



มาตรฐานอาชีพและคุณวุฒิวิชาชีพ
Occupational Standard and Professional Qualifications

สาขาวิชาชีพอุตสาหกรรมดิจิทัล สาขาเกม (Game)

จัดทำโดย สถาบันคุณวุฒิวิชาชีพ (องค์การมหาชน)
ร่วมกับ มหาวิทยาลัยศรีปทุม

1. ชื่อมาตรฐานอาชีพ

สาขาวิชาชีพอุตสาหกรรมดิจิทัล สาขาเกม (Game)

2. ประวัติการปรับปรุงมาตรฐาน

N/A

3. ทะเบียนอ้างอิง (Imprint)

N/A

4. ข้อมูลเบื้องต้น

เกม เป็นสิ่งที่สร้างขึ้นเพื่อสร้างความสนุกสนานบันเทิงให้กับผู้เล่น ซึ่งวิดีโอเกมเครื่องแรก ถูกสร้างขึ้นจากความเบื่อหน่ายในชีวิตการทำงานที่ทำอยู่ประจำของ นายวิลเลียม ฮิกกินบอธัม (William Higginbotham) นักฟิสิกส์นิวเคลียร์ ซึ่งสร้างจากคอมพิวเตอร์ที่ใช้คำนวณวิถีโค้งของจรวดมิสไซล์ โดยเรียกเกมนี้ว่า เทนนิสฟอรัท (Tennis for two) เป็นเกมที่ผู้เล่นจะใช้จอยส์ให้บอลสีเขียวข้ามไปยังอีกฝั่ง และในเวลาต่อมาวิดีโอเกมก็พัฒนาไปอีกขั้นโดย นาย ราล์ฟแบร์ (Ralph H. Baer) ได้ผลิตเครื่องเล่นในบ้าน ที่เล่นกันได้ 2 คน อีกทั้งในเวลาใกล้เคียงกัน นายอัลลัน อัลคอร์น (Allan Alcorn) (กันตพร วิรุบุตร, 2559) ได้พัฒนาเกมตู้ที่จำลองการเล่นปิงปองขึ้นมา ถือเป็นตู้เกมตู้แรกๆ ที่ได้รับความนิยมในยุคนั้น จากนั้นเกมก็ได้รับการพัฒนาขึ้นเรื่อย จากเกมตู้สู่เกมในเครื่องคอมพิวเตอร์ จากเกมแบบออฟไลน์พัฒนาเป็นเกมแบบออนไลน์ และกลายเป็นกีฬา (E-sport) ที่ได้รับการยอมรับและมีผู้คนให้ความสนใจเป็นจำนวนมาก (Krungsri, 2561) อุตสาหกรรมเกมกลายจึงกลายเป็นอุตสาหกรรมที่เติบโตอย่างรวดเร็วในช่วงหลายปีที่ผ่านมา

ประเทศไทยก็มีจำนวนผู้เล่นเกมอยู่จำนวนมาก ตั้งแต่ยุคของเกมตู้ที่ได้รับความนิยมอย่างมากคืออย่างเกม street fighter หรือเกมคอนโซลอย่างเครื่อง Family ที่ถือเป็นหนึ่งในเครื่องเกมที่ได้รับความนิยมอย่างมากในประเทศไทย ซึ่งเกมที่มีนิยมเล่นกัน เช่น เกมมาริโอ เกมเตอร์ดิส เป็นต้น แต่การเติบโตในอุตสาหกรรมเกมกลับไม่มากนัก เนื่องจากรูปแบบของเครื่องเกมที่ต้องใช้ตัวเกมในการเล่น ซึ่งตัวเกมส่วนใหญ่กลับเป็นของเถื่อน (ไทยรัฐ, 2557) แม้มีผู้เล่นมากแต่รายได้กลับไม่ไปถึงมือของผู้ผลิต ผู้ผลิตจึงไม่ต้องการลงทุนในไทย แต่เมื่อเทคโนโลยีมาพัฒนาให้เกิดแพลตฟอร์มใหม่ๆ มากขึ้น ทั้งเกมออนไลน์ เกมบนโทรศัพท์มือถือ เริ่มเข้ามามีอิทธิพล จึงก่อให้เกิดการเติบโตที่มากขึ้นตามมา จนกระทั่งเกมกลายเป็นกีฬาการแข่งขันจึงถือเป็นจุดเปลี่ยนสำคัญที่ช่วยขับเคลื่อนอุตสาหกรรมเกมในประเทศไทยให้ดีขึ้น

โดยอุตสาหกรรมเกมของประเทศไทยมีในปี 2560 มีการเติบโตอย่างต่อเนื่องที่ 10.3 พันล้านบาท ตลาดเกมไทยมีขนาดใหญ่ และ E-Sport ได้รับความนิยมมากที่สุด ในมีมูลค่าการใช้จ่ายในตลาดเกมอยู่ที่ 597 ล้านบาทต่อปี ในปี 2017 โดยมีอัตราการเติบโตต่อเนื่องที่ 25% ต่อปี โดยมีผู้เล่นเกมบนโทรศัพท์มือถือ 17 ล้านคิดเป็น 25% ของประชากรทั้งหมด ซึ่งจากการสำรวจของ Newzooพบว่ามากกว่าครึ่งมีอายุอยู่ที่ 21-35 ปี ซึ่งมูลค่าของตลาดเกมไทยนั้นอยู่ในอันดับที่ 20 ของโลก (Newzoo, 2018) ด้วยการเติบโตที่สูงขึ้นอย่างต่อเนื่อง จึงเกิดการสนับสนุนธุรกิจเกี่ยวกับเกมมากยิ่งขึ้น ตลาดจึงมีความต้องการบุคลากรที่เข้าใจเกี่ยวกับธุรกิจเกม ธุรกิจ E-Sport ผลิตเกม นักสร้างเกม ในจำนวนที่มากขึ้นตามไปด้วย

การเติบโตของอุตสาหกรรมเกม การพัฒนาของกีฬา E-Sport ก่อให้เกิดอาชีพใหม่ๆ ขึ้นในอุตสาหกรรมนี้ ทั้งนักพากย์เกม ผู้คุมเกม รวมไปถึงนักออกแบบพัฒนาเกม ซึ่งภาครัฐ ภาคเอกชน รวมไปถึงสถาบันต่างๆ หันมาให้ความสำคัญเกี่ยวกับธุรกิจนี้มากขึ้น โดยสถาบันการศึกษาหลายแห่งได้จัดหลักสูตรเพื่อพัฒนาบุคลากรในสายอาชีพเกี่ยวกับเกมมากขึ้น ตอบสนองต่อความต้องการของตลาด เช่น มหาวิทยาลัยศรีปทุม ได้เปิดคณะดิจิทัลมีเดีย ที่เปิดสาขาวิชาการออกแบบอินเตอร์แอคทีฟและเกม ที่เน้นพัฒนาเกม และคนในวงการเกม (สิริลักษณ์ เล่า, 2561) ให้สอดคล้องกับความต้องการที่เพิ่มขึ้นในปัจจุบัน เมื่อมีผู้มีความรู้ และทักษะในด้านนี้มากขึ้น สายอาชีพด้านนี้เริ่มมีให้เห็นเด่นชัดขึ้น จึงต้องมีการสร้างมาตรฐานอาชีพ เพื่อรองรับความรู้ และความเชี่ยวชาญ และทักษะทางด้านการผลิตเกม กำหนดสมรรถนะให้เป็นมาตรฐานสากล และเพื่อให้หน่วยงานทั้งภาครัฐและเอกชนมีความเข้าใจที่ถูกต้องตรงกันในเรื่องของอาชีพนี้ และเป็นประโยชน์ในการบริหารด้านแรงงานต่อไป

5. ประวัติการปรับปรุงมาตรฐานในแต่ละครั้ง

N/A

6. ครั้งที่

1

7. คุณวุฒิวิชาชีพที่ครอบคลุม (Professional Qualifications included)

สาขาวิชาชีพอุตสาหกรรมดิจิทัล

สาขาเกม

อาชีพนักพัฒนาโปรแกรมเกม ระดับ 4

8. คุณวุฒิวิชาชีพที่เกี่ยวข้อง (Related Professional Qualifications)

N/A

9. หน่วยสมรรถนะทั้งหมดในมาตรฐานอาชีพ (List of All Units of Competence within this Occupational Standards)

รหัสหน่วยสมรรถนะ	เนื้อหา
3011	เขียนโปรแกรมเกมโดยใช้ไลบรารีหรือเครื่องมือ (Libraries/Tools) ของเกมเอนจิน (Game Engine)
3012	ทดสอบ และแก้ไข ปัญหาที่เกิดขึ้นจากการเขียนโปรแกรมเกม
3013	เขียนโปรแกรมเพื่อเพิ่มประสิทธิภาพของเกม (Optimization)
3021	ใช้ระบบจัดการการเปลี่ยนแปลงที่เกิดขึ้นกับไฟล์ในการพัฒนาเกม (Version Control)

10. ระดับคุณวุฒิ

10.1 สาขาวิชาชีพอุตสาหกรรมดิจิทัล สาขาเกม อาชีพนักพัฒนาโปรแกรมเกม ระดับ 4

คุณลักษณะของผลการเรียนรู้ (Characteristics of Outcomes)

บุคคลที่มีคุณลักษณะของผลการเรียนรู้ที่พึงประสงค์ในอาชีพนักพัฒนาโปรแกรมเกม ระดับ 4 ผู้มีสมรรถนะทางเทคนิคในด้านการพัฒนาโปรแกรมเกม โดยใช้ Game Engine เป็นเครื่องมือ และสามารถทดสอบโปรแกรมเกมได้ รวมทั้งสามารถแก้ไขข้อผิดพลาด ที่เกิดขึ้นจากการพัฒนาโปรแกรมเกมได้ และให้คำแนะนำ ถ่ายทอด ประสบการณ์ ทักษะ และความรู้ พร้อมมีเจตคติที่ดี มีคุณธรรม จริยธรรมในการประกอบวิชาชีพ

การเลื่อนระดับคุณวุฒิวิชาชีพ (Qualification Pathways)

- คุณสมบัติของผู้ที่สามารถเข้ารับการประเมินคุณวุฒิวิชาชีพ สาขาวิชาชีพเทคโนโลยีสารสนเทศ และการสื่อสาร และดิจิทัลคอนเทนต์ สาขาเกม อาชีพนักพัฒนาโปรแกรมเกม ระดับ 4 ต้องมีประสบการณ์เกี่ยวกับนักพัฒนาโปรแกรมเกม หรือ ที่เกี่ยวข้องอย่างน้อย 1 ปี
 - ผู้ที่สำเร็จการศึกษาระดับประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.) หรือเทียบเท่า ในด้านการ พัฒนาโปรแกรมหรือที่เกี่ยวข้อง หรือ
 - ผู้ที่กำลังศึกษาระดับปริญญาตรี ในด้านการพัฒนาโปรแกรมหรือที่เกี่ยวข้อง หรือ
 - มีประสบการณ์ทำงานด้านนักพัฒนาโปรแกรมเกม หรือที่เกี่ยวข้องไม่น้อยกว่า 1 ปี
- ผู้ที่ผ่านการประเมินและได้รับการรับรองคุณวุฒิวิชาชีพ สาขาวิชาชีพอุตสาหกรรมดิจิทัล สาขาเกม อาชีพนักพัฒนาโปรแกรมเกม ระดับ 4
 - ผ่านเกณฑ์การประเมินตามหน่วยสมรรถนะของอาชีพนักพัฒนาโปรแกรมเกม (Game Programmer) ระดับ 4 จำนวน 4 หน่วย

ในกรณีต่ออายุหนังสือรับรองมาตรฐานอาชีพให้เป็นไปตามคู่มือสำหรับผู้เข้ารับการประเมินหรือคู่มือเจ้าหน้าที่สอบ

หลักเกณฑ์การต่ออายุหนังสือรับรองมาตรฐานอาชีพ

N/A

กลุ่มบุคคลในอาชีพ (Target Group)

ผู้ปฏิบัติงานด้านการพัฒนาโปรแกรมเกม หรืออื่นๆ ที่เกี่ยวข้อง

หน่วยสมรรถนะ (หน่วยสมรรถนะทั้งหมดของคุณวุฒิวิชาชีพนี้)

- 3011 เขียนโปรแกรมเกมโดยใช้ไลบรารีหรือเครื่องมือ (Libraries/Tools) ของเกมเอนจิน (Game Engine)
- 3012 ทดสอบ และแก้ไข ปัญหาที่เกิดขึ้นจากการเขียนโปรแกรมเกม
- 3013 เขียนโปรแกรมเพื่อเพิ่มประสิทธิภาพของเกม (Optimization)
- 3021 ใช้ระบบจัดการการเปลี่ยนแปลงที่เกิดขึ้นกับไฟล์ในการพัฒนาเกม (Version Control)

ตารางแผนผังแสดงหน้าที่

1. ตารางแสดงหน้าที่ 1

ประกาศใช้ ณ 19/04/2564

ตาราง 1 : FUNCTIONAL MAP แสดง KEY PURPOSE , KEY ROLES , KEY FUNCTION

ความมุ่งหมายหลัก Key Purpose	บทบาทหลัก Key Roles		หน้าที่หลัก Key Function	
	รหัส	คำอธิบาย	รหัส	คำอธิบาย
พัฒนาศักยภาพของบุคลากรในสาขาอาชีพเกมให้สามารถแข่งขันและเป็นที่ยอมรับในระดับสากล	30	ปฏิบัติงานด้านนักพัฒนาโปรแกรมเกม	301	ปฏิบัติการด้านการพัฒนาเกม
			302	ปฏิบัติการด้านการจัดเก็บการเปลี่ยนแปลงที่เกิดขึ้นกับไฟล์ (Version Control)

คำอธิบาย ตารางแผนผังแสดงหน้าที่เป็นแผนผังที่ใช้วิเคราะห์หน้าที่งานเพื่อให้ได้หน้าที่หลัก (Key Function)

2. ตารางแสดงหน้าที่ 1 (ต่อ)

ประกาศใช้ ณ 19/04/2564

ตาราง 2 : FUNCTIONAL MAP แสดง KEY FUNCTION , UNIT OF COMPETENCE , ELEMENT OF COMPETENCE

หน้าที่หลัก Key Function		หน่วยสมรรถนะ Unit of Competence		หน่วยสมรรถนะย่อย Element of Competence	
รหัส	คำอธิบาย	รหัส	คำอธิบาย	รหัส	คำอธิบาย
301	ปฏิบัติการด้านการพัฒนาเกม	3011	เขียนโปรแกรมเกมโดยใช้ไลบรารีหรือเครื่องมือ (Libraries/Tools) ของเกมเอนจิน (Game Engine)	30111	วิเคราะห์ความต้องการของเอกสารออกแบบ และจัดเตรียม Assets
				30112	เขียนโปรแกรมตามเอกสารออกแบบ (Design Document)
				30113	ทดสอบหาข้อผิดพลาดของโปรแกรม
		3012	ทดสอบ และแก้ไข ปัญหาที่เกิดขึ้นจากการเขียนโปรแกรมเกม	30121	แก้ไขข้อผิดพลาดของโปรแกรมที่ตรวจพบ
				30122	เขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming) ได้อย่างมีประสิทธิภาพ
		3013	เขียนโปรแกรมเพื่อเพิ่มประสิทธิภาพของเกม (Optimization)	30131	ใช้ Data Structures และ Variables ได้อย่างมีประสิทธิภาพ
				30132	ใช้ Statements และ ประยุกต์ใช้ Algorithms ได้อย่างมีประสิทธิภาพ
302	ปฏิบัติการด้านการจัดการเก็บการเปลี่ยนแปลง ที่เกิดขึ้นกับไฟล์ (Version Control)	3021	ใช้ระบบจัดการเก็บการเปลี่ยนแปลงที่เกิดขึ้นกับไฟล์ในการพัฒนาเกม (Version Control)	30211	ใช้งานระบบจัดการเก็บการเปลี่ยนแปลงที่เกิดขึ้นกับไฟล์ได้ (Version Control Management)
				30212	ใช้งานการแยกเป็นสาขาของระบบจัดการเก็บการเปลี่ยนแปลงที่เกิดขึ้นกับไฟล์ (Version Control Branching Management)

คำอธิบาย

ตารางแผนผังแสดงหน้าที่ (ต่อ) เป็นแผนผังที่ใช้วิเคราะห์หน้าที่งานหลังจากได้หน้าที่หลัก (Key Function) เพื่อให้ได้ หน่วยสมรรถนะ (Unit of Competence) และหน่วยสมรรถนะย่อย (Element of Competence)

1. รหัสหน่วยสมรรถนะ 3011
2. ชื่อหน่วยสมรรถนะ เขียนโปรแกรมเกมโดยใช้ไลบรารีหรือเครื่องมือ (Libraries/Tools) ของเกมเอนจิน (Game Engine)
3. ทบทวนครั้งที่ - / -
4. สร้างใหม่ ปรับปรุง
5. สำหรับชื่ออาชีพและรหัสอาชีพ (Occupational Classification)

อาชีพนักพัฒนาโปรแกรมเกม

6. คำอธิบายหน่วยสมรรถนะ (Description of Unit of Competency)

บุคคลที่ผ่านสมรรถนะนี้จะต้องมีความรู้ความเข้าใจ การใช้ไลบรารีหรือเครื่องมือ (Libraries/Tools) ของเกมเอนจิน ที่ใช้พัฒนาเกมได้ สามารถตรวจสอบความถูกต้องของ Assets ที่จะนำมาใช้ได้ และสามารถจัดทำเอกสารทางเทคนิค (Technical Document)

7. สำหรับระดับคุณวุฒิ

1	2	3	4	5	6	7	8
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

8. กลุ่มอาชีพ (Sector)

ผู้ปฏิบัติงานด้านการพัฒนาโปรแกรมเกม หรืออื่นๆ ที่เกี่ยวข้อง

9. ชื่ออาชีพและรหัสอาชีพอื่นที่หน่วยสมรรถนะนี้สามารถใช้ได้ (ถ้ามี)

N/A

10. ข้อกำหนดหรือกฎระเบียบที่เกี่ยวข้อง (Licensing or Regulation Related) (ถ้ามี)

N/A

11. สมรรถนะย่อยและเกณฑ์การปฏิบัติงาน (Elements and Performance Criteria)

สมรรถนะย่อย (Element)	เกณฑ์ในการปฏิบัติงาน (Performance Criteria)	วิธีการประเมิน (Assessment)
30111 วิเคราะห์ความต้องการของเอกสารออกแบบ และจัดเตรียม Assets	1. ระบุและรวบรวม Assets ทั้งหมด ที่ต้องใช้ในงาน 2. ตรวจสอบความถูกต้องของ Assets และ จัดเก็บ Assets	
30112 เขียนโปรแกรมตามเอกสารออกแบบ (Design Document)	1. เลือกใช้ ไลบรารีหรือเครื่องมือ (Libraries/Tools) ใน Game Engine 2. ใช้ ไลบรารีหรือเครื่องมือ (Libraries/Tools) และ Assets โดยกำหนดค่าได้ตรงตามเอกสารออกแบบ 3. เขียน Game Logic เพิ่มเติมได้ เอกสารออกแบบทางเทคนิค (Technical Document) 4. จัดทำเอกสาร (Programming Document)	

12. ความรู้และทักษะก่อนหน้าที่จำเป็น (Pre-requisite Skill & Knowledge)

มีความรู้และทักษะพื้นฐานการเขียนโปรแกรม สามารถวิเคราะห์เอกสารออกแบบและสามารถเขียนโปรแกรมตามเอกสารออกแบบได้ โดยใช้ เกมเอนจิน (Game Engine) เป็นเครื่องมือ และสามารถจัดทำเอกสารทางเทคนิค (Technical Document) ได้

13. ทักษะและความรู้ที่ต้องการ (Required Skills and Knowledge)

(ก) ความต้องการด้านทักษะ

1. ทักษะการเขียนโปรแกรมคอมพิวเตอร์
2. ทักษะการตรวจสอบความถูกต้องของ Assets และ จัดเก็บ Assets
3. ทักษะการใช้ โลบรารีหรือเครื่องมือ (Libraries/Tools) และ Assets โดยกำหนดค่าได้ตรงตามที่กำหนด
4. ทักษะการอ่าน Code จากเครื่องมือพื้นฐาน (Tools) ของเกมเอนจินที่เลือกใช้
5. ทักษะการเขียน Game Logic เพิ่มเติมได้ตรงกับตามที่กำหนด
6. ทักษะการจัดทำเอกสารทางเทคนิค (Technical Document)

(ข) ความต้องการด้านความรู้

1. มีความรู้เกี่ยวกับ Assets ทั้งหมดที่จำเป็นต้องใช้
2. มีความรู้เกี่ยวกับการเลือกใช้ Libraries/Tools ใน Game Engine ที่เลือกใช้

14. หลักฐานที่ต้องการ (Evidence Guide)

หลักฐานที่จะต้องกำหนดข้อแนะนำเกี่ยวกับการประเมินและควรที่จะใช้ประกอบร่วมกันกับเกณฑ์การปฏิบัติงาน (Performance Criteria) และทักษะและความรู้ที่ต้องการ (Required Skill and Knowledge)

(ก) หลักฐานการปฏิบัติงาน (Performance Evidence)

1. ไปบันทึกผลการทดสอบภาคปฏิบัติ

(ข) หลักฐานความรู้

1. ผลการสอบแบบทดสอบปรนัย

(ค) คำแนะนำในการประเมิน

1. เจ้าหน้าที่ที่สอบตรวจประเมินหลักฐานโดยพิจารณาจากร่องรอยหลักฐานที่เกี่ยวข้อง ทั้งหลักฐานด้านปฏิบัติงานและหลักฐานด้านความรู้

(ง) วิธีการประเมิน

1. แบบทดสอบปรนัย
2. แบบทดสอบภาคปฏิบัติ

15. ขอบเขต (Range Statement)

ขอบเขตอธิบายถึงขอบเขตของการปฏิบัติงาน และสภาพแวดล้อมอื่นๆ หรือสถานการณ์อื่นๆ ที่มีผลกระทบต่อการทำงานรวมถึงเครื่องมือ อุปกรณ์ เทคโนโลยี ทรัพยากรที่ใช้หรือข้อกำหนดอื่นๆ ที่เกี่ยวข้อง

(ก) คำแนะนำ

1. ในการปฏิบัติงานให้คำนึงถึงทักษะการวิเคราะห์ความต้องการของเอกสารออกแบบ และจัดเตรียม Assets โดยสามารถระบุ Assets ทั้งหมดและตรวจสอบความถูกต้องของ Assets ให้เหมาะสมกับเกมเอนจินที่ใช้ นอกจากนี้ยังต้องคำนึงถึงการเขียนโปรแกรมโดยใช้เกมเอนจินโดยต้องมีความรู้ความเข้าใจโลบรารีหรือเครื่องมือ (Libraries/Tools) ของเกมเอนจินที่เลือกใช้ สามารถเขียนโปรแกรม Logic ให้เข้ากับเกมเอนจินที่เลือกใช้อย่างเหมาะสม และสามารถจัดทำเกมเอกสารทางเทคนิค (Technical Document) ให้โปรแกรมเมอร์คนอื่นเข้าใจในงานที่ได้ทำไป

(ข) คำอธิบายรายละเอียด

1. เกมเอนจิน (Game Engine) หมายถึง โปรแกรมคอมพิวเตอร์เพื่อใช้สร้างวิดีโอเกม ซึ่งโปรแกรมเหล่านี้ถูกเขียนขึ้นมาจากภาษาคอมพิวเตอร์ต่างๆ อาทิ C, C++, JAVA เป็นต้น เกมเอนจินเปรียบเสมือนเป็นเครื่องมือใช้สร้างชิ้นส่วนต่างๆ ของระบบเกม ชิ้นส่วนเหล่านี้ จะถูกนำไปประกอบเข้าด้วยกัน และถูกทดสอบว่าสามารถทำงานร่วมกันได้อย่างไม่มีปัญหา ภายใต้สภาพแวดล้อมที่ถูกกำหนดไว้ เป็นกระบวนการในการสร้างเกมให้ออกมาสมบูรณ์ นักพัฒนาเกมจำนวนมากใช้เกมเอนจินในการสร้างเกมสำหรับเครื่องเล่น มือถือ ตลอดจนคอมพิวเตอร์พกพาการทำงานของเกมเอนจิน ประกอบด้วย การเรนเดอร์กราฟิก 2 มิติ 3 มิติ การตรวจสอบการชนระบบเสียง ภาษาสคริปต์ แอนิเมชัน ปัญญาประดิษฐ์ ระบบเน็ตเวิร์ค การสตรีม การจัดการหน่วยความจำ การรองรับภาษาที่แปล ตลอดจนระบบอื่นๆ อีกมากมาย
2. Programming Logic หมายถึง การเขียนโปรแกรมเชิงตรรกะ (อังกฤษ: Logic programming) เป็นแนวทางการเขียนโปรแกรม (programming paradigm) แบบหนึ่ง โดยกำหนดเซตคุณลักษณะ (attribute) ของคำตอบ แทนที่จะกำหนดขั้นตอนที่ทำให้ได้คำตอบ การเขียนโปรแกรมแบบนี้มีหลักการคือ ความจริง + กฎ = ผลลัพธ์ หลักการอื่นที่แตกต่างให้ดู การเขียนโปรแกรมตรรกะเชิงอุปนัย (Inductive logic programming)
3. โลบรารี (Library/Tool) หมายถึง คลังโปรแกรม หรือ ในทางวิทยาการคอมพิวเตอร์ คือส่วนที่รวบรวมกระบวนการ (process) และฟังก์ชันย่อย (subroutine) ต่างๆ

ซึ่งอาจจะรวมรหัสต้นฉบับ (source code) ที่จำเป็นต้องใช้ในการเขียนซอฟต์แวร์ หรือ ใช้ในการทำงานของโปรแกรมหนึ่ง Library/Tool ที่เกี่ยวกับการพัฒนาเกมมีมากมายหลายชนิดเช่น Graphic Artificial Intelligent (AI) และ Physic เป็นต้น

4. Logic หมายถึงตรรกศาสตร์ เป็นวิชาที่ว่าด้วยกฎเกณฑ์และเหตุผล การได้มาของผลภายใต้กฎเกณฑ์ที่กำหนดถือเป็นสาระสำคัญ ข้อความหรือการให้เหตุผลในชีวิตประจำวันสามารถสร้างเป็นรูปแบบที่ชัดเจนจนใช้ประโยชน์ในการสรุปความ ความสมเหตุสมผลเป็นที่ยอมรับกันอย่างกว้างขวาง
5. Asset หมายถึงสิ่งต่างๆ ที่นำเข้ามาในโปรเจกต์เพื่อใช้ในการพัฒนาเกม ไม่ว่าจะเป็น โมเดล สคริปต์ แอนิเมชัน Texture และอื่นๆ ล้วนถือเป็น Asset ทั้งสิ้น ซึ่ง Asset จะถูกเก็บไว้ในไดเรกทอรีที่ชื่อว่า Assets ภายในโปรเจกต์ ถ้าหากไม่ได้อยู่ในไดเรกทอรีนี้จะไม่สามารถใช้ในการพัฒนาเกมได้
6. เอกสารทางเทคนิค (Technical Document) หมายถึง เอกสารทางเทคนิคในทางวิศวกรรมซอฟต์แวร์เป็นคำศัพท์ที่ครอบคลุมเอกสารทั้งหมดที่เกี่ยวข้องกับการพัฒนาผลิตภัณฑ์ซอฟต์แวร์ และเอกสารประเภทต่างๆ นั้นถูกสร้างขึ้นผ่านวงจรการพัฒนาซอฟต์แวร์ทั้งหมด Software Development Lifecycle (SDLC) เอกสารนี้มีไว้เพื่ออธิบายการทำงานของฟังก์ชันต่างๆ ในซอฟต์แวร์ข้อมูลที่เกี่ยวข้องกับโปรเจกต์โดยรวม และการอภิปรายคำถามที่สำคัญทั้งหมดที่เกิดขึ้นระหว่างผู้มีส่วนได้เสียและผู้พัฒนาโปรแกรม
7. เอกสารออกแบบ (Design Document หรือ Software Design Document) หมายถึง เอกสารการออกแบบซอฟต์แวร์เป็นคำอธิบายที่เป็นลายลักษณ์อักษรของผลิตภัณฑ์ซอฟต์แวร์ที่นำออกแบบซอฟต์แวร์เขียนเพื่อให้คำแนะนำโดยรวมแก่ทีมพัฒนาซอฟต์แวร์เกี่ยวกับสถาปัตยกรรมของโครงสร้างซอฟต์แวร์ มักจะมาพร้อมกับไดอะแกรมสถาปัตยกรรมซอฟต์แวร์เพื่อระบุรายละเอียดคุณลักษณะของชิ้นส่วนของซอฟต์แวร์ฟังก์ชันเล็กๆ ของโครงสร้างซอฟต์แวร์เพื่อให้ทีมพัฒนาซอฟต์แวร์สามารถแบ่งงานกันเป็นทีมได้

16. หน่วยสมรรถนะร่วม (ถ้ามี)

N/A

17. อุตสาหกรรมร่วม/กลุ่มอาชีพร่วม (ถ้ามี)

N/A

18. รายละเอียดกระบวนการและวิธีการประเมิน (Assessment Description and Procedure)

1. แบบทดสอบปรนัย
2. แบบทดสอบภาคปฏิบัติ

1. รหัสหน่วยสมรรถนะ 3012
2. ชื่อหน่วยสมรรถนะ ทดสอบ และแก้ไข ปัญหาที่เกิดขึ้นจากการเขียนโปรแกรมเกม
3. ทบทวนครั้งที่ - / -
4. สร้างใหม่ ปรับปรุง

5. สำหรับชื่ออาชีพและรหัสอาชีพ (Occupational Classification)

อาชีพนักพัฒนาโปรแกรมเกม

6. คำอธิบายหน่วยสมรรถนะ (Description of Unit of Competency)

บุคคลที่ผ่านสมรรถนะนี้จะต้องมีความรู้ความเข้าใจ ในการทดสอบ และแก้ไข ปัญหาที่เกิดขึ้นจากการเขียนโปรแกรมเกม

7. สำหรับระดับคุณวุฒิ

1	2	3	4	5	6	7	8
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

8. กลุ่มอาชีพ (Sector)

ผู้ปฏิบัติงานด้านการพัฒนาโปรแกรมเกม หรืออื่นๆ ที่เกี่ยวข้อง

9. ชื่ออาชีพและรหัสอาชีพอื่นที่หน่วยสมรรถนะนี้สามารถใช้ได้ (ถ้ามี)

N/A

10. ข้อกำหนดหรือกฎระเบียบที่เกี่ยวข้อง (Licensing or Regulation Related) (ถ้ามี)

N/A

11. สมรรถนะย่อยและเกณฑ์การปฏิบัติงาน (Elements and Performance Criteria)

สมรรถนะย่อย (Element)	เกณฑ์ในการปฏิบัติงาน (Performance Criteria)	วิธีการประเมิน (Assessment)
30121 ทดสอบหาข้อผิดพลาดของโปรแกรม	1. ระบุวิธีการและเลือกใช้เครื่องมือในการทดสอบโปรแกรม 2. ค้นหาปัญหาและข้อผิดพลาดของโปรแกรม	
30122 แก้ไขข้อผิดพลาดของโปรแกรมที่ตรวจพบ	1. ระบุสาเหตุของปัญหาที่เกิดขึ้น 2. แก้ไขปัญหาโปรแกรมที่เกิดขึ้น	

12. ความรู้และทักษะก่อนหน้าที่จำเป็น (Pre-requisite Skill & Knowledge)

มีความรู้และทักษะในการทดสอบ และแก้ไข ปัญหาที่เกิดขึ้นจากการเขียนโปรแกรม

13. ทักษะและความรู้ที่ต้องการ (Required Skills and Knowledge)

(ก) ความต้องการด้านทักษะ

- ทักษะการเขียนโปรแกรมคอมพิวเตอร์
- ทักษะการอ่าน Code จากเครื่องมือพื้นฐาน (Tools) ของเกมเอนจินที่เลือกใช้
- ทักษะการแก้ไขปัญหาหรือข้อผิดพลาดของโปรแกรมที่เกิดขึ้น
- ทักษะการใช้เครื่องมือหาข้อผิดพลาดของโปรแกรมต่างๆ

(ข) ความต้องการด้านความรู้

- มีความรู้เกี่ยวกับการกำหนดวิธีการ ทฤษฎี และเครื่องมือ ที่ใช้ในการทดสอบโปรแกรม
- มีความรู้เกี่ยวกับการอ่าน Error Code จากเครื่องมือ
- มีความรู้เกี่ยวกับการใช้เทคนิคการทดสอบหาข้อผิดพลาดของโปรแกรม
- มีความรู้เรื่องเกี่ยวกับ Programming Logic

14. หลักฐานที่ต้องการ (Evidence Guide)

หลักฐานที่จะต้องกำหนดข้อแนะนำเกี่ยวกับการประเมินและควรที่จะใช้ประกอบรวมกันกับเกณฑ์การปฏิบัติงาน (Performance Criteria) และทักษะและความรู้ที่ต้องการ (Required Skill and Knowledge)

(ก) หลักฐานการปฏิบัติงาน (Performance Evidence)

1. ใบบันทึกผลการทดสอบภาคปฏิบัติ

(ข) หลักฐานความรู้ (Knowledge Evidence)

1. ผลการสอบแบบทดสอบปรนัย

(ค) คำแนะนำในการประเมิน

1. เจ้าหน้าที่ที่สอบตรวจประเมินหลักฐานโดยพิจารณาจากร่องรอยหลักฐานที่เกี่ยวข้อง ทั้งหลักฐานด้านปฏิบัติงานและหลักฐานด้านความรู้

(ง) วิธีการประเมิน

1. แบบทดสอบปรนัย

2. แบบทดสอบภาคปฏิบัติ

15. ขอบเขต (Range Statement)

ขอบเขตอธิบายถึงขอบเขตของการปฏิบัติงาน และสภาพแวดล้อมอื่นๆ หรือสถานการณ์อื่นๆ ที่มีผลกระทบต่อการทำงานรวมถึงเครื่องมือ อุปกรณ์ เทคโนโลยี ทรัพยากรที่ใช้หรือข้อกำหนดอื่นๆ ที่เกี่ยวข้อง

(ก) คำแนะนำ

1. ในการปฏิบัติงานการค้นหาและแก้ไขข้อผิดพลาดของโปรแกรมให้คำนึงถึงเครื่องมือและเทคนิคการค้นหาข้อผิดพลาดของเกมเอนจินที่เลือกใช้เป็นหลัก สามารถใช้เครื่องมือที่มีอยู่ในระบบปฏิบัติการ Microsoft Window ช่วยเหลือได้แต่ไม่จำเป็น การทดสอบข้อผิดพลาดของโปรแกรมในขั้นนี้จะเป็นการทดสอบความผิดพลาดของ Program Logic พื้นฐานไม่ลงลึกถึงระดับไลบรารีหรือเครื่องมือ (Libraries/Tools)

(ข) คำอธิบายรายละเอียด

1. เกมเอนจิน (Game Engine) หมายถึง โปรแกรมคอมพิวเตอร์ที่ใช้สร้างวิดีโอเกม ซึ่งโปรแกรมเหล่านี้ถูกเขียนขึ้นมาจากภาษาคอมพิวเตอร์ต่างๆ อาทิ C, C++, JAVA เป็นต้น เกมเอนจินเปรียบเสมือนเป็นเครื่องมือใช้สร้างชิ้นส่วนต่างๆ ของระบบเกม ชิ้นส่วนเหล่านี้ จะถูกนำไปประกอบเข้าด้วยกัน และถูกทดสอบว่าสามารถทำงานร่วมกันได้อย่างไม่มีปัญหา ภายใต้สภาพแวดล้อมที่ถูกกำหนดไว้ เป็นกระบวนการในการสร้างเกมให้ออกมาสมบูรณ์ นักพัฒนาเกมจำนวนมากใช้เกมเอนจินในการสร้างเกมสำหรับเครื่องเล่น มือถือ ตลอดจนคอมพิวเตอร์พกพาการทำงานของเกมเอนจิน ประกอบด้วย การเรนเดอร์กราฟิก 2 มิติ 3 มิติ การตรวจสอบการชนระบบเสียง ภาษาสคริปต์ แอนิเมชัน ปัญญาประดิษฐ์ ระบบเน็ตเวิร์ค การสตรีม การจัดการหน่วยความจำ การรองรับภาษาที่แปล ตลอดจนระบบอื่นๆ อีกมากมาย
2. Programming Logic หมายถึง การเขียนโปรแกรมเชิงตรรกะ (อังกฤษ: Logic programming) เป็นแนวทางการเขียนโปรแกรม (programming paradigm) แบบหนึ่ง โดยกำหนดเขตคุณลักษณะ (attribute) ของคำตอบ แทนที่จะกำหนดขั้นตอนที่ทำให้ได้คำตอบ การเขียนโปรแกรมแบบนี้มีหลักการคือ ความจริง + กฎ = ผลลัพธ์ หลักการอื่นที่แตกต่างให้ดู การเขียนโปรแกรมตรรกะเชิงอุปนัย (Inductive logic programming)
3. ไลบรารี (Library/Tool) หมายถึง คลังโปรแกรม หรือ ในทางวิทยาการคอมพิวเตอร์ คือส่วนที่รวบรวมกระบวนการ (process) และฟังก์ชันย่อย (subroutine) ต่างๆ ซึ่งอาจจะรวมรหัสต้นฉบับ (source code) ที่จำเป็นต้องใช้ในการเขียนซอฟต์แวร์ หรือ ใช้ในการทำงานของโปรแกรมหนึ่ง Library/Tool ที่เกี่ยวกับการพัฒนาเกมมีมากมายหลายชนิดเช่น Graphic Artificial Intelligent (AI) และ Physic เป็นต้น
4. ทดสอบหาข้อผิดพลาดของโปรแกรม (Debugging) หรือการดีบัก หมายถึง กระบวนการค้นหาและแก้ไขข้อบกพร่องหรือปัญหาภายในโปรแกรมคอมพิวเตอร์ หลักสำคัญของการดีบักคือการใช้เครื่องมือตรวจสอบของเกมเอนจินที่ใช้ได้ รวมถึงเครื่องมืออื่นๆ ในระบบปฏิบัติการที่เลือกใช้เช่น Microsoft Window เทคนิคการดีบักมีหลากหลายสามารถใช้เทคนิคใดก็ได้ในการค้นหาข้อผิดพลาดของโปรแกรม ตัวอย่างเทคนิคการดีบักเช่น interactive debugging, control flow analysis, unit testing, integration testing, log file analysis, monitoring at the application or system level, memory dumps และเทคนิคอื่นๆ ที่เกี่ยวข้อง ในการดีบักในขั้นนี้จะทดสอบการดีบักเฉพาะส่วนของโปรแกรม Logic ที่ไม่ลงลึกถึงไลบรารีหรือเครื่องมือ (Libraries/Tools) ของเกมเอนจิน

16. หน่วยสมรรถนะรวม (ถ้ามี)

N/A

17. อุตสาหกรรมร่วม/กลุ่มอาชีพร่วม (ถ้ามี)

N/A

18. รายละเอียดกระบวนการและวิธีการประเมิน (Assessment Description and Procedure)

1. แบบทดสอบปรนัย
2. แบบทดสอบภาคปฏิบัติ

1. รหัสหน่วยสมรรถนะ 3013
2. ชื่อหน่วยสมรรถนะ เขียนโปรแกรมเพื่อเพิ่มประสิทธิภาพของเกม (Optimization)
3. ทบทวนครั้งที่ - / -
4. สร้างใหม่ ปรับปรุง

5. สำหรับชื่ออาชีพและรหัสอาชีพ (Occupational Classification)

อาชีพนักพัฒนาโปรแกรมเกม

6. คำอธิบายหน่วยสมรรถนะ (Description of Unit of Competency)

บุคคลที่ผ่านสมรรถนะนี้จะต้องมีความรู้ความเข้าใจ ในพื้นฐานการเขียนโปรแกรมอย่างถูกหลักการเพื่อให้โปรแกรมสามารถทำงานได้อย่างมีประสิทธิภาพสูงสุด

7. สำหรับระดับคุณวุฒิ

1	2	3	4	5	6	7	8
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

8. กลุ่มอาชีพ (Sector)

ผู้ปฏิบัติงานด้านการพัฒนาโปรแกรมเกม หรืออื่นๆ ที่เกี่ยวข้อง

9. ชื่ออาชีพและรหัสอาชีพอื่นที่หน่วยสมรรถนะนี้สามารถใช้ได้ (ถ้ามี)

N/A

10. ข้อกำหนดหรือกฎระเบียบที่เกี่ยวข้อง (Licensing or Regulation Related) (ถ้ามี)

N/A

11. สมรรถนะย่อยและเกณฑ์การปฏิบัติงาน (Elements and Performance Criteria)

สมรรถนะย่อย (Element)	เกณฑ์ในการปฏิบัติงาน (Performance Criteria)	วิธีการประเมิน (Assessment)
30131 เขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming) ได้อย่างมีประสิทธิภาพ	1. ระบุทฤษฎีการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming) 2. ประยุกต์ใช้ทฤษฎีการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming)	
30132 ใช้ Data Structures และ Variables ได้อย่างมีประสิทธิภาพ	1. ระบุทฤษฎีการใช้และข้อจำกัดของ Data Structures และ Variables 2. ใช้ Data Structures และ Variables ได้	
30133 ใช้ Statements และ ประยุกต์ใช้ Algorithms ได้อย่างมีประสิทธิภาพ	1. ระบุทฤษฎีการใช้และข้อจำกัดของ Statements และ Algorithms 2. ใช้ Statements และ Algorithms ได้	

12. ความรู้และทักษะก่อนหน้าที่จำเป็น (Pre-requisite Skill & Knowledge)

มีความรู้และทักษะในการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming) มีความเข้าใจการใช้งาน Data Structures และ Variables และ Statements

13. ทักษะและความรู้ที่ต้องการ (Required Skills and Knowledge)

(ก) ความต้องการด้านทักษะ

N/A

(ข) ความต้องการด้านความรู้

1. มีความรู้เกี่ยวกับทฤษฎีการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming)
2. มีความรู้เกี่ยวกับทฤษฎีการใช้และข้อจำกัดของ Data Structures และ Variables
3. มีความรู้เกี่ยวกับทฤษฎีการใช้และข้อจำกัดของ Statements และ Algorithms
4. มีความรู้เกี่ยวกับ Programming Logic

14. หลักฐานที่ต้องการ (Evidence Guide)

หลักฐานที่จะต้องกำหนดข้อแนะนำเกี่ยวกับการประเมินและควรที่จะใช้ประกอบรวมกันกับเกณฑ์การปฏิบัติงาน (Performance Criteria) และทักษะและความรู้ที่ต้องการ (Required Skill and Knowledge)

(ก) หลักฐานการปฏิบัติงาน (Performance Evidence)

N/A

(ข) หลักฐานความรู้ (Knowledge Evidence)

1. ผลการสอบแบบทดสอบปรนัย

(ค) คำแนะนำในการประเมิน

1. เจ้าหน้าที่สอบตรวจประเมินหลักฐานโดยพิจารณาจากร่องรอยหลักฐานที่เกี่ยวข้อง ทั้งหลักฐานด้านปฏิบัติงานและหลักฐานด้านความรู้

(ง) วิธีการประเมิน

1. แบบทดสอบปรนัย

15. ขอบเขต (Range Statement)

ขอบเขตอธิบายถึงขอบเขตของการปฏิบัติงาน และสภาพแวดล้อมอื่นๆ หรือสถานการณ์อื่นๆ ที่มีผลกระทบต่อการทำงานรวมถึงเครื่องมือ อุปกรณ์ เทคโนโลยี ทรัพยากรที่ใช้หรือข้อกำหนดอื่นๆ ที่เกี่ยวข้อง

(ก) คำแนะนำ

1. ในการปฏิบัติงานให้คำนึงถึงทักษะการเขียนโปรแกรมเพื่อเพิ่มประสิทธิภาพของเกม (Optimization) ที่ทำให้โปรแกรมคอมพิวเตอร์ทำงานเร็วขึ้นหรือใช้ทรัพยากรน้อยลง ในการทดสอบนี้ต้องมีความเข้าใจการใช้และข้อจำกัดของ Data Structures, Variables และ Statements รวมถึงสามารถประยุกต์ใช้ Algorithms เพื่อให้โปรแกรมที่เขียนใช้ทรัพยากร น้อยที่สุด

(ข) คำอธิบายรายละเอียด

1. เกมเอนจิน (Game Engine) หมายถึง โปรแกรมคอมพิวเตอร์ที่ใช้สร้างวิดีโอเกม ซึ่งโปรแกรมเหล่านี้ถูกเขียนขึ้นมาจากภาษาคอมพิวเตอร์ต่างๆ อาทิ C, C++, JAVA เป็นต้น เกมเอนจินเปรียบเสมือนเป็นเครื่องมือใช้สร้างชิ้นส่วนต่างๆ ของระบบเกม ชิ้นส่วนเหล่านี้ จะถูกนำไปประกอบเข้าด้วยกัน และถูกทดสอบว่าสามารถทำงานร่วมกันได้อย่างไม่มีปัญหา ภายใต้สภาพแวดล้อมที่ถูกกำหนดไว้ เป็นกระบวนการในการสร้างเกมให้ออกมาสมบูรณ์ นักพัฒนาเกมจำนวนมากใช้เกมเอนจินในการสร้างเกมสำหรับเครื่องเล่น มือถือ ตลอดจนคอมพิวเตอร์พกพาการทำงานของเกมเอนจิน ประกอบด้วย การเรนเดอร์กราฟิก 2 มิติ 3 มิติ การตรวจสอบการชนระบบเสียง ภาษาสคริปต์ แอนิเมชัน ปัญญาประดิษฐ์ ระบบเน็ตเวิร์ค การสตรีม การจัดการหน่วยความจำ การรองรับภาษาที่แปล ตลอดจนระบบอื่นๆ อีกมากมาย
2. Programming Logic หมายถึง การเขียนโปรแกรมเชิงตรรกะ (อังกฤษ: Logic programming) เป็นแนวทางการเขียนโปรแกรม (programming paradigm) แบบหนึ่ง โดยกำหนดเขตคุณลักษณะ (attribute) ของคำตอบ แทนที่จะกำหนดขั้นตอนที่ทำให้ได้คำตอบ การเขียนโปรแกรมแบบนี้มีหลักการคือ ความจริง + กฎ = ผลลัพธ์ หลักการอื่นที่แตกต่างให้ดู การเขียนโปรแกรมตรรกะเชิงอุปนัย (Inductive logic programming)
3. ไบบลิารี (Library/Tool) หมายถึง คลังโปรแกรม หรือ ในทางวิทยาการคอมพิวเตอร์ คือส่วนที่รวบรวมกระบวนการ (process) และฟังก์ชันย่อย (subroutine) ต่างๆ ซึ่งอาจจะรวมรหัสต้นฉบับ (source code) ที่จำเป็นต้องใช้ในการเขียนซอฟต์แวร์ หรือ ใช้ในการทำงานของโปรแกรมหนึ่ง Library/Tool ที่เกี่ยวกับการพัฒนาเกมมีมากมายหลายชนิดเช่น Graphic Artificial Intelligent (AI) และ Physic เป็นต้น
4. การเขียนโปรแกรมเชิงวัตถุ OOP (Object Oriented Programming) เป็นวิธีการเขียนโปรแกรม โดยอาศัยแนวคิดของวัตถุชิ้นหนึ่ง มีความสามารถในการปกป้องข้อมูล และการสืบทอดคุณสมบัติ อาลัน เคย์ (Alan Kay) ได้เสนอกฎ 5
 - o ทุกๆ สิ่งเป็นวัตถุ (Everything is an Object)
 - o โปรแกรม ก็คือ กลุ่มของวัตถุที่ส่งข่าวสารบอกกันและกันให้ทำงาน (A Program is a Bunch of Object Telling Each Other What to do by Sending Messages)
 - o ในวัตถุแต่ละวัตถุจะต้องมีหน่วยความจำและประกอบไปด้วยวัตถุอื่นๆ (Each Object has Its Own Memory Made Up of Other Objects)
 - o วัตถุทุกชนิดจะต้องจัดอยู่ในประเภทใดประเภทหนึ่ง (Every Object has a Type)
 - o วัตถุที่จัดอยู่ในประเภทเดียวกันย่อมได้รับข่าวสารเหมือนกัน (All Object of a Particular Type Can Receive the Same Messages)
5. Variable หมายถึง ตัวแปร ซึ่งหมายถึงตัวที่ผู้เขียนโปรแกรมตั้งขึ้น เพื่อใช้เก็บค่าที่ต้องการนำมาใช้งานในการเขียนโปรแกรม เพื่อทำการประมวลผลข้อมูล เก็บข้อมูลในหน่วยความจำขณะที่โปรแกรมทำงาน ประกอบด้วยชื่อตัวแปรและตำแหน่งที่เก็บ แต่ในการเรียกใช้งานส่วนใหญ่จะใช้ชื่อตัวแปรเป็นหลัก ตัวแปรใช้เก็บข้อมูลที่มีการเรียกใช้หลายครั้ง เช่น ในการคำนวณหลายอย่าง การเปรียบเทียบข้อมูลหลายครั้ง

6. Statement เป็นข้อความที่เขียนสั่งให้คอมพิวเตอร์ทำงานตามที่ต้องการด้วยภาษาคอมพิวเตอร์ภาษาใดภาษาหนึ่ง มี 2 แบบ คือ แบบข้อความสั่งทำการ (executable statement) และข้อความสั่งไม่ทำการ (non executable statement) ข้อความทั้งหมดที่เขียนสั่งตั้งแต่ต้นจนจบเรียกว่า ชุดคำสั่ง หรือ โปรแกรม (program) Script โดยปกติประกอบด้วยลำดับของ statements ถ้ามีมากกว่าหนึ่ง statement ผลลัพธ์ปรากฏขึ้นในครั้งหนึ่งเป็น statement execute
7. Data Structures หมายถึง ความสัมพันธ์ระหว่างข้อมูลที่อยู่ในโครงสร้างนั้นๆ รวมทั้งกระบวนการในการจัดการข้อมูลในโครงสร้าง หรือ การจัดเตรียมรูปแบบการเก็บข้อมูลในหน่วยความจำอย่างมีระเบียบแบบแผนการแทนข้อมูลให้อยู่ในรูปแบบที่ถูกต้อง ตลอดจนกรณีวิธีการเข้าถึงข้อมูลในโครงสร้างให้เป็นไปอย่างมีประสิทธิภาพ
8. Algorithms หมายถึง แนวคิดอย่างมีเหตุมีผลที่ผู้พัฒนาโปรแกรม โปรแกรมเมอร์ หรือนักวิเคราะห์ระบบ ใช้ในการอธิบายวิธีการทำงานอย่างเป็นขั้นตามลำดับในการที่จะพัฒนาโปรแกรมนั้นๆ ให้กับผู้ที่สนใจหรือผู้ที่เป็นเจ้าของงาน หรือผู้ที่รับผิดชอบได้ทราบถึงขั้นตอนต่างๆ ในการเขียนหรือพัฒนาโปรแกรมขณะเดียวกันสามารถช่วยให้ผู้ที่เป็นเจ้าของงาน หรือผู้ที่รับผิดชอบได้ตรวจสอบขั้นตอนต่างๆ ในการทำงาน และความถูกต้องในแต่ละขั้นตอนการทำงาน โดยผู้ที่เป็นเจ้าของงานหรือผู้ที่รับผิดชอบนั้นๆ ไม่จำเป็นต้องเขียนโปรแกรมเป็น
9. Optimizations ในด้านวิทยาศาสตร์คอมพิวเตอร์ หมายถึง การเพิ่มประสิทธิภาพโปรแกรมหรือการปรับให้เหมาะสมของซอฟต์แวร์เป็นกระบวนการของการปรับเปลี่ยนระบบซอฟต์แวร์เพื่อให้การทำงานบางอย่างมีประสิทธิภาพมากขึ้นหรือใช้ทรัพยากรเช่น CPU และ Memory น้อยลง โดยทั่วไปโปรแกรมคอมพิวเตอร์อาจได้รับการปรับให้เหมาะสมเพื่อให้สามารถเรียกใช้งานได้รวดเร็วขึ้นหรือเพื่อให้สามารถใช้งานได้โดยใช้น้อยความจำน้อยลงหรือทรัพยากรอื่นๆ หรือใช้พลังงานน้อยลง การ Optimizations มีหลายระดับหลายวิธี แต่ในการทดสอบนี้จะวัดความเข้าใจในระดับ Data Structure, Variables, Statements และ Algorithms โดยต้องมีความเข้าใจการใช้และข้อจำกัดของ Data Structures, Variables และ Statements ได้อย่างเหมาะสมกับการใช้งานรวมถึงต้องสามารถเลือกประยุกต์ใช้ Algorithms เพื่อให้โปรแกรมที่เขียนใช้ทรัพยากรน้อยที่สุด

16. หน่วยสมรรถนะร่วม (ถ้ามี)

N/A

17. อุตสาหกรรมร่วม/กลุ่มอาชีพร่วม (ถ้ามี)

N/A

18. รายละเอียดกระบวนการและวิธีการประเมิน (Assessment Description and Procedure)

1. แบบทดสอบปรนัย

1. รหัสหน่วยสมรรถนะ 3021
2. ชื่อหน่วยสมรรถนะ ใช้ระบบจัดการการเปลี่ยนแปลงที่เกิดกับไฟล์ในการพัฒนาเกม (Version Control)
3. ทบพวนครั้งที่ - / -
4. สร้างใหม่ ปรับปรุง

5. สำหรับชื่ออาชีพและรหัสอาชีพ (Occupational Classification)

อาชีพนักพัฒนาโปรแกรมเกม

6. คำอธิบายหน่วยสมรรถนะ (Description of Unit of Competency)

บุคคลที่ผ่านสมรรถนะนี้จะต้องสามารถใช้งานระบบที่จัดการการเปลี่ยนแปลงที่เกิดกับไฟล์ในการพัฒนาเกม (Version Control) ในระดับผู้ใช้งาน

7. สำหรับระดับคุณวุฒิ

1	2	3	4	5	6	7	8
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

8. กลุ่มอาชีพ (Sector)

ผู้ปฏิบัติงานด้านการพัฒนาโปรแกรมเกม หรืออื่นๆ ที่เกี่ยวข้อง

9. ชื่ออาชีพและรหัสอาชีพอื่นที่หน่วยสมรรถนะนี้สามารถใช้ได้ (ถ้ามี)

N/A

10. ข้อกำหนดหรือกฎระเบียบที่เกี่ยวข้อง (Licensing or Regulation Related) (ถ้ามี)

N/A

11. สมรรถนะย่อยและเกณฑ์การปฏิบัติงาน (Elements and Performance Criteria)

สมรรถนะย่อย (Element)	เกณฑ์ในการปฏิบัติงาน (Performance Criteria)	วิธีการประเมิน (Assessment)
30211 ใช้งานระบบจัดการการเปลี่ยนแปลงที่เกิดกับไฟล์ได้ (Version Control Management)	1. ใช้งานระบบจัดเก็บภายในของตนเอง 2. ใช้งานระบบจัดการการเปลี่ยนแปลงภายนอก (Remote Version Control)	
30212 ใช้งานการแยกเป็นสาขาของระบบจัดการการเปลี่ยนแปลงที่เกิดกับไฟล์ (Version Control Branching Management)	1. ใช้งานการแยกเป็นสาขา (Branching) ของระบบจัดเก็บภายในของตัวเอง 2. ใช้งานการแยกเป็นสาขา (Branching) ของระบบจัดเก็บภายนอกได้ (Remote Version Control)	

12. ความรู้และทักษะก่อนหน้าที่จำเป็น (Pre-requisite Skill & Knowledge)

มีความสามารถในระบบที่จัดการการเปลี่ยนแปลงที่เกิดกับไฟล์ที่เพิ่มเติมและแก้ไข (Version Control)

13. ทักษะและความรู้ที่ต้องการ (Required Skills and Knowledge)

(ก) ความต้องการด้านทักษะ

1. ทักษะการใช้ระบบจัดการการเปลี่ยนแปลงที่เกิดกับไฟล์ในการพัฒนาเกม (Version Control) ในระดับ Local หรือ ระดับผู้ใช้งานไม่ใช่คนดูแล Server

(ข) ความต้องการด้านความรู้

1. มีความรู้เกี่ยวกับทฤษฎีของระบบที่จัดการการเปลี่ยนแปลงที่เกิดกับไฟล์ที่เพิ่มเติมและแก้ไข (Version Control)

14. หลักฐานที่ต้องการ (Evidence Guide)

หลักฐานที่จะต้องกำหนดข้อกำหนดเกี่ยวกับการประเมินและควรที่จะใช้ประกอบรวมกันกับเกณฑ์การปฏิบัติงาน (Performance Criteria) และทักษะและความรู้ที่ต้องการ (Required Skill and Knowledge)

(ก) หลักฐานการปฏิบัติงาน (Performance Evidence)

1. ใบบันทึกผลการทดสอบภาคปฏิบัติ

(ข) หลักฐานความรู้ (Knowledge Evidence)

1. ผลการสอบแบบทดสอบปรนัย

(ค) คำแนะนำในการประเมิน

1. เจ้าหน้าที่สอบตรวจประเมินหลักฐานโดยพิจารณาจากร่องรอยหลักฐานที่เกี่ยวข้อง ทั้งหลักฐานด้านปฏิบัติงานและหลักฐานด้านความรู้

(ง) วิธีการประเมิน

1. แบบทดสอบปรนัย

2. แบบทดสอบภาคปฏิบัติ

15. ขอบเขต (Range Statement)

ขอบเขตอธิบายถึงขอบเขตของการปฏิบัติงาน และสภาพแวดล้อมอื่นๆ หรือสถานการณ์อื่นๆ ที่มีผลกระทบต่อการทำงานรวมถึงเครื่องมือ อุปกรณ์ เทคโนโลยี ทรัพยากรที่ใช้หรือข้อกำหนดอื่นๆ ที่เกี่ยวข้อง

(ก) คำแนะนำ

1. ในการปฏิบัติงานให้คำนึงถึงทักษะการใช้ ระบบจัดการการเปลี่ยนแปลงที่เกิดกับไฟล์ได้ (Version Control)

โดยจะทดสอบเฉพาะระบบจัดการการเปลี่ยนแปลงที่เกิดกับไฟล์ Git เท่านั้น และจะเป็นการทดสอบทักษะการใช้งาน Git ในระดับผู้ใช้งานไม่ใช่ผู้ดูแล Server

(ข) คำอธิบายรายละเอียด

1. Version Control หมายถึง ระบบที่จัดการการเปลี่ยนแปลงที่เกิดขึ้นกับไฟล์หนึ่งหรือหลายไฟล์เพื่อที่คุณสามารถเรียกเวอร์ชันใดเวอร์ชันหนึ่งกลับมาดูเมื่อไรก็ได้ ส่วนใหญ่จะใช้กับการเขียนโปรแกรมเป็นทีม การจัดการไฟล์ที่เกี่ยวข้องกับการเขียนโปรแกรม แต่จริงๆ แล้ว Version Control สามารถใช้ได้กับไฟล์ทุกชนิดในที่นี้ Version Control มีหลายชนิดด้วยกันแต่เราจะใช้ GIT ซึ่งเป็น Version Control ที่เป็น Standard ในปัจจุบัน ในการทดสอบในขั้นตอนนี้จะเป็นการทดสอบทักษะการใช้ Git ในระดับ Local หรือระดับผู้ใช้งานไม่ใช่ผู้ดูแล Server

16. หน่วยสมรรถนะร่วม (ถ้ามี)

N/A

17. อุตสาหกรรมร่วม/กลุ่มอาชีพร่วม (ถ้ามี)

N/A

18. รายละเอียดกระบวนการและวิธีการประเมิน (Assessment Description and Procedure)

1. แบบทดสอบปรนัย

2. แบบทดสอบภาคปฏิบัติ