

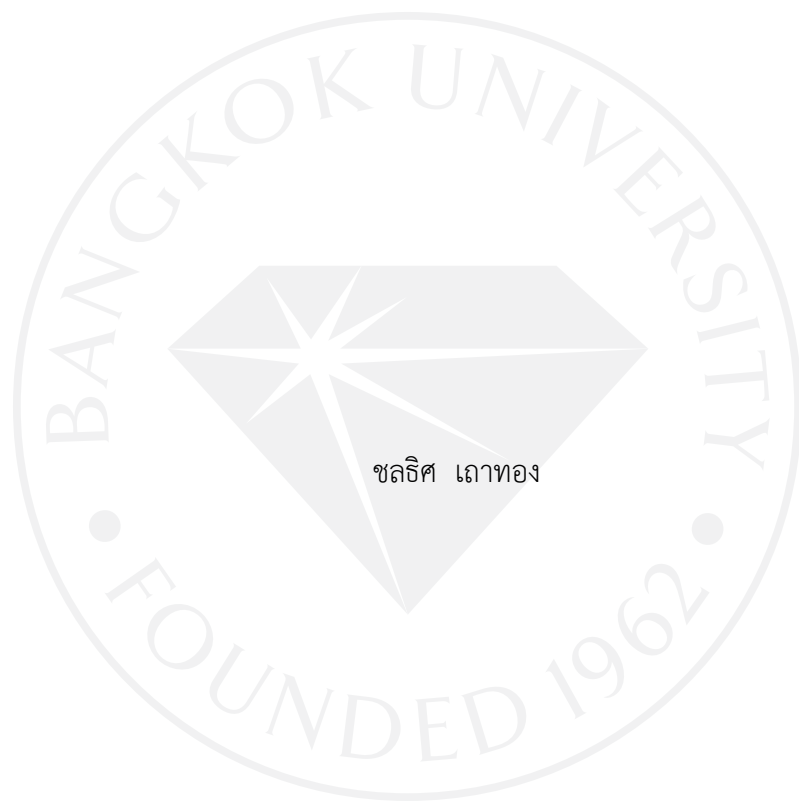
การแสดงผลแอนิเมชันจากฐานข้อมูลการจับความเคลื่อนไหวออนไลน์

Online Animation Displaying from Motion Capture Database



การแสดงผลแอนิเมชันจากฐานข้อมูลการจับความเคลื่อนไหวออนไลน์

Online Animation Displaying from Motion Capture Database



การค้นคว้าอิสระเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศและการจัดการ
มหาวิทยาลัยกรุงเทพ
ปีการศึกษา 2558



©2559

ชลธิศ เกาทอง

สงวนลิขสิทธิ์

บัณฑิตวิทยาลัย มหาวิทยาลัยกรุงเทพ
อนุมัติให้การค้นคว้าอิสระเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศและการจัดการ

เรื่อง การแสดงผลแอนิเมชันจากฐานข้อมูลการจับความเคลื่อนไหวออนไลน์

ผู้วิจัย ชลธิศ เกาทอง


ได้พิจารณาเห็นชอบโดย

อาจารย์ที่ปรึกษา


(ผู้ช่วยศาสตราจารย์ ดร.วรวัฒน์ เชิญสวัสดิ์)

ผู้เชี่ยวชาญ


(ดร.ถิรพล วงศ์สอาดสกุล)


(ดร.คันสนีย์ เทพปัญญา)

คณบดีบัณฑิตวิทยาลัย

29 พฤศจิกายน 2559

ชลธิศ เกาทอง. ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศและการจัดการ,
พฤษภาคม 2559, บัณฑิตวิทยาลัย มหาวิทยาลัยกรุงเทพ.

การแสดงผลแอนิเมชันจากฐานข้อมูลการจับความเคลื่อนไหวออนไลน์ (67 หน้า)

อาจารย์ที่ปรึกษา: ผู้ช่วยศาสตราจารย์ ดร.วรวัฒน์ เชิญสวัสดิ์

บทคัดย่อ

ปัจจุบันได้มีการนำเอาเทคโนโลยีการจับความเคลื่อนไหวหรือโมชันแคปเจอร์มาใช้ในการสร้างตัวละครแอนิเมชันให้มีความสมจริง ซึ่งเทคโนโลยีดังกล่าวเป็นการให้นักแสดงมาสวมใส่ชุดที่มีเซนเซอร์ติดไว้เพื่อบันทึกการเคลื่อนไหวของนักแสดง ทำให้ตัวละครแอนิเมชันมีการเคลื่อนไหวที่เหมือนจริงตามการเคลื่อนไหวของนักแสดง นอกจากนี้เทคโนโลยี Motion Capture ยังถูกนำมาประยุกต์ใช้ในงานอีกหลากหลายวงการ ถึงแม้ว่าปัจจุบันได้มีการจัดทำฐานข้อมูลโมชันแคปเจอร์ให้ใช้จำนวนมากแต่การเปิดดูการเคลื่อนไหวของไฟล์บันทึกความเคลื่อนไหวนั้น ๆ ยังมีความยุ่งยากพอสมควรซึ่งต้องทำการดาวน์โหลดไฟล์ข้อมูลทีบันทึกความเคลื่อนไหว และดาวน์โหลดโปรแกรมที่ใช้แสดงผล จึงสามารถใช้งานได้ ซึ่งใช้เวลาพอสมควรในการแสดงผลไฟล์ข้อมูลนั้น ๆ

งานวิจัยนี้ได้นำเสนอเทคนิคในการพัฒนาโปรแกรมที่สามารถแสดงผ่านเว็บเบราว์เซอร์ของไฟล์ข้อมูลบันทึกความเคลื่อนไหวบนเซิร์ฟเวอร์ในลักษณะของแอนิเมชันสามมิติที่มีการเคลื่อนไหวที่เหมือนจริงเพื่อตอบสนองความต้องการของผู้ใช้ในด้านความสะดวกสบายและความรวดเร็วในการใช้งาน ผู้วิจัยได้ทำการพัฒนาระบบแสดงผลบน 3 แพลตฟอร์มที่นิยมใช้ในการแสดงกราฟิก 3 มิติผ่านเว็บ ซึ่งได้แก่ Web 3D, Flash และ Unity 3D และได้เปรียบเทียบผลลัพธ์ของทั้งสามระบบทั้งในแง่ของความสมจริง ความปลอดภัยของข้อมูล และการนำไปใช้จริง

คำสำคัญ: การตรวจจับความเคลื่อนไหว, ตัวละครแอนิเมชัน, ไฟล์ข้อมูลบันทึกความเคลื่อนไหว, เว็บสามมิติ

Thaothong, C. M.S. (Information Technology and Management), November 2016,
Graduate School, Bangkok University.

Online Animation Displaying from Motion Capture Database (67 pp.)

Advisor: Asst.Prof.Worawat Choensawat, Ph.D.

ABSTRACT

Nowadays, utilizing a motion capture technology known as MoCap can help animators creating the realistic character animation. By using markers or sensors attached to a human body, the motion capture system can capture the movement of real actors for using in the animation process. This is because MoCap data provides accurate and semantically rich data. So far, several free MoCap databases have been provided on the Internet such as CMU database. However, for users to find their desired motion, they have to download the motion data and then use third party software to display the downloaded motion which is time-consuming and difficulties in work.

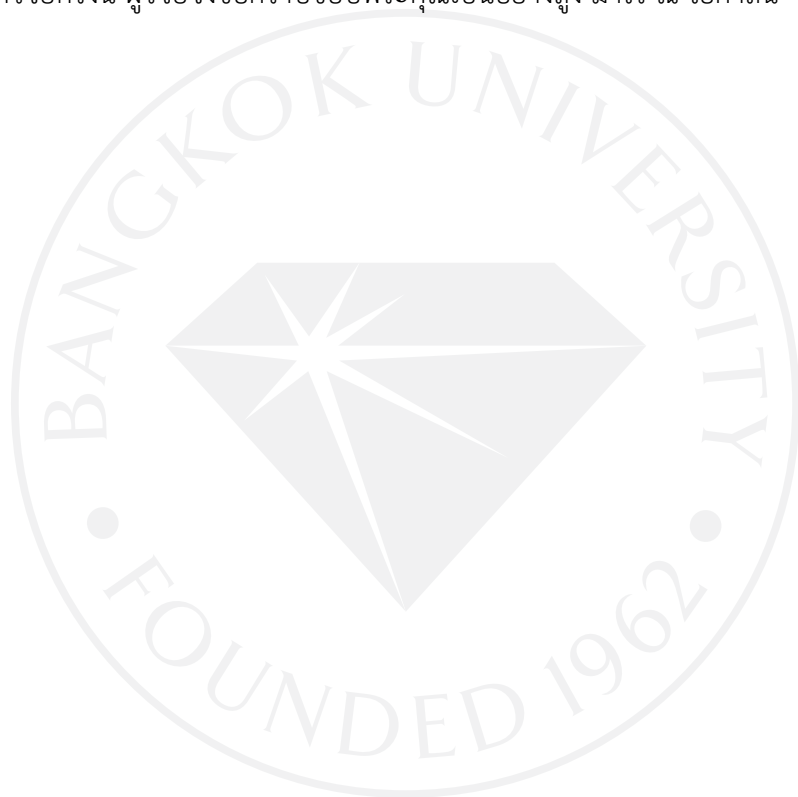
This research presents in the technical development program. That can generate animation character from motion capture database on the server. Users can visual 3D animation of the actual movement contains in motion capture data over the internet via any standard web browsers. Researchers have developed a display system on three popular platform used to display 3D graphics over the Web, which is the Web 3D, Flash and Unity 3D and compare the results of the three systems in terms of realism. Information security and implementing.

Keywords: Motion Capture, Animation Character, Motion Capture Database, Web 3D

กิตติกรรมประกาศ

การค้นคว้าอิสระในครั้งนี้ สำเร็จลุล่วงได้ด้วยความกรุณาจาก ผู้ช่วยศาสตราจารย์ ดร.วรวัฒน์ เชิญสวัสดิ์ อาจารย์ที่ปรึกษาการค้นคว้าอิสระและ ผู้ช่วยศาสตราจารย์ ดร.กิงกาญจน์ สุขคณาภิบาล ซึ่งได้ให้ความรู้ การชี้แนะแนวทาง การศึกษา ตรวจสอบและแก้ไขข้อบกพร่องในงาน ตลอดจนการให้ คำปรึกษาซึ่งเป็นประโยชน์ ในการวิจัยจนงานวิจัยครั้งนี้มีความสมบูรณ์ครบถ้วนสำเร็จไปได้ด้วยดี รวมถึงอาจารย์ท่านอื่น ๆ ที่ได้ถ่ายทอดวิชาความรู้ให้ และสามารถนำวิชาการต่าง ๆ มาประยุกต์ใช้ในการ ศึกษาวิจัยครั้งนี้ ผู้วิจัยจึงขอกราบขอบพระคุณเป็นอย่างสูง มาไว้ ณ โอกาสนี้

ชลธิศ เกาทอง



สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ง
บทคัดย่อภาษาอังกฤษ	จ
กิตติกรรมประกาศ	ฉ
สารบัญตาราง	ฌ
สารบัญภาพ	ญ
บทที่ 1 บทนำ	
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของการวิจัย	2
1.3 ขอบเขตของงานวิจัย	3
1.4 คำถามของงานวิจัย	4
1.5 ประโยชน์ที่คาดว่าจะได้รับ	4
1.6 คำนิยามศัพท์เฉพาะ	4
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	
2.1 ตัวละครแอนิเมชัน (Character Animation)	5
2.2 การตรวจจับความเคลื่อนไหว (Motion Capture)	6
2.3 รูปแบบไฟล์บันทึกข้อมูลความเคลื่อนไหว (Motion Capture Data Format)	9
2.4 เว็บ 3 มิติ (Web 3D)	15
2.5 โปรแกรมสำหรับพัฒนาที่ใช้ในการอ่านไฟล์ข้อมูลบันทึกความเคลื่อนไหว และโปรแกรมที่ใช้ในการแสดงผลบนเว็บไซต์แบบสามมิติ	16
2.6 ความปลอดภัยของข้อมูล	20
บทที่ 3 วิธีการดำเนินงานการวิจัย	
3.1 ขั้นตอนการดำเนินงานวิจัย	22
3.2 เครื่องมือที่ใช้ในการศึกษา	23
3.3 สถาปัตยกรรมของระบบ	23
3.4 การออกแบบหน้าจอการใช้งาน	26
3.5 การป้องกันความปลอดภัยของไฟล์จับความเคลื่อนไหว	27
3.6 การพัฒนาระบบบนแพลตฟอร์ม Unity 3D	28
3.7 การพัฒนาระบบบนแพลตฟอร์ม Flash	29

สารบัญ (ต่อ)

	หน้า
บทที่ 4 ผลการดำเนินงาน	
4.1 การเริ่มต้นการใช้งานระบบ BVH Player	31
4.2 หน้าจอการแสดงผลตัวละครแอนิเมชัน 3 มิติ	32
4.3 การใช้งานหน้าจอแสดงผลแอนิเมชัน 3 มิติ	33
4.4 เกณฑ์การให้คะแนนของแบบประเมิน	40
4.5 สรุปผลความพึงพอใจของระบบ	41
บทที่ 5 สรุปผล อภิปรายผล และข้อเสนอแนะ	
5.1 สรุปผลการศึกษา	43
5.2 อภิปรายผล	44
5.3 ปัญหาของระบบ	45
5.4 แนวทางแก้ปัญหา	46
5.5 ข้อเสนอแนะ และแนวทางการพัฒนาต่อ	46
บรรณานุกรม	47
ภาคผนวก ก หนังสือยินยอมเข้าร่วมการวิจัย	49
ภาคผนวก ข แบบประเมินความพึงพอใจของระบบ	51
ภาคผนวก ค JavaScript Source Code For BVH File Parsing	54
ประวัติผู้เขียน	67
เอกสารข้อตกลงว่าด้วยการอนุญาตให้ใช้สิทธิ์ในรายงานการค้นคว้าอิสระ	

สารบัญตาราง

	หน้า
ตารางที่ 1.1: ฟอรัมที่ใช้ในการเคลื่อนไหว	2
ตารางที่ 2.1: เปรียบเทียบข้อดีและข้อเสียการบันทึกความเคลื่อนไหวเมื่อใช้ในงานแอนิเมชัน	8
ตารางที่ 2.2: รูปแบบไฟล์ข้อมูลบันทึกความเคลื่อนไหว (Motion Capture File Format)	10
ตารางที่ 4.1: แสดงระดับผลการประเมินความพึงพอใจของระบบ	40
ตารางที่ 4.2: แสดงระดับผลการประเมิน	41
ตารางที่ 4.3: แสดงแบบประเมินความพึงพอใจของระบบ	41
ตารางที่ 4.4: ตารางสรุปผลการประเมินความพึงพอใจของระบบ	42
ตารางที่ 5.1: การเปรียบเทียบโปรแกรมที่ใช้ในการพัฒนาระบบ	44



สารบัญภาพ

	หน้า
ภาพที่ 2.1: การใช้เทคโนโลยีตรวจจับความเคลื่อนไหวในการพัฒนาเกมส์กีฬา (EA-Sport Game)	6
ภาพที่ 2.2: ตัวอย่างโครงสร้างกระดูกของรูปแบบไฟล์ BVH	11
ภาพที่ 2.3: เป็นภาพโครงสร้างแบบลำดับชั้นของส่วนต่าง ๆ ของร่างกายของไฟล์บันทึกความ	12
ภาพที่ 2.4: ตัวอย่างโค้ดส่วน Header ของไฟล์บันทึกความเคลื่อนไหวรูปแบบ BVH	13
ภาพที่ 2.5: ตัวอย่างโค้ดส่วน Body ของไฟล์บันทึกความเคลื่อนไหวรูปแบบ BVH	14
ภาพที่ 2.6: ภาพตัวอย่างละครสามมิติและโครงสร้างกระดูกที่ใช้ควบคุมการเคลื่อนไหว	15
ภาพที่ 2.7: เว็บจีแอล (WebGL: Web Graphics Library)	16
ภาพที่ 2.8: ตัวอย่างโปรแกรมสำหรับการแสดงผลจากไฟล์บันทึกการเคลื่อนไหว ในภาษา C++	17
ภาพที่ 2.9: ตัวอย่างการนำไฟล์บันทึกความเคลื่อนไหวมาใส่ในตัวละครในโปรแกรม Unity 3D	18
ภาพที่ 2.10: หน้าจอการทำงานของโปรแกรม NetBeans IDE 8.1	19
ภาพที่ 2.11: ตัวอย่างโปรแกรมการแสดงผลแอนิเมชันที่พัฒนาโดยใช้ NetBeans IDE 8.1	20
ภาพที่ 2.12: ตัวอย่าง URL ที่ถูกเข้ารหัสแล้ว	21
ภาพที่ 3.1: วิธีการสร้างไฟล์บันทึกความเคลื่อนไหวแบบ BVH	24
ภาพที่ 3.2: Flow Chart ของระบบแสดงผลแอนิเมชันจากไฟล์บันทึกความเคลื่อนไหว	25
ภาพที่ 3.3: Flow Chart การใช้งานระบบแสดงผลแอนิเมชันจากไฟล์บันทึกความเคลื่อนไหว	25
ภาพที่ 3.4: ฐานข้อมูลการตรวจจับความเคลื่อนไหว	26
ภาพที่ 3.5: หน้าจอแสดงผลแอนิเมชันจากไฟล์การตรวจจับความเคลื่อนไหว	27
ภาพที่ 3.6: เครื่องมือ Avatar Configuration	27
ภาพที่ 3.7: เครื่องมือ Animator	28
ภาพที่ 3.8: โปรแกรม Visual Studio	29
ภาพที่ 3.9: หน้าจอแสดงผลบนแพลตฟอร์ม Flash	29
ภาพที่ 3.10: หน้าจอแสดงผลบนแพลตฟอร์ม Flash	30
ภาพที่ 4.1 หน้าจอฐานข้อมูลการตรวจจับความเคลื่อนไหว	31
ภาพที่ 4.2: หน้าจอแสดงผลตัวละครแอนิเมชัน 3 มิติ	33
ภาพที่ 4.3: การแสดงผลแอนิเมชัน ตัวควบคุมขนาด BoneSize ที่ขนาดเล็กที่สุด	33
ภาพที่ 4.4: การแสดงผลแอนิเมชัน ตัวควบคุมขนาด BoneSize ที่ขนาดใหญ่ที่สุด	34
ภาพที่ 4.5: การแสดงผลแอนิเมชัน ตัวควบคุมขนาด Scale ที่ขนาดเล็กที่สุด	34
ภาพที่ 4.6: การแสดงผลแอนิเมชัน ตัวควบคุมขนาด Scale ที่ขนาดใหญ่ที่สุด	35
ภาพที่ 4.7: การแสดงผลแอนิเมชัน ตัวควบคุมความเร็ว Speed ที่ความเร็ว ช้าสุด	35

สารบัญภาพ (ต่อ)

	หน้า
ภาพที่ 4.8: การแสดงผลแอนิเมชัน ตัวควบคุมตำแหน่งในแนวแกน X -200 (PositionX)	36
ภาพที่ 4.9: การแสดงผลแอนิเมชัน ตัวควบคุมตำแหน่งในแนวแกน X 200 (PositionX)	36
ภาพที่ 4.10: การแสดงผลแอนิเมชัน ตัวควบคุมตำแหน่งในแนวแกน Y -200 (PositionY)	37
ภาพที่ 4.11: การแสดงผลแอนิเมชัน ตัวควบคุมตำแหน่งในแนวแกน Y 200 (PositionY)	37
ภาพที่ 4.12: การแสดงผลแอนิเมชัน ตัวควบคุมตำแหน่งในแนวแกน Z 200 (PositionZ)	38
ภาพที่ 4.13: การแสดงผลแอนิเมชัน ตัวควบคุมตำแหน่งในแนวแกน Z 200 (PositionZ)	38
ภาพที่ 4.14: Seek Bar ตัวควบคุมการเล่นแอนิเมชัน	39
ภาพที่ 4.15: การควบคุมกล้องและมุมมองของฉาก	39
ภาพที่ 4.16: การควบคุมกล้องและมุมมองของฉาก ที่เปลี่ยนจาก ภาพที่ 4.15	40



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันเทคโนโลยีด้านมัลติมีเดียมีการเปลี่ยนแปลงอย่างรวดเร็ว งานสร้างสรรค์หลายด้านมีการใช้เทคโนโลยีเข้ามาช่วย ทำให้สิ่งต่าง ๆ ที่เคยอยู่ในความฝันสามารถแสดงออกมาให้เห็นเป็นจริงได้ เทคโนโลยีการตรวจจับความเคลื่อนไหว (Motion Capture) เป็นอีกเทคโนโลยีหนึ่ง ที่ช่วยให้ผู้กำกับภาพยนตร์สามารถเนรมิตรคิดค้นตัวละครต่าง ๆ ให้ออกมาแสดงในภาพยนตร์ได้อย่างมหัศจรรย์ ไม่ว่าจะเป็นการเคลื่อนไหวของตัวละครแอนิเมชันในการ์ตูน วิดีโอเกมส์ ระบบจำลองต่าง ๆ หรือ ในการวิจัยวิเคราะห์ระบบร่างกายของมนุษย์ เป็นต้น ซึ่งทำให้งานที่ออกมามีความราบรื่นและสมจริง เป็นเทคโนโลยีที่นิยมใช้ในด้านงานความบันเทิงจำนวนมาก เพราะทำให้สร้างสรรค์ผลงานได้รวดเร็วมากยิ่งขึ้นระยะเวลาในการทำงานที่น้อยลง ทำให้เทคโนโลยีการตรวจจับความเคลื่อนไหวนี้มีความสำคัญมากในปัจจุบันและอนาคต จากตัวอย่างภาพยนตร์เรื่อง Dawn of The Planet of The Apes ที่ใช้เทคโนโลยีการตรวจจับความเคลื่อนไหวมาใช้ในการสร้างภาพยนตร์เพื่อให้ตัวละครมีการเคลื่อนไหวที่สมจริงมากยิ่งขึ้น

ในปัจจุบันได้มีฐานข้อมูล Motion Capture ที่อนุญาตให้นำไปใช้แบบไม่มีค่าใช้จ่ายเช่น ฐานข้อมูล Carnegie Mellon Mocap Database [Cite] และฐานข้อมูล Pollick lab Body Movement Library [Cite] แต่การสืบค้นในฐานข้อมูลขนาดใหญ่เพื่อหาโมชันที่ต้องการไม่ใช่สิ่งที่ย่างย เนื่องจากต้องดาวน์โหลดไฟล์ข้อมูลและใช้โปรแกรมเฉพาะในการแสดงผล ซึ่งเป็นการสิ้นเปลืองเวลาอย่างมากความสำคัญอีกประการหนึ่งของการมีระบบที่สามารถแสดงผลข้อมูล Motion Capture ในรูปแบบแอนิเมชันก่อนที่ผู้ใช้จะทำการดาวน์โหลดคือ เรื่องสิทธิในการเข้าถึงข้อมูล เช่น ข้อมูลบางชุดเป็นลิขสิทธิ์ของนักแสดง โดยผู้ใช้ต้องได้รับอนุญาตก่อนที่จะดาวน์โหลดข้อมูลดังกล่าวได้ในกรณีเช่นนี้การมีระบบแสดงผลจึงเป็นสิ่งที่จำเป็นที่ให้ผู้รู้ว่าข้อมูลดังกล่าวเป็นอย่างไร ก่อนที่จะร้องขอหรือซื้อข้อมูลดังกล่าว ดังนั้นระบบการแสดงผลข้อมูล Motion Capture จึงต้องคำนึงถึงด้านความปลอดภัยของข้อมูลด้วย

ปัจจุบันได้มีการใช้ฟอรัมในการบันทึกการเคลื่อนไหวอยู่หลายฟอรัม แต่ซึ่งแบ่งออกได้เป็นสองกลุ่มใหญ่คือระบบที่เก็บข้อมูลการเคลื่อนไหวในระบบพิกัดบอกตำแหน่งเป็นสามมิติ (3D Coordinates System) และระบบที่เก็บข้อมูลในรูปแบบข้อมูลโครงสร้างเชิงลำดับชั้น ในตารางที่ 1.1 เป็นการแสดงฟอรัมหลักบางส่วนที่มีการใช้ในการบันทึกความเคลื่อนไหว

ตารางที่ 1.1: รูปแบบที่ใช้ในการจับความเคลื่อนไหว

ชื่อและนามสกุลของไฟล์	บริษัท/ผู้พัฒนา	ประเภทของฟอร์แมต
TRC	Motion Analysis	3D coordinates system
C3D	Developed by Dr. Andrew Dainis	3D coordinates system
ASF & AMC	Accliam	Skeleton hierarchy information
BVH	BioVision	Skeleton hierarchy information
HTR	Motion Analysis	Skeleton hierarchy information

จากตารางที่ 1.1 รูปแบบที่ได้รับความนิยมคือ BVH (BioVision Hierarchical Data) เนื่องจากเป็นไฟล์ที่นิยมใช้ในการเก็บข้อมูลที่เป็นารจับความเคลื่อนไหวและมีความเป็นสากลซึ่งเครื่องมือหลาย ๆ ตัวที่ทำงานด้าน Graphic สามารถรองรับ File Format BVH ก็คือ *.bvh เป็นสกุลไฟล์ที่เก็บข้อมูลเกี่ยวกับ Motion Capture ของสิ่งมีชีวิต โดยจะมีรูปแบบการเก็บเป็น Skeleton ของสิ่งมีชีวิต แล้วเก็บข้อมูลว่า Bone แต่ละชิ้นจะมีการขยับอย่างไรในเฟรมใดบ้าง

จากปัญหาดังกล่าวผู้เขียนได้นำเสนอระบบโดยมีสถาปัตยกรรมระบบแบบที่สามารถแสดงผลข้อมูลของไฟล์บันทึกการเคลื่อนไหวออกมาในรูปแบบการเคลื่อนไหวของตัวละครสามมิติที่สมจริง และสามารถใช้งานบนอินเทอร์เน็ตได้โดยเปิดผ่านเว็บเบราว์เซอร์ ไม่ต้องลงอาศัยโปรแกรมอื่น ๆ เพื่อแก้ไขปัญหาดังกล่าว ที่เน้นความสะดวก รวดเร็ว ง่ายต่อการใช้งาน มีความน่าเชื่อถือความปลอดภัยของไฟล์บันทึกความเคลื่อนไหวและประหยัดค่าใช้จ่ายในการทำงาน ได้ไฟล์ข้อมูลการเคลื่อนไหวที่เหมาะสมกับงาน

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 เพื่อศึกษาเทคโนโลยีการบันทึกความเคลื่อนไหวและโครงสร้างของไฟล์ข้อมูลบันทึกความเคลื่อนไหว

1.2.2 เพื่อพัฒนาระบบที่แสดงการเคลื่อนไหวของตัวละคร จากไฟล์ข้อมูลบันทึกความเคลื่อนไหวบนอินเทอร์เน็ต

1.2.3 เพื่อให้การแสดงผลไฟล์บันทึกข้อมูลความเคลื่อนไหว ได้สมจริงสะดวกและรวดเร็วมากยิ่งขึ้น

1.2.4 เพื่อวิเคราะห์ความปลอดภัยของไฟล์บันทึกข้อมูลความเคลื่อนไหวที่อยู่บนเซิร์ฟเวอร์

1.3 ขอบเขตของงานวิจัย

1.3.1 แผนการดำเนินการ

1) ศึกษาค้นคว้ารวบรวมข้อมูลและทำความเข้าใจเกี่ยวกับเทคโนโลยีการตรวจจับความเคลื่อนไหว

2) ศึกษาเครื่องมือโปรแกรมที่ใช้ในการตรวจจับความเคลื่อนไหว

3) ศึกษาโปรแกรมที่ใช้ในการพัฒนาระบบ ได้แก่ NetBeans และ WebGL

4) วิเคราะห์และออกแบบโปรแกรมที่สามารถนำการเคลื่อนไหวมาใส่ในตัวละครแอนิเมชัน

5) พัฒนาโปรแกรมตามที่ได้ออกแบบ

6) ทดสอบและปรับปรุงแก้ไขตัวโปรแกรม

7) จัดทำรายงานการค้นคว้าอิสระและคู่มือการใช้งานของระบบ

1.3.2 ขอบเขตของระบบ

1) ระบบต้องสามารถอ่านไฟล์ข้อมูลบันทึกความเคลื่อนไหวที่อยู่บนเซิร์ฟเวอร์และแสดงผลผ่านตัวละครแอนิเมชันบนเว็บที่เป็นสามมิติได้

2) ระบบต้องสามารถควบคุมการแสดงผลได้ เช่น มีตัวควบคุมตำแหน่งการแสดงความเร็ว ขนาดของตัวละครแอนิเมชัน มีปุ่มเล่น ปุ่มหยุด ปุ่มเริ่มต้นใหม่ ปุ่มเล่นไปข้างหน้า หรือถอยหลังทีละเฟรม เป็นต้น

3) ระบบทั้งหมดทำงานอยู่บนอินเทอร์เน็ต โดยเปิดผ่านเว็บเบราว์เซอร์ตามคุณสมบัติที่กำหนดเท่านั้น

4) การแสดงผลของตัวละครแอนิเมชันต้องมีการเคลื่อนไหวตามไฟล์บันทึกการเคลื่อนไหวได้อย่างสมจริง

5) ไฟล์บันทึกความเคลื่อนไหวที่อยู่บนเซิร์ฟเวอร์มีการกำหนดสิทธิ์การเข้าถึง

6) ระบบต้องมีความปลอดภัยในการเก็บรักษาข้อมูล

1.4 คำถามของงานวิจัย

1.4.1 สามารถพัฒนาระบบที่แสดงผลข้อมูลการเคลื่อนไหวของตัวละครผ่านเซิร์ฟเวอร์ได้หรือไม่

1.4.2 ความปลอดภัยของไฟล์ข้อมูลบันทึกความเคลื่อนไหวที่แสดงผลอยู่ในระดับใด

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 ได้ระบบที่สามารถอ่านไฟล์ข้อมูลบันทึกความเคลื่อนไหวบนเซิร์ฟเวอร์และแสดงผลผ่านตัวละครแอนิเมชันบนเว็บสามมิติ

1.5.2 ได้เรียนรู้เกี่ยวกับเทคโนโลยีการจับภาพความเคลื่อนไหวโครงสร้างของไฟล์ข้อมูลบันทึกความเคลื่อนไหว และการทำงานของเว็บสามมิติ

1.5.3 ได้การแสดงผลของไฟล์บันทึกความเคลื่อนไหวเพื่อนำไปใช้ในการสร้างภาพยนตร์แอนิเมชันหรือเกมส์ ได้สะดวกและรวดเร็วยิ่งขึ้น

1.5.4 ได้ระบบที่มีความปลอดภัยของข้อมูลและน่าเชื่อถือ

1.6 คำนิยามศัพท์เฉพาะ

1.6.1 การตรวจจับความเคลื่อนไหว (Motion Capture) หมายถึง เทคโนโลยีที่ใช้ในการตรวจจับการเคลื่อนไหว ถูกนำมาใช้ในการสร้างภาพยนตร์ การ์ตูนสามมิติ หรือเกมส์สามมิติโดยใช้เซ็นเซอร์ติดตามร่างกายของนักแสดง เพื่ออ่านและแปลค่าความเคลื่อนไหวเข้าสู่คอมพิวเตอร์ทำให้ตัวละครสามมิติที่สร้างขึ้นสามารถแสดงท่าทางได้อย่างสมจริง

1.6.2 ไฟล์ข้อมูลบันทึกความเคลื่อนไหว หมายถึง ไฟล์ที่เก็บข้อมูลเกี่ยวกับการตรวจจับความเคลื่อนไหวของสิ่งมีชีวิตโดยมีรูปแบบการเก็บเป็นโครงกระดูกของสิ่งมีชีวิตแล้วเก็บข้อมูลว่ากระดูกแต่ละชิ้นขยับอย่างไรในเฟรมไหนบ้าง

1.6.3 เว็บเบราว์เซอร์ (Web Browser) หมายถึง โปรแกรมคอมพิวเตอร์ที่ผู้ใช้สามารถดูข้อมูลและโต้ตอบกับข้อมูลสารสนเทศที่จัดเก็บในหน้าเว็บไซต์ที่สร้างด้วยภาษาเฉพาะเช่นภาษาเอชทีเอ็มแอล (html) ที่จัดเก็บไว้ที่ระบบบริการเว็บหรือเว็บเซิร์ฟเวอร์

1.6.4 แอนิเมชัน (Animation) หมายถึง การสร้างภาพเคลื่อนไหวด้วยการนำภาพนิ่งมาเรียงลำดับกันและแสดงผลอย่างต่อเนื่องแบบเฟรมต่อเฟรม (Key Frame Animation)

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

งานวิจัยนี้เป็นการแสดงผลแอนิเมชันจากไฟล์ข้อมูลบันทึกความเคลื่อนไหวที่อยู่บนเซิร์ฟเวอร์ และแสดงผลผ่านตัวละครแอนิเมชันสามมิติผ่านเว็บเบราว์เซอร์ ซึ่งพิจารณาถึงความปลอดภัยของข้อมูลเป็นส่วนสำคัญโดยทำการทบทวนเอกสารและงานวิจัยที่เกี่ยวข้อง ได้แก่ ตัวละครแอนิเมชัน (Character Animation) การตรวจจับความเคลื่อนไหว (Motion Capture) รูปแบบไฟล์บันทึกข้อมูลความเคลื่อนไหวแต่ละประเภท เว็บจีแอลที่ใช้ในการแสดงผล (WebGL) และความปลอดภัยของข้อมูลซึ่งรายละเอียดของเอกสารและงานวิจัยที่เกี่ยวข้อง มีตามลำดับดังต่อไปนี้

2.1 ตัวละครแอนิเมชัน (Character Animation)

2.1.1 ความหมายของตัวละครแอนิเมชัน

ตัวละครแอนิเมชัน คือ กระบวนการทำให้ตัวละครมีชีวิต โดยกระบวนการนี้จะเกิดจากการวาดรูปด้วยมือแบบสองมิติ การปั้นด้วยดินเหนียวหรือการสร้างด้วยคุณสมบัติต่าง ๆ ที่ใช้ในเทคนิคคอมพิวเตอร์กราฟิกเพื่อให้เกิดการเคลื่อนไหว (Yu, Yang, Luo, Li, Brandt & Metaxas, 2016)

จากงานวิจัยข้างต้นเห็นได้ว่าตัวละครแอนิเมชัน (Character Animation) นั้นคือ ตัวละครที่ถูกสร้างขึ้นด้วยวิธีต่าง ๆ แล้วใช้เทคนิคคอมพิวเตอร์กราฟิกทำให้ตัวละครนั้นสามารถมีชีวิตและเคลื่อนไหวได้ตามที่ต้องการ

2.1.2 ความสำคัญของตัวละครแอนิเมชัน

ตัวละครของคน (Humanlike Character) มีความสำคัญเป็นอย่างมากต่องานคอมพิวเตอร์แอนิเมชัน ตัวละครเหล่านี้เป็นส่วนประกอบที่มีความสำคัญต่อการนำไปประยุกต์ใช้ที่หลากหลาย ดังเช่น คอมพิวเตอร์ เกมส์ ภาพยนตร์ที่ใช้เทคนิคพิเศษในการถ่ายทำ (Cinematic Special Effects) การสอนในรูปแบบสภาวะเสมือนจริงที่จำลองโดยเทคนิคทางคอมพิวเตอร์ (Virtual Reality Training) และงานศิลปะที่ต้องใช้การแสดงออกทางสีหน้า (Artistic Expression) เป็นต้น (Reitsma & Pollard, 2007)

จากงานวิจัยข้างต้นเห็นได้ว่าความสำคัญของตัวละครแอนิเมชันนั้น มีความสำคัญมากต่อการใช้งานในด้านต่าง ๆ เช่น งานด้านความบันเทิง ไม่ว่าจะเป็น เกมส์ ระบบจำลอง ภาพยนตร์ หรือการ์ตูน เป็นต้น ต่างก็ใช้ตัวละครแอนิเมชันเป็นส่วนประกอบหลัก ๆ เพื่อให้งานนั้น ๆ ออกมาสมบูรณ์มีชีวิตชีวาเสมือนจริงมากที่สุด ดังนั้นตัวละครแอนิเมชันจึงมีความสำคัญมาก เช่นตัวละครแอนิเมชัน

จากเรื่อง มาดากัสกา (Madagascar) ซึ่งแอนิเมชันในหลาย ๆ เรื่องจะมีตัวละครแอนิเมชันเป็นองค์ประกอบหลักสำคัญในการดำเนินเนื้อเรื่อง

2.2 การตรวจจับความเคลื่อนไหว (Motion Capture)

2.2.1 ความหมายการตรวจจับความเคลื่อนไหว

การตรวจจับความเคลื่อนไหว หมายถึง ข้อมูลที่บันทึกการเคลื่อนไหว และทำการแปลงการเคลื่อนไหวนั้นให้อยู่ในรูปแบบดิจิทัล การสร้างภาพเคลื่อนไหวให้กับตัวละครสามมิติ (3D Character Animation) ได้มามีบทบาทในอุตสาหกรรมบันเทิงอย่างมาก ไม่ว่าจะเป็นการสร้างภาพยนตร์สามมิติ หรือการทำเกมส์คอมพิวเตอร์ โดยวิธีหนึ่งที่นิยมใช้กันมากก็คือ การสร้างความเคลื่อนไหวโดยวิธีโมชันแคปเจอร์ (Motion Capture) ซึ่งคือการตรวจจับความเคลื่อนไหวของจุดสำคัญต่าง ๆ บนตัวละคร ซึ่งมีข้อดีกว่าการสร้างภาพเคลื่อนไหวด้วยมืออย่างมาก เพราะช่วยลดระยะเวลาในการสร้าง และยังให้การเคลื่อนไหวที่สมจริงกว่า (ดีชกรณั ดันเจริญ, 2554)

การใช้เทคโนโลยีในการตรวจจับความเคลื่อนไหวโดยส่วนใหญ่จะใช้ในด้านอุตสาหกรรมบันเทิง เช่น ภาพยนตร์ แอนิเมชัน หรือการพัฒนาเกมส์ เป็นต้น จากภาพที่ 2.1 เป็นการใช้อุปกรณ์ตรวจจับความเคลื่อนไหวในการพัฒนาเกมส์ด้านกีฬาเพื่อให้ตัวละครในเกมส์มีการเคลื่อนไหวที่สมจริงมากยิ่งขึ้น

ภาพที่ 2.1: การใช้เทคโนโลยีการตรวจจับความเคลื่อนไหวในการพัฒนาเกมส์กีฬา (EA-Sport Game)



ที่มา: Ben, D. (2010). *Sport sims motion capture technology*. Caerleon: Usk Valley.

2.2.2 ประเภทของการบันทึกความเคลื่อนไหว

การบันทึกความเคลื่อนไหวแบ่งได้เป็น 3 ประเภทหลัก ๆ ตามเทคโนโลยีที่ใช้ ดังนี้

- 1) การบันทึกความเคลื่อนไหวโดยชุดเชิงกล (Electro-mechanical Motion Capture System) ประเภทนี้ใช้กับการจับการเคลื่อนไหวของมนุษย์ โดยเฉพาะชุดที่สร้างขึ้นพิเศษให้มนุษย์สวมใส่สำหรับการบันทึกความเคลื่อนไหว ชุดจะมีลักษณะเป็นโครงสร้างที่เชื่อมต่อกันมีตัวต้านทานปรับค่าได้ในกรณีการวัดการหมุนของจุดต่าง ๆ ตามข้อต่อสำคัญของร่างกาย การรู้มุมการหมุนของจุดต่าง ๆ ทำให้สามารถรู้ท่าทางการเคลื่อนไหวของผู้สวมชุดได้ (ดีชกรณั ตันเจริญ, 2554)
- 2) การบันทึกความเคลื่อนไหวโดยใช้แม่เหล็ก (Magnetic Motion Capture System) ใช้เซ็นเซอร์ติดตั้งไว้ตามส่วนต่าง ๆ เพื่อวัดคลื่นแม่เหล็กไฟฟ้าความถี่ต่ำที่ออกมาจากแหล่งกำเนิดคลื่นที่ติดตั้งไว้บริเวณที่บันทึก เซ็นเซอร์แต่ละตัวจะถูกต่อสายมาเข้ายังวงจรวัดค่าเพื่อหาตำแหน่งของเซ็นเซอร์แต่ละตัวในสนามแม่เหล็ก วงจรวัดค่าส่งข้อมูลไปเข้าเครื่องคอมพิวเตอร์เพื่อแสดงตำแหน่งและการหมุนในรูปแบบสามมิติ (Ricci & Formica, 2014)
- 3) การบันทึกความเคลื่อนไหวโดยใช้กล้องวิดีโอ (Optical Motion Capture System) ผู้แสดงใส่ชุดที่มีจุดสะท้อนแสง (Reflective Dots) มีกล้องจับภาพอยู่โดยรอบ จุดสะท้อนแสงจะใช้เพื่อวิเคราะห์การเคลื่อนไหวของผู้แสดง ตัวอย่างที่ใช้ในงานด้านดนตรี เช่น การตรวจจับการเคลื่อนไหวของคอนดักเตอร์ (Conductor) และพัฒนามาใช้เพื่องานทางการแพทย์ (Biomedical) เช่น การตรวจจับการบาดเจ็บในกีฬาหรือการวิเคราะห์สมรรถภาพของนักกีฬา เป็นต้น ข้อดีคือผู้แสดงสามารถเคลื่อนไหวได้อย่างอิสระ ไม่มีสายมาโยงกับอุปกรณ์สามารถใช้กับวัตถุขนาดใหญ่หรือวัตถุหลายชิ้นได้ และยังได้ข้อมูลที่มีความละเอียด ส่วนข้อเสีย อาจเกิดการรบกวนของแสงจากภายนอก จุดสะท้อนแสงอาจแตกหักได้เนื่องจากผลกระทบของการแสดง ทำให้สูญเสียข้อมูลบางส่วน สามารถเพิ่มเติมแก้ไขข้อเสียข้อมูลได้โดยการประมาณตำแหน่งจุดที่หายไป นอกจากนั้นนักแสดงจะต้องใส่ชุดที่มีจุดสะท้อนแสงและใช้เวลาในการประมวลผลค่อนข้างนาน (Schubert, Eggensperger, Gkogkidis, Hutter, Ball & Burgard, 2016)

สามารถสรุปความหมายของการตรวจจับความเคลื่อนไหวได้ว่า การใช้เครื่องมือชุดสวมใส่ (Body Suit) ที่มีเซ็นเซอร์ หรือ มาร์คเกอร์ติดตั้งที่ชุดตามจุดสำคัญต่าง ๆ ซึ่งเครื่องมือที่ใช้ในการตรวจจับให้นักแสดงทำการแสดงโดยสวมใส่ชุดเอาไว้ เมื่อเพิ่มส่วนของใบหน้า นิ้วมือ และรายละเอียดของท่าทางลงไปแล้ว จะทำให้ได้ข้อมูลที่จัดเก็บมีคุณภาพมากขึ้น

2.2.3 ความสำคัญของการตรวจจับความเคลื่อนไหว

เทคโนโลยีการตรวจจับความเคลื่อนไหว สามารถนำมาประยุกต์ใช้ในงานทางการแพทย์ได้ เช่น การทำอุปกรณ์ถ่วงมือที่บันทึกข้อมูลความเคลื่อนไหวที่ใช้ในการผ่าตัดแบบเสมือนจริง ในงานวิจัยได้ศึกษาการพัฒนาอุปกรณ์ตรวจจับความเคลื่อนไหวแบบพิเศษขึ้นมาเพื่อใช้ในการผ่าตัดเสมือนจริง

ซึ่งอุปกรณ์ที่ใช้ในการตรวจจับความเคลื่อนไหวต้องมีความแม่นยำสูงเพื่อนำไปใช้ในการสอนการผ่าตัดแบบเสมือนจริง (Cheng, Guo & Gao, 2010)

เทคโนโลยี Optical Motion Capture เป็นเคล็ดลับในการสร้างงานแอนิเมชัน (Animation) การพัฒนาเกมส์ (Games) และคอมพิวเตอร์กราฟิก (Computer Graphics) ที่ใช้ในการจับการเคลื่อนไหวของนักแสดงจริงก่อนที่จะนำไปใช้ขับเคลื่อนตัวละคร โดยสามารถลดระยะเวลาและต้นทุนกว่าการทำงานในแบบแอนิเมตเฟรมต่อเฟรม (Key Frame Animation) ได้เป็นอย่างมาก (สุวิชัย พรธธา, 2552)

การประยุกต์ใช้เทคโนโลยีตรวจจับความเคลื่อนไหวในงานทางการแพทย์ ในปัจจุบันมีการใช้รูปแบบจำลองการผ่าตัดเสมือนจริงจำนวนมากในการเรียนการสอนของนักศึกษาแพทย์ แต่อุปกรณ์สำหรับการผ่าตัดเสมือนจริงนั้นมีราคาแพงเกินไป เนื่องจากอุปกรณ์นั้นเอาไว้ใช้ในการสร้างภาพยนตร์ จึงได้จัดทำอุปกรณ์ที่เป็นถูงมือสำหรับการผ่าตัดเสมือนจริงโดยเฉพาะในราคาที่ไม่สูงจนเกินไปและมีความแม่นยำเพื่อให้นักศึกษาแพทย์เอาไว้ใช้ในการศึกษา (Cao, Gao, Wang & Li, 2016)

จากงานวิจัยข้างต้นเห็นได้ว่าการตรวจจับความเคลื่อนไหวเป็นเทคโนโลยีที่ทำให้การสร้างสรรคผลงานรวดเร็วมากยิ่งขึ้นประหยัดต้นทุนในการทำงาน และยังได้ตัวละครแอนิเมชันที่มีการเคลื่อนไหวที่ราบรื่น เสมือนจริงมากที่สุด ซึ่งการสร้างแอนิเมชันสามารถทำได้ทั้ง 2 แบบได้แก่ การสร้าง แอนิเมชันแบบเฟรมต่อเฟรม (Key Frame Animation) และการใช้เทคโนโลยีตรวจจับความเคลื่อนไหวในการสร้างแอนิเมชัน วิธีการเลือกใช้งานจะขึ้นอยู่กับงานที่ทำ เพราะเทคโนโลยีการตรวจจับความเคลื่อนไหวก็ยังมีข้อจำกัดบางประการ ดังตารางที่ 2.1 คือตารางเปรียบเทียบข้อดีและข้อเสียของการใช้เทคโนโลยีบันทึกความเคลื่อนไหวในการสร้างแอนิเมชัน

ตารางที่ 2.1: เปรียบเทียบข้อดีและข้อเสียของการบันทึกความเคลื่อนไหวเมื่อใช้งานแอนิเมชัน

ข้อดีของการใช้งานการบันทึกความเคลื่อนไหวในงานแอนิเมชัน	ข้อเสียของการใช้งานการบันทึกความเคลื่อนไหวในงานแอนิเมชัน
1) มีความรวดเร็วและเก็บผลเป็นแบบ Real-time	1) ใช้อุปกรณ์และโปรแกรมที่มีลักษณะพิเศษที่ใช้สำหรับเก็บรวบรวมและประมวลผล
2) สามารถทำได้หลายรูปแบบและหลายวิธี	2) ราคาของอุปกรณ์และโปรแกรมนั้นราคาสูงเกินกว่าที่บุคคลทั่วไปจะสามารถจ่ายได้ในธุรกิจขนาดเล็ก

(ตารางมีต่อ)

ตารางที่ 2.1 (ต่อ): เปรียบเทียบข้อดีและข้อเสียของการบันทึกความเคลื่อนไหวเมื่อใช้ในงานแอนิเมชัน

ข้อดีของการใช้งานการบันทึกความเคลื่อนไหวในงานแอนิเมชัน	ข้อเสียของการใช้งานการบันทึกความเคลื่อนไหวในงานแอนิเมชัน
3) ใช้เวลาในการทำงานน้อยกว่าแบบเฟรมต่อเฟรม	3) การเคลื่อนไหวที่ไม่เป็นไปตามกฎหรือเสมือนจริงจะไม่สามารถจัดเก็บได้
4) การเคลื่อนไหวที่ซับซ้อนและการเคลื่อนไหวที่เสมือนจริงในเรื่องของน้ำหนักและการแลกเปลี่ยนของแรงสามารถสร้างขึ้นมาได้อย่างเหมือนจริง	4) เมื่อมีปัญหาเกิดขึ้น การแก้ปัญหาด้วยการทำการบันทึกความเคลื่อนไหวใหม่อีกครั้งง่ายกว่าการแก้ไขข้อมูลที่มีอยู่
5) เมื่อใช้เทคนิคแบบดั้งเดิมในการสร้างแอนิเมชัน ตัวงานทั้งหมดจะไม่ได้แปรผันตามความซับซ้อนหรือความยาวของชุดข้อมูลในระดับเดียวกัน	5) กระบวนการทำงานของการบันทึกความเคลื่อนไหวอาจต้องการสภาพแวดล้อมที่พิเศษเฉพาะสำหรับการทำงาน

จากตารางที่ 2.1 การใช้เทคโนโลยีตรวจจับความเคลื่อนไหวจะเหมาะกับงานที่มีการเคลื่อนไหวที่มนุษย์สามารถเลียนแบบได้ การเคลื่อนไหวที่ออกมาจะมีความสมจริงต่างจากการสร้างแอนิเมชันแบบเฟรมต่อเฟรมซึ่งจะเหมาะสำหรับตัวละครแอนิเมชันที่มนุษย์ไม่สามารถเคลื่อนไหวได้

2.3 รูปแบบไฟล์บันทึกข้อมูลความเคลื่อนไหว (Motion Capture Data Format)

รูปแบบไฟล์ที่ใช้ในการบันทึกความเคลื่อนไหวมีหลากหลายประเภท ซึ่งจะแบ่งไปตามลักษณะการใช้งาน แต่ละรูปแบบจะมีความแตกต่างกัน วิธีการเลือกใช้งานแต่ละรูปแบบจะขึ้นอยู่กับความเหมาะสมของงานนั้น ๆ รูปแบบไฟล์ข้อมูลบันทึกความเคลื่อนไหว (Motion Capture File Format) ซึ่งมีหลากหลายรูปแบบแต่แบบที่ได้รับความนิยมคือ BioVision Hierarchical หรือที่เรียกว่า BVH ดังตารางที่ 2.2 จะแสดงให้เห็นถึงรูปแบบไฟล์ข้อมูลบันทึกความเคลื่อนไหว (Motion Capture File Format) แต่ละแบบ

ตารางที่ 2.2: รูปแบบไฟล์ข้อมูลบันทึกความเคลื่อนไหว (Motion Capture File Format)

File Extension	Associated Company/ Description	File Format Reference
ASC	Ascension	NO LINK
ASF & AMC	Acclaim	http://www.darwin3d.com/gamedev/acclaim.zip
ASK & SDL	BioVision/Alias	NO LINK
BVA & BVH	BioVision	http://www.biovision.com/bvh.html
BRD	LambSoft Magnetic Format	http://www.dcs.shef.ac.uk/~mikem/fileformats/brd.html
C3D	Biomechanics, Animation and Gait Analysis	http://www.c3d.org/c3d_format.htm
CSM	3D Studio Max, Character Studio	http://www.dcs.shef.ac.uk/~mikem/fileformats/csm.html
DAT	Polhemous	NO LINK
GTR, HTR & TRC	Motion Analysis	http://www.cs.wisc.edu/graphics/Courses/cs-838-1999/Jeff/ {HTR.html, TRC.html}
MOT & SKL	Acclaim-Motion Analysis	(Under Development - http://www.cs.wisc.edu/graphics/Courses/cs-838-1999/Jeff/SKL-MOT.html)

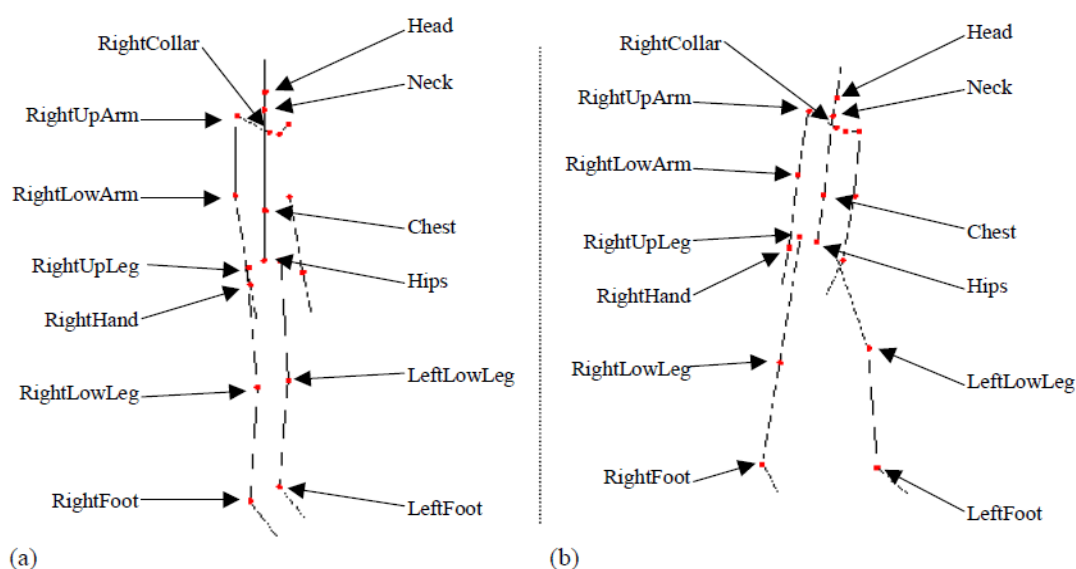
ที่มา: Meredith, M., & Maddock, S. (2001). Motion capture file formats explained. *Department of Computer Science, University of Sheffield, 211, 241-244.*

2.3.1 ไฟล์บันทึกความเคลื่อนไหวแบบ BioVision: BVH (BioVision Hierarchical Data)

1) ความหมายของไฟล์บันทึกข้อมูลความเคลื่อนไหว (BVH File) คือไฟล์รูปแบบที่เกี่ยวกับการตรวจจับความเคลื่อนไหว (Motion Capture) BVH ย่อมาจากคำว่า Biovision Hierachy มีคำแปลว่า มโนภาพเกี่ยวกับสิ่งมีชีวิตที่มีโครงสร้างเป็นลำดับชั้น โดยมีรูปแบบ

การเก็บข้อมูลอย่างเป็นลำดับขั้น (*.bvh) เป็นสกุลไฟล์ที่เก็บข้อมูลเกี่ยวกับการตรวจจับความเคลื่อนไหวของสิ่งมีชีวิต โดยจะมีรูปแบบการเก็บโครงกระดูกของสิ่งมีชีวิต แล้วเก็บข้อมูลว่า กระดูก แต่ละชิ้นจะขยับอย่างไรในเฟรมไหนบ้าง ดังภาพที่ 2.2 ตัวอย่างโครงสร้างกระดูกรูปแบบไฟล์ BVH ภาพ (a) คือ ตำแหน่งเริ่มต้น และ ภาพ (b) คือ เฟรมแรกของการเคลื่อนไหว

ภาพที่ 2.2: ตัวอย่างโครงสร้างกระดูกของรูปแบบไฟล์ BVH

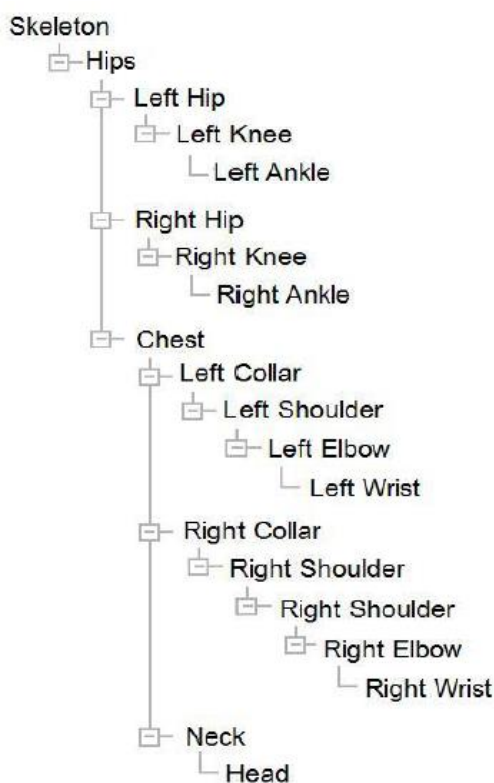


ที่มา: Meredith, M., & Maddock, S. (2001). Motion capture file formats explained. *Department of Computer Science, University of Sheffield, 211, 241-244.*

1) ความสำคัญของไฟล์บันทึกข้อมูลความเคลื่อนไหวรูปแบบ BVH คือการนำเอาไฟล์ที่มีข้อมูลบันทึกความเคลื่อนไหวไปใส่กับตัวละครแอนิเมชันสามมิติได้ จะทำให้ตัวละครเหล่านั้นสามารถขยับท่าทางได้เหมือนกับสิ่งมีชีวิตจริง ๆ มากขึ้นทั้งยังช่วยประหยัดเวลาในการทำแอนิเมชัน

ไฟล์ BVH เป็นไฟล์ข้อมูลบันทึกการเคลื่อนไหว รูปแบบข้อมูลแบบ BVH เป็นรูปแบบที่นิยมใช้ในการทำการบันทึกความเคลื่อนไหวเพราะเป็นรูปแบบที่แบ่งข้อมูลโครงกระดูกเป็นส่วน ๆ ทั้งข้อมูลขนาดและข้อมูลการหมุน ไฟล์ BVH จัดเก็บข้อมูลจุดหมุน 18 จุดบริเวณข้อต่อต่าง ๆ ดังภาพที่ 2.3 เป็นภาพโครงสร้างแบบลำดับขั้นของส่วนต่าง ๆ ของร่างกายของไฟล์บันทึกความเคลื่อนไหวแบบ BVH

ภาพที่ 2.3: เป็นภาพโครงสร้างแบบลำดับชั้นของส่วนต่าง ๆ ของร่างกายของไฟล์บันทึกความเคลื่อนไหวแบบ BVH



ไฟล์ BVH ประกอบด้วย 2 ส่วน ในส่วนแรกคือส่วนของ Header ซึ่งอธิบายเกี่ยวกับ ลำดับชั้นและค่าของขนาดโครงกระดูกแต่ละส่วน โดยเริ่มต้นส่วน Header ด้วยคีย์เวิร์ดที่ว่า Hierarchy จากนั้นบรรทัดต่อมาตามด้วยคีย์เวิร์ด ROOT แล้วตามด้วยชื่อของโครงกระดูกส่วนที่เป็น ROOT โดยแต่ละส่วนของโครงกระดูกจะมีข้อมูล OFFSET ซึ่งเป็นตำแหน่งของโครงกระดูกเพื่อใช้ในการกำหนดขนาดของโครงกระดูก ข้อมูล CHANNELS เป็นตัวบอกว่าข้อมูลที่เราจะเก็บในแต่ละเฟรมของภาพมีอะไรบ้าง ซึ่งในส่วนของ ROOT จะเก็บข้อมูลตำแหน่งและข้อมูลการหมุน กระดูกส่วนอื่นเก็บเฉพาะข้อมูลการหมุนเท่านั้น ข้อมูลของข้อต่อส่วนถัดไปจะใช้คำว่า JOINT แล้วตามด้วยชื่อข้อต่อ โดยเมื่อถึงข้อต่อสุดท้ายเช่น ข้อมือ ข้อเท้า จะจบด้วยคำว่า End Site แล้วตามด้วยตำแหน่งของส่วนปลายของกระดูกท่อนั้น ดังภาพที่ 2.4 ตัวอย่างโค้ดส่วน Header ของไฟล์บันทึกความเคลื่อนไหวรูปแบบ BVH

ภาพที่ 2.4: ตัวอย่างโค้ดส่วน Header ของไฟล์บันทึกความเคลื่อนไหวรูปแบบ BVH

```

HIERARCHY
ROOT Hips
{
  OFFSET 0.00 0.00 0.00
  CHANNELS 6 Xposition Yposition Zposition Zrotation Xrotation Yrotation
  JOINT Chest
  {
    OFFSET 0.000000 4.570000 0.000000
    CHANNELS 3 Zrotation Xrotation Yrotation
    JOINT Neck
    {
      OFFSET 0.000000 17.620001 0.000000
      CHANNELS 3 Zrotation Xrotation Yrotation
      JOINT Head
      {
        OFFSET 0.000000 5.190000 0.000000
        CHANNELS 3 Zrotation Xrotation Yrotation
        End Site
        {
          OFFSET 0.000000 4.140008 0.000000
        }
      }
    }
  }
  JOINT LeftCollar
  {
    OFFSET 1.060000 15.330000 01.760000
    CHANNELS 3 Zrotation Xrotation Yrotation
    JOINT LeftShoulder
    {
      OFFSET 5.00000 0.000000 0.000000
      CHANNELS 3 Zrotation Xrotation Yrotation
      JOINT LeftElbow
      {
        OFFSET 0.000000 -11.90000 0.000000
        CHANNELS 3 Zrotation Xrotation Yrotation
      }
    }
  }
}

```

ไฟล์ตัวอย่างข้อมูลที่บันทึกความเคลื่อนไหวในรูปแบบ BVH ส่วนแรกของไฟล์จะเริ่มต้นด้วยคำว่า “HIERARCHY” บรรทัดต่อมาจะใช้คำว่า “ROOT” และชื่อของกระดูกที่เป็นรากของลำดับชั้นโครงสร้างกระดูก คำว่า “ROOT” จะบ่งบอกถึงการเริ่มต้นของโครงสร้างลำดับชั้นโครงสร้างกระดูก คำที่สำคัญต่อมาคือ “OFFSET” จะบ่งบอกตำแหน่งการวางของกระดูก คำต่อมาคือ “CHANNEL” จะบ่งบอกถึงตำแหน่งและทิศทางการหมุนของกระดูก คำต่อมาคือ “JOINT” จะเป็นส่วนที่บอกชื่อของกระดูกชั้นนั้น ๆ ได้แก่ Chest, Neck, Head, LeftCollar, LeftShoulder, LeftElbow, LeftWrist, RightCollar, RightShoulder, RightElbow, RightWrist, LeftHip LeftKnee, LeftAnkle, RightHip, RightKnee และ RightAnkle ส่วนสุดท้ายจะเป็นส่วนที่บ่งบอกถึงตำแหน่งการเคลื่อนไหวของกระดูกจะมีคำเริ่มต้นว่า “Motion” ทัดมาจะเป็นคำว่า “Frame” ซึ่งจะบ่งบอกถึงจำนวนเฟรมทั้งหมด และ “Frame Time” จะบอกถึงจำนวนวินาทีต่อเฟรม ถึงแม้ว่า ไฟล์ BVH ที่มีส่วนประกอบของโครงสร้างกระดูกหลาย ๆ ส่วน ปกติแล้วไฟล์ BVH 1 ไฟล์จะมีแค่โครงสร้างกระดูก 1 ร่าง

ในส่วนที่สองคือ ส่วนข้อมูลการเคลื่อนไหว โดยเริ่มต้นด้วยคีย์เวิร์ดคำว่า MOTION ตามด้วย คำว่า Frames ตามด้วยตัวเลขเพื่อบอกจำนวนเฟรมของการเคลื่อนไหว และ Frame Time เพื่อบอก ข้อมูลการแสดงผลภาพของเฟรมหนึ่ง จากนั้นจะเป็นส่วนของข้อมูลการเคลื่อนไหวซึ่งจะเรียงตาม CHANNELS ในส่วนของ Header ดังภาพที่ 2.5 เป็นภาพตัวอย่างโค้ดส่วน Body ของไฟล์บันทึก ความเคลื่อนไหวรูปแบบ BVH ซึ่งเป็นส่วนสำคัญในการบันทึกความเคลื่อนไหว ในการบอกตำแหน่ง ของการเคลื่อนไหวตามส่วนต่าง ๆ

ภาพที่ 2.5: ตัวอย่างโค้ดส่วน Body ของไฟล์บันทึกความเคลื่อนไหวรูปแบบ BVH

```

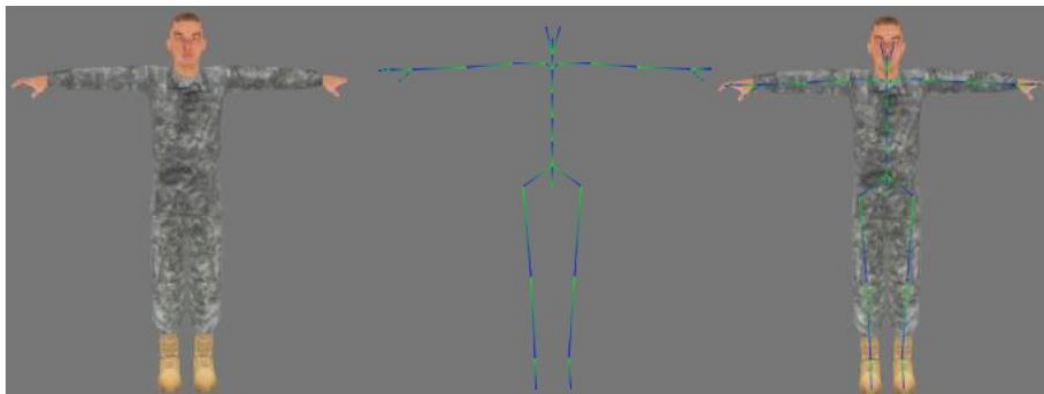
MOTION
Frames: 653
Frame Time: 0.033333333
0.010345 34.729090 0.833995 -2.011510 -0.789648 6.904627 0.300505 6.938093 6.294344
0.019510 34.709489 0.832247 -1.936929 -0.658205 7.120351 0.193165 6.828914 6.287553
-0.053442 34.629365 0.883424 -2.601870 -0.698811 6.810140 0.796100 6.678196 6.357690
-0.159605 34.549199 0.956530 -3.539522 -0.883551 6.251682 1.703863 6.599179 6.461394
-0.184719 34.550149 0.939362 -3.702774 -0.700956 6.218389 1.872036 6.413827 6.492455
-0.149785 34.576969 0.815703 -3.319940 0.161257 6.580933 1.435449 5.805309 6.477915
-0.104799 34.479727 0.673946 -2.723958 1.739070 6.686864 0.675289 4.468923 6.440978
-0.207228 34.432785 0.754104 -3.217537 1.392317 5.860462 1.067374 4.591159 6.492552
-0.183310 34.443926 0.749431 -2.952485 1.299278 6.003497 0.880509 4.653276 6.462714
-0.147189 34.439365 0.744721 -2.649856 1.259517 6.138421 0.711710 4.654959 6.438468
-0.142128 34.424449 0.740165 -2.651403 1.260172 6.032897 0.770421 4.629993 6.447950
-0.172582 34.405228 0.732831 -2.877269 1.348089 5.755015 0.932531 4.531355 6.475778
-0.195373 34.398136 0.725651 -2.980006 1.468486 5.500891 0.958394 4.424141 6.484501
-0.218449 34.408671 0.751088 -3.120900 1.315279 5.089189 1.096732 4.532299 6.499672
-0.246720 34.417304 0.792512 -3.381454 1.021527 4.663176 1.373553 4.695896 6.531378
.....

```

2.3.1 การจำลองการเคลื่อนไหวจากไฟล์ข้อมูลการบันทึกความเคลื่อนไหวแบบ BVH

ในการสร้างงานแอนิเมชัน การออกแบบการเคลื่อนไหวให้กับตัวละครมีความยากลำบากและกินเวลามาก จึงทำให้เกิดแนวคิดที่จะใช้การจำลองการเคลื่อนไหวของมนุษย์มาใส่ในตัวละครที่สร้างขึ้น ซึ่งทำให้เกิดความสะดวกในการออกแบบการเคลื่อนไหวให้กับตัวละครมากขึ้น ในงานวิจัยนี้จะแสดงผลแอนิเมชัน 3 มิติ จากไฟล์ข้อมูลบันทึกความเคลื่อนไหวแบบ BVH ที่อยู่บนเซิร์ฟเวอร์ จะใช้โครงกระดูกมาแมพกับตัวละครแอนิเมชันเพื่อให้มีความเคลื่อนไหวตามไฟล์บันทึกความเคลื่อนไหว แต่ตัวละครแอนิเมชันกับโครงกระดูกต้องมีขนาดแต่ละส่วนเท่า ๆ กัน ถ้าแตกต่างกันมากจะทำให้การเคลื่อนไหวคลาดเคลื่อน ดังภาพที่ 2.6 ภาพตัวอย่างละครสามมิติและโครงกระดูกที่ใช้ควบคุมการเคลื่อนไหว

ภาพที่ 2.6: ภาพตัวอย่างละครสามมิติและโครงกระดูกที่ใช้ควบคุมการเคลื่อนไหว



ระบบไฟล์ BVH มีการอ้างอิงตำแหน่งข้อต่อตามร่างกาย 18 จุดเพื่ออ้างอิงข้อต่อของตัวละคร การสร้างตัวละครขนาดของโครงกระดูกแต่ละท่อนจะถูกกำหนดไว้ในส่วนของ Header ของไฟล์ และเมื่อมีการเคลื่อนไหวข้อต่อส่วนใด ส่วนที่เป็น Children Nodes ของข้อต่อส่วนนั้นจะถูกพาไปด้วยเช่น หมุนหัวไหล่ข้างซ้ายไปให้อยู่ในลักษณะกางแขน ส่วนของข้อศอกและข้อมือก็จะถูกพาไปด้วย

2.4 เว็บ 3 มิติ (Web 3D)

2.4.1 เว็บสามมิติ คือ เว็บที่มีการแสดงผลภาพทางหน้าจอทั้งแนวแกน X, Y และ Z ซึ่งมีมุมมองเป็นแบบสามมิติสามารถเห็นความกว้าง ความยาว และความลึกของภาพได้ ทำให้มีมิติที่น่าสนใจและสมจริงซึ่งเหมาะสำหรับการแสดงผลในปัจจุบันที่ทุกอย่างต้องดูเสมือนจริงมากที่สุดจึงทำให้ เว็บสามมิตินั้นได้รับความนิยมเป็นอย่างมาก เว็บไซต์แบบ 3 มิติ ซึ่งผู้ใช้งานจะสามารถมองเห็นได้หลายมุมมองสามารถหมุนมุมมองได้ จึงทำให้เว็บไซต์นั้น ๆ มีความน่าสนใจและมีมิติการนำเสนอที่มากขึ้น

2.4.2 เว็บจีแอล (WebGL: Web Graphics Library)

เว็บจีแอล (WebGL) หรือไลบรารี กราฟิกส์บนเว็บ คือ รูปแบบการใช้งาน canvas ของ HTML เพื่อใช้งานเอพีไอสำหรับกราฟิกส์สามมิติสำหรับเว็บเบราว์เซอร์โดยไม่จำเป็นต้องติดตั้งปลั๊กอินส์เพิ่ม สเป็กของเว็บจีแอลยังอยู่ในขั้นตอนการตราฟต์ โดยบริหารและจัดการโดยกลุ่มโครโนส เว็บจีแอลมีพื้นฐานอยู่บน OpenGL ES 2.0 และให้อินเตอร์เฟซสำหรับงานกราฟิกส์ 3 มิติ โดยการทำงานของ HTML5 canvas element และการเข้าถึงอินเตอร์เฟซของ Document Object Model Interfaces ตัวอย่างเว็บไซต์ที่ใช้ ไลบรารี กราฟิกส์บนเว็บหรือที่เรียกว่า WebGL ซึ่งเป็น

ไลบรารีที่ช่วยในการแสดงผลกราฟิกแบบสามมิติบนเว็บไซต์ ดังภาพที่ 2.7 เป็นภาพตัวอย่างการใช้ไลบรารี กราฟิกบนเว็บหรือ WebGL ซึ่งทำให้วัตถุที่อยู่บนเว็บมีการเคลื่อนไหว

ภาพที่ 2.7: เว็บจีแอล (WebGL: Web Graphics Library)



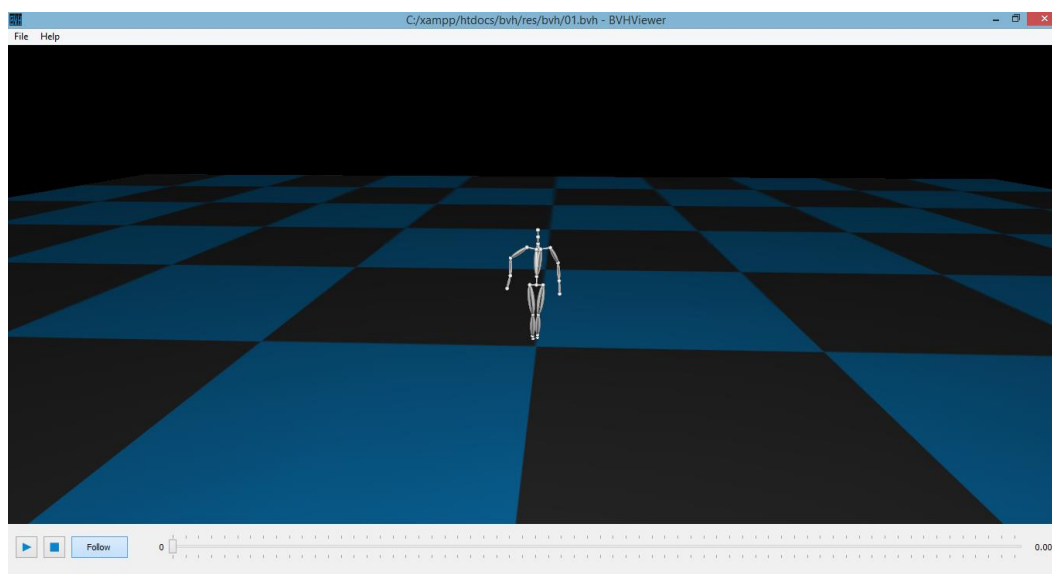
2.5 โปรแกรมสำหรับพัฒนาที่ใช้ในการอ่านไฟล์ข้อมูลบันทึกความเคลื่อนไหวและโปรแกรมที่ใช้ในการแสดงผลบนเว็บไซต์แบบสามมิติ

ภาษาของโปรแกรมที่สามารถอ่านไฟล์ข้อมูลบันทึกความเคลื่อนไหวมีอยู่ด้วยกันหลายภาษา เช่น C++, JAVA, JavaScript และ Python เป็นต้น ซึ่งในงานวิจัยนี้จะยกตัวอย่างวิธีการอ่านไฟล์บันทึกความเคลื่อนไหวของภาษา C++ และ JavaScript

2.5.1 โปรแกรมที่ใช้พัฒนาระบบ วิธีการอ่านไฟล์ที่บันทึกข้อมูลความเคลื่อนไหวและการแสดงผลในภาษา C++ โปรแกรมมีชื่อว่า Bloodshed Dev-C++ Version 4.9.9.2. ซึ่งเป็นโปรแกรมที่ออกแบบมาเพื่อช่วยให้ผู้ที่ทำการเขียนโปรแกรมใช้ในการสร้างโปรแกรม โดยจะมี อีดีเตอร์ (Editor) สำหรับเขียนโค้ดของโปรแกรมและมีตัวแปลภาษามาให้พร้อม ปัจจุบันมีการออกชุดพัฒนามาหลายรุ่น และเปลี่ยนแปลงรวดเร็วมาก เช่น Microsoft C/C++, Microsoft Visual C# และ Microsoft Visual C++ .NET เป็นต้น ซึ่งชุดพัฒนาแต่ละตัวมีวิธีการนำไปใช้งานที่แตกต่างกัน อย่างไรก็ตามการเขียนโปรแกรมภาษาซีไม่ว่าจะเป็น IDE ใด ก็มีหลักการและวิธีการในการเขียนที่คล้ายคลึงกันจะต่างกันบ้างตรงรายละเอียดบางอย่างที่เพิ่มขึ้น หรือพัฒนาให้ง่ายในการเขียนโปรแกรมโดย Bloodshed Dev-C++ เวอร์ชัน 4.9.9.2 ซึ่งสามารถเขียนได้ทั้งภาษา C และภาษา C++ เป็นชุดพัฒนาขึ้นมาเพื่อใช้เป็นฟรีแวร์และทำงานภายใต้ระบบปฏิบัติการ Windows และใช้ได้กับ Windows

ทุกรุ่น โปรแกรมสำหรับการแสดงผลแอนิเมชันในภาษา C++ จะต้องดาวน์โหลดซอฟต์แวร์ก่อนที่จะเลือกไฟล์บันทึกความเคลื่อนไหว ดังภาพที่ 2.8 เป็นโปรแกรมสำหรับการแสดงผลจากไฟล์บันทึกการเคลื่อนไหวโดยใช้การเขียนโปรแกรมในภาษา C++

ภาพที่ 2.8: ตัวอย่างโปรแกรมสำหรับการแสดงผลจากไฟล์บันทึกการเคลื่อนไหว ในภาษา C++



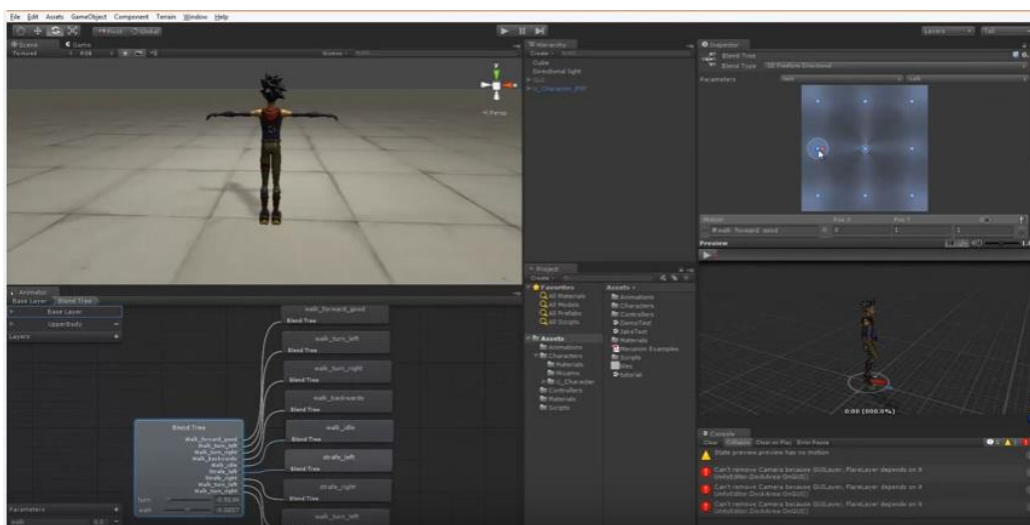
2.5.2 โปรแกรมที่ใช้ในพัฒนาระบบการแสดงผลผ่านเว็บไซต์ 3 มิติ โดยตัวโปรแกรมนั้นจะสามารถ Export ออกมาเป็นไฟล์ HTML5 ที่มีไลบรารี กราฟิกบนเว็บเป็นส่วนสำคัญในการแสดงผล ได้แก่ Unity 3D, Flash และ Netbeans เป็นต้น ซึ่งในบทนี้จะกล่าวถึงที่มาและความสำคัญของแต่ละโปรแกรม เหตุผลที่เลือกโปรแกรมในการนำมาใช้พัฒนาระบบอ่านไฟล์ตัวจับความเคลื่อนไหว

1) โปรแกรม Unity 3D คือเกมเอนจินสำหรับการสร้างเกม ที่ถูกพัฒนาขึ้นมาอย่างต่อเนื่อง ตั้งแต่ในช่วงแรก ๆ ที่รองรับการพอร์ตเกมลงบน Windows, OS X และเว็บเท่านั้น แต่ปัจจุบันได้เพิ่มความสามารถในการพอร์ตลงบน iOS, Android และแพลตฟอร์มอื่น ๆ เกือบทุกแพลตฟอร์มปัจจุบันโปรแกรม Unity สามารถทำงานได้ทั้งบน Windows และ OS X ซึ่งมีเวอร์ชันฟรีให้ใช้ด้วย

การพัฒนาเกม 3 มิติ มีเกมเอนจินสำหรับพัฒนาเกม 3 มิติ และช่วยอำนวยความสะดวกให้กับนักพัฒนาเป็นอย่างมาก เช่น Dark Basic, Torque Game Engine เป็นต้น การใช้งานเอนจินต่าง ๆ และพบว่า เอนจินที่เหมาะสมกับผู้เริ่มต้นและเหมาะที่จะใช้ในการเรียนการสอนมากที่สุดคือ Unity หน้าจอการทำงานของโปรแกรม Unity 3D ซึ่งในส่วนการออกแบบหน้าจอจะทำได้ง่ายเนื่องจากมีเครื่องมือสำหรับการออกแบบทุกอย่างเหมาะสมสำหรับการออกแบบเกมส์เป็นอย่างมาก ใน

ส่วนเบื้องหลังจะเป็นการเขียนโค้ดโดยใช้ภาษา C# ในการทำงานของแต่ละฟังก์ชัน ทีมพัฒนาทั้งขนาดใหญ่และขนาดเล็ก ให้ความสนใจกับ Unity 3D เพราะเป็นตัวช่วยให้สามารถสร้างเกมได้อย่างรวดเร็ว และคุ้มค่านั่นเอง ดังภาพที่ 2.9 ภาพตัวอย่างการนำไฟล์บันทึกความเคลื่อนไหวมาใส่ในตัวละครโดยใช้โปรแกรม Unity 3D

ภาพที่ 2.9: ตัวอย่างการนำไฟล์บันทึกความเคลื่อนไหวมาใส่ในตัวละครโดยใช้โปรแกรม Unity 3D



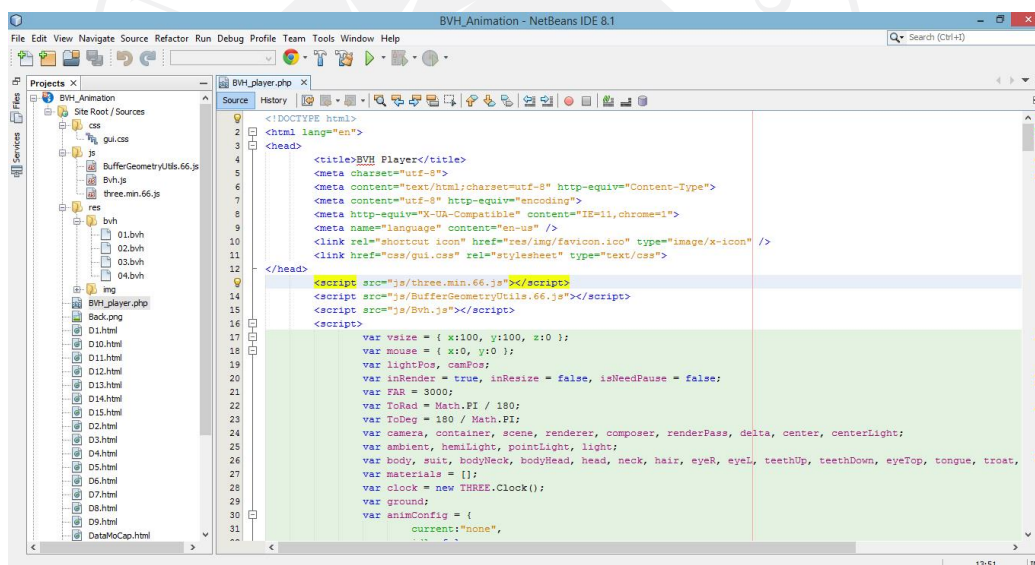
2) โปรแกรม Flash Builder 4.6 เป็นซอฟต์แวร์ที่ช่วยในการสร้างสื่อมัลติมีเดีย ภาพเคลื่อนไหว (Animation) ภาพกราฟิกที่มีความคมชัดเนื่องจากเป็นกราฟิกแบบเวกเตอร์ (Vector) สามารถเล่นเสียงและวิดีโอแบบสตรีมได้สามารถสร้างงานให้โต้ตอบกับผู้ใช้ (Interactive Multimedia) มีฟังก์ชันสำหรับการเขียนโปรแกรม (Action Script) ทำงานในลักษณะ CGI โดยเชื่อมต่อการเขียนโปรแกรมภาษาอื่น ๆ ได้มากมาย เช่น ภาษา PHP, JSP, ASP, ASP.NET, C/C++, C# และ JAVA เป็นต้น โดยเฉพาะข้อดีของโปรแกรม Flash คือ ความสามารถในการบีบอัดไฟล์ให้มีขนาดเล็ก มีผลทำให้แสดงผลได้อย่างรวดเร็ว นอกจากนั้นยังแปลงไฟล์ไปอยู่ในฟอร์แมตอื่นได้ เช่น avi, gif, wav, bmp, jpg, gif และ png เป็นต้น

โปรแกรม Flash เริ่มมีชื่อเสียงประมาณปี พ.ศ. 2539 จนถึง ปัจจุบันได้ถูกนำมาใช้งานอย่างแพร่หลาย โดยเฉพาะเทคโนโลยีเว็บ ทำให้การนำเสนอทำได้ที่น่าสนใจ นอกจากนั้น โปรแกรม Flash ยังสามารถสร้างแอปพลิเคชัน (Application) เพื่อใช้ทำงานต่าง ๆ รองรับการใช้งานกับอุปกรณ์ที่ เชื่อมต่อกับระบบเครือข่ายอินเทอร์เน็ต และทำงานได้กับหลาย ๆ แพลตฟอร์ม (Platform)

3) โปรแกรม Netbeans คือโปรแกรมสำหรับพัฒนาแอปพลิเคชันด้วยภาษาจาวา เป็นเครื่องมือสำหรับโปรแกรมเมอร์ที่จะใช้พัฒนา Application ด้วยภาษา Java โปรแกรม NetBeans นั้นเป็นโปรแกรมประเภท OpenSource software โดยผู้ใช้งานไม่จำเป็นต้องเสียเงินเพื่อซื้อมาใช้งาน

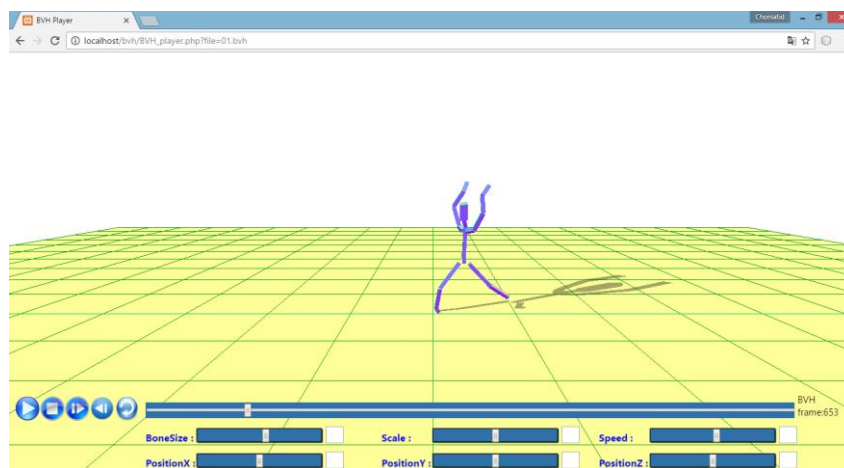
ปัจจุบัน NetBeans ได้รับความนิยมมากยิ่งขึ้น ได้รับการพัฒนาให้มีความสามารถเพิ่มขึ้น สามารถพัฒนาโปรแกรมโดยใช้ภาษาอื่น ๆ ได้อีกหลากหลายโดยติดตั้งโปรแกรมเสริม(Add-on)ได้จาก เว็บไซต์ หรือผ่านตัวอัปเดตเซนเตอร์ (Update Center) ของ NetBeans เช่น (C/C++), Web Application, Java EE, Mobility(Java ME), Java FX, Java Script และ PHP เป็นต้น ในเวอร์ชัน 6.0 เป็นต้นไปมีการรวมโปรแกรมเสริมต่าง ๆ ที่สำคัญเข้าในตัวติดตั้งของ NetBeans โดยสามารถเลือกติดตั้งได้ภายหลัง

ภาพที่ 2.10: หน้าจอการทำงานของโปรแกรม NetBeans IDE 8.1



ข้อดีของโปรแกรมนี้อีกคือ โปรแกรม NetBeans นั้นทำงานแยกส่วนต่าง ๆ ออกจากกันเป็น Module จึงทำให้สามารถนำ Module ต่าง ๆ ที่มีผู้ที่ได้พัฒนาต่อเติมมาติดตั้งเพิ่มเติมในภายหลังได้ ใช้งานได้ทั้งบนระบบปฏิบัติการ Windows และ Mac OS X การที่จะแสดงผลเป็นเว็บเบราว์เซอร์ Chrome ต้องติดตั้ง Netbeans Connector ก่อนจึงจะสามารถแสดงผลสามมิติได้

ภาพที่ 2.11: ตัวอย่างโปรแกรมการแสดงผลแอนิเมชันที่พัฒนาโดยใช้ NetBeans IDE 8.1



จากภาพที่ 2.31: เป็นตัวอย่างโปรแกรมการแสดงผลแอนิเมชันจากไฟล์บันทึกความเคลื่อนไหวบนเซิร์ฟเวอร์ที่พัฒนาโดยใช้โปรแกรม NetBeans IDE 8.1 ซึ่งในงานวิจัยชิ้นนี้ก็ใช้โปรแกรม NetBeans ในการพัฒนาโปรแกรมการแสดงผลแอนิเมชันจากฐานข้อมูลบันทึกความเคลื่อนไหวบนเซิร์ฟเวอร์เป็นหลัก เหตุผลที่เลือก NetBeans IDE 8.1 เพราะเป็นโปรแกรมที่สามารถแสดงผลผ่านเว็บไซต์ ซึ่งสามารถรองรับการทำงานของ WebGL จึงทำให้เว็บไซต์ที่แสดงผลออกมาเป็นแบบสามมิติ ตรงกับความต้องการของผู้พัฒนา ส่วนภาษาที่ใช้ในการพัฒนาจะมี HTML เป็นตัวแสดงผลผ่านเว็บไซต์ ภาษา CSS สำหรับการตกแต่งข้อความ หน้าอินเตอร์เฟซของโปรแกรม และภาษา JavaScript สำหรับฟังก์ชันการอ่านไฟล์บันทึกความเคลื่อนไหว การสร้างโครงกระดูก และการเคลื่อนไหวของตัวละคร

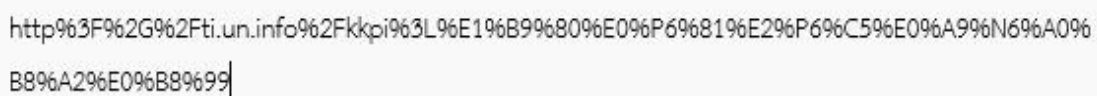
2.6 ความปลอดภัยของข้อมูล

ในงานวิจัยชิ้นนี้ได้ทำระบบแสดงผลแอนิเมชันจากไฟล์บันทึกความเคลื่อนไหวที่อยู่บนเซิร์ฟเวอร์ ทำให้ข้อมูลของระบบทั้งหมดสามารถถูกเข้าถึงได้จากบุคคลภายนอก จึงต้องมีการวิเคราะห์ระบบการรักษาความปลอดภัยของข้อมูล เนื่องจากระบบทั้งหมดถูกพัฒนาโดยใช้ภาษา JavaScript ซึ่งเป็นแบบ Client Side Script จึงทำให้ผู้ใช้งานสามารถดู Source Code ของระบบและรู้ที่อยู่ของไฟล์บันทึกความเคลื่อนไหวได้ จากปัญหาดังกล่าวผู้พัฒนาจึงใช้วิธีการเข้ารหัส URL เพื่อไม่ให้ผู้เข้ามาใช้งานระบบสามารถเข้าถึงไฟล์บันทึกความเคลื่อนไหว

2.6.1 การเข้ารหัส URL

การเข้ารหัส URL คือการปกปิดข้อมูลอย่างหนึ่งของระบบซึ่งจะทำให้ URL อ่านได้ยากกว่าเดิม เป็นวิธีที่นิยมใช้ เพื่อป้องกันการเข้าถึงข้อมูล ซึ่งจะสามารถป้องกันการเข้าถึงข้อมูลได้ในระดับหนึ่ง ลักษณะของ URL ที่ถูกเข้ารหัสแล้ว จะมีข้อความบางส่วนกลายเป็นเครื่องหมายเปอร์เซ็นต์ และ ตัวเลขผสมอักษร 2 หลักแทรกอยู่ (Cheng et al., 2010) ดังภาพที่ 2.12 เป็นตัวอย่าง URL ที่ถูกเข้ารหัสไว้

ภาพที่ 2.12: ตัวอย่าง URL ที่ถูกเข้ารหัสแล้ว



http%3F%2G%2Fti.un.info%2Fkpi%3L%E1%B9%80%E0%P6%81%E2%F6%5E0%A9%N6%A0%B8%A2%E0%B8%99

บทที่ 3

วิธีการดำเนินงานการวิจัย

3.1 ขั้นตอนการดำเนินงานวิจัย

การวิจัยนี้ทำเพื่อสร้างการแสดงผลแอนิเมชันจากไฟล์ข้อมูลที่บันทึกความเคลื่อนไหวบนเซิร์ฟเวอร์และการวิเคราะห์ความปลอดภัยของข้อมูล เพื่อแก้ปัญหาการแสดงผลแอนิเมชันที่มีในปัจจุบันซึ่งต้องทำการโหลดซอฟต์แวร์มาติดตั้งภายในเครื่องก่อนและโหลดไฟล์บันทึกความเคลื่อนไหวมาใช้ในการแสดงผล ในงานวิจัยนี้ได้ทำเพื่อให้มีความสะดวกในการใช้งาน น่าเชื่อถือและแสดงผลได้รวดเร็วมากยิ่งขึ้นโดยมีขั้นตอนการดำเนินการบนแพลตฟอร์ม Web 3D ดังนี้

3.1.1 การรวบรวมข้อมูล

ศึกษาและค้นคว้างานที่เกี่ยวข้องกับเทคโนโลยีการจับภาพเคลื่อนไหว การแสดงผลแบบสามมิติบน Web Browser ที่สามารถใช้งานได้กับ WebGL และการใช้โปรแกรมในการสร้างระบบการแสดงผลแอนิเมชันจากไฟล์บันทึกความเคลื่อนไหว

3.1.2 การออกแบบ

การออกแบบระบบสำหรับใช้แสดงผลของไฟล์บันทึกความเคลื่อนไหวรูปแบบ BVH บน Web Browser ซึ่งสามารถอ่านไฟล์บันทึกความเคลื่อนไหวรูปแบบ BVH และแสดงการเคลื่อนไหวผ่านตัวละครแอนิเมชันสามมิติ ที่ได้เตรียมไว้ โดยไฟล์ทั้งหมดอยู่บนเซิร์ฟเวอร์ และออกแบบหน้าจอสำหรับการแสดงผลแบบสามมิติ

3.1.3 การพัฒนาระบบ

พัฒนาระบบโดยการเขียนโปรแกรมด้วยภาษา Java Script, Html 5 และ CSS ผ่านโปรแกรม Netbeans ซึ่งเป็นโปรแกรมที่สามารถทำงานได้บน Web Browser อ่านไฟล์บันทึกความเคลื่อนไหวรูปแบบ BVH แล้วแปลงข้อมูลที่ได้มาใส่ในตัวละครแอนิเมชันสามมิติที่เตรียมไว้ มีการควบคุมการเล่นของตัวละครได้

3.1.4 การทดสอบ

การทดสอบจะทดสอบโดยใช้ไฟล์บันทึกความเคลื่อนไหวรูปแบบ BVH ที่อยู่บนเซิร์ฟเวอร์เป็นตัวทดลอง ซึ่งระบบต้องสามารถอ่านไฟล์บันทึกความเคลื่อนไหวรูปแบบ BVH ได้และตัวละครแอนิเมชันต้องมีการเคลื่อนไหวที่ราบรื่นสมจริงตามไฟล์ต้นฉบับและทดสอบความปลอดภัยการเข้าถึงไฟล์บันทึกความเคลื่อนไหวที่อยู่บนเซิร์ฟเวอร์

3.2 เครื่องมือที่ใช้ในการศึกษา

3.2.1 ซอฟต์แวร์ที่ใช้ในการพัฒนา

- | | |
|---------------------------------|----------------|
| 1) NetBeans IDE | Version: 8.1 |
| 2) XAMPP Control Panel (Apache) | Version: 3.2.2 |

เหตุผลที่เลือกใช้ NetBeans เพราะเป็นโปรแกรมที่พัฒนาแอปพลิเคชันด้วยภาษาจาวา และยังสามารถพัฒนาภาษาอื่น ๆ ได้อีกหลากหลายโดยติดตั้งโปรแกรมเสริม (Add-on) ได้จากเว็บไซต์ หรือผ่านตัวอัปเดตเซนเตอร์ (Update Center) ของ NetBeans เช่น Java Script, PHP และ HTML5 เป็นต้น ซึ่งในงานวิจัยชิ้นนี้ ต้องการแสดงผลแอนิเมชันจากไฟล์บันทึกความเคลื่อนไหวที่อยู่บนเซิร์ฟเวอร์ จึงจำเป็นต้องใช้ภาษาที่สามารถทำงานบนเว็บสามมิติได้

3.2.3 เว็บเบราว์เซอร์ที่ใช้ทดสอบ

- | | |
|----------------------|-------------|
| 1) Internet Explorer | Version: 11 |
| 2) Firefox | Version: 45 |
| 3) Chrome | Version: 49 |

เว็บเบราว์เซอร์ที่ใช้ในการทดสอบการวิจัยนี้เป็นเว็บที่สามารถรองรับการทำงานของ WebGL ได้เท่านั้น ซึ่งเวอร์ชันที่ได้กล่าวถึงด้านบนเป็นเวอร์ชันขั้นต่ำที่สามารถรองรับการทำงานของ WebGL และสามารถแสดงผลแบบสามมิติได้

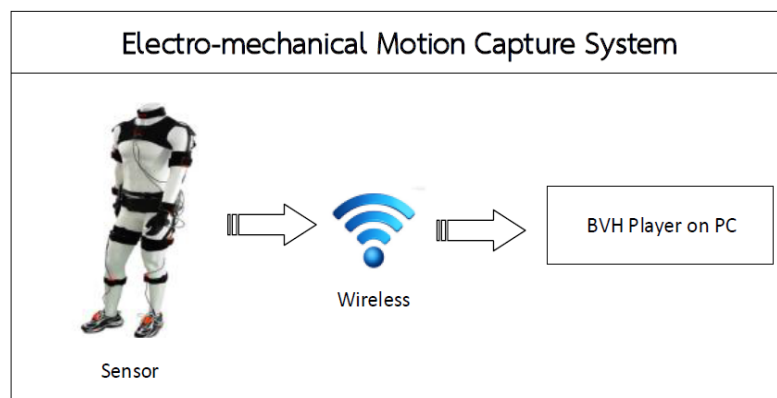
3.3 สถาปัตยกรรมของระบบ

สถาปัตยกรรมระบบอธิบายตั้งแต่การสร้างไฟล์บันทึกความเคลื่อนไหวแบบ BVH ไปจนถึงการแสดงผลผ่านทางเว็บเบราว์เซอร์โดยไฟล์บันทึกความเคลื่อนไหวแบบ BVH ที่ระบบอ่านเป็นไฟล์ที่อยู่บนเซิร์ฟเวอร์

3.3.1 วิธีการสร้างไฟล์บันทึกความเคลื่อนไหวแบบ BVH

โดยการสร้างไฟล์บันทึกความเคลื่อนไหวแบบ BVH นั้นใช้นักแสดงมาสวมชุดที่มีเซ็นเซอร์ติดไว้ตามตัว ซึ่งเซ็นเซอร์ที่ติดไว้ทำการส่งสัญญาณผ่าน WiFi ไปที่เครื่องคอมพิวเตอร์ที่มีโปรแกรมอัดวิดีโอสำหรับการบันทึกการเคลื่อนไหวของนักแสดง จากนั้นก็ได้ไฟล์บันทึกความเคลื่อนไหวแบบ BVH ดังภาพที่ 3.1 ได้อธิบายถึงวิธีการสร้างไฟล์บันทึกความเคลื่อนไหวแบบ BVH

ภาพที่ 3.1: วิธีการสร้างไฟล์บันทึกความเคลื่อนไหวแบบ BVH

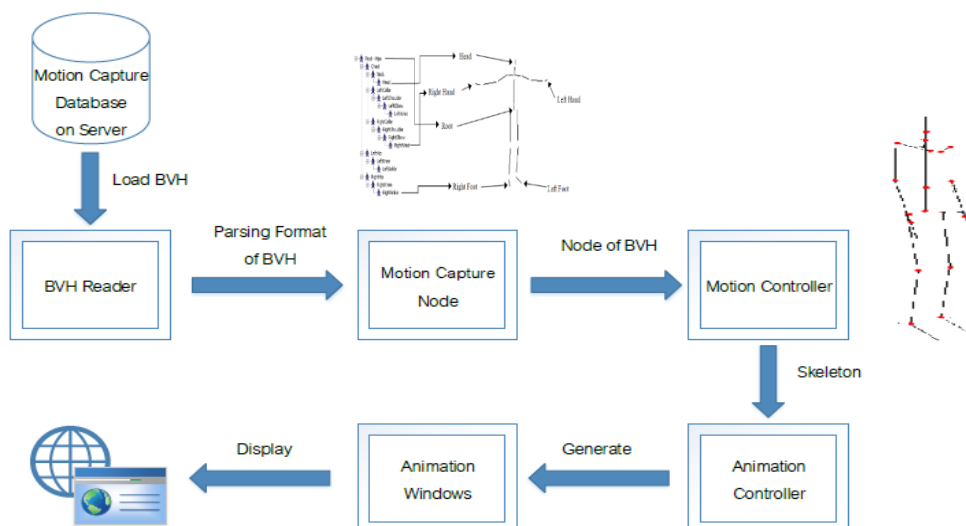


3.3.2 ขั้นตอนการสร้างระบบแสดงผลแอนิเมชันผ่านเว็บสามมิติจากไฟล์ฐานข้อมูลบันทึกความเคลื่อนไหวที่อยู่บนเซิร์ฟเวอร์ ซึ่งมีขั้นตอนดังนี้

- 1) นำไฟล์บันทึกความเคลื่อนไหวแบบ BVH ที่มีอยู่ ไปเก็บไว้บนเซิร์ฟเวอร์ที่ได้จัดเตรียมไว้
- 2) สร้างเว็บไซต์ที่สามารถแสดงผลแอนิเมชันแบบสามมิติได้ โดยใช้ภาษา HTML5 และ CSS ในการพัฒนา
- 3) สร้างระบบที่อ่านไฟล์บันทึกความเคลื่อนไหวแบบ BVH โดยใช้ภาษา Java Script ในการพัฒนา
- 4) นำค่าที่ได้จากการอ่านไฟล์บันทึกความเคลื่อนไหวมาสร้างตัวละครแอนิเมชันและแมพการเคลื่อนไหวทำทางให้ตรงกับไฟล์
- 5) นำตัวละครแอนิเมชันที่ได้เตรียมไว้มาแสดงผลที่เว็บไซต์แบบสามมิติ
- 6) สร้างตัวควบคุมการเล่น ตำแหน่งการแสดงผล และขนาดของตัวละครแอนิเมชัน

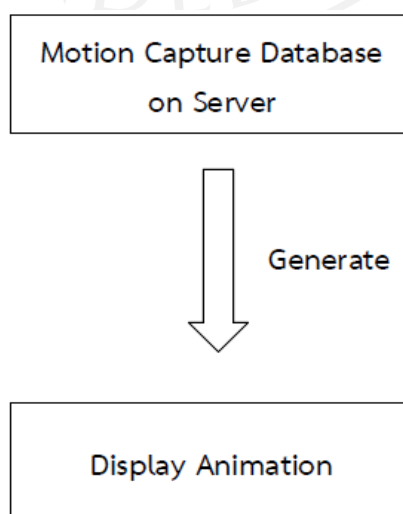
3.3.3 การอธิบายโครงสร้างระบบการแสดงผลแอนิเมชันจากไฟล์ฐานข้อมูลการตรวจจับความเคลื่อนไหวที่อยู่บนเซิร์ฟเวอร์ซึ่งได้อธิบายขั้นตอนตั้งแต่การสร้างตัวอ่านไฟล์ BVH จากนั้นนำรูปแบบที่ได้ส่งไปที่ Motion Capture Node นำ Node แต่ละ Node ไปที่ตัวควบคุมการเคลื่อนไหวหรือ Motion Controller จากนั้นได้โครงกระดูกมาใส่การควบคุมแอนิเมชันสุดท้ายก็สร้าง Animation Windows หรือ เว็บสามมิติที่ใช้ในการแสดงผล ดังภาพที่ 3.2 เป็นภาพ Flow Chart ของระบบแสดงผลแอนิเมชันจากไฟล์บันทึกความเคลื่อนไหวที่อยู่บนเซิร์ฟเวอร์

ภาพที่ 3.2: Flow Chart ของระบบแสดงผลแอนิเมชันจากไฟล์บันทึกความเคลื่อนไหว



3.3.4 อธิบายโครงสร้างการทำงานของผู้ใช้ระบบแสดงผลอนิเมชันจากไฟล์บันทึกความเคลื่อนไหว แบ่งการทำงานออกเป็น 2 ส่วน 1) ผู้ใช้งานต้องเลือกไฟล์บันทึกความเคลื่อนไหวจากฐานข้อมูลบันทึกความเคลื่อนไหวที่อยู่บนเซิร์ฟเวอร์ เพื่อใช้ในการแสดงผลแอนิเมชัน 2) ส่วนของการแสดงผลอนิเมชันจากไฟล์บันทึกความเคลื่อนไหวที่เลือกมา ดังภาพที่ 3.3 Flow Chart การใช้งานระบบแสดงผลอนิเมชันจากไฟล์บันทึกความเคลื่อนไหว

ภาพที่ 3.3: Flow Chart การใช้งานระบบแสดงผลอนิเมชันจากไฟล์บันทึกความเคลื่อนไหว



3.4 การออกแบบหน้าจอการใช้งาน

หน้าจอการใช้งานของระบบหลัก ๆ มีอยู่ด้วยกันทั้งหมด 2 หน้าดังนี้

3.4.1 หน้าจอ: ฐานข้อมูลการตรวจจับความเคลื่อนไหว ประกอบด้วย 1 ส่วน คือ ตารางฐานข้อมูลการตรวจจับความเคลื่อนไหว ซึ่งมี รหัสไฟล์ ชื่อไฟล์ ประเภท ปุ่มกดแสดงผลการเคลื่อนไหว คำอธิบาย และปุ่มสำหรับเอาไว้อ่านไฟล์

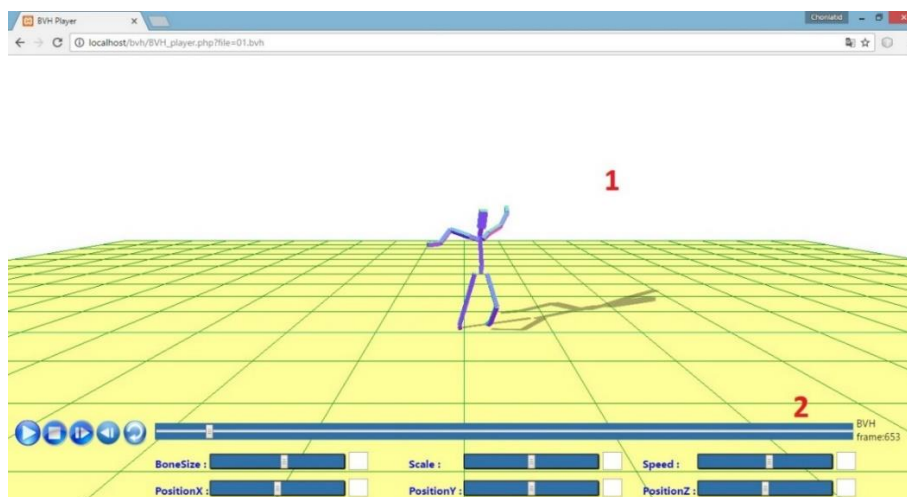
ภาพที่ 3.4: ฐานข้อมูลการตรวจจับความเคลื่อนไหว

Motion ID	Name	Category	BVH	Description	Download
M001	Tai Chi	dance	⬇	Description	⬇
M002	Krabi	dance	⬇	Description	⬇
M003	Swing and Hang	action	⬇	Description	⬇
M004	Sit and Stand	action	⬇	Description	⬇
M005	Walk	action	⬇	Description	⬇
M006	Run	action	⬇	Description	⬇
M007	jump	action	⬇	Description	⬇
M008	punch/strike	action	⬇	Description	⬇
M009	bend over	action	⬇	Description	⬇
M010	swordplay	action	⬇	Description	⬇
M011	wash self	action	⬇	Description	⬇
M012	expressive arms	dance	⬇	Description	⬇
M013	sideways arabesque	dance	⬇	Description	⬇
M014	sideways arabesque	dance	⬇	Description	⬇
M015	quasi-cou-de-pied	dance	⬇	Description	⬇

3.4.2 หน้าจอ: การแสดงผลตัวละครแอนิเมชัน 3 มิติ ประกอบด้วย 2 ส่วน คือ

- 1) ส่วนแสดงผลตัวละครแอนิเมชัน 3 มิติ
- 2) ส่วนควบคุมการเล่นและการแสดงผลของตัวละครแอนิเมชัน 3 มิติ มีปุ่มควบคุมการเล่นทั้งหมด 5 ปุ่มได้แก่ (1) ปุ่มเล่น (2) ปุ่มหยุด (3) ปุ่มเดินหน้าที่ละเฟรม (4) ปุ่มถอยหลังทีละเฟรม (5) ปุ่มย้อนกลับ และบาร์ควบคุมการแสดงผลทั้งหมด 7 บาร์ ได้แก่ (1) ตัวควบคุมขนาดของกระดูก (2) ตัวควบคุมสเกล (3) ตัวควบคุมความเร็ว (4) ตัวควบคุมตำแหน่งในแกน X (5) ตัวควบคุมตำแหน่งในแนวแกน Y (6) ตัวควบคุมตำแหน่งในแนวแกน Z (7) ตัวควบคุมการเล่นการเคลื่อนไหว

ภาพที่ 3.5: หน้าจอแสดงผลแอนิเมชันจากไฟล์การตรวจจับความเคลื่อนไหว



จากภาพที่ 3.5 หน้าจอการแสดงผลแอนิเมชัน 3 มิติจากไฟล์การตรวจจับความเคลื่อนไหว คือหน้าที่มาจากการกดปุ่ม Play ของหน้าฐานข้อมูลการตรวจจับความเคลื่อนไหวซึ่งในหน้านั้นแสดง การเคลื่อนไหวตามไฟล์ที่เลือก

3.5 การป้องกันความปลอดภัยของไฟล์จับความเคลื่อนไหว

การพัฒนาบบบนแพลตฟอร์ม Web 3D ซึ่งระบบการแสดงผลแอนิเมชันนี้จะทำงานอยู่บน เซิร์ฟเวอร์ ผู้พัฒนาจึงได้เข้ารหัส URL หรือที่เรียกว่าการ Encoding URL ที่อยู่ไฟล์จับความเคลื่อนไหว เพื่อป้องกันความปลอดภัยของข้อมูล ดังภาพที่ 3.6 เป็นโค้ดส่วนการเข้ารหัสที่อยู่ของไฟล์จับความเคลื่อนไหว แต่เนื่องจากการเป็นการทำงานแบบ Client Side Script คือการทำงานจากทางฝั่งของผู้ใช้งาน จึงไม่สามารถป้องกันข้อมูลได้ทั้งหมด

ภาพที่ 3.6: โค้ดส่วนการเข้ารหัสที่อยู่ไฟล์จับความเคลื่อนไหว

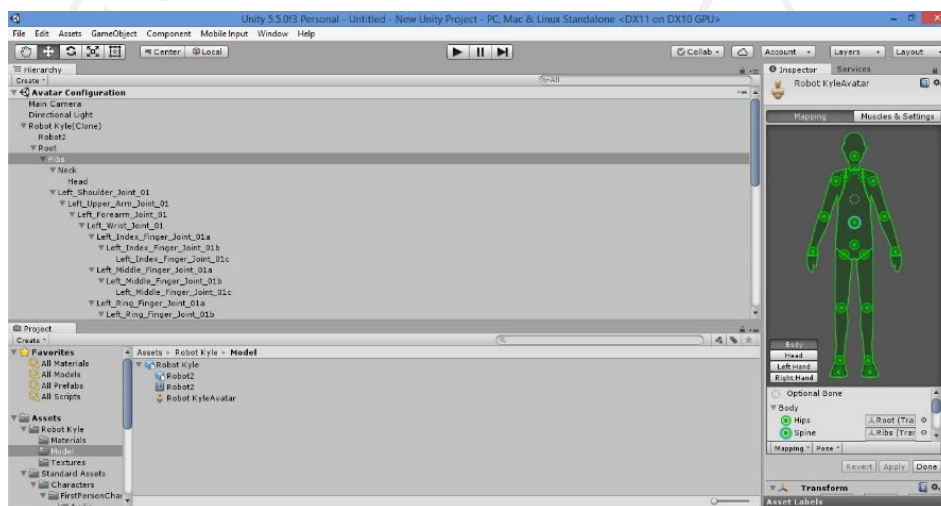
```
BVH.Reader.prototype = {
  constructor: BVH.Reader,
  _keyStr : "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/",
  load: function(fname){
    fname = this.decode("Ly9sb2NhbGhvc3QvYnZoL3Jlcy9idmgu") + fname;
    this.type = fname.substring(fname.length-3, fname.length);
  }
}
```

3.6 การพัฒนาระบบบนแพลตฟอร์ม Unity 3D

การพัฒนาระบบบนแพลตฟอร์ม Unity 3D จะมีการใช้เครื่องมือและเทคโนโลยีที่แตกต่างจากการพัฒนาบนแพลตฟอร์ม Web 3D และ Flash เนื่องจาก Unity 3D จะมีเครื่องมือที่รองรับการพัฒนาแบบสามมิติ เทคโนโลยีที่ใช้ในการพัฒนาบนแพลตฟอร์ม Unity 3D จะมีการอธิบายถึงวิธีการพัฒนาระบบและเครื่องมือที่ใช้ในการพัฒนาบนแพลตฟอร์ม Unity 3D ดังนี้

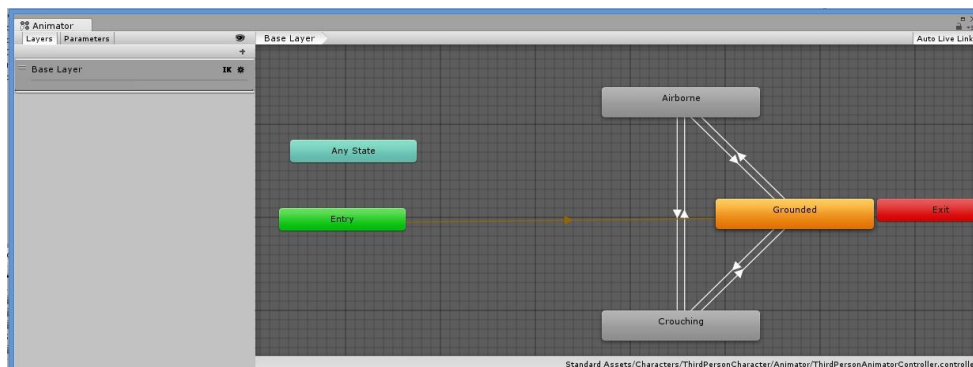
3.6.1 การตั้งค่าตัวละครให้เข้ากับไฟล์ข้อมูลบันทึกความเคลื่อนไหว ซึ่ง Unity 3D จะมีเครื่องมือที่เรียกว่า Avatar Configuration เอาไว้สำหรับกำหนดจุดต่าง ๆ ของตัวละครหรือที่เรียกว่าการ Mapping ในส่วนของข้อต่อต่าง ๆ ดังภาพที่ 3.7 เป็นการตั้งค่าในส่วนของข้อต่อต่าง ๆ ของตัวละครเพื่อให้เข้ากับไฟล์จับความเคลื่อนไหว

ภาพที่ 3.7: เครื่องมือ Avatar Configuration



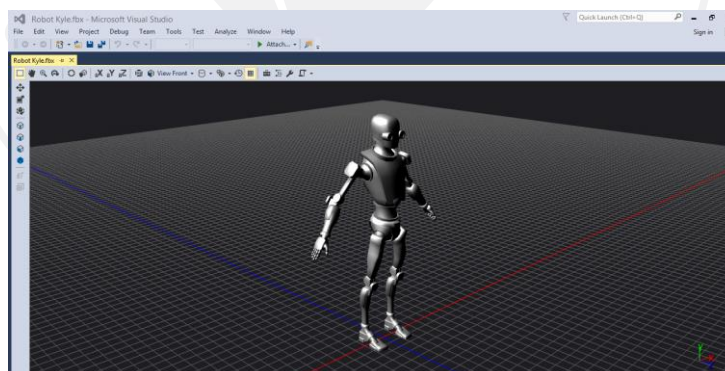
3.6.2 การใส่การเคลื่อนไหวให้กับตัวละคร Unity 3D จะมีเครื่องมือที่ชื่อว่า Animator เป็นการกำหนดการเคลื่อนไหวหรือเงื่อนไขต่าง ๆ ให้กับตัวละครได้ ซึ่งสามารถควบคุมการทำงานและเงื่อนไขต่าง ๆ ได้โดยผ่านภาษา C# ดังภาพที่ 3.8 เครื่องมือ Animator ที่ใช้ในการใส่การเคลื่อนไหวให้ตัวละคร

ภาพที่ 3.8: เครื่องมือ Animator ใช้ควบคุมการเคลื่อนไหว



3.6.3 การแสดงผลการเคลื่อนไหวและเครื่องมือที่ใช้ในการพัฒนาระบบแสดง Unity 3D จะมีการเชื่อมต่อกับ Visual Studio ซึ่งการที่จะพัฒนาระบบต่าง ๆ บนแพลตฟอร์ม Unity 3D จะต้องติดตั้งโปรแกรมเสริมที่ชื่อว่า Visual Studio เป็นโปรแกรมสำหรับการเขียนโค๊ดหรือคำสั่งต่าง ๆ เพื่อควบคุมการทำงานของ Unity 3D ดังภาพที่ 3.9 เป็นภาพโปรแกรม Visual Studio

ภาพที่ 3.9: โปรแกรม Visual Studio ที่ใช้ในการแสดงผล



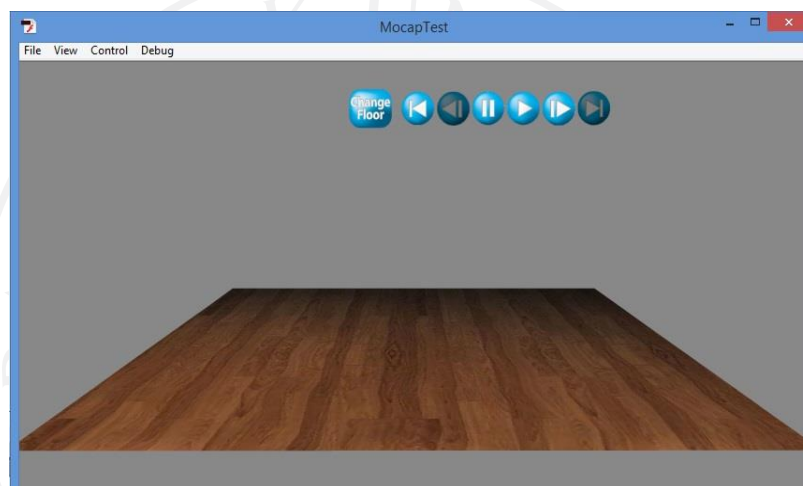
3.7 การพัฒนาระบบบนแพลตฟอร์ม Flash

การพัฒนาระบบบนแพลตฟอร์ม Flash จะมีการใช้เครื่องมือและเทคโนโลยีที่แตกต่างจากการพัฒนาบนแพลตฟอร์ม Web 3D และ Unity 3D เนื่องจาก Flash จะไม่มีเครื่องมือที่รองรับการพัฒนาแบบสามมิติ จึงต้องมีการใช้ไลบรารีที่ชื่อว่า Paper Vision 3D ซึ่งเป็น ไลบรารีที่เอาไว้สำหรับพัฒนาระบบที่เป็นสามมิติ เทคโนโลยีที่ใช้ในการพัฒนาบนแพลตฟอร์ม Flash จะมีการอธิบายถึงวิธีการพัฒนาระบบและเครื่องมือที่ใช้ในการพัฒนาบนแพลตฟอร์ม Flash ดังนี้

3.7.1 เทคโนโลยีที่ใช้และไลบรารีเสริมสำหรับการพัฒนาระบบแบบสามมิติบนแพลตฟอร์ม Flash จะใช้โปรแกรม Flash Builder ซึ่งเป็นโปรแกรมที่ใช้เขียนด้วยภาษา Action Script

3.7.2 เทคโนโลยีที่ใช้ในการออกแบบหน้าจอและแสดงผลการเคลื่อนไหวสามมิติ จะต้องติดตั้งโปรแกรม Flash Professional ซึ่งเป็นโปรแกรมที่ใช้ในการออกแบบหน้าจอและไลบรารีสำคัญที่ชื่อว่า paperVision3D ซึ่งเป็นไลบรารีสำหรับการแสดงผลสามมิติบนแพลตฟอร์ม Flash ดังภาพที่ 3.10 หน้าจอการแสดงผลบนแพลตฟอร์ม Flash

ภาพที่ 3.10: หน้าจอการแสดงผลบนแพลตฟอร์ม Flash



บทที่ 4

ผลการดำเนินงาน

หลังจากที่ได้สร้างระบบการแสดงผลการเคลื่อนไหวของตัวละครแอนิเมชัน 3 มิติ จากฐานข้อมูลการตรวจจับการเคลื่อนไหวบนเซิร์ฟเวอร์ในส่วนต่าง ๆ ตามแผนงานที่วางไว้ซึ่งสามารถอธิบายผลการดำเนินงานรายงานการวิจัยได้ดังนี้

4.1 การเริ่มต้นการใช้งานระบบ BVH Player

เมื่อเข้ามาที่เว็บไซต์ BVH Player หน้าแรกเป็นหน้าฐานข้อมูลการตรวจจับความเคลื่อนไหวซึ่งในหน้า Motion Capture Database เก็บรวบรวมไฟล์บันทึกความเคลื่อนไหวรูปแบบ BVH ไว้บนเซิร์ฟเวอร์ซึ่งมีไฟล์บันทึกความเคลื่อนไหวเก็บไว้และมีข้อมูลของไฟล์บันทึกความเคลื่อนไหวแสดงในตารางดังต่อไปนี้

- 4.1.1 รหัสไฟล์บันทึกความเคลื่อนไหว รูปแบบ BVH
- 4.1.2 ชื่อไฟล์บันทึกความเคลื่อนไหว รูปแบบ BVH
- 4.1.3 ประเภทของไฟล์บันทึกความเคลื่อนไหว รูปแบบ BVH
- 4.1.4 ปุ่มกดเล่น เพื่อไปหน้าการแสดงผล
- 4.1.5 ปุ่ม Description แสดงรายละเอียดของไฟล์ BVH
- 4.1.6 ปุ่มดาวน์โหลดสำหรับดาวน์โหลดไฟล์ BVH ออกมาไว้ที่เครื่องได้

ดังภาพที่ 4.1 ผู้ใช้สามารถดูชื่อไฟล์ ประเภทของไฟล์ รายละเอียดของไฟล์ สามารถกดแสดงผลแอนิเมชันเพื่อไปหน้าแสดงผลแอนิเมชัน 3 มิติและสามารถดาวน์โหลดไฟล์บันทึกความเคลื่อนไหวได้

ภาพที่ 4.1: หน้าจอฐานข้อมูลการตรวจจับความเคลื่อนไหว



The screenshot shows a web browser window displaying the 'Motion Capture Database'. The page features a table with the following columns: Motion ID, Name, Category, BVH, Description, and Download. The table lists 15 motion capture files, all categorized as 'dance'. Each row includes a play button icon in the BVH column and a download icon in the Download column.

Motion ID	Name	Category	BVH	Description	Download
M001	motion 1	dance		Description	
M002	motion 2	dance		Description	
M003	motion 3	dance		Description	
M004	motion 4	dance		Description	
M005	motion 5	dance		Description	
M006	motion 6	dance		Description	
M007	motion 7	dance		Description	
M008	motion 8	dance		Description	
M009	motion 9	dance		Description	
M010	motion 10	dance		Description	
M011	motion 11	dance		Description	
M012	motion 12	dance		Description	
M013	motion 13	dance		Description	
M014	motion 14	dance		Description	
M015	motion 15	dance		Description	

4.2 หน้าจอการแสดงผลตัวละครแอนิเมชัน 3 มิติ

หน้าจอการแสดงผลแบ่งเป็น 2 ส่วนหลัก ดังแสดงในภาพที่ 4.2 ประกอบด้วย 2 ส่วน ดังนี้

4.2.1 ส่วนของการควบคุมการแสดงผลแอนิเมชัน

ส่วนของการควบคุมการแสดงผลแอนิเมชันได้มีการออกแบบปุ่มและบาร์สำหรับควบคุมการมีปฏิสัมพันธ์ระหว่างผู้ใช้งานกับตัวละครแอนิเมชันดังต่อไปนี้

- 1) ปุ่ม Play เอาไว้สำหรับกดเล่นแอนิเมชัน
- 2) ปุ่ม Stop เอาไว้สำหรับกดหยุดแอนิเมชัน
- 3) ปุ่ม Next เอาไว้สำหรับกดเล่นที่ละเฟรม
- 4) ปุ่ม Previous เอาไว้สำหรับกดถอยหลังที่ละเฟรม
- 5) ปุ่ม Return เอาไว้สำหรับเริ่มต้นใหม่
- 6) Seek Bar เอาไว้สำหรับควบคุมการเล่น
- 7) BoneSize Bar สำหรับควบคุมขนาดของกระดูกได้
- 8) Scale Bar สำหรับควบคุมสเกลของตัวละครแอนิเมชันได้
- 9) Speed Bar สำหรับควบคุมความเร็วของการเล่นได้
- 10) PositionX Bar, PositionY Bar และ PositionZ Bar สำหรับย้ายตำแหน่งของ

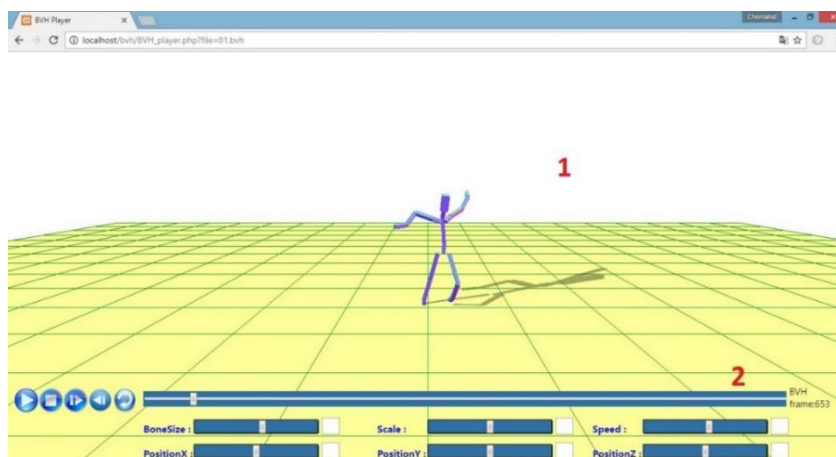
ตัวละครในแกน X, Y และ Z

การออกแบบจากที่กล่าวมานี้ผู้ใช้งานสามารถควบคุมการแสดงผลการเล่นโดยการกดปุ่มได้ ไม่ว่าจะเป็นการกดเล่น กดหยุด กดเดินหน้าหรือถอยหลังที่ละเฟรม และกดเริ่มเล่นใหม่ได้ ในส่วนของบาร์ที่ใช้ในการปรับรูปแบบที่ใช้ในการแสดงสามารถปรับขนาดของกระดูก สเกลของตัวละคร ความเร็วของการเคลื่อนไหว และตำแหน่งที่แสดงผลในแกน X, Y หรือ Z ได้

4.2.2 ส่วนของการแสดงผลตัวละครแอนิเมชัน

ส่วนของการแสดงผลตัวละครแอนิเมชัน ได้มีการออกแบบตัวละครจากวัตถุสามมิติให้มีรูปร่างเหมือนมนุษย์เพื่อให้แสดงผลการเคลื่อนไหวตามไฟล์บันทึกความเคลื่อนไหวที่ได้เลือกจากตารางฐานข้อมูลบันทึกความเคลื่อนไหว ในหน้า Motion Capture Database

ภาพที่ 4.2: หน้าจอการแสดงผลตัวละครแอนิเมชัน 3 มิติ

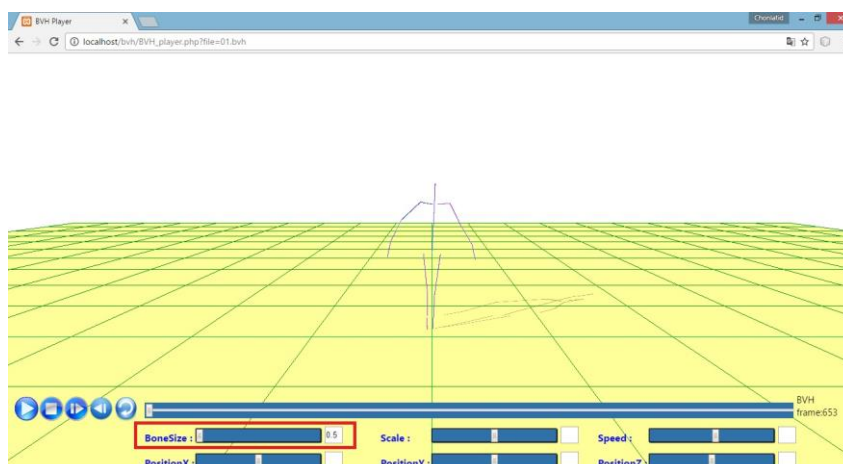


4.3 การใช้งานหน้าจอแสดงผลแอนิเมชัน 3 มิติ

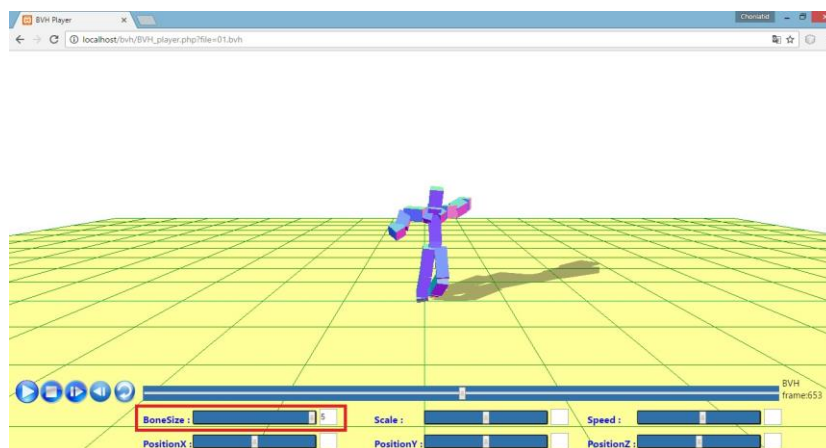
ดังที่ได้กล่าวถึงในหัวข้อที่ 4.2.1 ผู้ใช้สามารถควบคุมการแสดงผลแอนิเมชันได้ทั้งขนาดของกระดูก สเกลของตัวละคร ความเร็วของการเคลื่อนไหว และตำแหน่งที่แสดงผลของตัวละครในแนวแกน X,Y และ Z ซึ่งอธิบายวิธีการใช้งานการควบคุมในส่วนหน้าจอแสดงผลแอนิเมชันดังนี้

- บาร์ควบคุมขนาดกระดูกของตัวละครแอนิเมชันแสดงในกรอบสี่เหลี่ยม ในภาพที่ 4.3 บาร์ที่ชื่อว่า Bone Size มีขนาดเล็กที่สุดตั้งแต่ 0.5 เท่าของขนาดจริงตามไฟล์บันทึกความเคลื่อนไหวไปจนถึงขนาดใหญ่สุด 5 เท่าของขนาดจริงตามไฟล์บันทึกความเคลื่อนไหวซึ่งสามารถควบคุมได้ตามความต้องการ ดังภาพที่ 4.3 และ 4.4 แสดงให้เห็นถึงบาร์ควบคุมขนาดกระดูก (Bone Size) ของตัวละครแอนิเมชัน 3 มิติ จากขนาดเล็กที่สุดไปจนถึงขนาดใหญ่มากที่สุด

ภาพที่ 4.3: การแสดงผลแอนิเมชัน บาร์ควบคุมขนาด BoneSize ที่ขนาดเล็กที่สุด

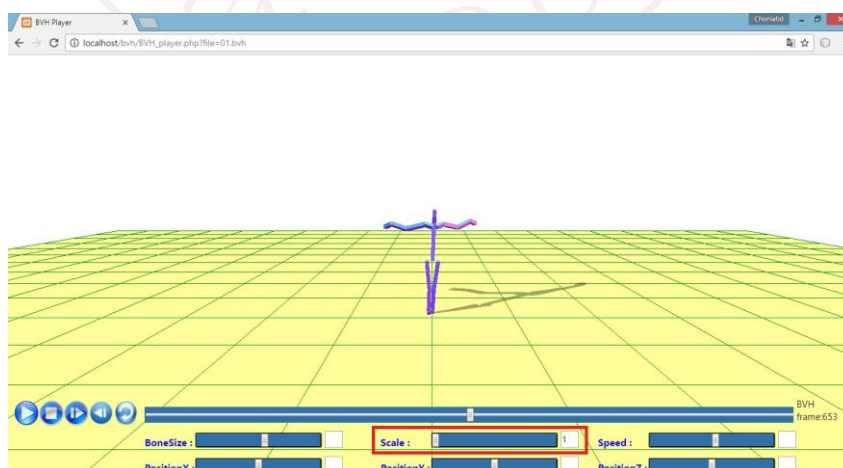


ภาพที่ 4.4: การแสดงผลแอนิเมชัน บาร์ควบคุมขนาด Bone Size ที่ขนาดใหญ่ที่สุด

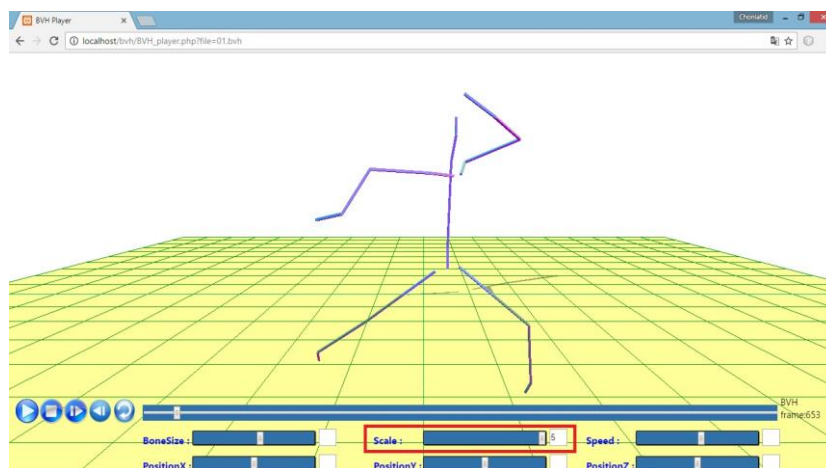


- บาร์ควบคุมขนาดสเกลของตัวละครแอนิเมชัน 3 มิติ บาร์ที่ชื่อว่า Scale มีขนาดเล็กที่สุดที่ 1 เท่าของขนาดจริงตามไฟล์บันทึกความเคลื่อนไหวและขนาดใหญ่ที่สุด 5 เท่าของขนาดจริงตามไฟล์บันทึกความเคลื่อนไหวซึ่งสามารถควบคุมได้ตามต้องการดังภาพที่ 4.5 และ 4.6 แสดงให้เห็นถึงบาร์ควบคุมขนาดสเกล (Scale) ของตัวละครแอนิเมชัน 3 มิติ จากขนาดเล็กที่สุด 1 เท่าของขนาดจริงตามไฟล์บันทึกความเคลื่อนไหวไปจนถึงขนาดใหญ่ที่สุด 5 เท่าของขนาดจริงตามไฟล์บันทึกความเคลื่อนไหว

ภาพที่ 4.5: การแสดงผลแอนิเมชัน บาร์ควบคุมขนาด Scale ที่ขนาดเล็กที่สุด

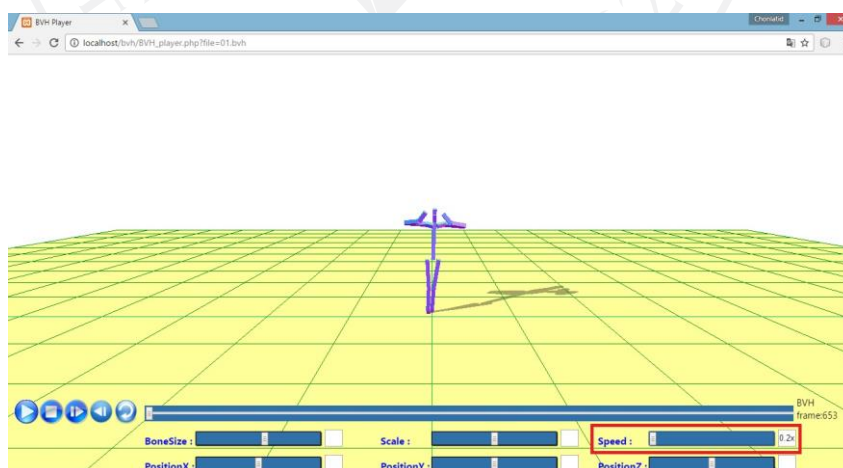


ภาพที่ 4.6: การแสดงผลแอนิเมชัน บาร์ควบคุมขนาด Scale ที่ขนาดใหญ่ที่สุด



- บาร์ควบคุมความเร็วในการแสดงผลของตัวละครแอนิเมชัน 3 มิติ บาร์ที่ชื่อว่า Speed มีความเร็วการเคลื่อนไหวตั้งแต่ 0.2 เท่าของความเร็วจริง ไปจนถึง 3 เท่าของความเร็วจริง (0.2x ถึง 3x) ดังภาพที่ 4.7 แสดงให้เห็นถึงบาร์ควบคุมความเร็วการเคลื่อนไหว (Speed) ของตัวละครแอนิเมชัน 3 มิติ จากความเร็ว ช้าสุด 0.2 เท่าของความเร็วจริง

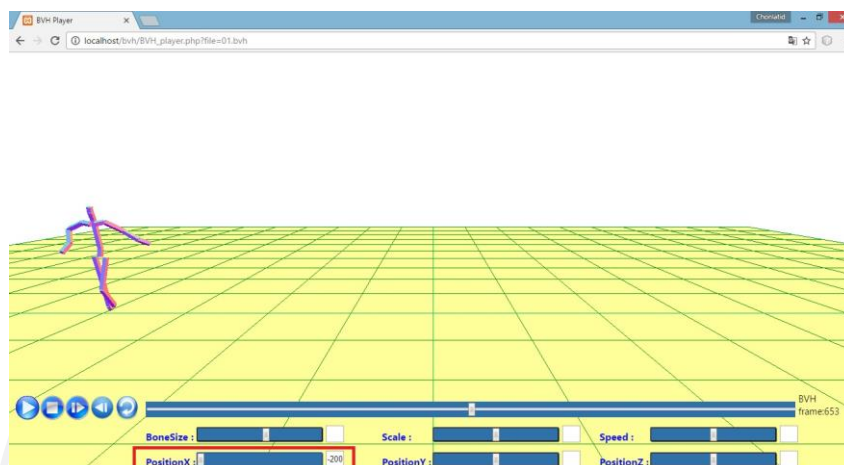
ภาพที่ 4.7: การแสดงผลแอนิเมชัน บาร์ควบคุมความเร็วการเคลื่อนไหว Speed ที่ความเร็ว ช้าสุด



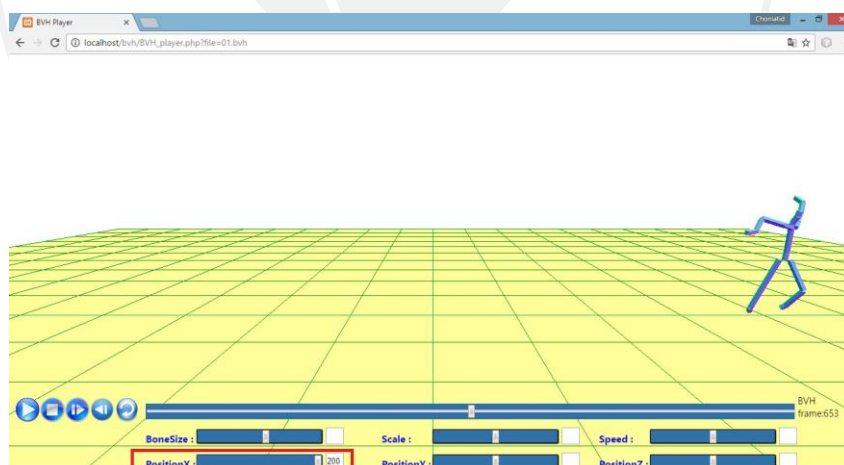
- บาร์ควบคุมตำแหน่งในการแสดงผลของตัวละครแอนิเมชัน 3 มิติ ในแกน X หรือแนวนอน โดยบาร์ที่ชื่อว่า PositionX มีค่าตั้งแต่ -200px ไปจนถึง 200px ดังภาพที่ 4.8 และ 4.9 แสดงให้เห็น

ถึงบาร์ควบคุมตำแหน่งการแสดงผลของตัวละครแอนิเมชัน 3 มิติ ในแนวแกน X จากด้านซ้ายสุดไปด้านขวาสุดมีค่าตั้งแต่ -200px ไปจนถึง 200px

ภาพที่ 4.8: การแสดงผลแอนิเมชัน บาร์ควบคุมตำแหน่งในการแสดงผลในแนวแกน X -200px (PositionX)

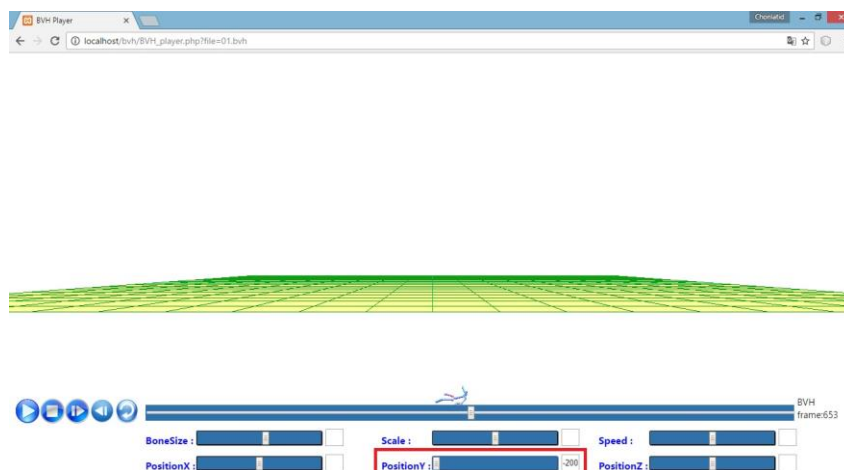


ภาพที่ 4.9: การแสดงผลแอนิเมชัน ตัวควบคุมตำแหน่งในแนวแกน X 200px (PositionX)

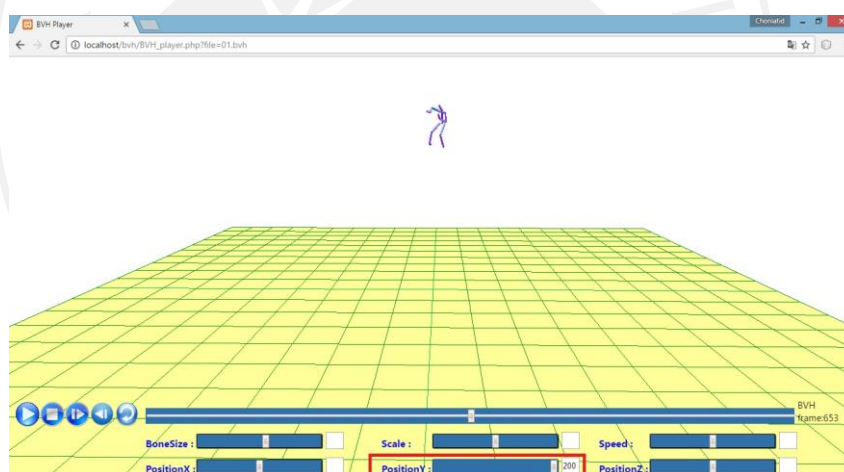


- บาร์ควบคุมตำแหน่งในการแสดงผลของตัวละครแอนิเมชัน 3 มิติ ในแกน Y หรือแนวตั้ง โดยบาร์ที่ชื่อว่า PositionY มีค่าตั้งแต่ -200px ไปจนถึง 200px ดังภาพที่ 4.10 และ 4.11 แสดงให้เห็นถึงบาร์ควบคุมตำแหน่งการแสดงผลของตัวละครแอนิเมชัน 3 มิติ ในแนวแกน Y จากด้านล่างสุดไปด้านบนสุดมีค่าตั้งแต่ -200px ไปจนถึง 200px

ภาพที่ 4.10: การแสดงผลแอนิเมชัน บาร์ควบคุมตำแหน่งในแนวแกน Y -200px (PositionY)

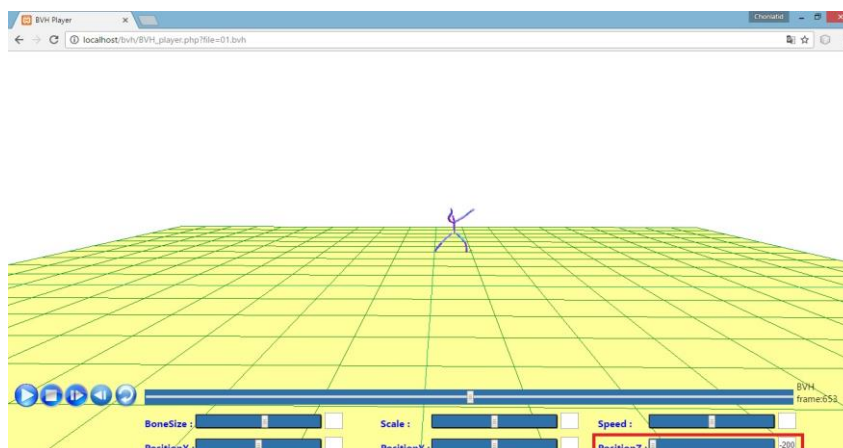


ภาพที่ 4.11: การแสดงผลแอนิเมชัน บาร์ควบคุมตำแหน่งในแนวแกน Y 200px (PositionY)

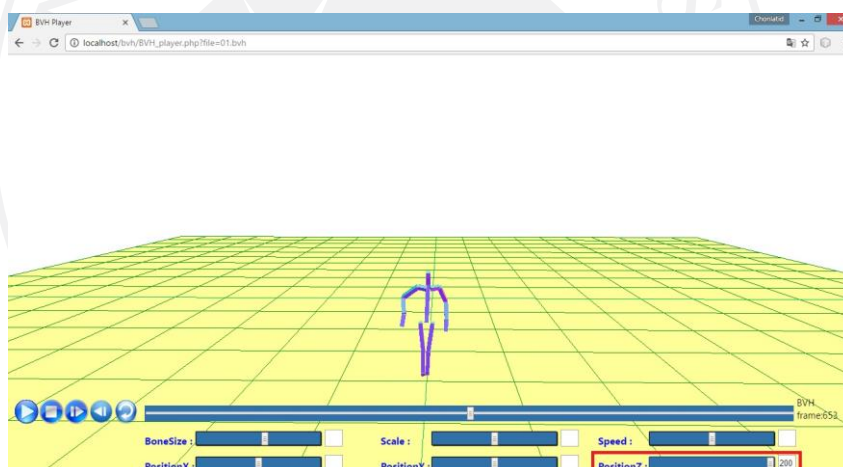


- บาร์ควบคุมตำแหน่งในการแสดงผลของตัวละครแอนิเมชัน 3 มิติ ในแกน Z หรือแนวลึก โดยบาร์ที่ชื่อว่า PositionZ มีค่าตั้งแต่ -200px ไปจนถึง 200px ดังภาพที่ 4.12 และ 4.13 แสดงให้เห็นถึงบาร์ควบคุมตำแหน่งการแสดงผลของตัวละครแอนิเมชัน 3 มิติ ในแนวแกน Z จากด้านไกลสุดไปด้านใกล้สุดมีค่าตั้งแต่ -200px ไปจนถึง 200px

ภาพที่ 4.12: การแสดงผลแอนิเมชัน บาร์ควบคุมตำแหน่งในแนวแกน Z 200px (PositionZ)

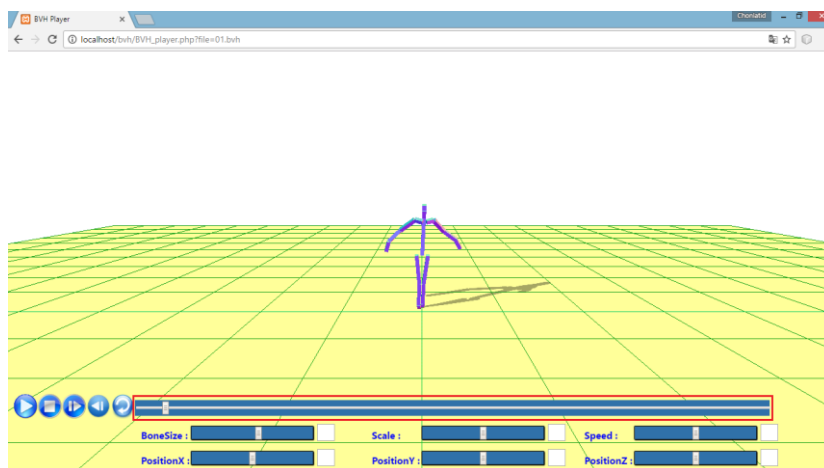


ภาพที่ 4.13: การแสดงผลแอนิเมชัน บาร์ควบคุมตำแหน่งในแนวแกน Z 200px (PositionZ)



- บาร์ควบคุมการเล่นแอนิเมชันหรือที่เรียกว่า Seek Bar เป็นบาร์ที่บอกถึงระยะเวลาในการเล่นแอนิเมชันนั้น ๆ ว่าการแสดงผลเพิ่งเริ่มต้นหรือใกล้จบ โดยการกดปุ่มที่เป็นรูปไอคอน Play หลังจากนั้น Seek Bar ขยับตามเฟรมการเคลื่อนไหวของตัวละคร และผู้ใช้งานสามารถเลื่อน Seek Bar โดยตัวละครมีการเคลื่อนไหวตามการขยับของ Seek Bar ดังภาพที่ 4.14 แสดงให้เห็นถึง Seek Bar เป็นบาร์ที่ควบคุมการเล่นของตัวละครแอนิเมชัน

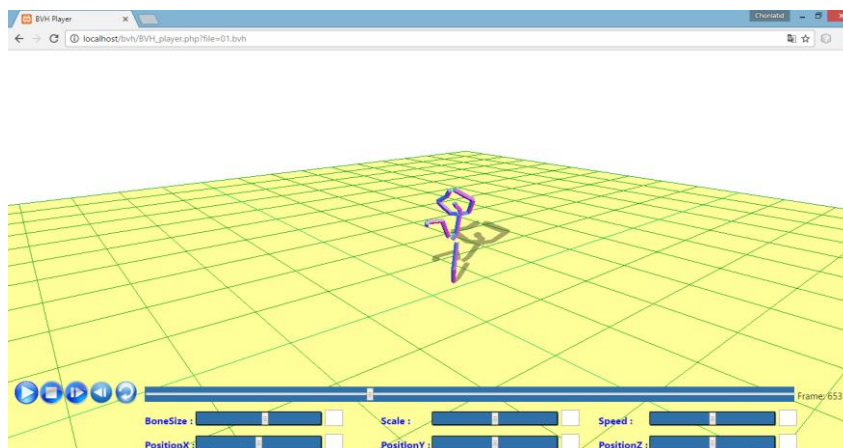
ภาพที่ 4.14: Seek Bar บาร์ควบคุมการเล่นแอนิเมชัน



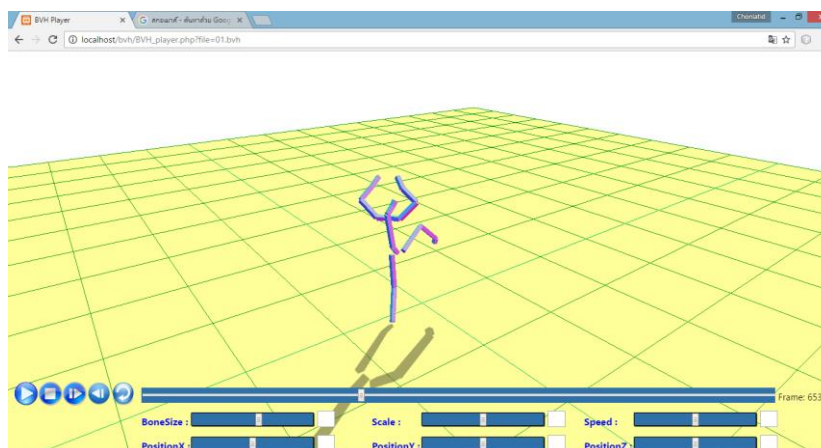
- ส่วนหน้าจอการแสดงผลของระบบ จากระบบที่ได้จัดทำมานี้แสดงผลในรูปแบบของเว็บไซต์ที่เป็นสามมิติ ซึ่งเว็บสามมิตินี้มีมุมมองที่หลากหลายซึ่งในระบบการแสดงผลนี้สามารถควบคุมมุมมองได้ ดังภาพที่ 4.15 และ ภาพที่ 4.16 เป็นการเปลี่ยนมุมมองโดยสังเกตได้จากเงาของตัวละครที่เปลี่ยนไป การควบคุมบนหน้าจอแสดงผลมีดังนี้

- 1) กดคลิกซ้ายค้างที่เมาส์เป็นการหมุนมุมมองไปรอบ ๆ ตัวละคร
- 2) กดคลิกขวาค้างที่เมาส์เป็นการขยับกล้อง
- 3) สกอลเมาส์เลื่อนไปด้านหน้า เป็นการ ซูมเข้า
- 4) สกอลเมาส์เลื่อนไปด้านหลัง เป็นการ ซูมออก

ภาพที่ 4.15: การควบคุมกล้องและมุมมองของฉาก



ภาพที่ 4.16: การควบคุมกล้องและมุมมองของฉาก ที่เปลี่ยนจาก ภาพที่ 4.15



4.4 เกณฑ์การให้คะแนนของแบบประเมิน

เนื่องจากงานวิจัยชิ้นนี้ได้มีการทำระบบสำหรับการใช้งาน จึงต้องมีการประเมินผลงานของระบบในด้านต่าง ๆ เช่น ความสวยงามของหน้าจอ และการทำงานเป็นต้น ดังตารางที่ 4.1 และ 4.2 เป็นตารางแสดงระดับผลการประเมินระดับความพึงพอใจของระบบแสดงผลแอนิเมชันจากไฟล์บันทึกความเคลื่อนไหวที่อยู่บนเซิร์ฟเวอร์

ตารางที่ 4.1: แสดงระดับผลการประเมินความพึงพอใจของระบบ

ระดับผลการประเมิน	ระดับความพึงพอใจ
มากที่สุด	ความพึงพอใจของระบบมากที่สุด
มาก	ความพึงพอใจของระบบชั้นมาก
ปานกลาง	ความพึงพอใจของระบบปานกลาง
น้อย	ความพึงพอใจของระบบน้อย
น้อยที่สุด	ความพึงพอใจของระบบน้อยที่สุด

จากตารางที่ 4.1 มีเกณฑ์การประเมินระดับความพึงพอใจแบ่งเป็น 5 ระดับ ได้แก่ 1) ความพึงพอใจของระบบมากที่สุด 2) ความพึงพอใจของระบบชั้นมาก 3) ความพึงพอใจของระบบปานกลาง 4) ความพึงพอใจของระบบน้อย 5) ความพึงพอใจของระบบน้อยที่สุด

ตารางที่ 4.2 แสดงระดับผลการประเมินความพึงพอใจของระบบ

คะแนนความพึงพอใจ	ผลการประเมิน
4.50 – 5.00	มากที่สุด
3.50 – 4.49	มาก
2.50 – 3.49	ปานกลาง
1.50 – 2.49	น้อย
0.00 – 1.49	น้อยที่สุด

จากตารางที่ 4.2 มีเกณฑ์การให้คะแนนความพึงพอใจแบ่งออกเป็น 5 ช่วง ได้แก่ 1) 4.5–5 พึงพอใจมากที่สุด 2) 3.5–4.49 พึงพอใจมาก 3) 2.5–3.49 พึงพอใจปานกลาง 4) 1.5–2.49 พึงพอใจน้อย 5) 0–1.49 พึงพอใจน้อยที่สุด

ตารางที่ 4.3 แสดงแบบประเมินความพึงพอใจของระบบ

รายละเอียด	ระดับความพึงพอใจ				
	5	4	3	2	1
1. รูปแบบหน้าตา (Interface)					
1.1 ความเหมาะสมของการจัดวางรูปแบบ มีความสะดวกและ					
1.2 ความสวยงาม และความเรียบร้อยของรูปแบบ					
2. การทำงานของตัวระบบ					
2.1 ใช้งานได้ง่าย ไม่ซับซ้อน					
2.2 ระบบสามารถทำงานได้รวดเร็วทันใจผู้ใช้ระบบ					
2.3 ระบบมีสิ่งจำเป็นที่ผู้ใช้ระบบต้องการ					

จากตารางที่ 4.3 เป็นการแสดงตารางแบบประเมินที่ใช้ในการประเมิน โดยมีทั้งหมด 6 ข้อ และช่องการให้คะแนน 5 ช่อง โดยผู้ที่ทำการประเมินต้องทำเครื่องหมายถูกไว้ในช่องที่เตรียมไว้ให้

4.5 สรุปผลความพึงพอใจของระบบ

สรุปผลความพึงพอใจของระบบการแสดงผลข้อมูลการตรวจจับความเคลื่อนไหวผ่านตัวละครแอนิเมชัน 3 มิติ จากการเก็บรวบรวมข้อมูลแบบประเมินความพึงพอใจและความคิดเห็น จากบุคคลทั่วไป จำนวน 30 คนแบ่งเป็นผู้ชาย 14 คนและผู้หญิง 16 คน

นำข้อมูลจากแบบสอบถามความพึงพอใจต่อระบบมาวิเคราะห์หาระดับความพึงพอใจต่อระบบโดยใช้วิธีการคำนวณการหาค่าเฉลี่ยและส่วนเบี่ยงเบนมาตรฐาน (Standard Deviation: S.D.,S,s) เพื่อกำหนดระดับความพึงพอใจที่มีต่อระบบการแสดงผลแอนิเมชันจากไฟล์บันทึกความเคลื่อนไหว โดยใช้สูตรการหาส่วนเบี่ยงเบนมาตรฐาน (Standard Deviation: S.D.)

$$S.D. = \sqrt{\frac{n \sum x^2 - (\sum x)^2}{n(n-1)}}$$

ตารางที่ 4.4: ตารางสรุปผลความพึงพอใจของระบบ

รายละเอียด	ค่าเฉลี่ย	ส่วนเบี่ยงเบนมาตรฐาน	ระดับความพึงพอใจ
1. รูปแบบหน้าตา (Interface)			
1.1 ความเหมาะสมของการจัดวางรูปแบบ มีความสะดวก	4.23	0.57	มาก
1.2 ความสวยงาม และความเรียบร้อยของรูปแบบ	3.9	0.71	มาก
2. การทำงานของตัวระบบ			
2.1 ใช้งานได้ง่าย ไม่ซับซ้อน	4.23	0.63	มาก
2.2 ระบบสามารถทำงานได้รวดเร็วทันใจผู้ใช้งานระบบ	4.1	0.61	มาก
2.3 ระบบมีสิ่งจำเป็นที่ผู้ใช้งานระบบต้องการ	4	0.64	มาก

จากการสรุปผลสำรวจความพึงพอใจต่อระบบการแสดงผลแอนิเมชันจากไฟล์บันทึกความเคลื่อนไหวรูปแบบ BVH ซึ่งจากผลการประเมินแบ่งออกเป็นส่วนของรูปแบบหน้าตาของระบบและการทำงานของตัวระบบ ระบบมีความเหมาะสมของการจัดวางรูปแบบ ความสะดวกในการใช้งานและสามารถจดจำได้ง่าย ความสวยงามความเรียบร้อยของรูปแบบในการนำเสนอ การใช้งานที่เอื้อประโยชน์ และความสมบูรณ์ของระบบต่อผู้ใช้งานในภาพรวมของระบบอยู่ในระดับความพึงพอใจมาก

บทที่ 5

สรุปผล อภิปรายผล และข้อเสนอแนะ

5.1 สรุปผลการศึกษา

จากการศึกษาพบว่าภาษาที่ใช้ในการพัฒนาระบบสำหรับแสดงผลกราฟิก 3D บนเว็บเบราว์เซอร์ ที่นิยมใช้กันมีอยู่ด้วยกันทั้งหมด 4 ภาษา ซึ่งประกอบด้วยภาษา C++, Python, Java Applets, Flash + Action Script และ Java Script + HTML5 สำหรับภาษา C++, Python และ Java เป็นการทำระบบที่ใช้งานบนเครื่อง Client โดยต้องโหลดซอฟต์แวร์มาติดตั้งภายในเครื่องต่างจาก Java Script ที่ระบบทั้งหมดทำงานอยู่บนเซิร์ฟเวอร์โดยเปิดผ่านเว็บเบราว์เซอร์ซึ่งสามารถใช้งานได้โดยไม่ต้องติดตั้งอะไรเพิ่มเติม

เพื่อให้เห็นถึงข้อดีข้อเสียของเครื่องมือและภาษาต่าง ๆ การพัฒนาในโครงการนี้ได้ทดลองพัฒนาโดยใช้เครื่องมือที่แตกต่างกันดังนี้

5.1.1 การพัฒนาบน Unity 3D ในการพัฒนาระบบแสดงผลแอนิเมชันจากการจับความเคลื่อนไหว แต่เนื่องจากโปรแกรม Unity 3D เหมาะสมกับการสร้างเกมส์ โดยภาษาหลักที่เหมาะสมในการพัฒนาคือภาษา C# ซึ่งมีความยากในการเขียนระบบเพื่ออ่านค่าจากไฟล์บันทึกความเคลื่อนไหว เนื่องจาก Unity 3D เป็นโปรแกรมที่มีเครื่องมือเหมาะสำหรับการสร้างตัวละครแอนิเมชันและนำเข้าไฟล์บันทึกความเคลื่อนไหวมาแมพเข้ากับตัวละคร ส่วนด้านความปลอดภัยของไฟล์ฐานข้อมูลจับความเคลื่อนไหวจะไม่สามารถป้องกันได้ เนื่องจาก Unity 3D ใช้ HTML5+WebGL ในการพัฒนาระบบแสดงผลแอนิเมชันออนไลน์ซึ่งเป็นการทำงานทางฝั่งของผู้ใช้งานจึงไม่สามารถป้องกันการเข้าถึงได้

5.1.2 การพัฒนาบนแพลตฟอร์ม Flash ในการพัฒนา เนื่องจากโปรแกรม Flash Builder ใช้พื้นฐานภาษา Action Script การที่พัฒนาระบบแสดงผลแอนิเมชันบนเว็บเบราว์เซอร์แบบสามมิติจำเป็นต้องลงตัวเสริม (Flash Professional) เพื่อทำให้รองรับไลบรารีกราฟิกบนเว็บและแสดงผลผ่านเว็บได้ นอกจากนี้การแสดงผลจากโปรแกรม Flash ไม่ซัพพอร์ตการแสดงผลบน IOS ในด้านความปลอดภัยของไฟล์จับความเคลื่อนไหว โปรแกรม Flash มีตัวป้องกันโค้ดที่ชื่อว่า Sandbox ซึ่งทำให้ผู้ใช้งานไม่สามารถดูโค้ดที่อยู่ไฟล์ได้

5.1.3 การพัฒนาระบบบนแพลตฟอร์ม Web 3D ภาษา Java Applets มาเป็นโปรแกรม Netbeans ภาษา Java Script+HTML5 เพื่อให้สามารถใช้งานได้บนเว็บเบราว์เซอร์และรองรับการทำงานของ WebGL

5.2 อภิปรายผล

5.2.1 จากการทดสอบระบบการแสดงผล Motion Capture ผ่านเว็บเบราว์เซอร์ 3 มิติ ผู้พัฒนาระบบสามารถสรุปผลการทำงานของระบบได้ว่า ผู้ใช้งานระบบ (User) สามารถกระทำงานได้ทุกฟังก์ชันงาน ได้แก่

1) การอ่านข้อมูลไฟล์ Motion Capture ที่อยู่บนเซิร์ฟเวอร์ และนำมาแสดงผลผ่านตัวละครแอนิเมชัน 3 มิติ

2) การควบคุมการเล่นของตัวละครแอนิเมชัน 3 มิติ

3) การจัดตำแหน่งการแสดงผลของตัวละครแอนิเมชัน 3 มิติ

5.2.2 จากการทดสอบการแสดงผลบนเว็บเบราว์เซอร์ต่าง ๆ กันซึ่ง Chrome สามารถแสดงผลได้ดีที่สุด เนื่องจากผู้พัฒนาใช้ Chrome เป็นเว็บเบราว์เซอร์ต้นแบบในการทดสอบการแสดงผลแอนิเมชัน ในส่วนของเว็บเบราว์เซอร์ Firefox และ Internet Explorer สามารถแสดงผลได้แต่มีตำแหน่งของวัตถุคลาดเคลื่อนไป

5.2.3 การใช้โปรแกรมสำหรับในการพัฒนาระบบ ซึ่งในงานวิจัยนี้ได้ทดลองใช้ทั้งหมด 3 แพลตฟอร์ม ได้แก่ Web 3D, Unity 3D และ Flash จากแพลตฟอร์มที่กล่าวมาทั้งหมดสามารถใช้ในการพัฒนาระบบแสดงผลแอนิเมชันจากไฟล์ตรวจจับความเคลื่อนไหวได้ซึ่งในงานวิจัยนี้วิเคราะห์และเปรียบเทียบการใช้งานของแต่ละแพลตฟอร์ม ดังตารางที่ 5.1 เป็นตารางการเปรียบเทียบแพลตฟอร์มที่ใช้ในการพัฒนา

ตารางที่ 5.1: การเปรียบเทียบโปรแกรมที่ใช้ในการพัฒนาระบบ

คุณสมบัติ/ชื่อโปรแกรม	Web3D	Unity3D	Flash
1.ภาษาและเทคโนโลยีที่ใช้	HTML5, WebGL, JavaScript3D	HTML5, WebGL, C#, Animator	Action Script, paperVision3D, Flash Professional
2.การแสดงผลแอนิเมชันแบบ Realtime	Real Time	ไม่ Real Time	Real Time
3.การป้องกันความปลอดภัยของข้อมูล	Encode URL	Encode URL	Sandbox

(ตารางมีต่อ)

ตารางที่ 5.1 (ต่อ): การเปรียบเทียบโปรแกรมที่ใช้ในการพัฒนาระบบ

คุณสมบัติ/ชื่อโปรแกรม	Web3D	Unity3D	Flash
4.แพลตฟอร์มที่รองรับ	Web Browser (ที่รองรับ WebGL)	Web Browser (ที่รองรับ WebGL)	Web Browser แต่ไม่รองรับการแสดงผลบน IOS

จากตารางที่ 5.1 การเปรียบเทียบโปรแกรมที่ใช้ในการพัฒนาระบบ สามารถสรุปได้ดังนี้ จากโปรแกรมที่กล่าวมาข้างต้นสามารถพัฒนาระบบแสดงผลแอนิเมชันจากไฟล์บันทึกความเคลื่อนไหวได้ทุกโปรแกรม แต่มีง่ายยากแตกต่างกันไปตามความเหมาะสมซึ่งผู้พัฒนาได้เลือกโปรแกรม NetBeans ในการพัฒนาระบบเพราะเป็นโปรแกรมที่ไม่มีค่าใช้จ่ายในการใช้งาน ซึ่งต่างจาก Flash Builder และตัวโปรแกรมสามารถรองรับภาษา Java Script, HTML5, PHP และ CSS ซึ่งสามารถติดตั้งโปรแกรมเสริมได้สะดวก

ในส่วนการแสดงผลโปรแกรม NetBeans รองรับการทำงานของ Web Server, WebGL โดยการติดตั้ง NetBeans Connector ทำให้สามารถระบบแสดงผลแอนิเมชันจากไฟล์บันทึกความเคลื่อนไหวผ่านเว็บแบบสามมิติได้

5.2.4 การทดสอบความปลอดภัยของระบบ ซึ่งระบบแสดงผลแอนิเมชันจากไฟล์บันทึกความเคลื่อนไหวได้ทำการเข้ารหัส URL ทำให้ผู้ใช้งานเข้าถึงไฟล์ข้อมูลที่บันทึกความเคลื่อนไหวที่อยู่บนเซิร์ฟเวอร์ได้ยากขึ้นในระดับหนึ่ง แต่ไม่สามารถป้องกันได้อย่างสมบูรณ์ เนื่องจากระบบเป็นการทำงานแบบ Client Side Script ซึ่งเป็นการทำงานจากฝั่งของผู้ใช้งาน จึงไม่สามารถป้องกันได้ทั้งระบบ

5.2.5 จากการสรุปผลสำรวจความพึงพอใจต่อระบบการแสดงผลแอนิเมชันจากไฟล์บันทึกความเคลื่อนไหวรูปแบบ BVH ซึ่งจากผลการประเมิน ระบบมีความเหมาะสม สวยงาม การใช้งานที่เอื้อประโยชน์ และความสมบูรณ์ของระบบต่อผู้ใช้งานในภาพรวมของระบบอยู่ในระดับความพึงพอใจมาก

5.3 ปัญหาของระบบ

5.3.1 เว็บเบราว์เซอร์ที่ไม่รองรับการทำงานของ WebGL ไม่สามารถแสดงผลแอนิเมชันได้

5.3.2 การแสดงผลแอนิเมชันจากไฟล์บันทึกความเคลื่อนไหวบางไฟล์มีตำแหน่งและขนาดไม่ตรงตามที่กำหนดไว้

5.3.3 เว็บเบราว์เซอร์ที่ต่างกัน ให้การแสดงผลของหน้าจอเช่น บาร์ควบคุมมีรูปแบบที่แตกต่างกัน

5.4 แนวทางแก้ปัญหา

5.4.1 ต้องใช้เว็บเบราว์เซอร์ที่รองรับการทำงานของ WebGL หรือ Web 3D ตามที่ได้กำหนดไว้ในรายงาน

5.4.2 มีเครื่องมือที่ใช้ในการควบคุมตำแหน่งการแสดงผล และ ขนาดของตัวละครแอนิเมชัน ซึ่งสามารถปรับได้ตามความต้องการ

5.5 ข้อเสนอแนะ และแนวทางการพัฒนาต่อ

จากข้อเสนอแนะที่ได้จากการทำแบบสอบถามความพึงพอใจในระบบการแสดงผลแอนิเมชันจากไฟล์บันทึกความเคลื่อนไหวบนเซิร์ฟเวอร์ สามารถสรุปได้ดังนี้

5.5.1 สามารถให้อ่านไฟล์ข้อมูลความเคลื่อนไหวจากเว็บไซต์ต่าง ๆ ได้

5.5.2 ทำให้ทุกไฟล์ที่บันทึกข้อมูลความเคลื่อนไหวที่นำมาแสดงผลมีขนาดและตำแหน่งที่เหมือนกันเป็นมาตรฐาน

5.5.3 เพิ่มรูปแบบต่าง ๆ ของตัวละครแอนิเมชัน 3 มิติ เช่น สี หรือ รูปร่าง เป็นต้น

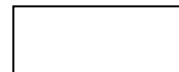
5.5.4 เพิ่มฉากหรือวัตถุต่าง ๆ เพื่อให้มีความน่าสนใจมากขึ้น

บรรณานุกรม

- ด้ชกรณั ตันเจริญ. (2554). เทคโนโลยีการตรวจจับการเคลื่อนไหวและการประยุกต์ใช้งาน. วารสาร
ปัญญาวิวัฒน์, 3(1), 113-122.
- สุวิชัย พรรษา. (2552). Motion capture จินตนาการที่สัมผัสได้. สืบค้นจาก
<http://suwichai.blogspot.com>.
- สิทธิชัย ทิพย์สิงห์. (ม.ป.ป.). บทเรียนออนไลน์การสร้างแอนิเมชันด้วย Flash. สืบค้นจาก
<http://www.kroojan.com/flash/content/flash-intro.html>.
- Ben, D. (2010). Sport sims motion capture technology. Caerleon: Usk Valley.
- Bodenheimer, B., Rose, C., Rosenthal, S., & Pella, J. (1997). The process of motion
capture: Dealing with the data. In Computer Animation and
Simulation'97 (pp. 3-18). New York: Springer Vienna.
- Cao, Q., Gao, C., Wang, Y., & Li, T. (2016, May). The motion capture data glove device
for virtual surgery. In Nanoelectronics Conference (INEC), IEEE
International (pp. 1-2). New York: IEEE Press.
- Cheng, K., Guo, R., & Gao, M. (2010, August). An optimizing chinese string matching
algorithm based on the url encoding. In Information Engineering (ICIE),
WASE International Conference on Vol. 1 (pp. 23-25). New York: IEEE Press.
- Edin, M. (2013). BVH file loading and displaying. Retrieved from
<http://www.gamedev.net>.
- Maestri, G. (2006). Digital character animation 3. New Jersey: New Riders.
- Menache, A. (2000). Understanding motion capture for computer animation and
video games. San Francisco: Morgan kaufmann.
- Meredith, M., & Maddock, S. (2001). Motion capture file formats explained.
Department of Computer Science, University of Sheffield, 211, 241-244.
- Phunsa, S., Potisarn, N., & Tirakoat, S. (2009, February). Edutainment--Thai art of self-
defense and boxing by motion capture technique. In Computer Modeling and
Simulation, ICCMS'09. International Conference on (pp. 152-155). New York:
IEEE Press.
- Reitsma, P. S., & Pollard, N. S. (2007). Evaluating motion graphs for character
animation. ACM Transactions on Graphics (TOG), 26(4), 18.

- Ricci, L., & Formica, D. (2014, November). Dynamic accuracy assessment of data-fusion techniques for wearable, inertial and magnetic based human motion capture. In IEEE SENSORS 2014 Proceedings (pp. 2215-2218). New York: IEEE Press.
- Schubert, T., Eggensperger, K., Gkogkidis, A., Hutter, F., Ball, T., & Burgard, W. (2016). Automatic bone parameter estimation for skeleton tracking in optical motion capture. *Proc. of the IEEE*, 1, 256-263.
- Yu, Q., & Terzopoulos, D. (1998, June). Synthetic motion capture for interactive virtual worlds. In *Computer Animation 98. Proceedings* (pp. 2-10). New York: IEEE Press.
- Yu, X., Yang, J., Luo, L., Li, W., Brandt, J., & Metaxas, D. (2016, March). Customized expression recognition for performance-driven cutout character animation. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)* (pp. 1-9). New York: IEEE Press.





หนังสือแสดงเจตนายินยอมเข้าร่วมการวิจัย

(Informed Consent Form)

วันที่..... เดือน..... พ.ศ.....

ข้าพเจ้า..... อายุ.....ปี

ขอแสดงเจตนายินยอมเข้าร่วมโครงการวิจัยเรื่อง

การศึกษาการแสดงผลแอนิเมชันจากไฟล์ข้อมูลที่บันทึกความเคลื่อนไหวบนเซิร์ฟเวอร์

ข้าพเจ้าได้รับทราบรายละเอียดเกี่ยวกับที่มาและจุดมุ่งหมายของโครงการวิจัยฯ และข้อมูลของผู้เข้าร่วมโครงการถูกเก็บรักษาไว้เป็นรายบุคคลโดยไม่เปิดเผยต่อสาธารณะ โดยมีเพียงการรายงานผลการวิจัยต่อคณะอาจารย์

ผู้ทำวิจัยได้ชี้แจงรายละเอียดขั้นตอนต่าง ๆ ที่ต้องปฏิบัติในการทดลองใช้แอปพลิเคชันฯ เพื่อนำผลการทดลองปฏิบัติไปวิเคราะห์และสรุปผลต่อไป

ข้าพเจ้ายินดีที่ได้เข้าร่วมโครงการและลงชื่อไว้เป็นหลักฐานเพื่อใช้ประโยชน์ต่องานวิจัยต่อไปข้าพเจ้ายินยอมที่เข้าร่วมในโครงการดังกล่าว

ลงชื่อ.....ผู้เข้าร่วมการวิจัย
(.....)



BVH Player System

คำชี้แจง แบบสอบถาม

1. เพื่อให้ผู้ใช้งานระบบ BVH Player ได้ทำการประเมินตัวระบบ และนำเสนอข้อเสนอแนะมายังผู้พัฒนาระบบ
2. โปรดเติมเครื่องหมาย ✓ และกรอกข้อความให้สมบูรณ์

ส่วนที่ 1 ข้อมูลทั่วไปของผู้ตอบแบบสอบถาม

1. เพศ ชาย หญิง
2. อายุ
 น้อยกว่า 10 ปี 11 – 15 ปี 16 – 20 ปี 21 – 25 ปี 26 – 30 ปี
 31 – 35 ปี 35 – 40 ปี 41 – 45 ปี 45 – 50 ปี มากกว่า 51 ปี
3. วุฒิการศึกษา ต่ำกว่าปริญญาตรี ปริญญาตรี สูงกว่าปริญญาตรี

ส่วนที่ 2 ความพึงพอใจต่อระบบ

ระดับ 5 = มากที่สุด 4 = มาก 3 = ปานกลางหรือพอใช้ 2 = น้อย 1 = ต้องปรับปรุงแก้ไข

รายละเอียด	ระดับความพึงพอใจ				
	5	4	3	2	1
1. รูปแบบหน้าต่าง (Interface)					
1.3 ความเหมาะสมของการจัดวางรูปแบบ มีความสะดวกและ					
1.2 ความสวยงาม และความเรียบร้อยของรูปแบบ					
2. การทำงานของตัวระบบ					
2.3 ใช้งานได้ง่าย ไม่ซับซ้อน					
2.2 ระบบสามารถทำงานได้รวดเร็วทันใจผู้ใช้งานระบบ					
2.3 ระบบมีสิ่งจำเป็นที่ผู้ใช้งานระบบต้องการ					

ส่วนที่ 3 ข้อเสนอแนะ

3.1 สิ่งที่ท่านพึงพอใจในระบบ BVH Player

.....

.....

.....

3.2 สิ่งที่คุณควรเสนอแนะนำไปพัฒนาระบบ BVH Player

.....

.....

.....

ขอขอบคุณในความร่วมมือ
ผู้พัฒนาระบบ BVH Player





ภาคผนวก ค

JavaScript Source Code For BVH File Parsing

```
var BVH = { REVISION:'0.1a'};
BVH.TO_RAD = Math.PI / 180;
window.URL = window.URL || window.webkitURL;
BVH.Reader = function(){
    this.debug = true;
    this.type = "";
    this.data = null;
    this.root = null;
    this.numFrames = 0;
    this.secsPerFrame = 0;
    this.play = false;
    this.channels = null;
    this.lines = "";
    this.speed = 1;
    this.nodes = null;
    this.frame = 0;
    this.oldFrame = 0;
    this.startTime = 0;
    this.position = new THREE.Vector3( 0, 0, 0 );
    this.scale = 1;
    this.tmpOrder = "";
    this.tmpAngle = [];
    this.skeleton = null;
    this.bones = [];
    this.boneSize = 1.5;
    this.material = new THREE.MeshNormalMaterial();
},
BVH.Reader.prototype = {
    constructor: BVH.Reader,
```

```

    _keyStr :
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"
    =",
    load:function(fname){
        fname = this.decode("Ly9sb2NhbGhvc3QvYnZoL3Jlcy9idm9v") + fname;
        this.type = fname.substring(fname.length-3,fname.length);

        var _this = this;
            var xhr = new XMLHttpRequest();
            xhr.open( 'GET', fname, true );
            if(this.type === 'bvh'){ // direct from file
                xhr.onreadystatechange = function(){ if ( this.readyState
=== 4 ){ _this.parseData(this.responseText.split(/\s+/g));}};
            }
            xhr.send( null );
        },
        parseData:function(data){
            this.data = data;
            this.channels = [];
            this.nodes = [];
            var done = false;
            while (!done) {
                switch (this.data.shift()) {
                case 'ROOT':
                    if(this.root !== null) this.clearNode();
                    this.root = this.parseNode(this.data);
                    this.root.position.copy(this.position);
                    this.root.scale.set(this.scale,this.scale,this.scale);
                    if(this.debug){
                        this.addSkeleton( this.nodes.length );
                    }
                }
            }
        }
    }

```

```

        break;
    case 'MOTION':
        this.data.shift();
        this.numFrames = parseInt( this.data.shift() );
        this.data.shift();
        this.data.shift();
        this.secsPerFrame = parseFloat(this.data.shift());
        done = true;
    }
}
this.getNodeList();
this.startTime = Date.now();
this.play = true;
},
reScale:function (s) {
    this.scale = s;
    this.root.scale.set(this.scale,this.scale,this.scale);
},
rePosition:function (v) {
    this.position = v;
    this.root.position.x = v.x;
    this.root.position.y = v.z;
    this.root.position.z = v.z;
},
getNodeList:function () {
    var n = this.nodes.length, node, s = "";
    for(var i=0; i<n; i++){
        node = this.nodes[i];
        s += node.name + " _ " + i + "<br>"
    }
},

```

```

addSkeleton:function ( n ) {
    this.skeleton = new THREE.Object3D();
    this.bones = [];
    var n = this.nodes.length, node, bone;
    var geo = new THREE.CubeGeometry( this.boneSize, this.boneSize, 1);
    //ขนาดจำนวนวัตถุ
    geo.applyMatrix( new THREE.Matrix4().makeTranslation( 0, 0, 0.5 ) );
    //ระยะห่างวัตถุ
    for(var i=0; i<n; i++){
        node = this.nodes[i];
        if ( node.name !== 'Site' ){
            bone = new THREE.Mesh(geo, this.material);
            bone.castShadow = true;
            bone.rotation.order = 'XYZ';
            bone.name = node.name;
            this.skeleton.add(bone);
            this.bones[i] = bone;
        }
    }
    scene.add( this.skeleton );
},
updateSkeleton:function ( ) {
    var mtx, node, bone;
    var n = this.nodes.length;
    var target;
    for(var i=0; i<n; i++){
        node = this.nodes[i];
        bone = this.bones[i];
        if ( node.name !== 'Site' ){
            mtx = node.matrixWorld;
            bone.position.setFromMatrixPosition( mtx );

```

```

        if(node.children.length){
            target = new
THREE.Vector3().setFromMatrixPosition( node.children[0].matrixWorld );
            bone.lookAt(target);
            bone.rotation.z = 0;

            if(bone.name==="Head")bone.scale.set(this.boneSize*1,this.boneSize*1,
BVH.DistanceTest(bone.position, target)*(this.boneSize*0.5));
            else
bone.scale.set(this.boneSize,this.boneSize,BVH.DistanceTest(bone.position,
target));
        }
    }
},
transposeName:function(name){
    if(name==="hip") name = "Hips";
    if(name==="abdomen") name = "Spine1";
    if(name==="chest") name = "Chest";
    if(name==="neck") name = "Neck";
    if(name==="head") name = "Head";
    if(name==="lCollar") name = "LeftCollar";
    if(name==="rCollar") name = "RightCollar";
    if(name==="lShldr") name = "LeftUpArm";
    if(name==="rShldr") name = "RightUpArm";
    if(name==="lForeArm") name = "LeftLowArm";
    if(name==="rForeArm") name = "RightLowArm";
    if(name==="lHand") name = "LeftHand";
    if(name==="rHand") name = "RightHand";
    if(name==="lFoot") name = "LeftFoot";
    if(name==="rFoot") name = "RightFoot";

```

```

if(name==="lThigh") name = "LeftUpLeg";
if(name==="rThigh") name = "RightUpLeg";
if(name==="lShin") name = "RightLowLeg";
if(name==="rShin") name = "LeftLowLeg";
// leg
if(name==="RightHip") name = "RightUpLeg";
if(name==="LeftHip") name = "LeftUpLeg";
if(name==="RightKnee") name = "RightLowLeg";
if(name==="LeftKnee") name = "LeftLowLeg";
if(name==="RightAnkle") name = "RightFoot";
if(name==="LeftAnkle") name = "LeftFoot";
// arm
if(name==="RightShoulder") name = "RightUpArm";
if(name==="LeftShoulder") name = "LeftUpArm";
if(name==="RightElbow") name = "RightLowArm";
if(name==="LeftElbow") name = "LeftLowArm";
if(name==="RightWrist") name = "RightHand";
if(name==="LeftWrist") name = "LeftHand";
if(name==="rcollar") name = "RightCollar";
if(name==="lcollar") name = "LeftCollar";
if(name==="rtoes") name = "RightToe";
if(name==="ltoes") name = "LeftToe";
if(name==="upperback") name = "Spine1";
return name;
},
parseNode:function(data){
var name, done, n, node, t;
name = data.shift();
name = this.transposeName(name);
node = new THREE.Object3D();
node.name = name;

```



```

done = false;
while ( !done ) {
    switch ( t = data.shift() ) {
        case 'OFFSET':
            node.position.set( parseFloat( data.shift()
), parseFloat( data.shift() ), parseFloat( data.shift() ) );
            node.offset = node.position.clone();
            break;
        case 'CHANNELS':
            n = parseInt( data.shift() );
            for ( var i = 0; 0 <= n ? i < n : i > n; 0
<= n ? i++ : i-- ) {
                this.channels.push({ node: node,
prop: data.shift() });
            }
            break;
        case 'JOINT':
        case 'End':
            node.add( this.parseNode(data) );
            break;
        case '}':
            done = true;
    }
}
this.nodes.push(node);
return node;
},
clearNode:function(){
    var i;
    if(this.nodes){
        for (i=0; i<this.nodes.length; i++){

```

```

        this.nodes[i] = null;
    }
    this.nodes.length = 0;
    if(this.bones.length > 0){
    for ( i=0; i<this.bones.length; i++){
        if(this.bones[i]){
            this.bones[i].geometry.dispose();
        }
    }
    this.bones.length = 0;
    scene.remove( this.skeleton );
    }
    }
},
animate:function(){
    var ch;
    var n = this.frame % this.numFrames * this.channels.length;
    var ref = this.channels;
    var isRoot = false;
    for ( var i = 0, len = ref.length; i < len; i++) {
        ch = ref[ i ];
        if(ch.node.name === "Hips") isRoot = true;
        else isRoot = false;
        switch ( ch.prop ) {
            case 'Xrotation':
                this.autoDetectRotation(ch.node, "X",
                parseFloat(this.data[n]));
                break;
            case 'Yrotation':
                this.autoDetectRotation(ch.node, "Y",
                parseFloat(this.data[n]));

```

```

        break;
    case 'Zrotation':
        this.autoDetectRotation(ch.node, "Z",
parseFloat(this.data[n]));
        break;
    case 'Xposition':
        if(isRoot) ch.node.position.x =
ch.node.offset.x + parseFloat(this.data[n])+ this.position.x;
        else ch.node.position.x =
ch.node.offset.x + parseFloat(this.data[n]);
        break;
    case 'Yposition':
        if(isRoot) ch.node.position.y =
ch.node.offset.y + parseFloat(this.data[n])+ this.position.y;
        else ch.node.position.y =
ch.node.offset.y + parseFloat(this.data[n]);
        break;
    case 'Zposition':
        if(isRoot) ch.node.position.z =
ch.node.offset.z + parseFloat(this.data[n])+ this.position.z;
        else ch.node.position.z =
ch.node.offset.z + parseFloat(this.data[n]);
        break;
    }
    n++;
}
if(this.bones.length > 0) this.updateSkeleton();
},
autoDetectRotation:function(Obj, Axe, Angle){
    this.tmpOrder+=Axe;
    var angle = Angle * BVH.TO_RAD;

```

```

    if(Axe === "X")this.tmpAngle[0] = angle;
    else if(Axe === "Y")this.tmpAngle[1] = angle;
    else this.tmpAngle[2] = angle;
    if(this.tmpOrder.length===3){
        var e = new THREE.Euler( this.tmpAngle[0], this.tmpAngle[1],
this.tmpAngle[2], this.tmpOrder );
        Obj.setRotationFromEuler(e);
        Obj.updateMatrixWorld();
        this.tmpOrder = "";
        this.tmpAngle.length = 0;
    }
},
update:function(){
    if ( this.play ) {
        this.frame = (((Date.now() - this.startTime) /
this.secsPerFrame / 1000) *this.speed)| 0;
        if(this.oldFrame!==0){this.frame += this.oldFrame; var
nt = this.frame * (1045 / this.numFrames);
document.getElementById("seekslider").value = nt;}
        if(this.frame > this.numFrames ){this.frame =
0;this.oldFrame=0; }
        this.animate();
    }
},
next:function(){
    this.play = false;
    this.frame ++;
    if(this.frame > this.numFrames ) this.frame = 0;
    this.animate();
},
prev:function(){

```

```

    this.play = false;
    this.frame --;
    if(this.frame<0)this.frame = this.numFrames;
    this.animate();
},
replay:function(){
    this.play = false;
    if(this.frame > 0){this.frame = 0;this.oldFrame=0;}
    this.animate();
},
decode: function (input) {
    var output = "";
    var chr1, chr2, chr3;
    var enc1, enc2, enc3, enc4;
    var i = 0;
    input = input.replace(/[^A-Za-z0-9\+\=\]/g, "");
    while (i < input.length) {
        enc1 = this._keyStr.indexOf(input.charAt(i++));
        enc2 = this._keyStr.indexOf(input.charAt(i++));
        enc3 = this._keyStr.indexOf(input.charAt(i++));
        enc4 = this._keyStr.indexOf(input.charAt(i++));
        chr1 = (enc1 << 2) | (enc2 >> 4);
        chr2 = ((enc2 & 15) << 4) | (enc3 >> 2);
        chr3 = ((enc3 & 3) << 6) | enc4;
        output = output + String.fromCharCode(chr1);
        if (enc3 !== 64) {
            output = output + String.fromCharCode(chr2);
        }
        if (enc4 !== 64) {
            output = output + String.fromCharCode(chr3);
        }
    }
}

```

```
    }  
    output = Base64._utf8_decode(output);  
    return output;  
  },  
  seekto:function(){  
    this.play = false;  
    var skt = parseFloat(this.numFrames *  
(document.getElementById("seekslider").value / 1045));  
    this.frame = parseInt(skt);  
    this.animate();  
  }  
}  
BVH.DistanceTest = function( p1, p2 ){  
  var x = p2.x-p1.x;  
  var y = p2.y-p1.y;  
  var z = p2.z-p1.z;  
  var d = Math.sqrt(x*x + y*y + z*z);  
  if(d<=0)d=0.1;  
  return d;  
}
```

ประวัติผู้เขียน

ชื่อ-นามสกุล: ชลธิศ เกาทอง

อีเมล: chonlatid.thao@bumail.net

ประวัติการศึกษา: ปริญญาตรี วิทยาศาสตร์บัณฑิต สาขาวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยกรุงเทพ



มหาวิทยาลัยกรุงเทพ

ข้อตกลงว่าด้วยการอนุญาตให้ใช้สิทธิในวิทยานิพนธ์/สารนิพนธ์

วันที่ 18 เดือน มกราคม พ.ศ. 2560

ข้าพเจ้า (นาย/นาง/นางสาว) ชลธิศ อกาทอง อยู่บ้านเลขที่ 705

ซอย ตากสิน 23 ถนน สมเด็จพระเจ้าตากสิน ตำบล/แขวง สำเหร่

อำเภอ/เขต ธนบุรี จังหวัด กรุงเทพฯ รหัสไปรษณีย์ 10600

เป็นนักศึกษาของมหาวิทยาลัยกรุงเทพ รหัสประจำตัว 7580700099

ระดับปริญญา ตรี โท เอก

หลักสูตร วิทยาศาสตรมหาบัณฑิต สาขาวิชา เทคโนโลยีสารสนเทศและการจัดการ

คณะ วิทยาศาสตร์และเทคโนโลยี ซึ่งต่อไปนี้เรียกว่า “ผู้อนุญาตให้ใช้สิทธิ” ฝ่ายหนึ่ง และ

มหาวิทยาลัยกรุงเทพ ตั้งอยู่เลขที่ 119 ถนนพระราม 4 แขวงพระโขนง เขตคลองเตย
กรุงเทพมหานคร 10110 ซึ่งต่อไปนี้เรียกว่า “ผู้ได้รับอนุญาตให้ใช้สิทธิ” อีกฝ่ายหนึ่ง

ผู้อนุญาตให้ใช้สิทธิ และ ผู้ได้รับอนุญาตให้ใช้สิทธิ ตกลงทำสัญญากันโดยมีข้อความดังต่อไปนี้

ข้อ 1. ผู้อนุญาตให้ใช้สิทธิขอรับรองว่าเป็นผู้สร้างสรรค์และเป็นผู้มีสิทธิแต่เพียงผู้เดียวในงานสารนิพนธ์/
วิทยานิพนธ์หัวข้อ การแสดงผลแอนิเมชันจากฐานข้อมูล การวัดความเคลื่อนไหวออนไลน์

ซึ่งถือเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิทยาศาสตรมหาบัณฑิต ของมหาวิทยาลัยกรุงเทพ
(ต่อไปนี้เรียกว่า “สารนิพนธ์/วิทยานิพนธ์”)

ข้อ 2. ผู้อนุญาตให้ใช้สิทธิตกลงยินยอมให้ผู้ได้รับอนุญาตให้ใช้สิทธิโดยปราศจากค่าตอบแทนและไม่มี
กำหนดระยะเวลาในการนำสารนิพนธ์/วิทยานิพนธ์ ซึ่งรวมถึงแต่ไม่จำกัดเพียงการทำซ้ำ ดัดแปลง เผยแพร่
ต่อสาธารณชน ให้เข้าต้นฉบับหรือสำเนา งาน ให้ประโยชน์อันเกิดจากลิขสิทธิ์แก่ผู้อื่น อนุญาตให้ผู้อื่นใช้
สิทธิโดยจะกำหนดเงื่อนไขอย่างหนึ่งอย่างใดด้วยหรือไม่ก็ได้ ไม่ว่าทั้งหมดหรือเพียงบางส่วน หรือการ
กระทำอื่นใดในลักษณะทำนองเดียวกัน

ข้อ 3. หากกรณีมีข้อขัดแย้งในปัญหาสิทธิในสารนิพนธ์/วิทยานิพนธ์ระหว่างผู้อนุญาตให้ใช้สิทธิกับ
บุคคลภายนอกก็ดี หรือระหว่างผู้ได้รับอนุญาตให้ใช้สิทธิกับบุคคลภายนอกก็ดี หรือมีเหตุขัดข้องอื่น ๆ
เกี่ยวกับลิขสิทธิ์ อันเป็นเหตุให้ผู้ได้รับอนุญาตให้ใช้สิทธิไม่สามารถนำงานนั้นออกทำซ้ำ เผยแพร่ หรือโฆษณา
ได้ ผู้อนุญาตให้ใช้สิทธิยินยอมรับผิดชอบและชดเชยค่าเสียหายแก่ผู้ได้รับอนุญาตให้ใช้สิทธิในความเสียหาย
ต่าง ๆ ที่เกิดขึ้นแก่ผู้ได้รับอนุญาตให้ใช้สิทธิทั้งสิ้น

สัญญาที่สร้างขึ้นสองฉบับ มีข้อความเป็นอย่างเดียวกัน คู่สัญญาได้อ่านและเข้าใจข้อความในสัญญาโดยละเอียดแล้ว จึงได้ลงลายมือชื่อให้ไว้เป็นสำคัญต่อหน้าพยาน และเก็บรักษาไว้ฝ่ายละฉบับ

ลงชื่อ.....ผู้อนุญาตให้ใช้สิทธิ
(จลรัตน์ เกษทอง)

ลงชื่อ.....ผู้ได้รับอนุญาตให้ใช้สิทธิ
(อาจารย์อภิญญา จุลพิสิฐ)
ผู้อำนวยการสำนักหอสมุดและศูนย์การเรียนรู้

ลงชื่อ.....พยาน
(ผู้ช่วยศาสตราจารย์กฤติกา ลีมลาวลัย)
รองคณบดีบัณฑิตวิทยาลัย

ลงชื่อ.....พยาน
(ดร.ถิรพล วงศ์อาดสกุล)
ผู้อำนวยการหลักสูตร/ ผู้รับผิดชอบหลักสูตร