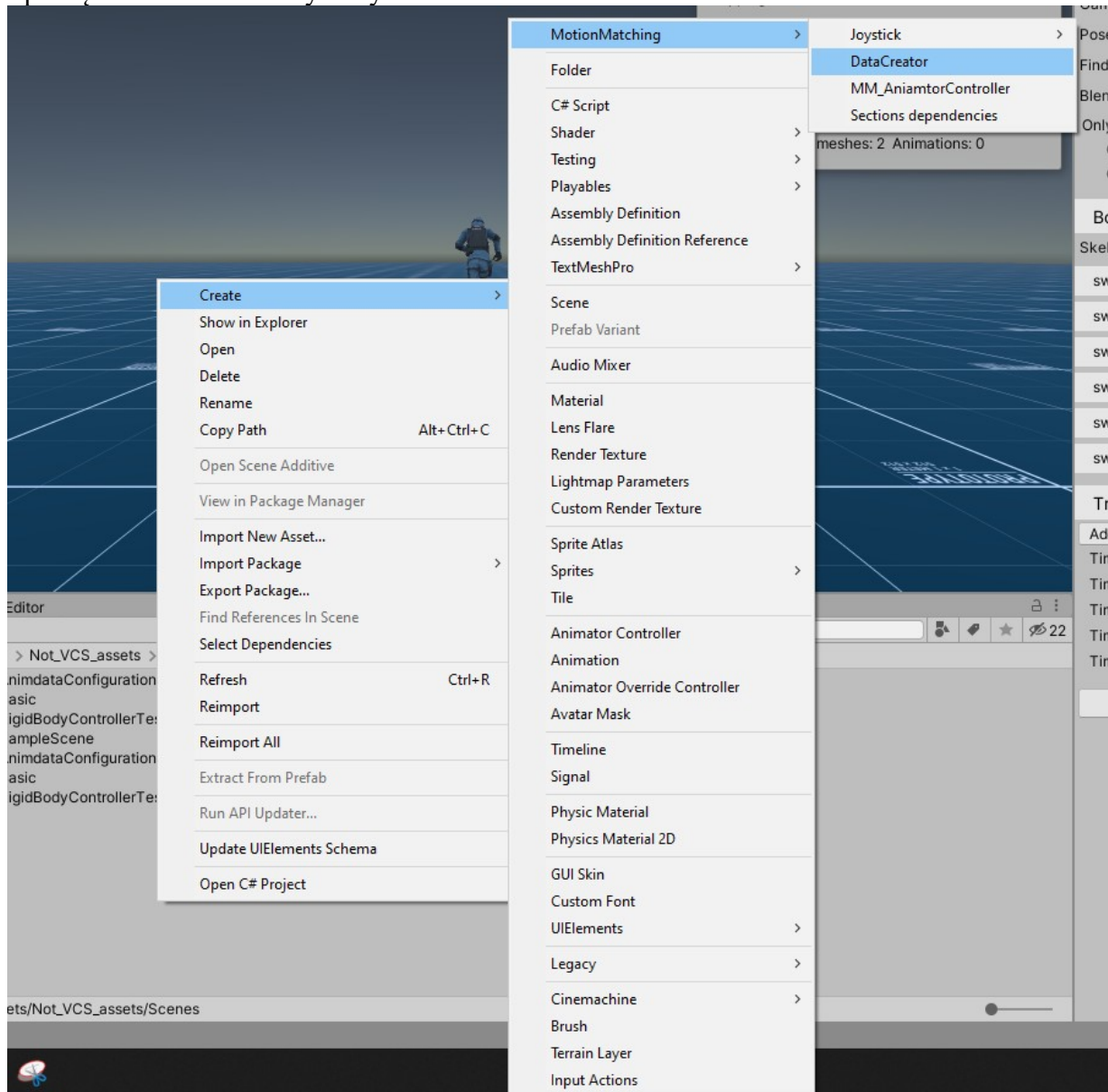
Każda AnimationData może posiadać dodatkowe 31 sekcji o dowolnych nazwach. Sekcje można dodawać z poziomu inspektora, za pomocą SectionDependencies i Motion Matching Animator Controller oraz MotionMatchingDataEditor.

Data type options – wyświetlane są tu informacje specyficzne dla typu AnimationData w celu informacyjnym.

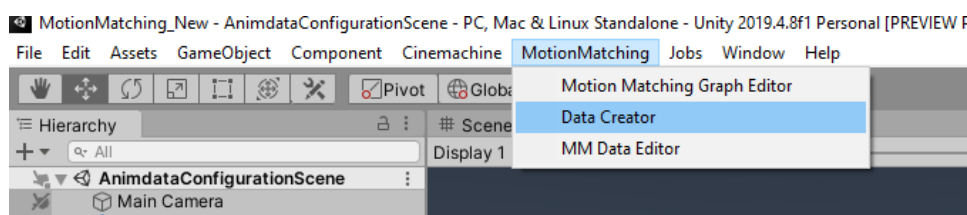
Contact Points – wyświetlane są tu informacje o posiadanych przez AnimationData punktów kontaktów (dodajemy je za pomocą Motion Matching Data Editor).

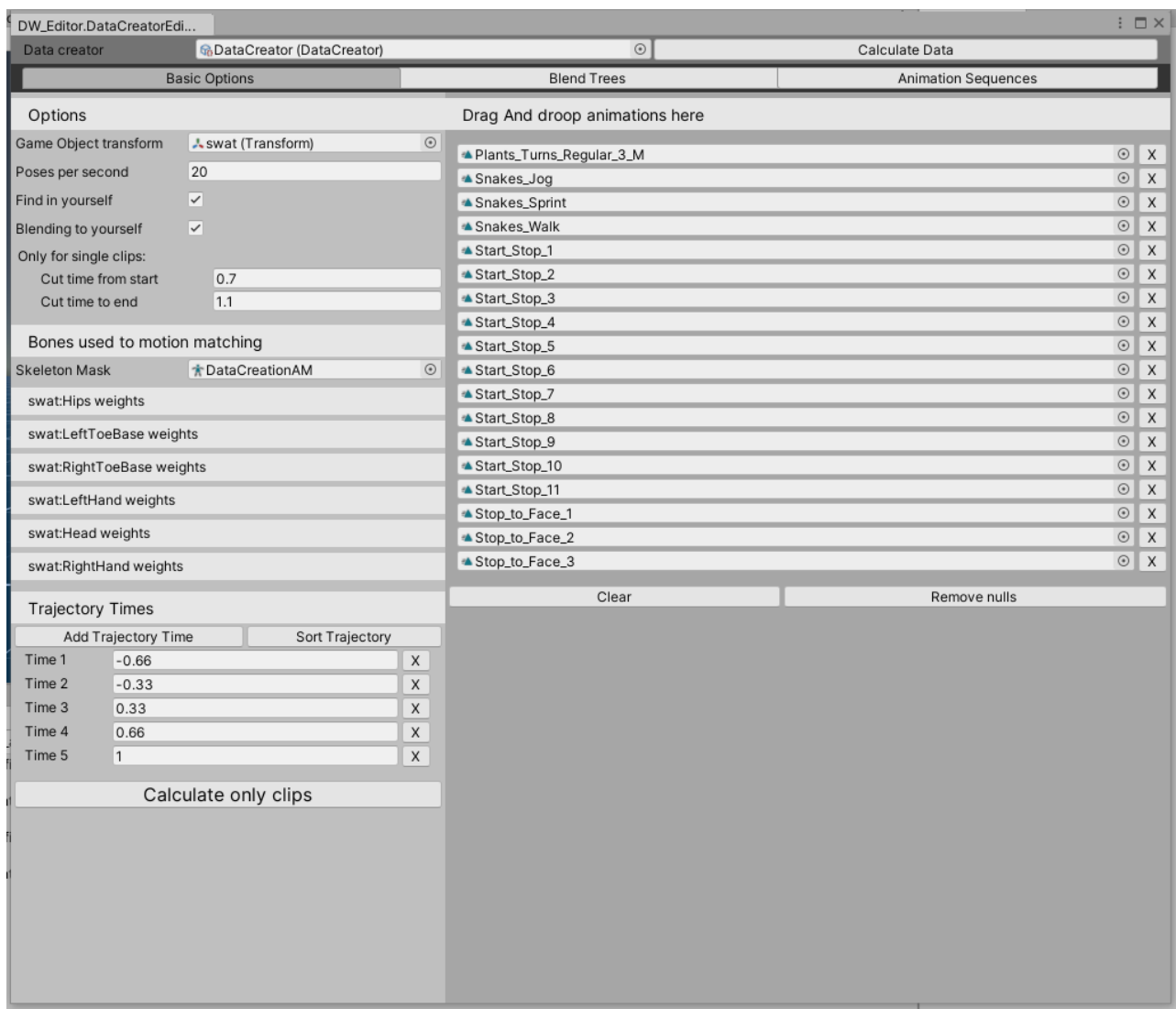
4. Tworzenie **AnimationData** – instrukcja obsługi **Motion Matching Data Creator**

Jest to edytor za pomocą którego tworzymy assety AnimationData. Edytor ten pozwala na tworzenie 3 typów AnimationData (SingleAnimation, BlendTree, AnimationSequence). Rozpoczęcie tworzenia zaczynamy od utworzenia assetu kreatora:



Dwukrotnie klikając na utworzony asset powinien pojawić się Data Creator Editor. Jeżeli edytor nie został utworzony w górnym menu wberamy Motion Matching, a następnie Data Creator Editor.





W górnej części edytora znajduje się pole na aktualnie otwarty edytor, oraz przycisku po kliknięciu którego rozpocznie się przeliczanie i tworzenie wszystkich animacji. Przed rozpoczęciem obliczania należy poprawnie ustawić wszystkie opcje.

Pod spodem mamy do wyboru trzy typy edytorów:

Basic Options – w tym menu edytora ustawiamy podstawowe opcje z jakimi mają być przeliczane animacje. Dodajemy tu animacje po przeliczeniu których utworzy się AnimationData typu SingleAnimation. Animacje dodajemy poprzez upuszczenie wybranych animacji na pole „Drag and drop animations here”. Znajdować się tu powinny długie Mocapy oraz zapętlone „cut clips”.

Po lewej stronie ustawiamy podstawowe opcje z jakimi mają zostać przeliczone animacje.

Game Object transform – animacje przeliczane są poprzez ich szybkie odtwarzanie za pomocą systemu Unity AnimationPlayables. W tym polu umieszczamy obiekt na którym mają zostać odtworzone animacje i przeprowadzone obliczenia.

Poses per second – ile klatek danych na podstawie których będzie działał MotionMatching ma zostać utworzonych w jednej sekundzie animacji. Im więcej klatek tym lepsze rezultaty lecz mniejsza wydajność, wartość ta powinna zawierać się w przedziale od 10 do 30.

Find in yourself – jakie powinno być domyślne ustawienie dla obliczanych AnimationData.

Blending to yourself – jakie powinno być domyślne ustawienie dla obliczanych AnimationData.

Cut time from start – oznacza jaki przedział czasu powinien być oznaczony w sekcji Never checking od początku animacji.

Cut time from start – oznacza jaki przedział czasu powinien być oznaczony w sekcji Never checking od koniec Animacji – Cut time from start do końca animacji.

Cut time from start/end powinny być używane tylko przy długich animacjach pochodzących z sesji motion capture, które nie są zapętlone. Dzięki ustawieniu tych opcji będziemy mieli pewność że nie będziemy sprawdzać animacji na jej początku i końcu gdzie przeliczona trajektoria klatek znajdujących się w tych miejscach może być niepoprawna.

Bone used to motion matching

Skeleton mask – jest to utworzony asset Avatar Mask z zaznaczonymi tymi kośćmi z których zostanie utworzona informacja o pozie w poszczególnych klatkach AnimationData. Wybranych kości nie powinno być za dużo 5 w zupełności wystarcza. Powinny być zaznaczone kości dłoni, stopy, kość szyi kość miednicy. Nie zawsze więcej zaznaczonych kości daje lepsze rezultaty.

Po umieszczeniu **Skeleton mask** i **Game Object transform** który posiada szkielet na którym została wykonana maska wyświetli się lista wybranych kości.

Trajectory times

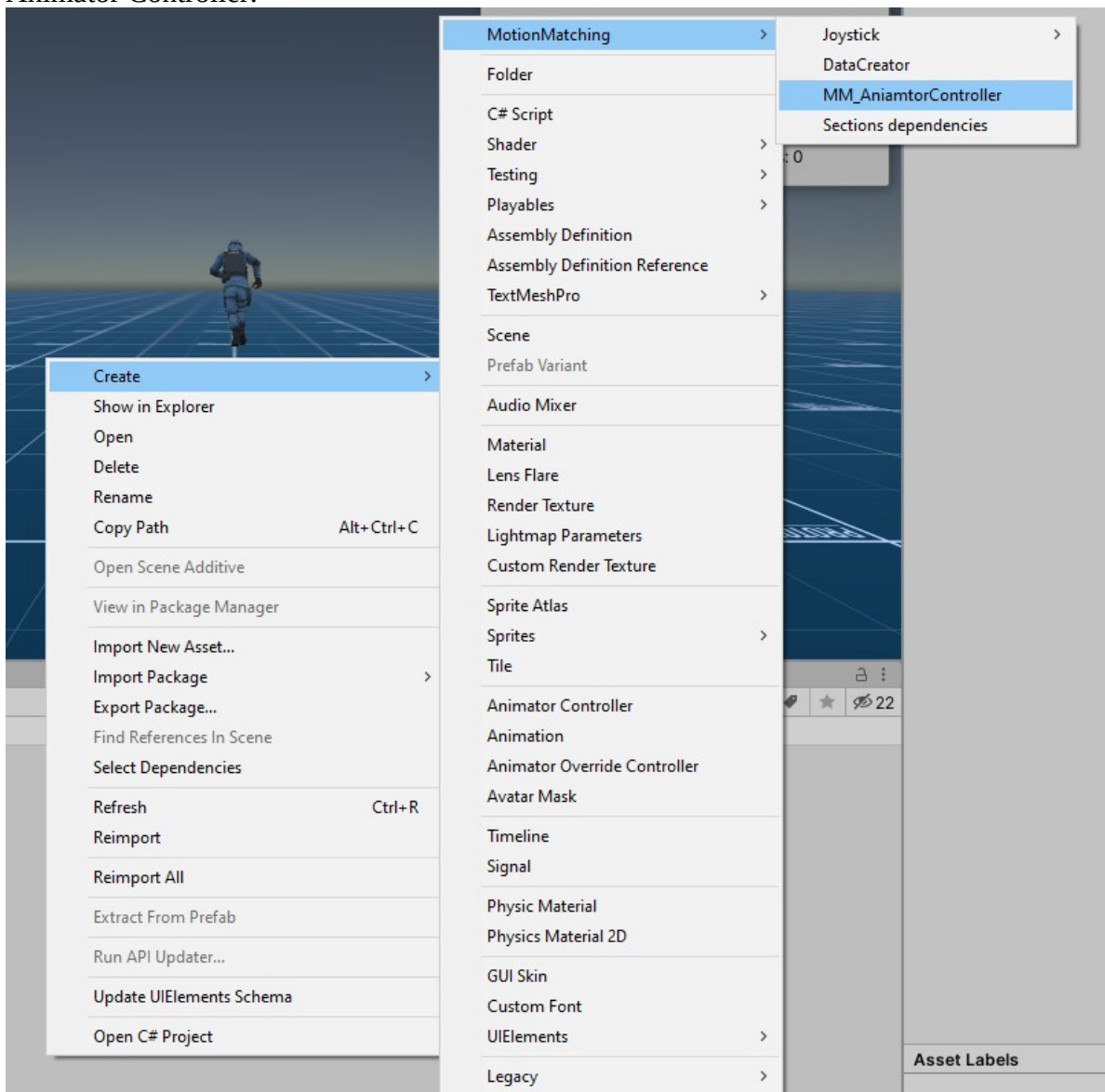
Dodajemy tuja czasy w jakich powinna być utworzona trajektoria w każdej klatce danych znajdujących się w AnimationData.

Nie należy przesadzać z ich ilością. Im więcej punktów trajektorii tym więcej obliczeń i mniejsza wydajność, 5 punktów wystarczy z czego dwa powinny mieć czas ujemny.

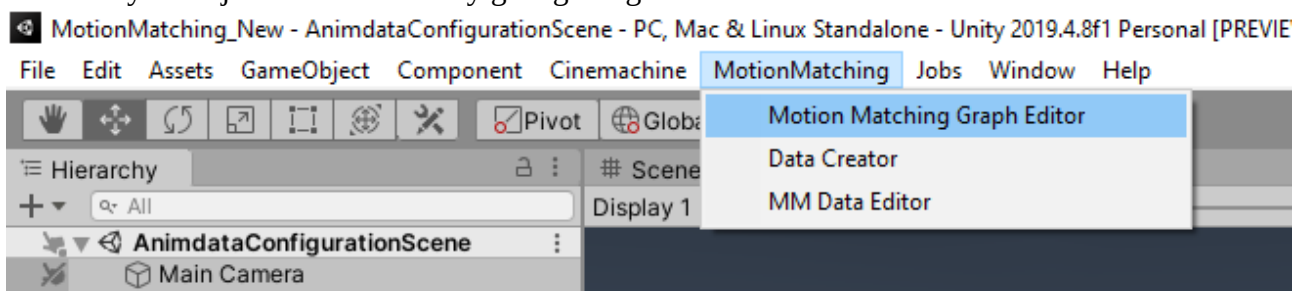
Przycisk **Calcualte Only Clips** powoduje że obliczane będą tylko animacje z Basic Options bez animacji z menu: **BlendTrees** i **Animation Sequences**.

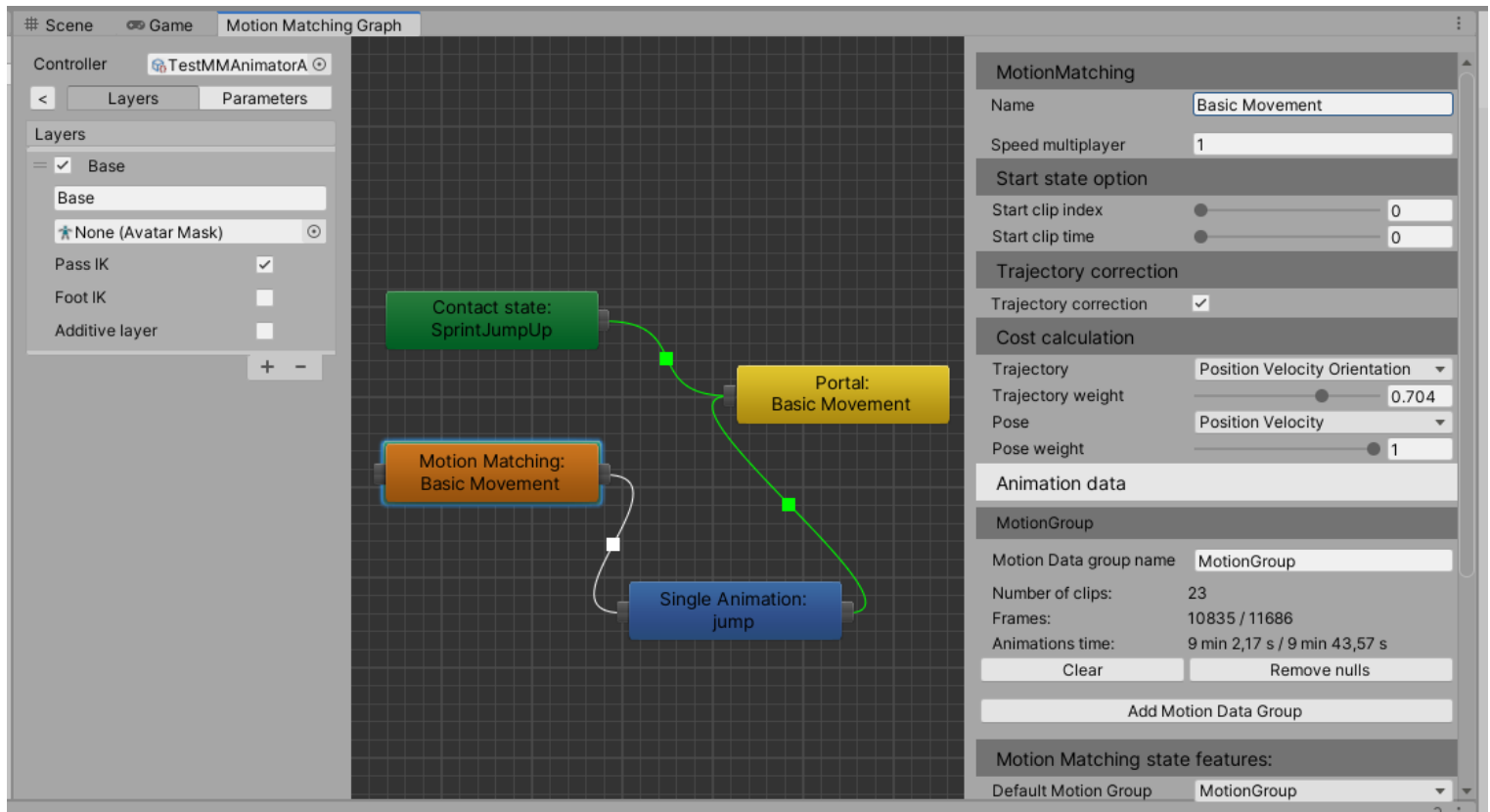
5. Motion Matching Graph Editor – Tworzenie Motion Matching Animator Controller – odpowiednik Animator Controller.

Pracę z Motion Matching Graph Editor rozpoczynamy od utworzenia assetu Motion Matching Animator Controller:



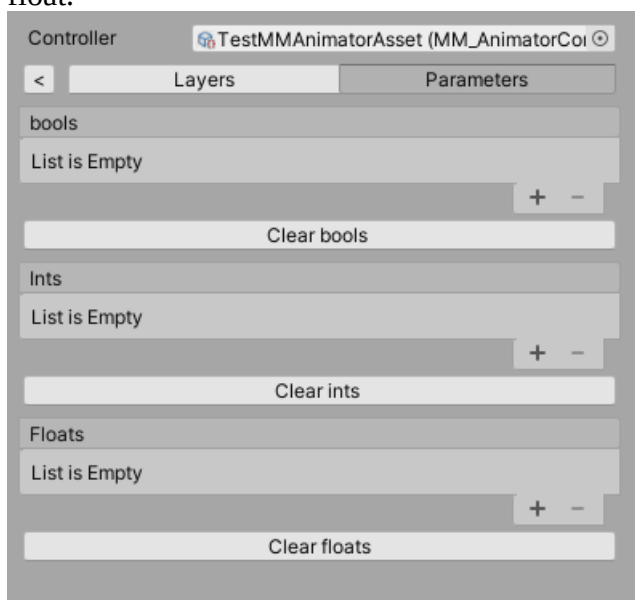
Motion Matching Graph Editor powinien zostać otworzony po dwukrotnym kliknięciu w w utworzony asset jeżeli nie otwieramy go z górnego menu:





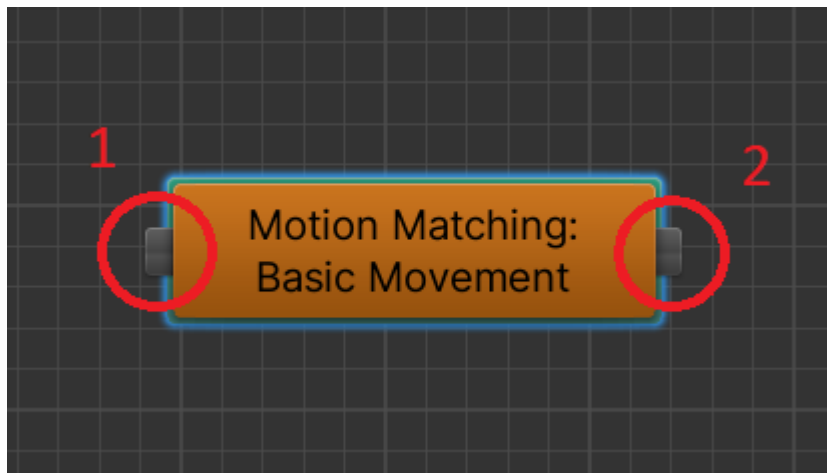
MMGE posiada 3 pola:

1. Left Space – dzieli się ono na dwa menu:
 1. Layers – widoczne powyżej, wyświetlane są tutaj informacje o warstwach, również możemy tworzyć nowe warstwy i ustawiać ich opcje.
 2. Parameters – ustawiamy tutaj parametry za pomocą których będziemy ustawiać warunki przejść pomiędzy stanami. Aktualnie możliwe jest dodawanie parametrów typu bool, int i float.



3. Przycisk oznaczony znakiem „<” pozwala na zwinięcie menu do lewej strony. Jego ponowne otwarcie wykonuje się kliknięciem bardzo blisko lewej krawędzi edytora.

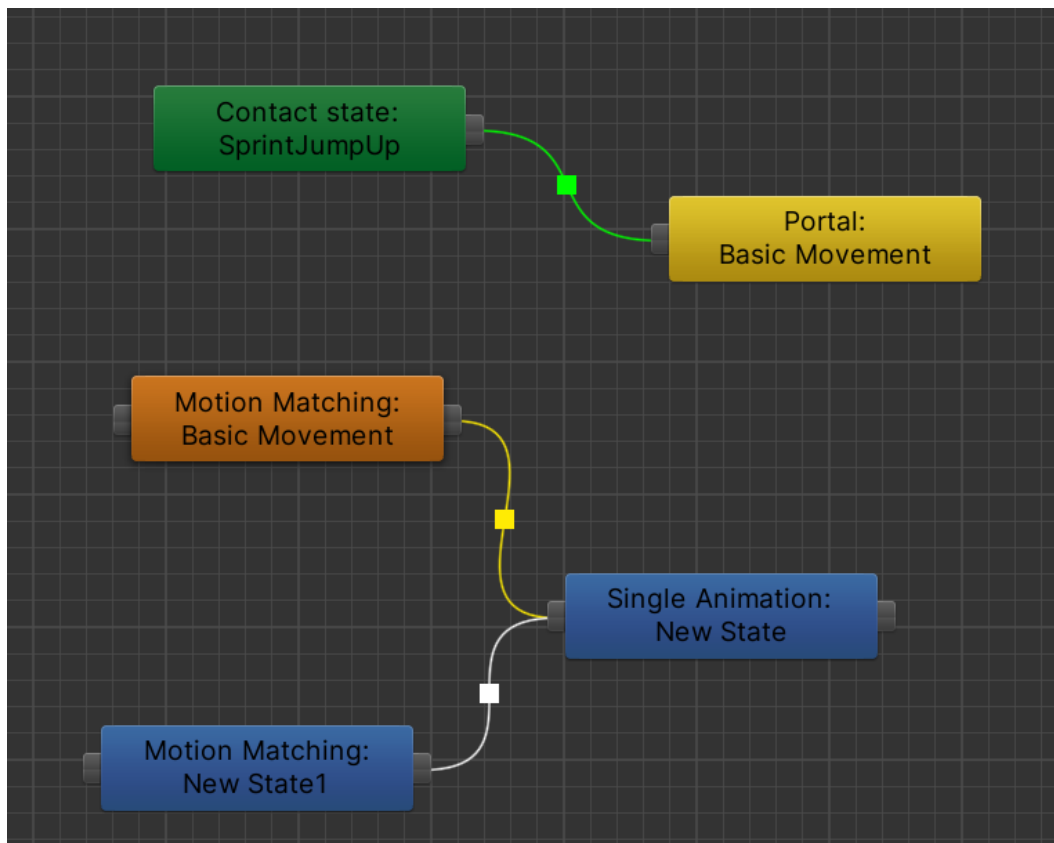
2. Graph Space – tworzymy tutaj wizualną reprezentację stanów oraz przejść pomiędzy nimi. Graf animacji z założenia tworzymy od lewej do prawej.



- 1 – wejście stanu (input)
2 – wyjście stanu (output)

Aktualnie możemy utworzyć 4 typy stanów:

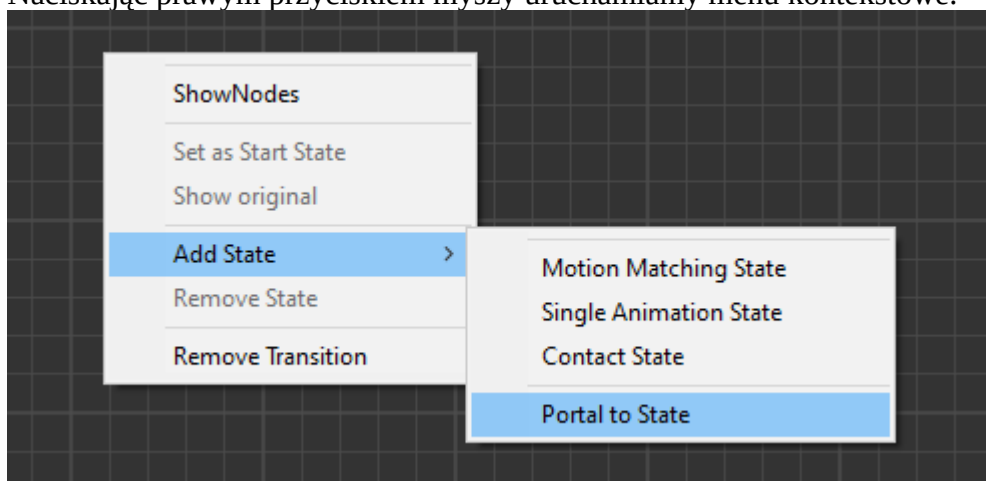
1. Motion Matching State – jest to stan realizujący ideę MotionMatchingu.
2. Single Animation State – jest to stan w którym odtwarzana jest pojedyncza animacja (można to zmienić) wybrana ze wszystkich znajdujących się w tym stanie animacji na podstawie motion matchingu.
3. Contact State – jest to stan który może posiadać tylko wyjścia. Do stanu możemy wejść tylko z poziomu kodu używając do tego specjalnej metody komponentu MotionMatching.
Contact state dzieli się na dwa kolejne stany (zmienia się to w Right Space)
 1. Contact – postać jest przemieszczana pomiędzy ustalonymi wcześniej punktami kontaktów.
 2. Impact – animacja wybierana jest na podstawie przekazanego Impact Pointu (aktualnie w fazie testowej).
4. Portal State – jest to stan który graficznie reprezentuje inny stan. Portal state nie posiada wyjść.



Kolor pomarańczowy oznacza że stan jest stanem startowym wybranej warstwy i od niego rozpoczyna się działanie systemu. Stanem startowym mogą być stany typu SingleAnimation oraz MotionMatching.

Kolorem żółtym mamy oznaczone portale, kolorem zielonym stany Contact/Impact, natomiast kolorem niebieskim pozostałe stany. Połączenia pomiędzy stanami SingleAnimation, Contact oraz MotionMatching są koloru białego, na żelono oznaczone są połączenia z portalami. Kolorem żółtym oznaczone jest aktualnie wybrane połączenie.

Naciskając prawym przyciskiem myszy uruchamiamy menu kontekstowe.



ShowNodes – przenosi widok w Graph Space do środka ciężkości wszystkich stanów.

Set as Start State – ustawia wybrany stan jako startowy

Show original – jeżeli aktualnie wybrany stan jest typu Portal przenosi widok do oryginalnego stanu.

Add State – pozwala na tworzenie stanów określonych typów

Remove State – usuwa wybrany stan

Remove Transition – usuwa wybrane połączenie

Połączenia pomiędzy stanami tworzymy poprzez naciśnięcie lewego przycisku myszy i puszczeniu go gdy kursor wskazuje stan z którym chcemy się połączyć. Pomiedzy dwoma różnymi stanami możemy posiadać jedno połączenie jeżeli stan ma już połączenie z innym stanem, a chcemy mieć możliwość powrotu do niego musimy użyć Portal State.

Przejścia wybieramy poprzez kliknięcie ka kwadrat znajdujący się na ich środku.

3. Right Space – jest miejsce w którym wyświetlane są opcje aktualnie wybranego elementu w Graph Space.

PORTAL STATE:



Portal node state selection

State name

Basic Movement
New State
New State1

Możemy w nim wybrać którego stanu jest to portal.

Stany **Motion Matching**, **Single Animation** oraz **Contact**, część opcji mają identyczne, oraz każdy z nich ma swoje własne specyficzne dla typu ustawienia.

ELEMENTY WSPÓLNE:



MotionMatching

Name Basic Movement

Speed multiplayer 1

Na samym początku wyświetlany jest typ wybranego stanu, możemy tu zmienić nazwę stanu, oraz szybkość z jaką będą odtwarzane animacje.

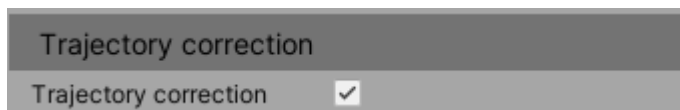


Start state option

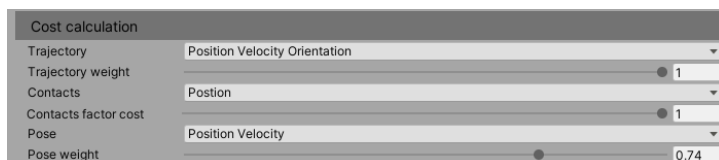
Start clip index 0

Start clip time 0

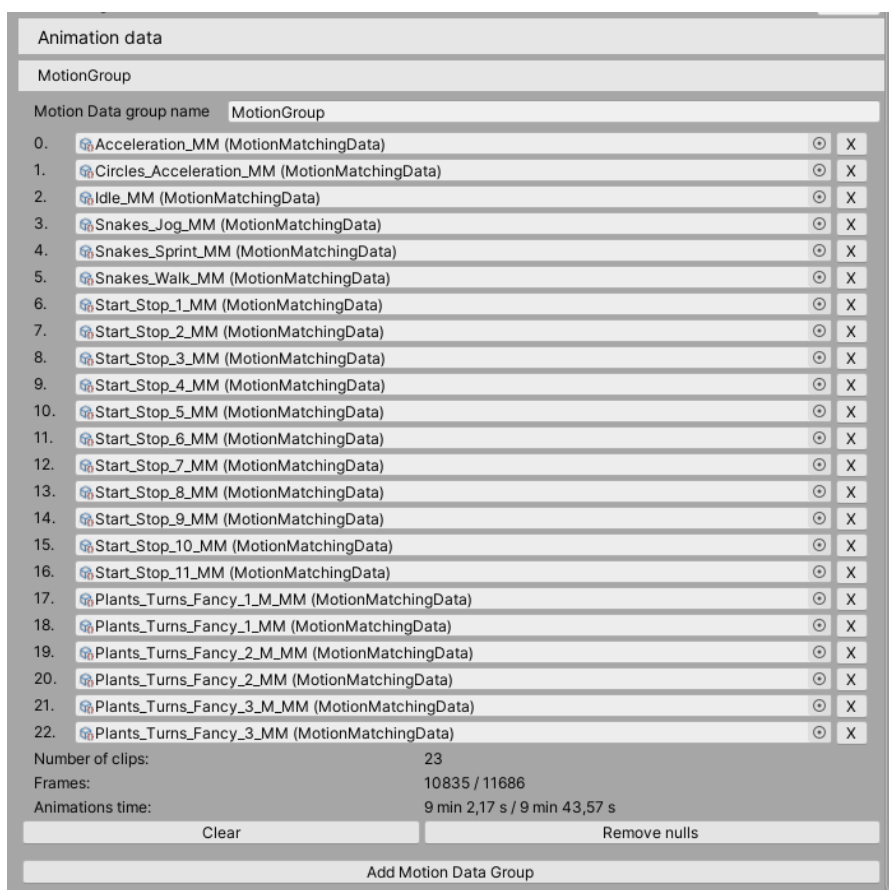
Start state option są to opcje pojawiające się tylko w stanie startowym. Możemy ustawić tu index startowy animacji, oraz czas w którym rozpocznie się jej odtwarzanie.



Jeżeli opcja Trajectory Correction jest zaznaczona, będzie następować korekcja trajektorii, jej opcje ustawia się w komponencie Motion Matching i tam zostaną one omówione. Korekcja trajektorii może być wykonana w Motion Matching State oraz Single Animation State.



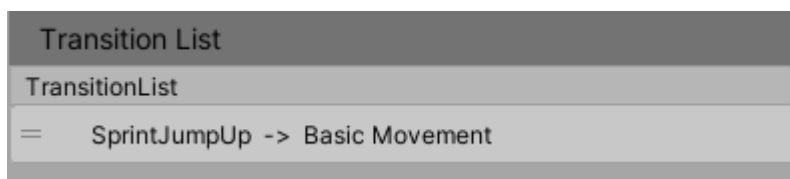
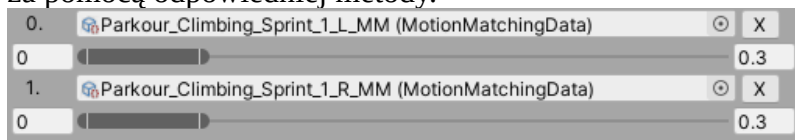
W opcjach Cost calculation możemy wybrać metody kosztów oraz ich wagi, różne typy stanów mogą mieć więcej lub mniej opcji do ustawienia w tej sekcji.



Animation Data – tutaj umieszczamy assety AnimationData z których będzie korzystał wybrany stan. Stan typu Motion Matching może mieć kilka grup danych pomiędzy którymi można się przełączać. Możemy z nich skorzystać jeżeli chcemy mieć w jednej grupie animacje tylko z normalny poruszaniem się a w drugiej animacje poruszania się gdy nasz bohater jest zmęczony. Zmieniania aktualnej grupy dokonujemy z poziomu kodu, odpowiednią metodą komponentu Motion Matching. Po kliknięciu na tą belkę możemy zwinąć listę animacji. Poniżej listy wyświetla się podsumowanie danej grupy: ilość

animacji, łączna ilość klatek (klatki używane przez stan / wszystkie klatki w Animation Data) oraz łączny czas animacji (używanych klatek/wszystkich klatek animacji) Animation data do grup dodajemy poprzez upuszczenie ich na belkę z nazwą grupy. Stany Single animation oraz Contact mogą posiadać tylko jedną grupę i nie ma możliwości dodania kolejnych.

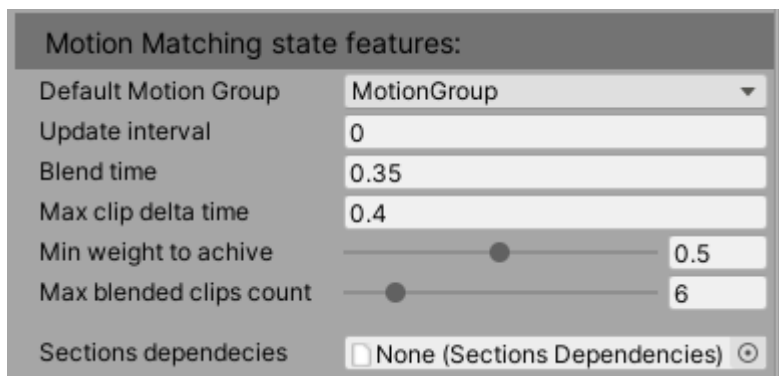
Dodatkowo w stanach typu Contact oraz Single Animation mamy możliwość gdzie będzie wyszukiwana animacja jeżeli zdecydujemy się na przejście do stanu bezpośrednio z kodu, za pomocą odpowiedniej metody.



W sekcji Transition List mamy listę przejść z wybranego stanu, możemy w niej zmieniać kolejność w jakiej będą sprawdzane.

Ostatnią grupą opcji jest **State Features**, każdy z typów stanów(MotionMatching, SingleAnimation oraz Contact) posiadają różne opcje w tej sekcji.

MOTION MATCHING STATE FEATURES



Default Motion Group – grupa w której będą wyszukiwane animacje po wejściu do stanu.

Update interval – co jaki czas będzie wykonywane szukanie kolijenej najlepszej animacji.

Jeżeli wynosi zero szukanie będzie wykonywane z częstotliwością równą aktualnym klatkom na sekundę z jaką działa program.

Blend time – czas w jakim będą blendowały się kolejne wylosowane animacje.

Max clip delta time – jeżeli podczas wyszukiwania trafimy do tej samej animacji i różnica pomiędzy czasem aktualnie odtwarzanej animacji i wylosowanej będzie mniejsza lub równa tej opcji nie nastąpi blendowanie do wylosowanej animacji.

Min weight to achieve – jest to waga jaką musi osiągnąć wylosowana animacja zanim jej waga ponownie zacznie maleć. Waga ta jest jeszcze przed normalizacją. Ustawienie jej wartości na większą od zera powoduje że blendowanie pomiędzy animacjami jest bardziej płynne, i nie występuje efekt szybkiego przeskakiwania pomiędzy animacjami.

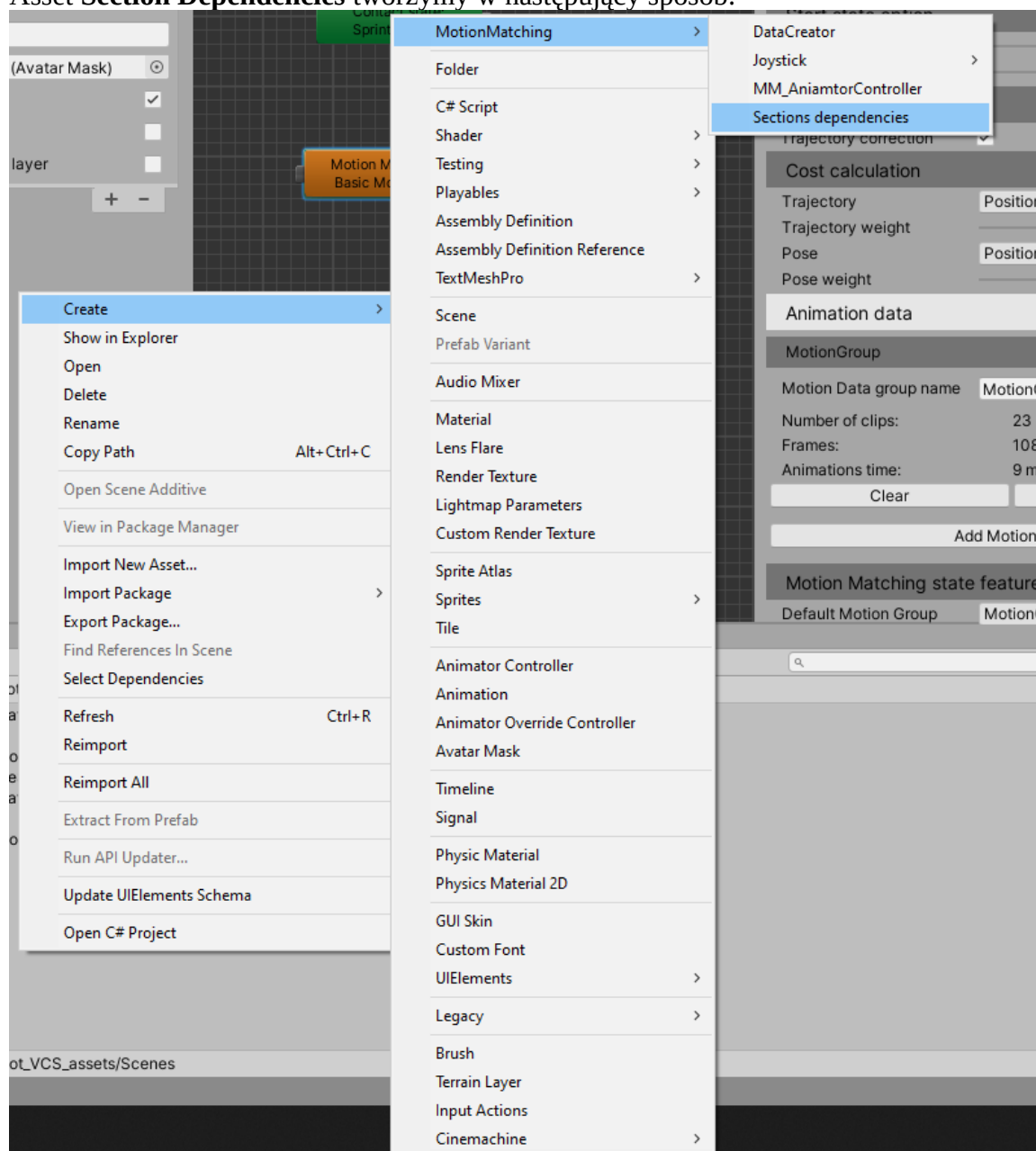
Max blended clips count – określa maksymalną ilość animacji jaka może być jednocześnie

blendowana. Jeżeli aktualnie blendowanych jest tyle animacji ile wynosi ta opcja nie następuje wyszukiwanie kolejnej animacji. Podczas testowania wyszło że wartość ta nie powinna przekraczać 10, i nie powinna być mniejsza niż 4. Możliwe jest ustawienie większej ilości.

Section Dependencies:

Motion Matching polega na ciągłym przeszukiwaniu animacji i wybieraniu najlepszego miejsca pasującego do aktualnej sytuacji. Wyszukiwanie to polega na obliczaniu kosztu do każdej z klatek danych i wybraniu tej o najmniejszym koszcie.

Section Dependencies (zależności sekcji) jest to asset za pomocą którego możemy wpływać na koszt przeszukiwanych klatek w zależności od tego w do których sekcji należy aktualny czas aktualnie odtwarzanej animacji i do których sekcji należą przeszukiwane klatki danych. Asset **Section Dependencies** tworzymy w następujący sposób:



Asset ten możemy edytować poprzez jego wybór w oknie Project, i edytowanie go w Inspektorze. Inspektor **Section Dependencies** prezentuje się następująco:

SectionsDependencies

Add Section

0. Always

Section X

Section name Section

Cost to section Cost weight

Section	1	X
Section_1	1.2	X
Section_2	0.8	X

Add section info

Section_1 X

Section name Section_1

Cost to section Cost weight

Section_2	1.2	X
-----------	-----	---

Add section info

Section_2 X

Section name Section_2

Cost to section Cost weight

Section	0.95	X
---------	------	---

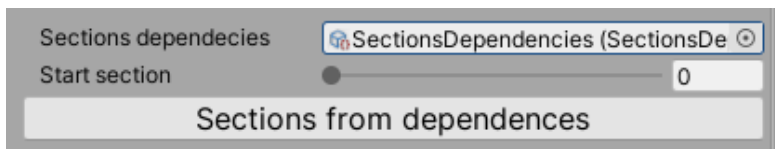
Add section info

Przyciskiem „Add Section” dodajemy nową sekcję. Następnie klikając na belkę z nazwą sekcji rozwijamy opcje jakie możemy ustawić, możemy zmienić jej nazwę, oraz ustalić koszty przejścia do wybranych sekcji.

Za pomocą przycisku „Add Section Info” dodajemy informację w której możemy ustawić do której sekcji koszt ma mieć ustaloną wagę. Z rozwijanej listy możemy wybrać sekcję dla której koszt ustawiamy. Jeżeli koszt sekcji jest mniejszy od 1 oznacza to że klatka należąca do tej sekcji ma większą szansę na zostanie wybraną. Odpowiednio koszt większy od 1 oznacza że klatka należąca do tej sekcji ma mniejszą szansę na wybranie. Jeżeli klatki należą do kilku sekcji jednocześnie, koszt przejścia do tych sekcji będzie ustalony, finalny koszt przejścia do tej klatki będzie iloczynem kosztów sekcji, np. finalny koszt klatki należącej do Section_1 i Section_2 obliczany z sekcji Section będzie wynosił:

$$\text{FinalCost} = \text{FrameCost} * \text{CostWeight}_{\text{Section}_1} * \text{CostWeight}_{\text{Section}_2}$$

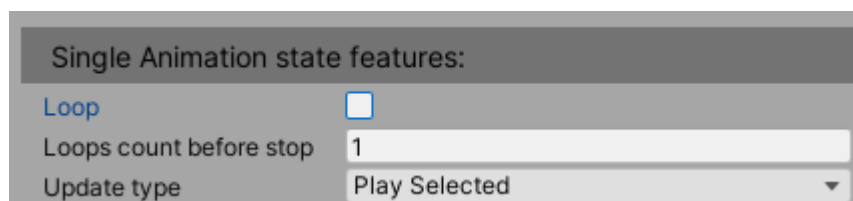
Po umieszczeniu assetu **Section Dependencies** w polu w MotionMatching state features pojawiają się dodatkowe opcje:



StartSection – jest to numer sekcji która będzie standardowo ustawiana podczas wejścia do stanu.

Sections from dependencies – jest to przycisk po wciśnięciu którego w każdej AnimationData w stanie zostaną dodane sekcje utworzone w assecie **Section Dependencies**.

SINGLE ANIMATION STATE FEATURES:



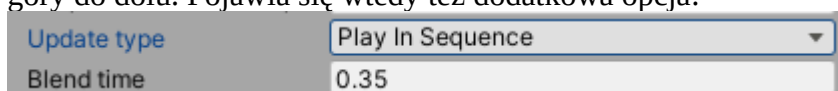
Loop – jeżeli zaznaczone wybrana AnimationData będzie odtwarzać się w pętli (o ile należąca do niej animacja jest zapętlona).

Loops count before stop – ilość petli jaką wykona wybrana AnimationData zanim się zatrzyma

Dwie powyższe opcje mogą nie działać!! do sprawdzenia.

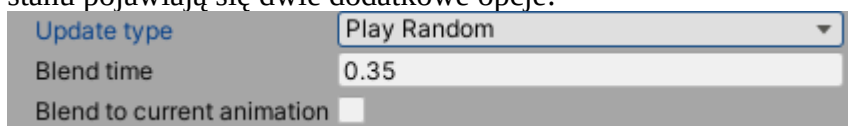
Update type – jest to typ z jakim zachodzą aktualizacje animacji danego stanu. Aktualnie istnieją 3 typy aktualizowania:

1. Play Selected – odtwarzana jest jedna wybrana na wejściu animacja.
2. Play in Sequence – animacje odtwarzane są po kolei zgodnie z ich ułożeniem od góry do dołu. Pojawia się wtedy też dodatkowa opcja:



Blend time – jest to czas blendowania pomiędzy animacjami.

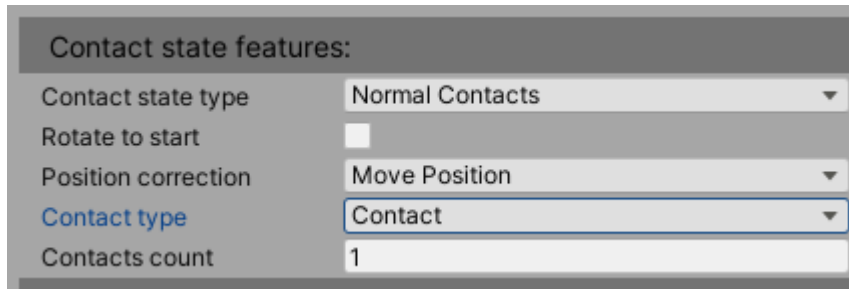
3. **Play Random** – po skończeniu odtwarzania wylosowanej animacji w Single Animation State jest losowana losowo kolejna animacja do odtworzenia. Animacja nie jest losowana za pomocą Motion Matchingu. Po wybraniu tego typu aktualizacji stanu pojawiają się dwie dodatkowe opcje:



Blend time – jest to czas blendowania pomiędzy animacjami.

Blend to current animation – jeżeli jest zaznaczona, wylosowana losowo animacja może być tą samą która jest aktualnie odtwarzana. W przeciwnym wypadku na pewno nie zostanie wylosowana ta sama animacja.

CONTACT STATE FEATURES



Contact state features:	
Contact state type	Normal Contacts
Rotate to start	<input type="checkbox"/>
Position correction	Move Position
Contact type	Contact
Contacts count	1

Contact state type – możemy tutaj wybrać typ contact stanu który był już wspomniany wcześniej. Aktualne typy to **Normal Contacts** oraz **Impacts**.

Zasada działania Contact State:

Po wejściu do Contact State wybierana jest najlepsza animacja na podstawie kosztu trajektorii, pozy oraz kosztu punktów kontaktu. Następnie na podstawie ustawień Contact State Features, animowany obiekt jest przesuwany odpowiednio pomiędzy przekazanymi punktami kontaktów. Jednak zanim zaczniemy używać Contact State musimy najpierw przygotować odpowiednio Animation Data które są umieszczone w stanie, czyli dodać do nich i przeliczyć punkty kontaktów. Robi się to za pomocą **Motion Matching Data Editor** (więcej o tym przy opisie Motion Matching Data Editor).

Do Contact State możemy wejść tylko z poziomu kodu. Jest to spowodowane tym że przy wejściu do Contact State musimy przekazać do niego odpowiednią ilość punktów kontaktu zależną od ustawień stanów oraz Animation Data umieszczonych w stanie.

Ważne !!!!!!!

Wszystkie Animation Data umieszczone w stanie typu Contact muszą mieć identyczną ilość kontaktów.

Rotate to start – jeżeli jest zaznaczone po wejściu do stanu animowany obiekt zostanie ustawiony w odpowiedniej rotacji do startowego punktu kontaktu.

Position Correction – jest to sposób w jaki będzie przesuwany obiekt pomiędzy punktami kontaktu. Aktualnie istnieją dwie metody:

Move position – obiekt jest przesuwany na podstawie obliczanej co klatkę delty pomiędzy przekazanym do stanu punktem kontaktów a punktem kontaktu w wybranej animacji. Jest dużo płynniejsza niż druga metoda ale czasami może powodować przeskoki przesuwanego obiektu.

Lerp position – obiekt przesuwany jest za pomocą funkcji Vector3.Lerp. Poruszanie się obiektu może nie być płynne, ale nie powoduje przeskoków animowanego obiektu

Contacts count – ilość wykorzystanych punktów kontaktu. Ilość ta odnosi się punktów znajdujących się pomiędzy pierwszym i ostatnim kontaktem w AnimationData. Pierwszy punkt nazywamy Start, a ostatni Land. Pomiędzy nim kontakty noszą nazwę Contact, i do nich odnosi się ta opcja.

Contact type – jest sposób w jaki będzie liczony koszt punktów kontaktu oraz, sposób w jaki będzie przesuwany obiekt w kontakt state. Aktualnie istnieje 5 typów, oraz możliwość

wyłączenia przesuwania (w przyszłości będzie możliwość implementacji własnego sposobu):

1. Start Contact – obiekt przesuwany jest do pierwszego a następnie pomiędzy kontaktami.
2. Contact – obiekt przesuwany jest tylko pomiędzy kontaktami
3. Contact Land – obiekt jest przesuwany pomiędzy kontaktami, a na koniec do ostatniego punktu.
4. Start Land – obiekt przesuwany jest najpierw do startu, a następnie do ostatniego punktu (Land).
5. Start Contact Land – obiekt przesuwany jest najpierw do startu, później pomiędzy kontaktami, a następnie do ostatniego punktu (Land).

Ilość kontaktów (Contact) ustawiana jest opcją **Contacts count**.

Jak powinna wyglądać tablica punktów kontaktów którą przekazujemy w metodzie **SwitchToContactState** komponentu **MotionMatching**?

Przekazywana lista kontaktów powinna zawsze posiadać punkt startowy, czyli punkt o indeksie 0 i końcowy znajdujący się na ostatnim miejscu listy. Jeżeli nie są one używane, co wynika z ustawień stanu (np. **Contact Type** jest równe **Contact**) to i tak przekazywana ilość kontaktów powinna wynosić ilość kontaktów plus 2 (start i koniec).

Opcje przejść pomiędzy stanami

The screenshot shows a configuration window titled "New State -> New State1". It contains several sections for configuring a transition:

- Transition options:** A dropdown menu set to "New option" with "+" and "-" buttons.
- Common options:** A "Blend time" slider set to 0.3.
- From state options:**
 - A checkbox "Check transition on max lenght" is unchecked.
 - Two "MotionMatchingData" sliders: "Parkour_Climbing_Sprint_1_L_MM" (range 1.2175 to 1.7) and "Parkour_Climbing_Sprint_1_R_MM" (range 1.21 to 1.7).
- To state options:**
 - Two "MotionMatchingData" sliders: "jumpOver_LH_MM" (range 0 to 0.9956) and "JumpOver_RH_MM" (range 0 to 1.0122).
- Option Conditions:**
 - Bool conditions:** A dropdown set to "New bool" and a dropdown set to "IS_TRUE".
 - Int conditions:** A dropdown set to "New int", a dropdown set to "GREATER", and a text input set to "54".
 - Float conditions:** A dropdown set to "New float", a dropdown set to "LESS_EQUAL", and a text input set to "0.4".

Na samej górze mamy możliwość ustawienia liczby opcji jaką ma posiadać stworzone przejście. Każda opcja może posiadać różne ustawienia.

Common options

W tej sekcji umieszczone są ustawienia dla wszystkich możliwych typów przejść. Aktualnie istnieją dwa 4 typy przejść pomiędzy typami stanów:

1. Single Animation → Single Animation
2. Single Animation → Motion Matching
3. Motion Matching → Single Animation
4. Motion Matching → Motion Matching

(Stany typu Contact są tutaj rozpatrywane jako Single Animation).

Aktualnie mam tu możliwość ustawienia jednej opcji a mianowicie czasu blendowania pomiędzy stanami.

Sekcja From State Option

Opcje te dotyczą stanu z którego wychodzimy, pojawiają się tutaj tylko w momencie jeśli stan jest typu SingleAnimation oraz Contact. W sekcji tej możemy wybrać w jakim przedziale czasu każdej animacji mają być sprawdzane warunki ustawiane niżej. Jeżeli opcja **Check transition on max Length** jest zaznaczona to warunki rozpoczną być sprawdzane w momencie gdy czas aktualnie odtwarzanej animacji będzie większy od tego czasu pomniejszonego o wcześniej ustawiony **blend time**.

Sekcja To State option

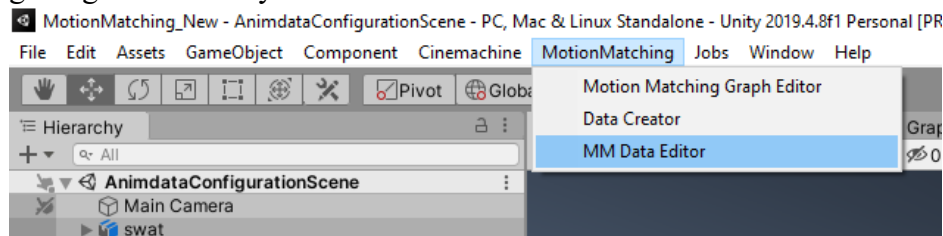
Opcje te dotyczą stanu z którego wychodzimy, pojawiają się tutaj tylko w momencie jeśli stan jest typu SingleAnimation oraz Contact. W sekcji tej możemy wybrać w jakim przedziale czasu każdej animacji będzie wyszukiwana najlepsza klatka do której rozpocznie się blendowanie.

Option Condition

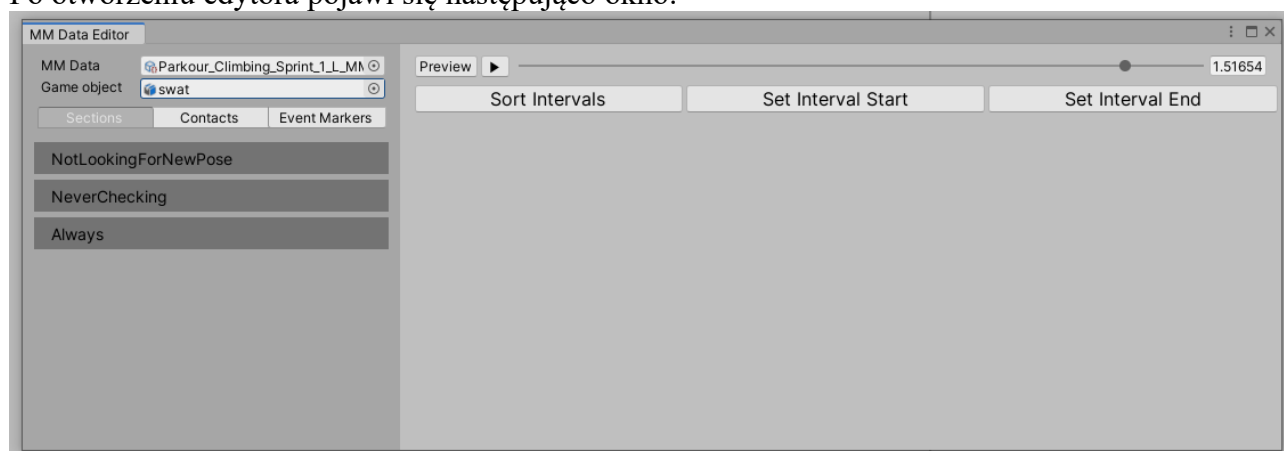
W sekcji tej ustawiamy warunki jakie muszą zostać spełnione aby rozpoczęło się przejście do kolejnego stanu. Wszystkie warunki muszą zostać spełnione.

6. Motion Matching Data Editor

Motion Matching Data Editor jest to edytor przeznaczony do edycji wybranych opcji AnimationData stworzonych przez Motion Matching Data Creator. Możemy w nim edytować sekcje AnimationData, tworzyć Contact oraz Impact points oraz dodawać tzw. Event Markers. Motion Matching Data Editor otwieramy za pomocą dwukliku na asset AnimationData lub z górnego menu Unity:



Po otwarciu edytora pojawi się następujące okno:



Dzieli się ono na dwie części. Po lewej stronie mam ogólne opcje, lub listę elementów aktualnie edytowanych ustawień. Po prawej stronie mam możliwość odtwarzania animacji (na scenie) oraz możliwość ustawiania opcji które wybraliśmy z lewej strony.

Lewa strona edytora:

MM Data – aktualnie edytowana AnimationData.

Game object – obiekt na którym będą odtwarzane animacje oraz na podstawie którego będą wykonywane obliczenia (Contact/Impact points).

Następnie mamy do wyboru co chcemy edytować (**Sections**, **Contact**, **Event Markers**).

Prawa strona edytora:

Przycisk **Preview** – za pomocą jego tworzymy graf animacji za pomocą którego będzie odtwarzana animacja. Dodatkowo animowany obiekt jest ustawiany w pozycji (0,0,0) (do edycji Animation Data zalecam utworzenie oddzielnej sceny). Jeżeli animacja jest aktualnie odtwarzana, zostanie zatrzymana i jej czas zostanie ustawiony na 0.

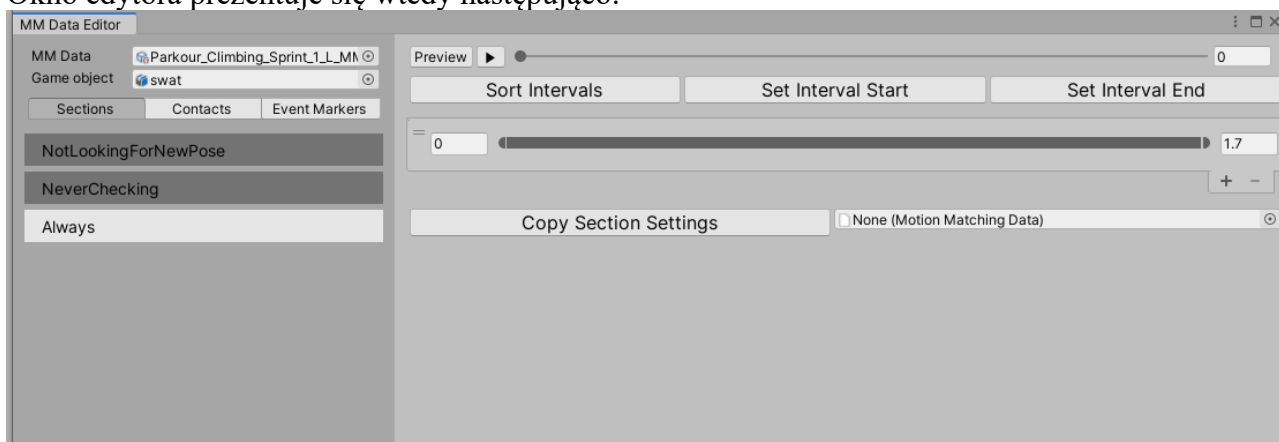
Przycisk **Play** – służy do rozpoczęcia odtwarzania animacji. Po wciśnięciu zmienia się na przycisk pauzowania animacji.

Slider służy do ręcznego przewijania animacji.

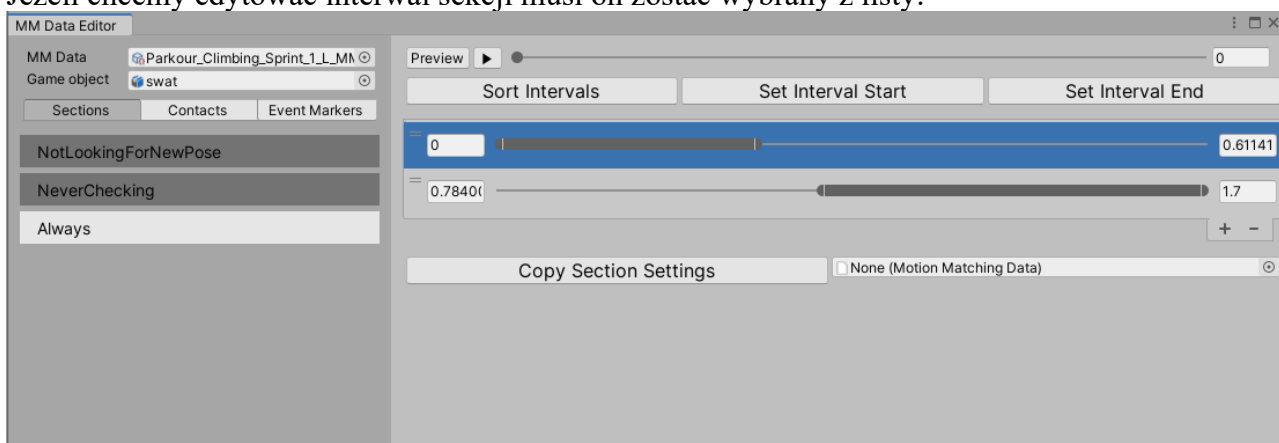
Jeżeli chcemy odtwarzać lub przeliczać kontakty w wybranej AnimationData musimy mieć ustawiony **Game Object**. W przeciwnym wypadku nie będzie możliwości przeliczenia jak równie odtwarzania animacji.

Opis edytowania Sekcji:

Po wybraniu z ToolBara **Sections** po lewej stronie zostaną wyświetlone istniejące sekcje w AnimationData. Po wybraniu sekcji z prawej strony zostaną wyświetlone przedziały danej sekcji. Okno edytora prezentuje się wtedy następująco:



Jeżeli chcemy edytować interwał sekcji musi on zostać wybrany z listy:



Przyciski znajdujące się ponad interwałami służą do:

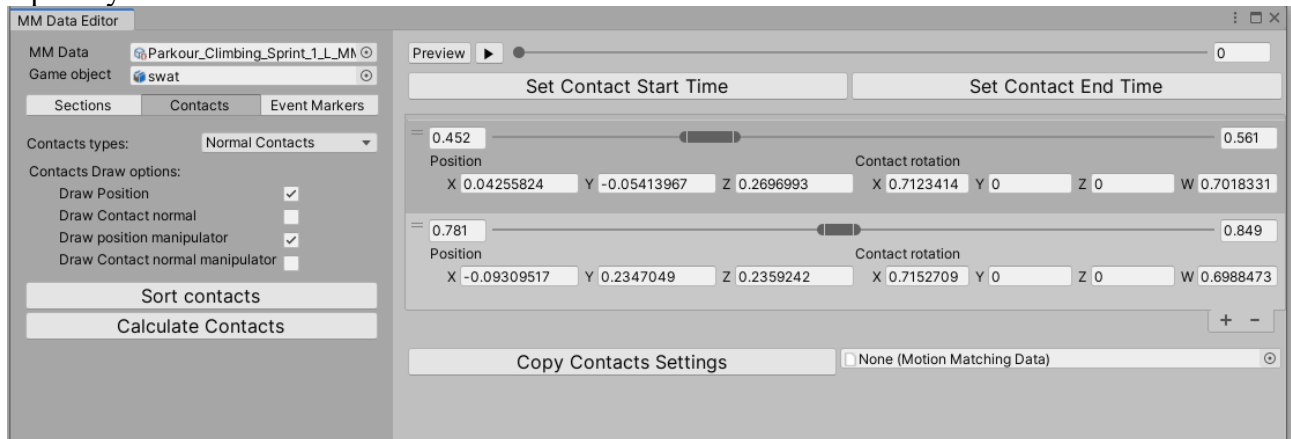
Sort intervals – sortuje interwały sekcji.

Set Interval Start – ustawia start wybranego interwału na aktualny czas animacji identyczny jak czas animacji na sliderze powyżej.

Set Interval End – ustawia koniec wybranego interwału na aktualny czas animacji identyczny jak czas animacji na sliderze powyżej.

Poniżej znajduje się przycisk za pomocą którego możemy skopiować ustawienia sekcji z innej AnimationData, przydaje się to w momencie gdy mamy AnimationData których animacje są „zmirrowane”.

Opis edytowania Contacts:



Po lewej stronie wyświetlają się opcje:

Contact types – możemy tu wybrać czy AnimationData będzie posiadała Contacty czy Impacty.

Draw Position – jeżeli zaznaczony w widoku sceny zostaną wyświetlone pozycje aktualnie przeliczonych Impac/Contact punktów AnimationData.

Draw Contact normal – jeżeli zaznaczony w widoku sceny zostaną wyświetlone kierunki normalnych aktualnie przeliczonych Impac/Contact punktów AnimationData.

Draw position manipulator – jeżeli zaznaczona wyświetlony zostanie w widoku sceny manipulator pozycji aktualnie wybranego po prawej stronie Contact/Impact punktu.

Draw contact normal manipulator – jeżeli zaznaczona wyświetlony zostanie w widoku sceny manipulator rotacji aktualnie wybranego po prawej stronie Contact/Impact punktu.

Przycisk Sort Contacts – sortuje punkty kontaktów ze względu na ich czas.

Przycisk Calculate contacts – oblicza punkty kontaktów w wybranej AnimationData zgodnie z ich ustawieniami.

Po prawej stronie mamy możliwość ustawiania opcji contact/impact punktów:

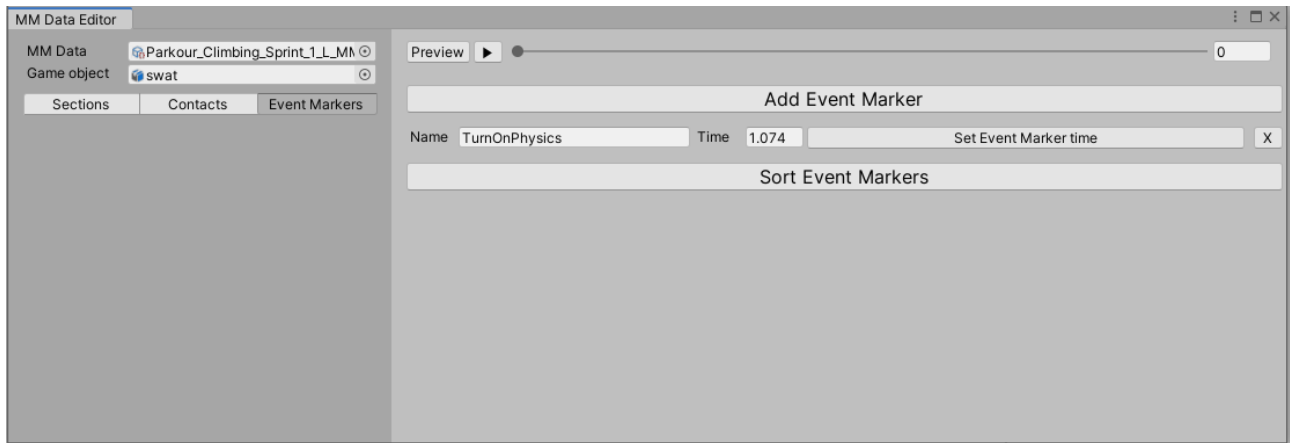
Przycisk **Set Contact Start Time** – służy do ustawiania startowego czasu kontaktu zgodnego z aktualnym czasem animacji.

Przycisk **Set Contact End Time** – służy do ustawiania końcowego czasu kontaktu zgodnego z aktualnym czasem animacji.

Przycisk **Copy Contact Settings** – kopiuje ustawienia kontaktów z wybranej AnimationData.

W liście aktualnych kontaktów możemy zobaczyć aktualne ustawienia poszczególnego kontaktu. Jeżeli któryś z nich został wybrany w widoku sceny zostaną wyświetlone odpowiednie manipulatory (o ile zostały zaznaczone po lewej stronie edytora). Widzimy tutaj początek i koniec czasu kontaktu, pozycję kontaktu (lokalna względem animowanego obiektu) oraz orientację (za pomocą której obliczany jest wektor normalny kontaktu). Ustawienia czasu możemy zmieniać za pomocą przycisków jak również za pomocą slidera. Ustawienia orientacji i pozycji zmieniamy za pomocą manipulatorów w widoku sceny. Jeżeli chcemy edytować konkretny punkt musi on zostać wybrany z listy wszystkich kontaktów.

Ustawienia Event Markers



Po lewej stronie nie ma żadnych dodatkowych opcji dla Event Markers.

Po prawej stronie mamy:

Przycisk **Add Event Marker** służący do dodawania nowego markera.

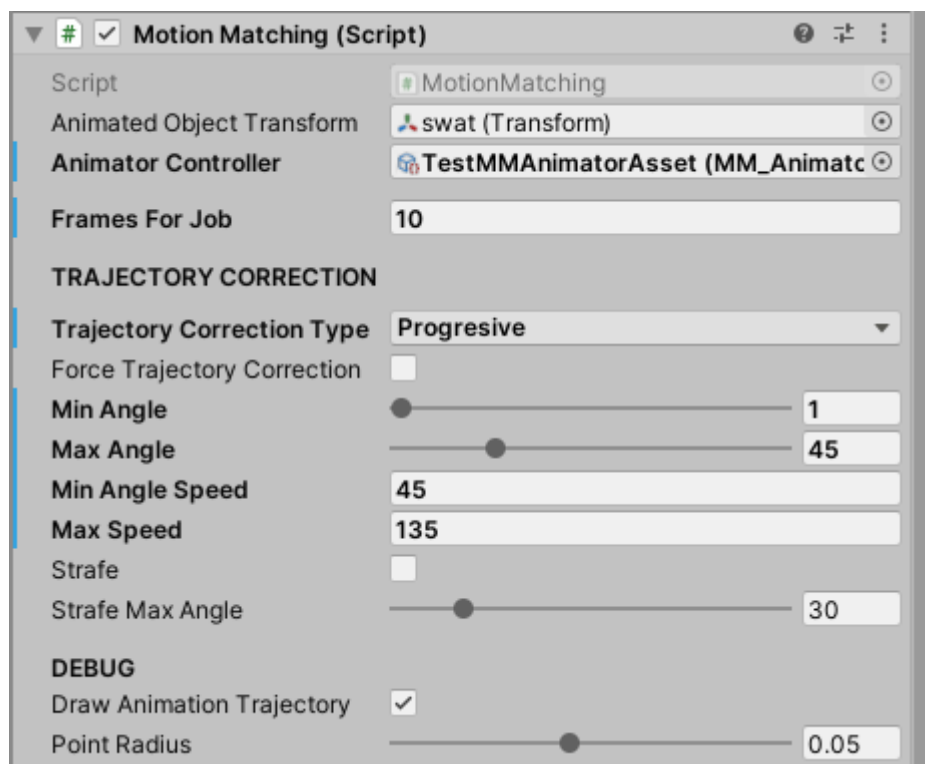
Następnie mamy listę markerów, z nazwą markera, jego czasem, przyciskami do ustawienia czasu zgodnego z aktualnym czasem animacji, oraz przycisk do usuwania markera.

Przycisk Sort Event Marker służy do posortowania markerów na podstawie ich czasu. Jeżeli markery nie zostaną posortowane może to spowodować błędy (zostanie to w przyszłości zmienione).

Event markery używamy za pomocą MotionMatchingStateBehavior, o tym więcej przy omawianiu komponentu Motion Matching.

7. Omówienie komponentu **Motion Matching**

Jest to główny komponent który realizuje logikę Motion Matching Animator Controller.



Animated Object Transform – jest to transform animowanego obiektu jeżeli, nie będzie ustawiony przyjmie transform obiektu do którego należy komponent.

Animator Controller – utworzony przez nas Motion Matching Animator Controller.

Frames for job – za pomocą tej opcji ustawiamy ile klatek animacji ma być liczonych w jednym wątku. Liczba ta nie powinna być zbyt mała, ponieważ obliczenie 100 klatek animacji nie zajmuje dużo czasu i nie chcemy tworzyć do tego 10 jobów, gdzie każdy z nich zajmie się 10 klatkami.

TRAJECTORY CORRECTION – są to opcje które ustawiamy aby system poprawiał trajektorię poruszania się animowanego obiektu zgodnie z trajektorią przekazaną przez gracza.

Trajectory correction type – jest sposób w jaki trajektoria jest poprawiana aktualnie istnieją 4 metody korekcji trajektorii:

1. Constant – animowany obiekt obracany jest ze stałą prędkością aby pierwszy punkt trajektorii aktualnie odtwarzanej animacji pokrywał się z pierwszym punktem trajektorii wymaganej przez gracza.
2. Progressive – działa identycznie jak constant z tą różnicą że prędkość obracania animowanego obiektu jest zależna od kąta pomiędzy kierunkiem do pierwszego punktu trajektorii animacji i trajektorii przekazywanej przez gracza.
3. Match Orientation Constant – animowany obiekt obracany jest ze stałą prędkością aby orientacja pierwszego punktu trajektorii animacji pokrywała się z orientacją pierwszego punktu trajektorii przekazanej przez gracza.
4. Match Orientation Progressive – kierunek obracania również wybierany jest na podstawie orientacji punktów z tym że prędkość obracania zależna jest od kąta pomiędzy tymi orientacjami.

*Pierwszy punkt trajektorii czyli pierwszy punkt którego czas jest większy od 0.

Force trajectory correction – jeżeli zaznaczone korekcja trajektorii będzie następować w stanach w których jest on wyłączona (tylko w stanach typu Single Animation oraz Motion Matching).

Min Angle i Max Angle – przedział w którym musi znajdować się kąt pomiędzy trajektoriami (obliczony zależnie od typu poprawy trajektorii) aby korekcja trajektorii nastąpiła.

Min Angle Speed – jest to prędkość w stopniach na sekundę z jaką ma być obracany animowany obiekt gdy kąt pomiędzy trajektoriami jest równy Min Angle.

Max Angle Speed – jest to prędkość w stopniach na sekundę z jaką ma być obracany animowany obiekt gdy kąt pomiędzy trajektoriami jest równy Max Angle. Jest też używana jako stała prędkość gdy typ poprawy trajektorii jest ustawiony na Constant lub Match Orientation Constant.

Strafe – jeżeli chcemy aby postać „strafe’owała” zaznaczamy tą opcję. Dodatkowo musimy przekazać z kodu do komponentu kierunek strafe.

Strafe Max Angle – jeżeli kąt pomiędzy kierunkiem strafe przekazany do komponentu a kierunkiem forward animowanego obiektu jest większy od tej wartości następuje poprawa trajektorii.

Za pomocą komponentu Motion Matching możemy dodawać do poszczególnych stanów MotionMatchingStateBehavior.

MotionMatchingStateBehavior jest abstrakcyjną klasą dzięki której możemy implementować logikę która będzie wykonywana tylko w stanie do którego zostanie dodany obiekt klasy dziedziczącej po niej. Klasa ta posiada 7 metod abstrakcyjnych które możemy zaimplementować, są to:

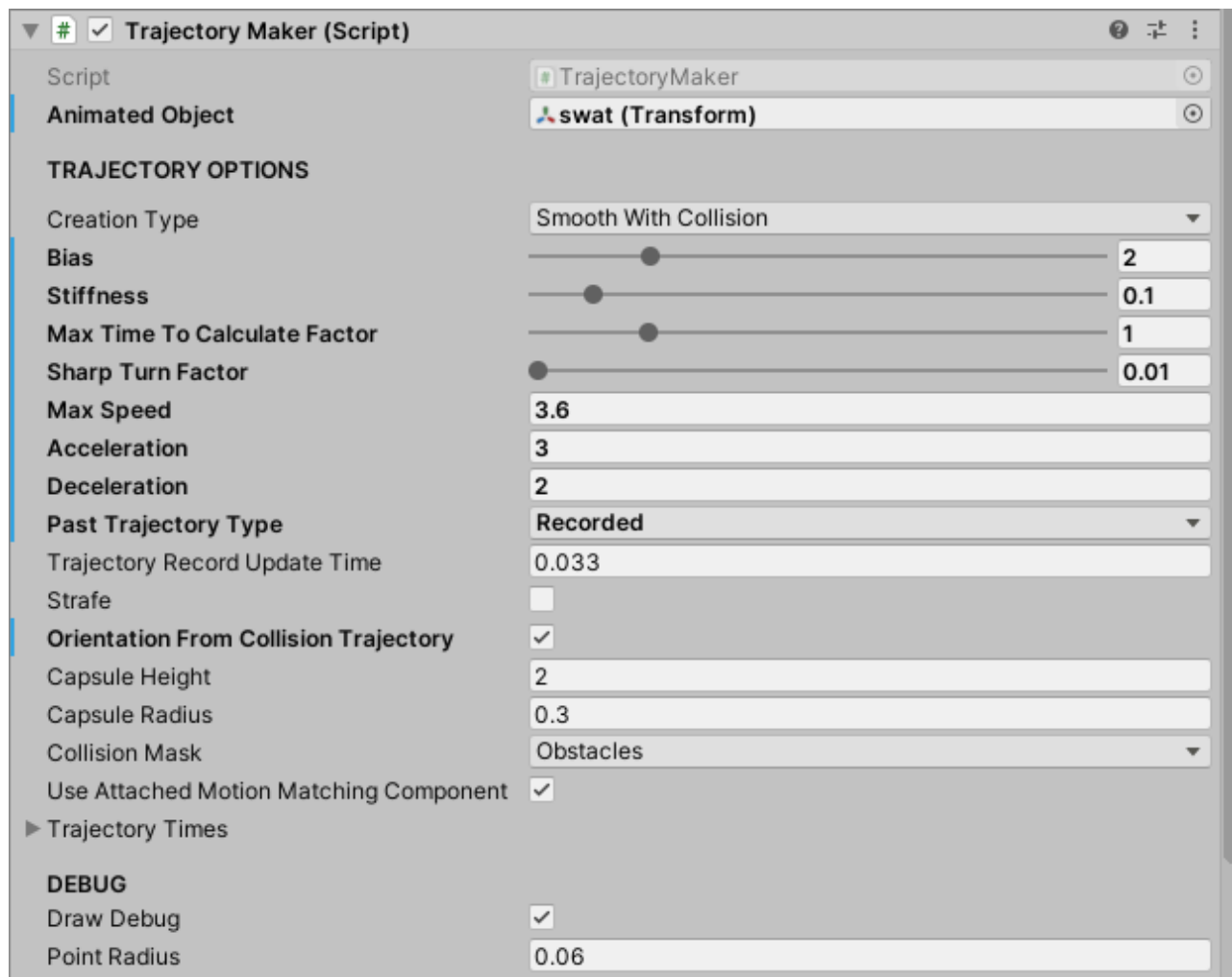
1. Enter() – jest wywoływana po wejściu do stanu.
2. Update() – odpowiedni Update z Mono Behavior.
3. LateUpdate() – odpowiednik Late Update z Mono Behavior.
4. OnStartOutTransition() – metoda ta jest wywoływana w momencie, gdy warunki przejścia do kolejnego stanu zostaną spełnione, jeszcze zanim wyjdziemy ze stanu.
5. Exit() – metoda jest wywoływana po wyjściu ze stanu, czyli w momencie gdy waga stanu w grafie animacji będzie wynosić zero (skończy się blendowanie stanów). Exit wykonuje się po OnStartOutTransition()
6. OnContactPointChange(int fromContactIndex, int toContactIndex) – metoda ta wykonuje się tylko w stanie typu Contact w momencie gdy następuje zmiana indexu aktualnego kontaktu do którego będzie przemieszczana postać.
7. CatchEventMarkers(string eventName) – w tej metodzie możemy implementować zachowania które powinny nastąpić w momencie gdy ustawione przez nas Event Marker (dodajemy je w Motion Matching Data Editor) został wyłapany.

Klasa ta zawiera również 3 pola:

1. logicState – referencja do aktualnego stanu w którym jesteśmy
2. mmAnimator – referencja od komponentu MotionMatching
3. transform – referencja do Transform animowanego obiektu

8. Omówienie komponentu **Trajectory Maker**

Trajectory Maker jest komponentem który tworzy trajektorię na podstawie której wybierana będzie kolejna animacja. Aby używać Trajectory Maker należy ustawić jego opcje odpowiednio do posiadanych animacji oraz przekazać do niego wartość **Input** (Vector3) na podstawie której będzie tworzona trajektoria.



Animated Object – jest to animowany obiekt z którego będzie tworzona trajektoria punktów z ujemnym czasem. Jeżeli nie jest ustawiony, przyjmuje wartość obiektu na którym znajduje się komponent.

Creation type – typ trajektorii jaki będzie tworzony. Aktualnie istnieją 4 typy trajektorii:

1. Constant
2. Constant with Collision
3. Smooth
4. Smooth with Collision

Bias – im wartość jest większa przyszłe punkty trajektorii szybciej będą osiągały docelową pozycję, i trajektoria będzie posiadała ostrzejsze łuki.

Stiffness – jeżeli wynosi 0 nie ma wpływu na trajektorię jeżeli wynosi jeden punkty będą osiągały swoje docelowe pozycje w tym samym czasie i trajektoria nie będzie układała się w łagodne łuki.

MaxTimeToCalculateFactor – jest to czas na podstawie którego będzie liczony współczynnik powodujący tworzenie trajektorii w kształcie łuków, im czas jest mniejszy tym łuki będą ostrzejsze i punkty szybciej będą osiągać swoją pozycję, im większy tym trajektoria będzie tworzyć się wolniej i łuki będą łagodniejsze.

SharpTurnFactor – jeżeli wynosi 0 nie ma wpływu na trajektorię, jeżeli wynosi 1 przy szybkich zmianach kierunku poruszania się punkty trajektorii będą poruszać się szybciej do oczekiwanej pozycji.

MaxSpeed – maksymalna prędkość dla jakiej ma być tworzona trajektoria.

Acceleration – przyspieszenie z jaką ma być tworzona trajektoria.

Deceleration – przyspieszenie z jaką ma spowalniać trajektoria gdy wektor Input jest zerowy.

PastTrajectoryType – typ przyszłej trajektorii dla punktów z ujemnym czasem. Aktualnie istnieją 2 typy przeszłej trajektorii:

1. Recorded – trajektoria jest nagrywana z parametru Animated Object.
2. Copy From Current Data – przeszła trajektoria jest kopiowana z aktualnie odtwarzanej animacji z komponentu Motion Matching.

Trajectory Record Update Time – jest to czas co jaki ma być nagrywany punkt trajektorii.

Strafe – jeżeli opcja jest zaznaczona orientacja punktów trajektorii będzie równa kierunkowi przekazanemu do komponentu (**StrafeDirection**) .

Orientation From Collision Trajectory – jeżeli opcja ta jest zaznaczona orientacja punktów trajektorii będzie równać się kierunkom do poszczególnych punktów po kolizji. Działa tylko z typem trajektorii **Smooth With Collision** i **Constant With Collision**.

Capsule Height – wysokość kapsuły dla której będzie obliczana trajektoria z kolizjami.

Capsule Radius – promień kapsuły dla której będzie obliczana trajektoria z kolizjami.

Collison Mask – maska dla której mają być sprawdzane kolizje.

UseAttachedMotionMatchingComponent – jeżeli opcja ta jest zaznaczona, czasy punktów trajektorii zostaną pobrane z komponentu MotionMatching znajdującego się na tym samym obiekcie co Trajectory Maker. Jeżeli komponent Motion Matching nie jest umieszczony na tym samym obiekcie możemy ręcznie wpisać czasy trajektorii do listy poniżej lub zsetować komponent MotionMatching do Trajectory Makera w metodzie Awake oddzielnego skryptu MonoBehaviour.