The Lego World

Members:

Name: YI DING (contact person)
Student ID: 300288746
Email: yidingcool@yahoo.co.nz

Name: MING MIN YING
Student ID: 300266387
Email: lucas.yingmm@gmail.com

Abstract:
This project is about real time animation, lighting and modelling. The project takes the form of a 3D space filled with Lego blocks, and a skeletal man (as seen in Assignment 2).
Each type of Lego block will correspond to different parts of the body. The user can control the skeleton using typical WASD control as seen in many video games. When the skeleton is in close proximity to Lego blocks he will attract them to himself, the blocks attaching to the corresponding part of the body. At the end of the simulation the skeletal man will have a full suit of Lego armor.

Objective of the project: To produce a real-time interactive environment involving a 3D skeleton and Lego block models.

Team Member	technical contributions	Responsibility of each member
YI DING (300288746)	1. Using hierarchy modelling to do some complex animation. 2.How to use AMC file in the game development. 3.Using affine transformation to do some complex animation.	1. YI DING is responsible for the animation parts of this project, it include the animation of the skeletal man and the animation of the interaction between skeletal man and Lego block. 2.Due to the reason that one of the team member KANE was not available(drop out of school) ,YI DING is also response for the camera control and building the Lego world(incudes ground, background and Placing the Lego block)
MING MIN YING(300266387)	1. Using the image lighting from the theory to practice.	MING MIN YING is responsible for implementing the image-base lighting for this project.

Member: YI DING

Topic	Interactive Animation
Technical problems	1. How to make the skeletal running and standing based on the keyboard control. 2. How to deal with position changes from standing->running->standing. 3. Animating Lego block transition from ground to skeletal man.
Summary of work and references	Methods: k-frame animation, 3D affine transformations, ASCII Motion Capture Data. Solution: 1. For Turning: By loading and running the AMC file to make the skeletal man have running and standing action. 2. For animating the standing->running->standing, use key frames to animate the position change between standing and running, running to standing. 3.The most interesting issue is the interaction between skeletal man and Lego blocks. When the distinct between the center of the block and root of the skeleton is less than a predefined value, the block will trigger a key frame animation. The start point will be at the center of the block, and the end point will be the relevant part of the body. This means the animation data of skeletal man and Lego Block need to be tracked ,stored and updated all the time. References VUW COMP308 lecture slide(T. Rhee,Kinematics,2015). VUW COMP308 Assignment2(COMP308 Assignment 2,2015) Siggraph 2015-real-time live https://www.youtube.com/watch?v=2LNZL0iPQIk(nico Gonzalez,2015) SIGGRAPH Asia 2015 - Computer Animation Festival Trailerhttps://www.youtube.com/watch?v=j1Z3hMQQqe4(SIGGRAPH,2015)

Contributions for this project	Team member MINGMIN YING tried very hard at implementing the image based lighting. Although he failed at the end, he have added and commented his code for image based lighting in the code we submmited. I(YI DING) have done my part properly, what you have seen in the demo is made by me, i also have created the video demo for this project.

Description of solution

Solution for skeletal man animation:

The skeletal will have an animation changes between standing and running while keyboard changing through ‘WASD’. To achieve this,i have used two AMC file(standing and running) download from the library online. When the keys are pressed, the skeleton will implement the corresponded AMC file. And also I have used key frame animation to fill the animation gap between running and standing.

Solution for interactive animation between skeleton and Lego block:

When the skeletal man is close enough to the Lego block ,the Lego block will be attracted by the skeletal man and will become to one part of the skeletal body. To achieve this ,I have calculated the animation information of skeletal man all the time, these information include middle point position and bone direction of each bone of the skeletal man. These information will be used for transferring, rotating and scaling the Lego block to the corresponded position of the body with a suitable angle.

Implementation details

Solution for skeletal man animation

Step1:Read the AMC files	I have download two AMC files called standing and running, then I read and store these two AMC file using the function in skeleton class.
	Example code in skeleton.hpp:
	std::vector<std::vector<bone>>m_bone_standactionset;
	std::vector<std::vector<bone>>m_bone_walkingactionset;

Step2:Create an active action set pointer	I have using a pointer pointe to the action set(standing action set or running action set),this active action will be used for the animation of skeletal man. Example code in skeleton.hpp: std::vector<std::vector<bone>>* m_bone_actionset;
Step3:change the pointer value while key pressed	While any key is pressed,the action set pointer while pointe to the running action set,then the skeletal man will have running action ,vice versa for the standing action. Example code in skeleton.cpp:: m_bone_actionset = &m_bone_runningactionset
Step4: add key frame animation to fill the gap of animation between running and standing.	To achieve the key frame animation we need to capture the two situation of the skeleton. The first situation will be the animation info of the skeleton when the key is released. And the second situation will be the first action of the standing set. Then I used these two action created a keyframe action set(contain 10 actions contains the rotation info of the bones). Then make the animation pointer pointe to the created keyframe action set.And after 10 steps,make the action set point to the standing action set. Now,we have a smooth action change between running and standing! See more detail in skeleton.cpp: Void Skeleton::createkeyframe(std::vector<std::vector<bone>>* actionset)

Solution for interactive animation between skeletal man and Lego block

Step1:catch the animation information from skeletal man.	For this project, the middle point of the bone position will be the place where the Lego blocks move to, and the direction of the bone will be the direction of the block as well.

I place the root of the skeletal man at the world coordinates(0,0),then I calculate the world position of the middle point of bones, and update these position info and bone direction all the time will rendering the skeletal man.
See more detail in skeleton.cpp and Result1:
comp308::vec3 position;(used to store middle point postion in skeleton.hpp)
void Skeleton::renderBone(bone *b, GLUquadric* q)

Step2: Create Lego block

All the Lego block have the same primitive model(created from obj file).When load the Lego model I have calculate the center position of the block and their width,heigh and length.

Step3: Set Lego block location and direction.

All the block will be drawing start at the (0,0) in world coordinates with a direction to the positive x axis.
The I used a function called setbody() which set the random position and size of the Lego block.
In this project, there are 31 bones for skeletal man, therefore I have created 31 Lego block corresponded to each bones.
See more detail in main.cpp:
Setbody();

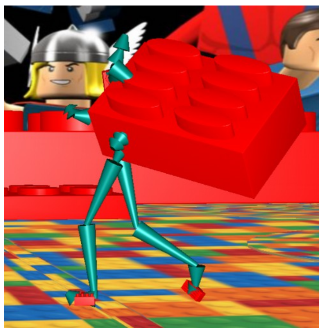
Step4: Moving the Lego block to the corresponded bone when the distance is close enough.

The distance are calculated by using the block position and the middle point of the bone position.
If the distance is close enough:
1. The position of the block will be changed to the bone position in 100 steps.
$P_{block}= P_{block} + (P_{bone}- P_{block})$*N/100(N will be increased from 1 to 100,P means the postion)
2. The direction of the block will be changed to the bone direction in 100 steps.
Rotation axis:
$R=cross(D_{block,}D_{bone});$
Rotation angle:
$Angle=degrees(acos(dot(D_{block,}D_{bone})));$
Then using:
glRotatef(angle*N/100,R.x,R.y,R.z);
(N will be increased from 1 to 100,D means the direction)
3. The size of the block will be changed to the predefined size in 100 steps.(width ,height,length):
$S_{length}=(1+(L_{bone}/L_{block}-1)*N/100);$
$S_{width}=(1+(W_{bone}/W_{block}-1)*N/100);$
$S_{height}=(1+(H_{bone}/H_{block}-1)*N/100);$
$glScalef(S_{length}, S_{height,} S_{width});$
See more detail in main.cpp:
draw();

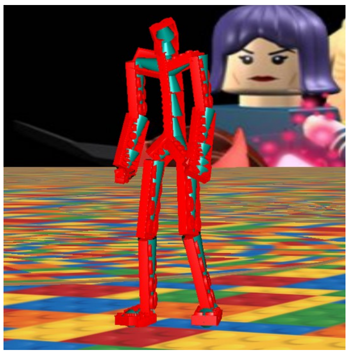
Demo of final output

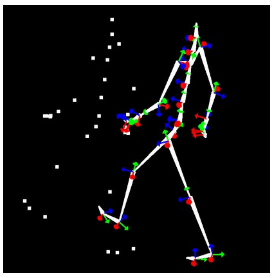
Start !

Left arm is empty


Left arm attract a block

Blocks is rotated and scaled and transferred to the arm finished.


Full suit of Lego block.


Result of capture the middle point position of the bones.

Conclusion and discussion

In conclusion ,I earned a lot experience with using affine transformations in OPENGL which is key part of most animation.

Limitation

1.I have tried to build a key frame animation for standing to running, but I choose to give up because the logic to create two key frames much more complicated and also the keyboard thread and render thread are separate, if I using two key frames, there will be some unpredicted conflictions. For example, when the skeletal man is rendering, the key is pressed and the action set will point to a new action set, however the skeletal man have not finished rendering, the data have been changed in the middle.

2.In my program,keyboardfuction will set running action for the skeletal man ,and the keyboardupfuction will set the standing action for the skeletalman.However,the GlUT version in ECS mechain is not the newest, the keybordup function is not working which will lead to the skeleton cannot have running action, because system called the keybordfunction and keybordupfuntion at same time, the keybordupfuntion will covered the result of keybordfuntion.

This is not actually a bug, and I have found an alternative way to run my programme on ECS machine. When u want skeletal man move, press SHIFT first, then press WASD to move. When u want to stop, release SHIFT first, then WASD. This is because SHIFT +WASD will make the input uppercase, this can be used to distinguish the action even both key function are called. But it still has some small problems in this way.

I have upload two versions, ECS version and General version. The General version will be work very well in personal computer with higher GLUT version.

3.When the skeletal man is moving, the body is moved as a whole thing while doing the running action, this will not happen in the real world.

Future works:

1. The block is moving to a bones linearly, it may be much more interesting to make the movement track curvilinearly by using CR-Spline or the other Spline algorithms.

2. Now the ground is flat,i can make the ground situation much more complicated. For example, some parts need you to climb up to access and some parts need to jump to access. It will make the program much more exciting.