D6.1 Animation pipeline and demo of new runtime rig

Grant Agreement nr	780470
Project acronym	SAUCE
Project start date (duration)	January 1st 2018 (36 months)
Document due:	29th June 2018
Actual delivery date	30th June 2018
Leader	IKinema
Reply to	a.pechev@ikinema.com
Document status	submission version

Project funded by H2020 from the European Commission

Project ref. no.	780470
Project acronym	SAUCE
Project full title	SAUCE
Document name	D6.1 Animation Pipeline And Demo Of New Runtime Rig
Security (distribution level)	PU
Contractual date of delivery	30th June 2018
Actual date of delivery	29th June 2018
Deliverable name	D6.1 Animation Pipeline And Demo Of New Runtime Rig
Type	DEM
Status & version	submission version
Number of pages	8
WP / Task responsible	IKinema
Other contributors	
Author(s)	IKinema
EC Project Officer	Ms. Cristina Maier, Cristina.MAIER@ec.europa.eu
Abstract	This document provides details centred around the deliverable D6.1 titled "Animation pipeline and demo fo the new runtime rig" within work package (WP) 6 of the SAUCE project. The document begins with a brief introduction and explanation of a typical IKinema workflow used for procedurally generating animation using Inverse Kinematics providing background context for the reader in section 2. This section will provide and overview of a typical pipeline and detail the important components and user input required while explaining each step. Sections 3 provides an introduction to the deliverable 6.1 (the subject of this document) within the context of WP6, in particular WP6 task 2 (WP6T2). It describes the work set out in the tasks included in WP6 and how the deliverable 6.1 fits into the bigger picture. Section 4 discusses the pipeline and the planned demonstration in more detail. It includes the user interface for the pipeline with key elements highlighted and described Section 5 compares the results of the pipeline development with the criteria set out in deliverable description within SAUCE project document.
Keywords	Full Body Inverse Kinematics, Procedural Animation
Sent to peer reviewer	Yes
Peer review completed	Yes
Circulated to partners	
Read by partners	
Mgt. Board approval	

Table of Contents

1 EXECUTIVE SUMMARY

This document provides details centred around the deliverable D6.1 titled "Animation pipeline and demo fo the new runtime rig" within work package (WP) 6 of the SAUCE project. The document begins with a brief introduction and explanation of a typical IKinema workflow used for procedurally generating animation using Inverse Kinematics providing background context for the reader in section 2. This section will provide and overview of a typical pipeline and detail the important components and user input required while explaining each step.

Sections 3 provides an introduction to the deliverable 6.1 (the subject of this document) within the context of WP6, in particular WP6 task 2 (WP6T2). It describes the work set out in the tasks included in WP6 and how the deliverable 6.1 fits into the bigger picture.

Section 4 discusses the pipeline and the planned demonstration in more detail. It includes the user interface for the pipeline with key elements highlighted and described

Section 5 compares the results of the pipeline development with the criteria set out in deliverable description within SAUCE project document.

2 BACKGROUND

IKinema offers a full body Inverse Kinematic (hereby referred to as IK) solver allowing for functionality such as procedurally animating characters at runtime, live retargeting from one character to another and offline animation generation.

The general IKinema workflow for procedural animation begins with creating an 'IKinema Rig' (or runtime rig) for a given character skeleton asset. The IKinema rig is made up of three main components:

- Bones to be posed by the IKinema Solver
- IKinema specific parameters
- IK Tasks (key bones we wish to drive by assigning targets)

The bones included in the IKinema rig are a subset of the bones included in the given character skeleton asset which the user wishes to procedurally animate.

The IKinema specific parameters set in the IKinema rig are characterisation information that are to be consumed by the solver. Tuning these parameters allows control in the overall look and feel of the generated poses.

For the set of bones included in the IKinema rig, IK Tasks are assigned to key bones within the hierarchy. Typical IKinema rigs are made up of multiple IK Tasks (11 for an average biped). Each of these IK Tasks are assigned an IK Demand that can either be automatically generated or user defined depending on the implementation. These IK Demands can take the form of a target rotation and position in what is referred to as constraint or world space. The IKinema solver acts to pose all of the bones included in the IKinema Rig in a way that allows the IK Tasks to reach their assigned IK Demands in space, with the 'look' and 'feel' of the generated pose governed by the IKinema Rig parameters.

The move from the forward kinematic (hereby referred to as FK), local space animation data of traditionally created animation assets to driving IK tasks in constraint space offers one very important distinction that is of key importance in the developments planned in tasks 3 and 4 of work package 6. It becomes much easier to modify generated IK Demands to account for things such as a 'world-awareness' where targets are often easily available in world space. Rather than driving the orientation of each bone of the skeleton (as is done in FK animation), the user only has to provide world space targets for each of the IK Tasks (11 in a typical biped) and the IKinema solver will generate a pose accordingly.

Given the setup of the runtime rig, IKinema's full body solver takes the following as input:

- Some existing joint-based animation asset
- IKinema Rig
- IK Demands

Given the above as input, IKinema's full body solver acts to generate a pose (made up of all the bones included in the IKinema Rig) which is influenced by the input animation but modified allowing the user defined IK Tasks to reach their user defined targets in space. The settings and parameters defined in the IKinema rig allow for subtle modifications to the fidelity of the resultant animation. This is a typical IKinema pipeline allows for any skeletal hierarchy to be procedurally animated. The full body nature of the solve is one of the key features offered by the technology in its current form.

3 INTRODUCTION

The deliverable (and subject of this report) D6.1 - "Animation Pipeline and Demo Of New Runtime Rig" exists within the context of Work Package 6 Task 2 (Hereby referred to as WP6T2) of the SAUCE project.

WP6T2 aims to develop novel algorithms for the animation pipeline, operating in constraint space. Joint based forward kinematic (hereby referred to as FK) animation will be mapped to constraint space where IK demands will be generated based on the input animation and further modified based on in-world state. The result is a procedurally modified animation adapting the input animation of the character to its environment. The move to constraint space (also referred to as world space) is significant as it is much more efficient to generate and modify IK task targets (IK demands) given in-world demands.

The input to the pipeline is animation metadata capturing various animation or subsets of animation assets. The input itself can be any traditional forward kinematic skeletal animation asset associated with the character the user wishes to animate.

Automatic transitional animations between any two joint based FK animation asset can be produced automatically by analysing the start and finish animation and generating IK demands to blend from one to the other smoothly. Passing these generated IK demands to the IKinema solver will allow for the production of transitional animations for switching from "any" animation to "any" animation. The result will be an animation engine that can drive any (historic) animation library with smooth transitions between animation assets that can be further adapted to the in-world state.

Deliverable D6.1 titled "Animation pipeline and demo of new runtime rig" is the initial development step in the overarching goals planned in WP6T2 above and extending to tasks 3 and 4. The deliverable itself will be a technical demo showcasing the real time Inverse Kinematic rig generation given an input of animation assets and associated mesh and skeleton. This will result in a virtual character whose animation can be procedurally modified during runtime based on the identification of key end-effectors and how the user wishes to drive them.

Current inverse kinematic methods for procedurally generating animation within the industry rely on probing world space using methods such as ray tracing to generate IK task targets for common behavior such as 'foot placement'. In this scenario, IK demands are generated based on the terrain local to the character for relevant IK Tasks (4 IK tasks, usually ball and toe of foot). IKinema's full body solver acts to procedurally modify the characters animation to plant the characters feet at the terrain. This is a simple example but generally effective as a base form of 'world-awareness'.

The work planned in WP6, specifically tasks 3 and 4 are mainly concerned with developing and extending current methodology used to generate and modify IK Demands. For instance, Deliverable 6.1, the subject of this document, represents an early stage which converts existing joint-based animation data in local space to constraint space automatically. The data converted to constraint space is used to drive an IKinema rig effectively reproducing the motion of the input animation using inverse kinematics. This is the basis for all future development which will explore IK Demand generation.

3.1 Main objectives and goals

Deliverable D6.1 is a technical demonstration of specific key features of the requirements set out in WP6T2. These key features are incorporated into a single tool and is the subject of the technical demonstration. The specific features demonstrated to satisfy the conditions of the deliverable D6.1 are, automatic runtime rig generation and automatic mapping of existing joint based animation to constraint space to drive aforementioned runtime rig procedurally.

The eventual goal of the pipeline as indicated in WP6T2, is an advancement in the state-of-the-art and to offer a non-trivial speed up in the time necessary to produce procedurally generated animation when compared to traditional methods. The deliverable D6.1 represents a significant step towards achieving this overarching goal.

The pipeline functionality is composed of two main blocks.

-Automatic runtime rig generation for a given character asset. Virtual markers are automatically generated and assigned to the character per bone.

-Generating data in constraint space to drive the runtime rig using Inverse Kinematics from user defined joint based animation assets.

The generated data is passed into the solver as IK demands along with the corresponding IK tasks assigned on the runtime rig. The solver then acts to pose the runtime rig such that the IK demands are met. The result is a procedurally generated animation clip which follows the motion of the user defined ordered animation set.

Deliverable D6.1 offers two particular advances in the current state-of-the art of the typically implemented procedural animation generation. The first is automatic runtime rig generation. This offers significant speed up in what would be the typical implementation of such a setup. The second is automatically mapping joint based FK animation to constraint space which is used to drive the generated rig. This serves as the basis for future development in WP6 where methodology used to generate smart IK demands will be investigated.

The pipeline demonstration is a guided tour through a typical use case resulting in a newly produced animation asset from a user defined set of assets. In practice this will take the form of a real-time walkthrough of the pipeline which allows a user to select an arbitrary number of animation assets, order them as desired and through the automatically generated Inverse Kinematic Runtime rig generate new animation assets.

4 MAIN CONTENTS OF THE DELIVERABLE

The tool used to facilitate the pipeline discussed in sections 3 and 3.1 has been developed and integrated into Autodesk Maya 2017 as it is an industry standard tool. As such, this will be the environment in which the demonstration is carried out. The integration into commonly used tooling is intended to provide an easy-to-use interface for developers and end-users. The software contains a user interface and a functional engine.

As described in Section 3.1 Main objectives and goals the pipeline demonstration is made up of two main components. The first is automatic generation of a runtime rig and the second is the methodology used to transform existing animation data into constraint space used to drive the rig. This system will provide the backbone for future planned development where transitional and semantically modified animation assets can be generated procedurally.

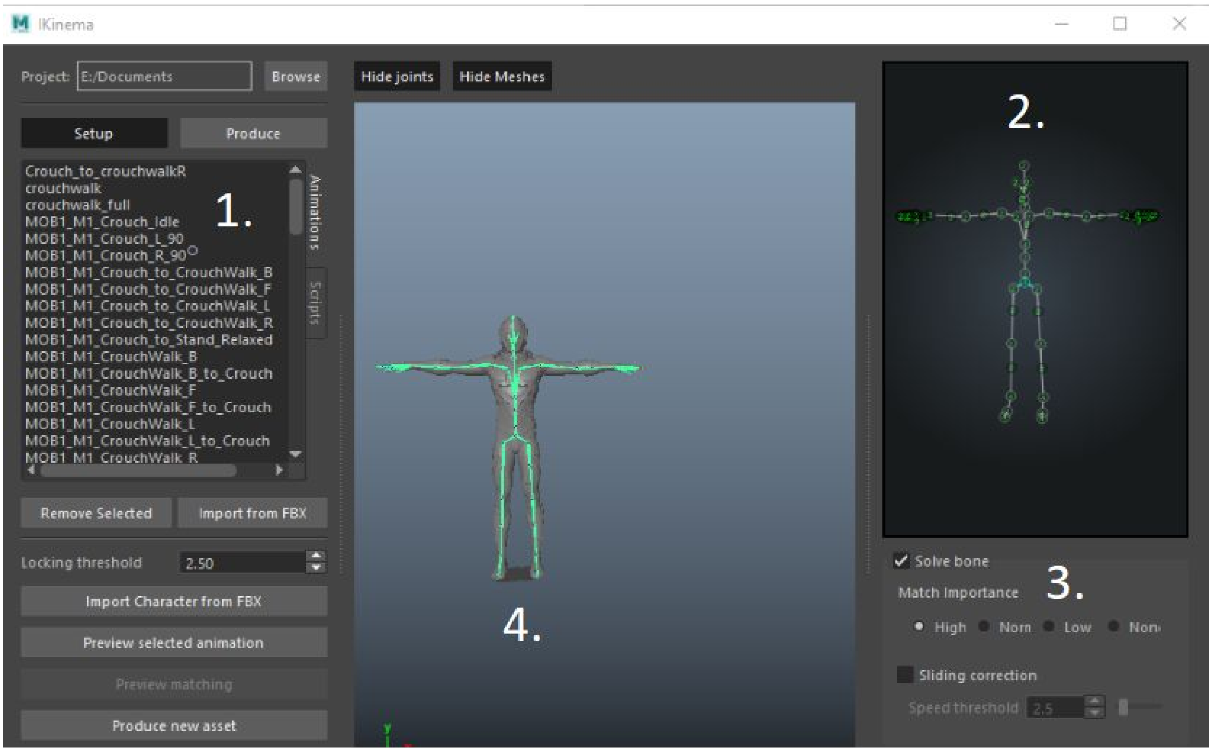

Image One - The user interface of the developed pipeline

Image one shows the user interface from which a user will operate the developed tool. Key functionality points are highlighted and described below.

1) This is the joint-based forward kinematic animation set the user imports into the tool. The user can select an ordered subset of animations from these imported assets. These will be mapped to constraint space.
2) This shows the automatically generated runtime rig
3) The user is able to tune the automatically generated runtime rig if needed. The parameters exposed to the user represent a streamlined version of the typical rig settings used allowing for ease of use and the 'artist's eye' to have an influence on the final result.
4) This is the preview viewport in which the user can preview the resulting procedurally generated animation

For the purpose of demonstration and delivery, the user interface and the technical functionality of the developed tool is highlighted through a step-by-step walkthrough of the pipeline. This will demonstrate a reasonable approximation of the user experience and power of the developed tool. The demonstration assumes the existence of an existing character asset to procedurally animate and an associated historic animation bank to convert to constraint space.

From the user interface, a new runtime rig will be automatically generated for the given character with virtual markers being automatically generated for that character prompted by the user. The runtime rig represents one point of entry allowing the user control over the final result through a

tuning process. The runtime rig is available to tune by the user which has an effect on the overall 'look' and 'feel' of the generated animation. In order to streamline the process, reasonable defaults are set for the automatically generated rig and as such requires minimal user input to achieve good results.

The next step of the pipeline will be selecting relevant animation assets. The selected animation assets are ordered by the user. The ordering is reflected in the resulting composite animation produced in the pipeline. Once the set of desired animations are chosen, a processing stage takes place that maps the forward kinematic animation data (of the user-selected animation assets) to constraint space where virtual markers are generated.

5 CONCLUSION

The pipeline developed succeeds in automatically generating both a runtime rig and data (from existing animation assets) to drive the rig using IKinema's full body solver. These two features cover the main goals set out for the deliverable demonstration D6.1 at this stage of WP6 namely, automatic runtime rig generation and mapping existing FK animation to constraint space.

The demonstration and eventual distribution of the pipeline as a package are intended to engage with industry professionals and obtain feedback on functionality and potential use-cases. With future development in mind, this pipeline will serve as the base to build upon. Flexibility in the design of the pipeline were of key importance. This is because within the context of SAUCE, this pipeline is to be extended in the pursuit of methodology to produce smart, reusable assets. Future research and development planned by IKinema within SAUCE revolves around procedural methodology to generate data to drive the runtime rig.