Scene Motion Capture Manual

Content

What is Scene Motion Capture?

Scene Motion Capture is additional tool for Animation Baker that allows you to capture any changes of transform of any objects in your scene and save it as Unity animation file. You can capture procedural or physical motion of any object including characters with any Mecanim transitions and IK influence, ragdolls etc.

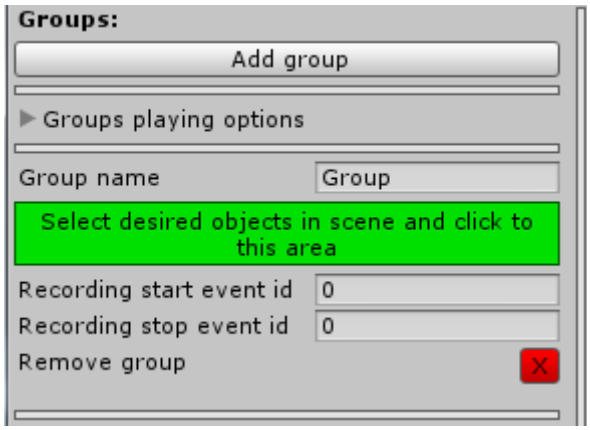
Usage

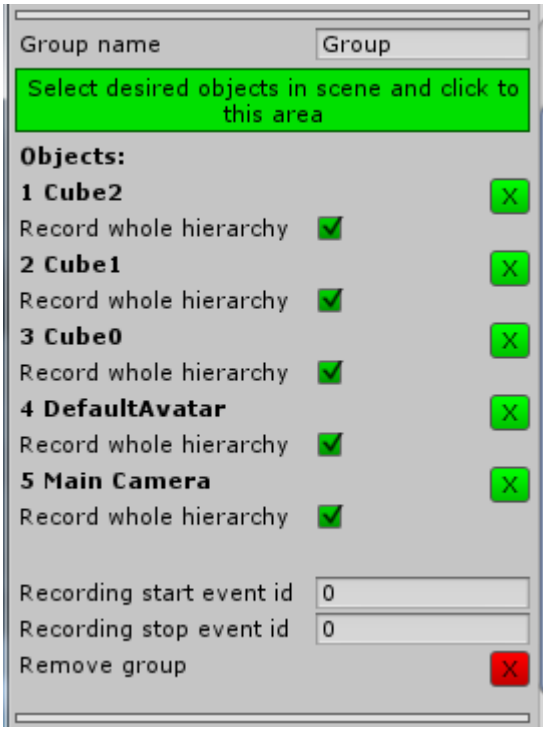
In order to start capture you have to assign desired objects in SMC window (execute "Window/AnimationBaker/Scene Motion Capture") and set some recording events for this such as pressing key to start recording. Scene Motion Capture has groups into which you can add your objects for capture, it also has events that causes recording or stops it. After this you can switch to play mode, press desired key and motion capture will started.The only sense of groups in that the each group may be controlled by individual events. For example you want that the capture for some characters in your scene controlled by F1 key and capture for some physical objects – by key F2, in this case you just have to create two groups in SMC window and assign your characters to one of them and physics objects to another one, and after that set desired key events for each of them. You can also add any number of events as well as groups. Each created event can be assigned to any group as "Start recording" event or "Stop recording" or both. When you finish capture for some groups SMC creates animation clips for each recorded object and also creates prefabs for each group, this prefabs are references to all recorded objects in scene but with already assigned resultant animation clips so you can easily use this prefabs in order to play recorded motions without unnecessarily assigning clips to objects manualy.

Object groups

Object groups mainly used for separation of motion capture of objects that are in different groups. Also resultant animation clips of objects from different groups saves into different folders (in its groups folders) as well as its prefabs. When you opened SMC window it's by default creates new object group into which you can add objects, but you always can add new group by pressing button "Add group". In order to add objects to the group just select them in hierarchy view and click on the green area labeled "Select desired objects in scene and click to this area", after this you will see that all selected objects appeared in current group. You can remove any object from the group by pressing cross button to the right of its name. Also you can see that below each object is a flag "Record whole hierarchy", if this flag is set then it's means that the hierarchy of this object will also be captured in other case only motion of root object will be recorded.

You can assign any available recording events for group through fields "Recording start event id" and "Recording stop event id". First field used to start recording for this group, the second respectively – for stopping recording. Enter event ids you want to this fields, event id displayed for each event inside its area.

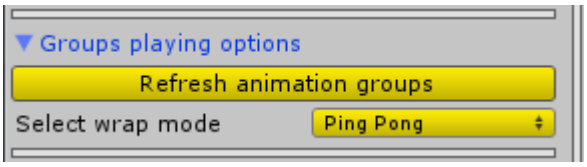


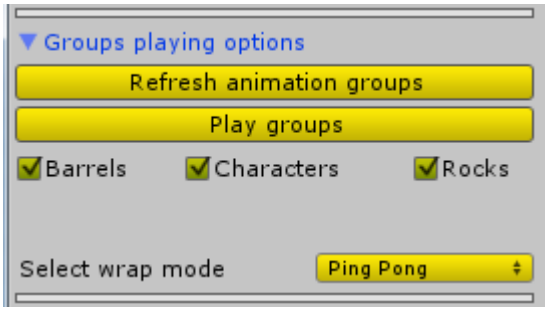


You can remove desired group by pressing red cross button to the right of label "Remove group".

Playing groups

Section "Groups" has foldout section "Groups playing options", this little menu helps you to play animation for recorded groups. In order to play some or all groups press button "Refresh animation groups" – this makes SMC scan "GroupsPrefabs" folder for created group prefabs. If any prefab founded "Play groups" button will apear as well as groups check-boxes with help of which you can determine what groups you want to play. Also you can set wrap mode for animation clips of objects of all groups by selecting wrap mode in popup menu "Select wrap mode". By pressing "Play groups" button you will start playing animations for all selected groups.





Events

Events uses for starting or stopping recording for desired groups. Event can be represented by sub events or be just an empty event that can be only called from script at runtime. By default when you opened SMC window it's creates event that activating by pressing key F12, i.e. this event has key sub event. If current event does not have key event, you can add it by pressing button "Add key event", you also can change key by pressing "Read key" button. You can also add time sub event to the current event by pressing button "Add time event" inside event area. Time sub event activates current event when passed in game time (since Time.unscaledTime) become larger than this value. You can combine both sub events by adding them to current event, in this case you will see square button, it can take two states :"OR" or "AND". This is nothing more than a logical ligament of this sub events. For example, if you have an event for F12 key and time sub event with value 10 and in same time logical ligament is "AND" then this means that current event will be activated when key F12 will be pressed during play mode but only after 10 seconds passed after play mode started. In the same way, if you switch button to "OR" state then this event will be activated by pressing key F12 or after 10 seconds passed after play mode started(even when key was not pressed). You can remove any sub event by pressing cross button to the rigth of its field.

<u>Note:</u> Should remember that if you remove both sub events from current event then this event could be called only from some script at runtime(what shows label "Can be activated only from script using ID" in the event area).

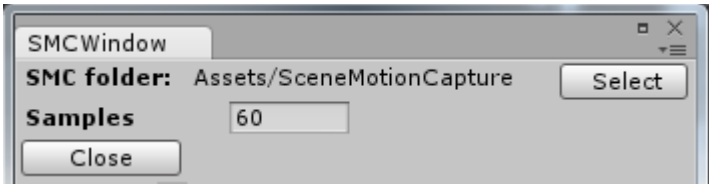
For example, if you want to the motion capture for some group has been recorded in between some time interval, then you should do as follows:
- add all your objects for capture into some group;
- create two events with only time sub events;
- set start recording time for one of events and end recording time to another;
- set start recording event id to "Recording start event id" field of the group and end recording event id set to the field "Recording stop event id".

Now if you switch to play mode SMC will make motion capture for assigned objects throughout all length of determined time interval.

You can remove event by pressing red cross button to the rigth of label "Remove event".

Settings

In the menu "Settings" you can set desired frame rate for sampling animation clips and specify path to SceneMotionCapture root folder in case if you put it away from default location.



Motion Capture data

After recording for some groups finished for all its objects creates animation clips that are saves to "Animations\'Group name'" folder inside SMC root folder, also creates prefabs for each group that contains all objects from this groups with assigned resultant animation clips, this prefabs saves to "GroupPrefabs" folder of root directory of SMC. Of course you can use obtained animation clips as you want, manually assign them to desired object and do any things, but created prefabs can help you to play whole group of objects as it was recorded.

Runtime Scene MotionCapture handling

You can add object groups and events also in play mode, for example, if you need to create new group with some events from objects that was created just now and needs to be added to the group for motion capture. You can handle this things by methods of SceneMoCap class.

Methods

SceneMoCap.AddEvent

public static int AddEvent()
public static int AddEvent(KeyCode key)
public static int AddEvent(float time)
public static int AddEvent(KeyCode key,float time, bool OR)

Properties:

key – key code of desired key, for example KeyCode.F12
time – time limit exceeding which event will be activated
OR – sub events logical ligament

Description:

Creates new event with desired properties for Scene Motion Capture and returns ID if this event. You can use this ID to activate/deactivate event or to change properties of event.

SceneMoCap.AddGroup

public static int AddGroup()
public static int AddGroup(string name)
public static int AddGroup(MoCapObject[] objects)
public static int AddGroup(string name,MoCapObject[] objects,int startEventId,int endEventId)
public static int AddGroup(string name,Transform[] objects,int startEventId,int endEventId)

Properties:
name – name of the group, must be unique
objects – motion capture objects list, may take array of MoCapObject class or array of Transform class
startEventId – id of event that is responsible for start of motion capture
endEventId – id of event that is responsible for stopping of motion capture

Description:

Creates new object group with desired objects and properties, returns ID of created group. You can use this ID to change properties of desired group or to add new objects to it.

SceneMoCap.SetGroupProperties

public static void SetGroupProperties(int groupId,string name,int startEventId,int endEventId)
public static void SetGroupProperties(int groupId,string name,Transform[] objects,int startEventId, int endEventId)
public static void SetGroupProperties(int groupId,Transform[] objects,int startEventId, int endEventId)
public static void SetGroupProperties(int groupId,int startEventId,int endEventId)
public static void SetGroupProperties(int groupId,Transform[] objects, bool addObjects)
public static void SetGroupProperties(int groupId,string name,Transform[] objects,int startEventId,int endEventId, bool addObjects)

Properties:
groupId - ID of the group properties of which you want to change
name – name of the group, must be unique

objects – list of transforms of game objects that you want to use for capture
startEventId – id of event that is responsible for start of motion capture
endEventId – id of event that is responsible for stopping of motion capture
addObjects – if true then passed objects will be added to desired group otherwise passed objects will replace current objects in group

Description:
Sets desired objects and properties for existing group with ID groupId.

SceneMoCap.SetEventProperties
public static void SetEventProperties(int eventId,KeyCode key)
public static void SetEventProperties(int eventId,float time)
public static void SetEventProperties(int eventId,KeyCode key,float time)
public static void SetEventProperties(int eventId,KeyCode key,float time,bool OR)

Properties:
eventId – ID of event properties of which you want to change
key – key code of desired key, for example KeyCode.F12
time – time limit exceeding which event will be activated
OR – sub events logical ligament

SceneMoCap.ActivateEvent
public static void ActivateEvent(int eventId)

Properties:
eventId – ID of event that you want to activate

SceneMoCap.DeactivateEvent
public static void DeactivateEvent(int eventId)

Properties:
eventId – ID of event that you want to deactivate

Classes

SceneMoCap.MoCapObject
Variables:
Transform objTransform – transform of the GameObject;
bool considerHierarchy – flag that responsible for recording hierarchy of this object;

Description:
This class used for storing game objects from scene inside object group.

MoCapObject usage example:

```
Transform[] objects;

void AddThisObjectsToGroup(){
if(objects = = null)
return;
if(objects.Length<1)
return;
MoCapObject[] groupObjects = new MoCapObject[objects.Length];
        for(int i=0;i<groupsObjects.Length;i++){
                groupsObjects[i] = new MoCapObject(objects[i],true);
        }
SceneMoCap.AddGroup(groupObjects);
}
```

SMCHelper

SMCHelper may be useful if you want to play your groups not just from SMC window but also from compiled application or just from play mode. In order to do this just drag into scene SMCHelper prefab from "SMCHelper" folder of SMC root directory or just assign SMCHelper.cs script to any game object in scene. It's needed only for assign existing groups prefabs, so drag all desired groups prefabs to the "Groups" array of SMCHelper.cs – that's it, now you can play your groups in playmode by call method SMCPlayer.PlayAnimationGroups() from any script.

SMCPlayer

SMCPlayer needs only to play your object groups just by call method SMCPlayer.PlayAnimationGroups in playing mode or in compiled application.

SMCPlayer.PlayAnimationGroups

```
public static void PlayAnimationGroups(bool rewrite)
public static void PlayAnimationGroups(bool rewrite,bool[] playGroups,WrapMode wrapMode)
```

Properties:

rewrite – if true then previous playing groups will be removed from scene
playGroups – array of flags of type bool that determines what groups should be played(if playGroups[0] equals true then it's means that group with ID 0 can be played etc.)
wrapMode – wrap mode for animations of all objects in played groups