Animation Production coursework Report

Xingyang Zhou xz15531

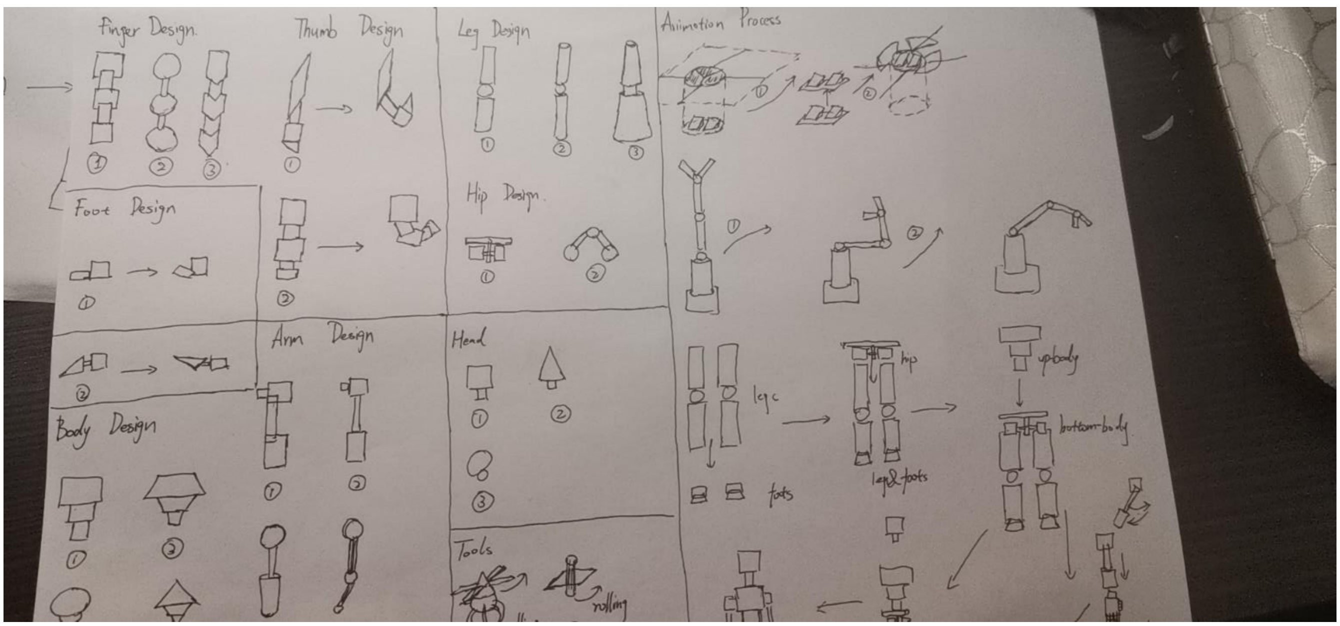
1. Introduction

This report is a demonstration about COMSM0013 animation production coursework one. It illustrate how I generate the 20 seconds humanoid robot build up animation. The story is in an automatic factory, a ploy rigid robot is assembling with the help of some tiny automatic machines. It was made by using basic animation skills and tools. With the help of some robotic dynamic materials, it looks kind of realistic.

2. Pre-production and planning

2.1. Components design

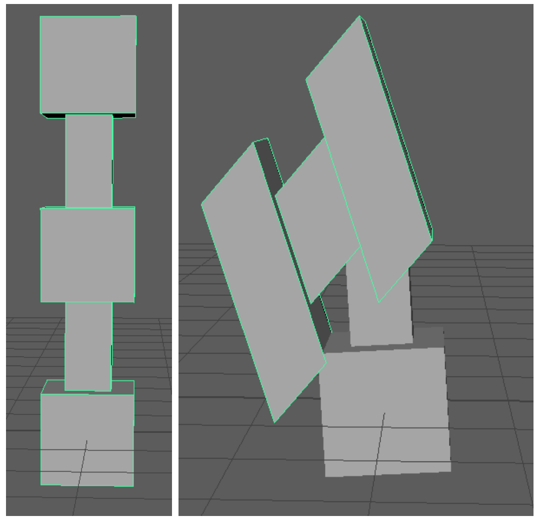
The initial sprite comes from the Minecraft game of Microsoft. It gives me the idea to use basic polygons to assemble my robot. In order to make this robot more human like, different joint components were created to make sure while it moving, it could make poses as accurate as possible. Basic on this idea I created serval simple components on my sketch, they look as follows:



Graph1. components sketch

A number of optional plans were created to prevent bad animation effect while processing. After try all different combinations, I decided to make a pure cubic robot apart from knee and elbow joints are spheres, because while cubic knee joints only have vertical and horizontal edge, which only make it looks properly while rotate horizontal or vertical. When rotate both horizontal and vertical in the same time, it looks joint stuck in each other. Another reason is, while two connect cubic joints rotate in same direction(horizontal or vertical) with a large rotation degree, there will be interspace between these two joints which makes them looks unconnected.

However, the other joints could use cubic because based on the human biological body structure, these joins either can't rotate in same direction nor they could only rotate in small degree.

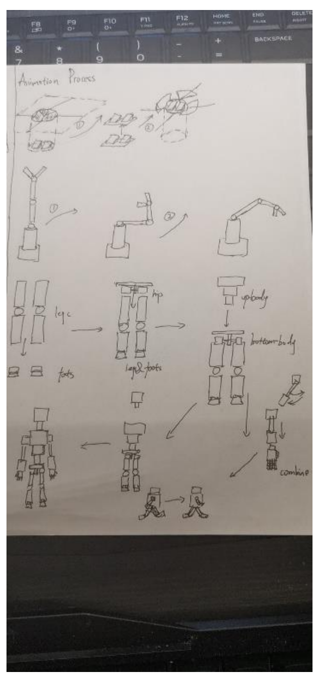


Graph 2 and 3. The cubic rotation issue

However, there are still issues of these cubic joints without using Maya skeleton, I have to change the translate of each components after each rotation. After read some robotic and human skeleton materials, I found the reason of my joints separate while rotating, it is because the distance from mass center to each edges are different. To fix that, it seems we have to blend face or restrict the rotation degree to a very narrow range. However, there is also another method to fix this issue, it will be discussed in the section 3.1.3.

2.2. Scene Design

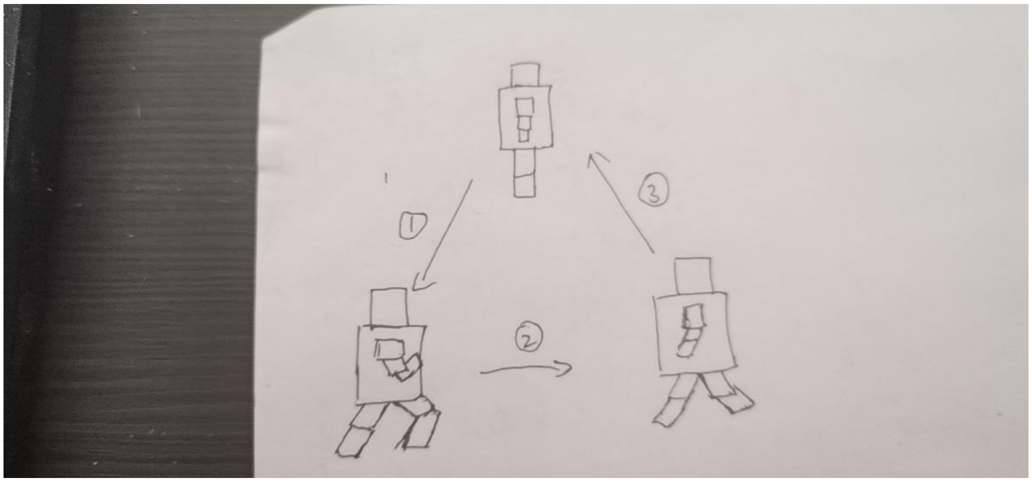
I drew a simple sketch of the animation scene, it looks like below:



Graph 4. Animation scene

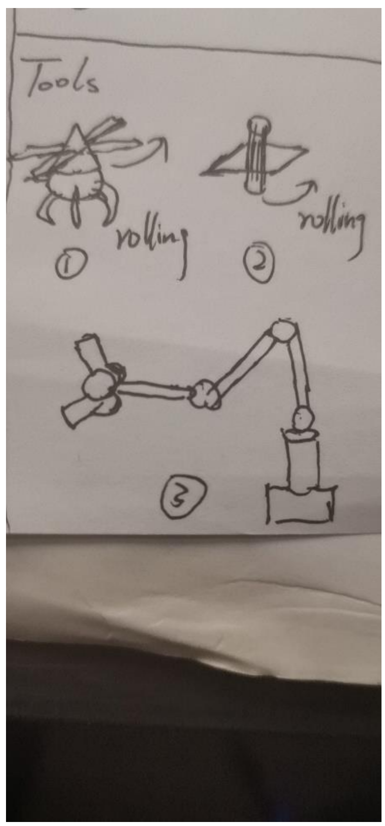
The assemble processing starts from the foot parts. These parts were lifted by an elevator from a well, the gates of the well will open then close during the lifting process. Following that, legs components will come and dock with the foot parts. Then the hip and upper body will be drop from above. For the leg parts, they will connect with main body by using different animation separately. One of the leg will drop down to pick up one freeze hand and the other will wait the hand walk and jump to the connect node.

As I design the hand with index finger, middle finger, ring finger and little finger have equal length, these four fingers could act like human legs and thumb could act like one of the arms. Thus, I have the idea to make this separate hand move towards one of the arms by itself.



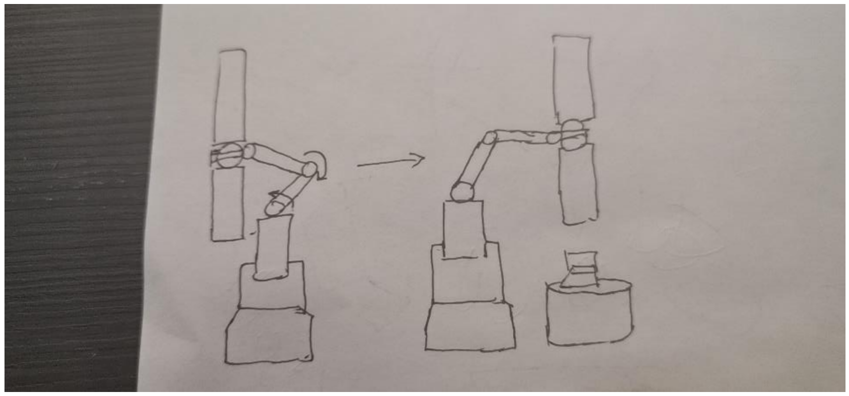
Graph 5.Finger animation sketch

With the idea above, I begin to create these model in Maya, but I noticed I need some tools to help these components move in the scene, otherwise the assembly processing won't looks realistic.



Graph 6. Helper construction tools

The top two are two kind of mini copters. They will help the robot components to lift up, and these components will drop down themselves. As we have gravity in realistic life, it won't be any physical violation if it drop automatically. The robotic arm was used to move legs from a convey belt to the assembly platform. As the move trajectory of the legs is a curve, using copters to lift them looks wired, thus, I made this robotic arm to grab them then move, by changing the translation and rotation of robotic arm components, It could looks that the robotic arm put the leg parts to the platform. A simple sketch illustrate below:



Graph 7. Leg construct sketch

3. Challenges and Highlights:

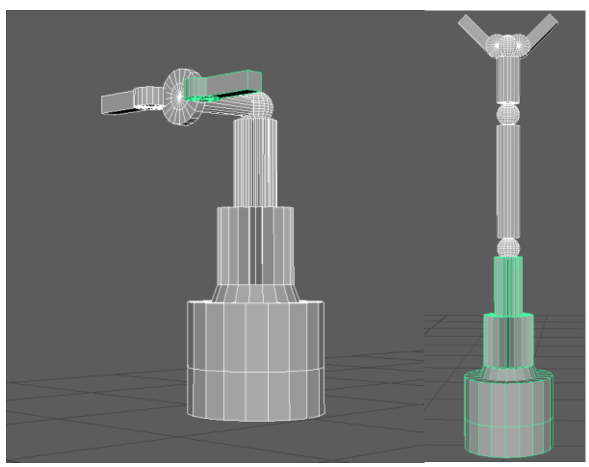
3.1. Challenges:

The main challenges are four parts, namely, creating robotic arm control system, control rotating copters, adjusting hand walking and jumping motion, camera setting. These four challenges will be explain and discuss in the following sections, each section will demonstrate advantages and shortcoming of my current solution.

(In following sections, set key frame will be a default method to save animation frame and won't be mentioned)

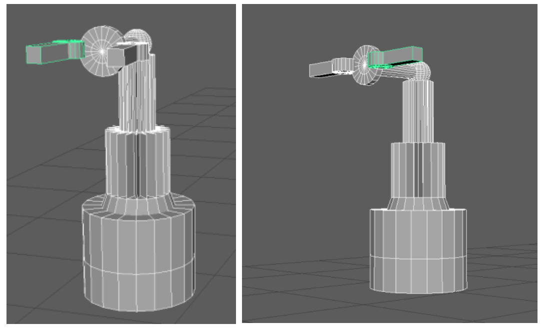
3.1.1. Creating robotic arm control system:

In this animation, two kinds of robotic arms were used, for convenience, they will called MK1(mark I) and MK2(mark II). MK1 was designed to release robot leg parts to a convey belt and MK2 aimed to pick up these leg parts and lift them to the construct platform to combine with foot components. They looks like below:



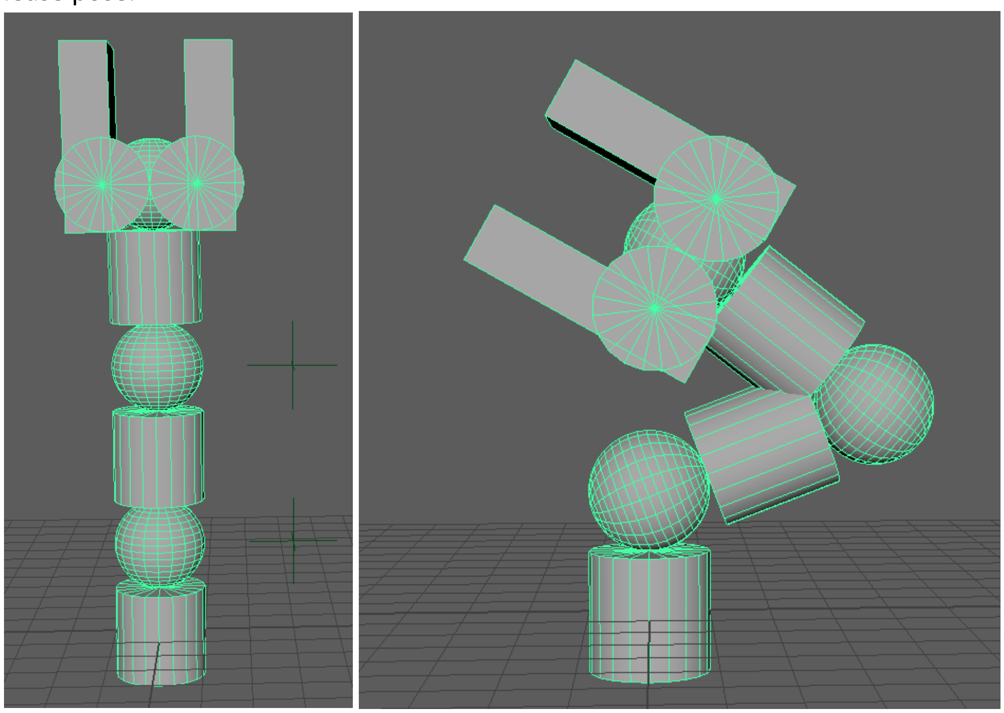
Graph 8 and 9 MK1 and MK2

MK1 is quite simple, because only the claw parts will rotate to release leg parts. To achieve that, the cubic part of the claw was parented to the cylinder part. As the pivot point of the cube goes to the center of the cylinder, the release animation could be create by set two different key frames with different cylinder rotation degrees.



Graph 10 and 11, MK1 claw close and open

However, MK2 is complicated, the control system of MK2 used both aim constrain and parent constrain. Although it could be directly created by using parent constrain, but I want to use aim constrain to help me move it conveniently and practice aim constrain. Each component was parent to the component below it in the initial pose, two independent locator were created next to the arm. Two sphere joints will be apply aim constrain to both those two locator, then, the arm pose could be control by adjust the position of those two locators. In addition, the top sphere join and claw still need manually adjustment to implement grab and release pose.

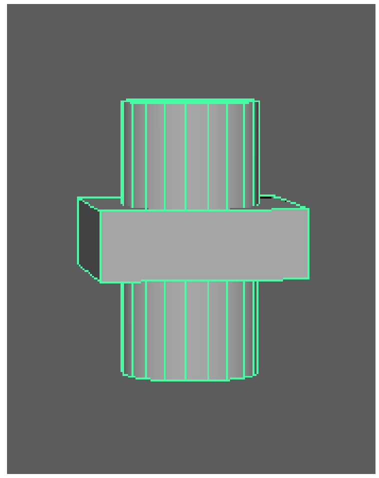


Graph 12 and 13 MK2 constrain design(shape change little bit later)

Although these constrain helps me adjust the arm pose, there still some shortcomings. Firstly, as and I apply one aim constrain locator to two linked object, the bottom joints will influence the rotation degree of top join, which means I can not adjust every possible pose by only move these two locator. This might not be a good idea, but at least it made my arm movement became easier, because I only need to adjust sphere a little bit after move the locators. The claw design succeed from MK1, but rotate the control cylinder, grab and release could be achieve.

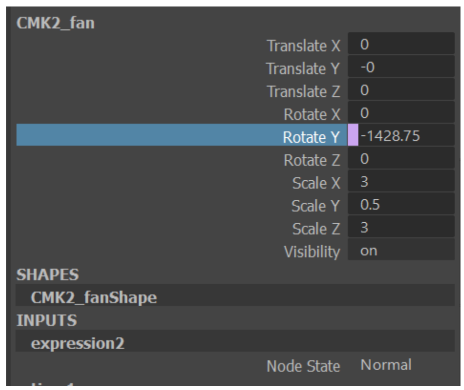
3.1.2. Control rotating copters:

The second challenge is to create and move tiny copters. To solve that, I used point constrain, parent constrain and animation expression.



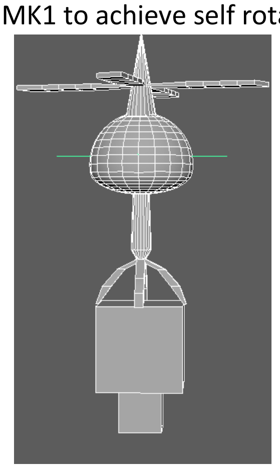
Graph 14, copter MK1

Two kinds of copters were used in this animation, they were name as MK1 and MK2, to differ from robotic arms, they will be called as CMK1 and CMK2 in the following parts of this report. CMK1 is quite simple, the center cylinder were parented to the mid cubic plane and a rotateY expression was assigned to the plane. With the help of this expression, CMK1 always rotate on Y direction, which looks like a small copter suspend in the air.



Graph 15, expression on CMK1

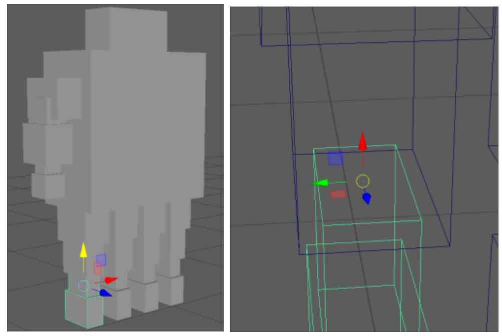
CMK2 is little bit harder than CMK1, it needs an extra point constrain to control the translate of the fan. The issue is that the payload CMK2 grabbed is not centrosymmetric, thus if we parent the bottom part to the fan, the claws will rotate and no longer holds the payload anymore. However, with the help of the point constrain, we could move the copter fan with the copter body. To achieve that, we only needs to point the fan to the locator and parent the locator to the body with copter body. In addition, the fan of CMK2 succeed the expression applied on MK1 to achieve self rotate.



Graph 16, Copter MK2 design

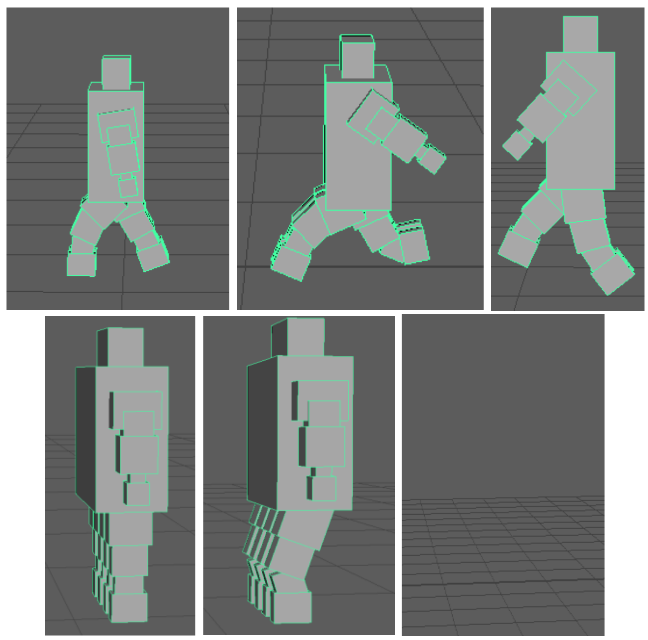
3.1.3. adjusting hand walking and jumping motion:

The most hard part is creating the hand walking and jumping motion. To achieve that, I tried two ways, the first is to use key frame and adjust the graphic editor, the second is to use motion path with point constrain. However, as the second method still need to adjust finger pose, it was discarded in the end.
To adjust the hand pose to make it walks like a human, firstly, every node of the hand need to parent to the node above it. Secondly, we need to change the pivot point of each node from the mass center to the most upper face of the component.



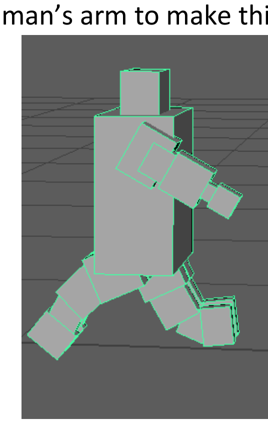
Graph 17, Change hand component pivot point

After that, we need to get six key frame for the hand walking and jumping animation.



Graph 18-23, hand motion key frame

The first four key frames are design for the walking processing. The screenshot based on different height, but we more focus on the finger pose(legs and arm of the hand man). After reading Disney animation rule, I set anticipation key frame for this hand man's arm to make this hand man has an anticipation.

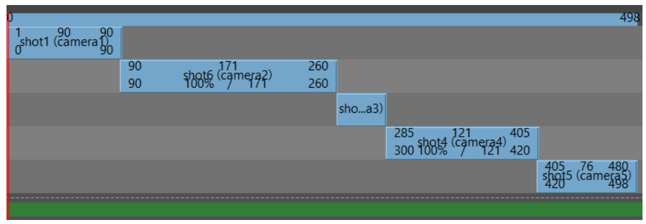


Graph 24 hand walk anticipation

For the jump motion, I rotate some finger components to simulate human squat. Thus in animation video, it looks that this hand man had a perpetration before it jumps up. The last key is use to hide the original hand. As I want to use a point constrain locator to ensure that hand and arm move together, I need to use a duplicate hand to avoid the constrain locator overwrite my keyframe. Thus I did a trick here, to hide the original one in key 405 and show the duplicate one in the same frame.

3.1.4. camera setting:

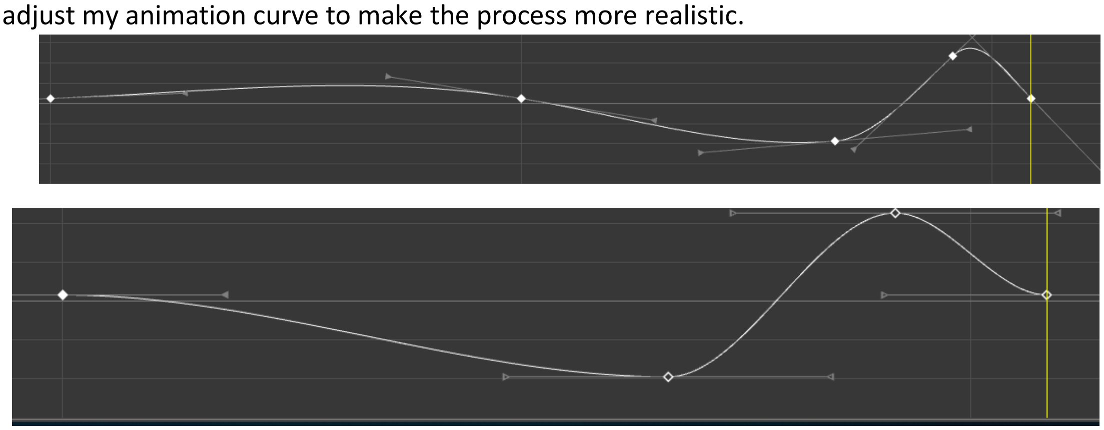
The idea or camera in this animation is like CCTV in real life. I just want camera freeze and switch from one to another, thus I create several cameras and use camera sequencer to control cameras switch at the right time.



Graph25 Camera sequencer

3.2. Highlights:

The highlights of this animation is applying Disney animation rules. I applied anticipation, Squash and stretch, slow in and slow out and kindly made a Disney walking process. I think the most successful part is dropping the body and head from copters. In this 1 second animation, I successfully made a kind of gravity system and a collision rebound mechanism by set key frame and adjust translate and the shape of the body. I also used graphic editor to adjust my animation curve to make the process more realistic.



Graph26 and 27 graphic editor adjustment.

To make sure I have a accelerate and decelerate process during the dropping animation. I finally use graph 27 instead of 26. Although the graph 26 looks more smooth, but it’s speed is not zero at each key frame. According to the Disney animation slow in and slow out rule, sometimes it’s more appropriate to use physics system to make animation. Moreover, in order to make animation consistent, most of the animation curve looks like this.

An attached mp4 file called highlight demonstrate my highlight, please check it out.

The convey belt was create by create mash network, with the help of a track curve and mash curve setting, it rolling automatically. Although it could be create by apply deform to cylinder. I think it works better with mash network animation.

4. Conclusion and discussion:

In conclusion, the methods discussed above are almost all method I used, though some of the components use same methods are not mentioned. After finishing this coursework. I found out the most important factors of creating animation are good story line, detailed and attractive characters and appropriate applying animation methods. The consistency of animations is more significant then blindly apply all possible animation skills.