

หลักสูตรฝึกอบรมฐานสมรรถนะ สาขาสาขาแอนิเมชันและเกม อาชีพนักพัฒนาโปรแกรมเกม ระดับ 4 (Game programmer)

วัตถุประสงค์ของหลักสูตร

1. เพื่อให้ผู้เข้ารับการอบรม มีความรู้เกี่ยวกับการเลือกใช้ Libraries/Tool รวมถึง Assets การอ่าน Error Code การเลือกใช้เครื่องมือในการทดสอบโปรแกรม การเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming) การใช้และข้อจำกัดของ Data Structures, Variables, Statement, Algorithms และทฤษฎีของระบบที่จัดการเก็บการเปลี่ยนแปลงที่เกิดกับไฟล์ (Version Control)
2. เพื่อให้ผู้เข้ารับการอบรม มีทักษะการเขียนโปรแกรมคอมพิวเตอร์ การอ่าน Code การเขียน Game Logic การแก้ไขปัญหาหรือหาข้อผิดพลาดของโปรแกรมที่เกิดขึ้น รวมถึงทักษะในการใช้ระบบจัดการเก็บการเปลี่ยนแปลงที่เกิดขึ้นกับไฟล์ในการพัฒนาเกม
3. เพื่อให้ผู้เข้ารับการอบรมนำความรู้ที่ได้รับจากการอบรมไปพัฒนาต่อยอดสร้างนวัตกรรม เสริมสร้างความเป็นมาตรฐานทางอาชีพ เป็นกำลังในการพัฒนามาตรฐานวิชาชีพให้มีศักยภาพสูงสุด

คุณสมบัติผู้เข้ารับการประเมิน

1. นักศึกษาที่กำลังศึกษาในระดับประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.) ถึงระดับอุดมศึกษา ในสถาบันการศึกษาภาครัฐและเอกชน ในสาขาที่เกี่ยวข้อง หรือบุคคลทั่วไปที่มีประสบการณ์ทำงานด้านการพัฒนาโปรแกรมเกม หรือที่เกี่ยวข้องไม่น้อยกว่า 1 ปี
2. มีบัตรประจำตัวประชาชน กรณีที่เป็นนักศึกษา จะต้องมียบัตรประจำตัวนักศึกษา และมีสัญชาติไทย
3. มีความรู้พื้นฐานในการใช้งานโปรแกรมที่เกี่ยวข้องกับหลักสูตร
4. สิทธิในการอบรม และประเมินมาตรฐานอาชีพได้ 1 อาชีพ/คน

หน่วยสมรรถนะที่ใช้ในอบรมและการประเมิน

1. เขียนโปรแกรมเกมโดยใช้ ไลบรารีหรือเครื่องมือ (Libraries/Tools) ของเกมเอนจิน (Game Engine)
2. ทดสอบ และแก้ไข ปัญหาที่เกิดขึ้นจากการเขียนโปรแกรมเกม
3. เขียนโปรแกรมเพื่อเพิ่มประสิทธิภาพของเกม (Optimization)
4. ใช้ระบบจัดการเก็บการเปลี่ยนแปลงที่เกิดกับไฟล์ในการพัฒนาเกม (Version Control)

จำนวนผู้เข้ารับการประเมิน

บุคคลทั่วไป หรือนักศึกษา จำนวนไม่น้อยกว่า 40 คน/รุ่น

ระยะเวลาการอบรม พร้อมสอบประเมิน

ฝึกอบรม จำนวน 3 วัน (18 ชั่วโมง)

สอบประเมิน จำนวน 1 วัน

- ภาคทฤษฎี 30 นาที
- ภาคปฏิบัติ 2 ชั่วโมง 30 นาที

เกณฑ์การผ่านการฝึกอบรม

1. ผู้เข้าอบรม ต้องเข้ารับการฝึกอบรมไม่น้อยกว่าร้อยละ 80 ของระยะเวลาฝึกอบรมตลอดหลักสูตร
2. ผู้เข้าอบรม ต้องทดสอบประเมินความรู้ภาคทฤษฎีด้วยแบบประเมินผลก่อนการฝึกอบรม (Pre-Test)
3. ผู้เข้าอบรม ต้องทดสอบประเมินความรู้ภาคทฤษฎีด้วยแบบประเมินผลหลังการฝึกอบรม (Post-Test)

เกณฑ์ผ่านไม่น้อยกว่าร้อยละ 70

เกณฑ์การผ่านการประเมิน

● "ผ่านการประเมิน"

หมายถึง ผู้เข้ารับการประเมิน มีผลคะแนน ภาคทฤษฎีไม่น้อยกว่าร้อยละ 70 และภาคปฏิบัติไม่น้อยกว่าร้อยละ 80

● "ไม่ผ่านการประเมิน"

หมายถึง ผู้เข้ารับการประเมิน มีผลคะแนน ภาคทฤษฎีน้อยกว่าร้อยละ 70 และภาคปฏิบัติน้อยกว่าร้อยละ 80

กำหนดการจัดกิจกรรมอบรม

วันที่ 1

09.00 – 12.00 น. บรรยายความรู้ หัวข้อ “เขียนโปรแกรมเกมโดยใช้ ไลบรารีหรือเครื่องมือ (Libraries/Tools) ของเกมเอนจิน (Game Engine)”

- การวิเคราะห์ความต้องการจากเอกสารออกแบบ (Design Document) และการจัดเตรียม Assets สำหรับสร้างเกม
- ฝึกปฏิบัติ (Workshop)

13.00 – 16.00 น. บรรยายความรู้ หัวข้อ “เขียนโปรแกรมเกมโดยใช้ ไลบรารีหรือเครื่องมือ (Libraries/Tools) ของเกมเอนจิน (Game Engine)” (ต่อ)

- การเขียนโปรแกรมตามเอกสารออกแบบ (Design Document)
- ฝึกปฏิบัติ (Workshop)

วันที่ 2

09.00 – 12.00 น. บรรยายความรู้ หัวข้อ “ทดสอบ และแก้ไข ปัญหาที่เกิดขึ้นจากการเขียนโปรแกรมเกม”

- การทดสอบหาข้อผิดพลาดของโปรแกรม
- การแก้ไขข้อผิดพลาดของโปรแกรมที่ตรวจพบ
- ฝึกปฏิบัติ (Workshop)

13.00 – 16.00 น. บรรยายความรู้ หัวข้อ “เขียนโปรแกรมเพื่อเพิ่มประสิทธิภาพของเกม (Optimization)”

- การเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming) ได้อย่างมีประสิทธิภาพ
- การใช้ Data Structures และ Variables ได้อย่างมีประสิทธิภาพ
- ฝึกปฏิบัติ (Workshop)

กำหนดการจัดกิจกรรมอบรม (ต่อ)

วันที่ 3

- 09.00 – 12.00 น. บรรยายความรู้ หัวข้อ “เขียนโปรแกรมเพื่อเพิ่มประสิทธิภาพของเกม (Optimization)” (ต่อ)
- การใช้ Statements และ ประยุกต์ใช้ Algorithms ได้อย่างมีประสิทธิภาพ
 - ฝึกปฏิบัติ (Workshop)
- 13.00 – 16.00 น. บรรยายความรู้ หัวข้อ “ใช้ระบบจัดการการเปลี่ยนแปลงที่เกิดกับไฟล์ในการพัฒนาเกม (Version Control)”
- การใช้งานระบบจัดการการเปลี่ยนแปลงที่เกิดกับไฟล์ได้ (Version Control Management)
 - การใช้งานการแยกเป็นสาขาของระบบจัดการการเปลี่ยนแปลงที่เกิดกับไฟล์ (Version Control Branching Management)
 - ฝึกปฏิบัติ (Workshop)

กำหนดการจัดกิจกรรมสอบประเมิน

วันที่ 1

- 09.00 – 10.00 น. แนะนำ และอธิบายกระบวนการประเมิน พร้อมแนวทางการสอบประเมิน
- 10.00 – 11.00 น. สอบภาคทฤษฎี
- 11.00 – 12.00 น. พักเที่ยง
- 12.00 – 15.30 น. สอบภาคปฏิบัติ

อุปกรณ์ที่ต้องใช้สำหรับอบรมและประเมิน อาชีพนักพัฒนาโปรแกรมเกม ระดับ 4 (Game programmer)

ลำดับ	อุปกรณ์	จำนวน	หมายเหตุ
1	คอมพิวเตอร์	1 เครื่อง	เครื่องคอมพิวเตอร์ 1 เครื่องต่อผู้เข้ารับการประเมิน 1 คน <ul style="list-style-type: none">• Quad-core Intel หรือ AMD processor, 2.5 GHz หรือที่เร็วกว่า NVIDIA GeForce 470 GTX หรือ AMD Radeon 6870 HD series card หรือ ที่มีคุณสมบัติสูงกว่า RAM 8 GB หรือ มากกว่า• ติดตั้งระบบปฏิบัติการ Window 10
2	ซอฟต์แวร์ที่ใช้ในการประเมิน	3 โปรแกรม	1. โปรแกรม Unity เวอร์ชัน 2018.3.13f1 หรือสูงกว่า 2. Git เวอร์ชันล่าสุด 2.21.0 หรือสูงกว่า 3. Visual Studio Code หรือโปรแกรมอื่น ๆ ที่มี ความสามารถใกล้เคียงกัน

****มหาวิทยาลัยฯ จัดเตรียมซอฟต์แวร์ เพื่อใช้ในการอบรมภาคปฏิบัติ และการสอบภาคปฏิบัติ****

แผนโครงสร้างหลักสูตร

หน่วยสมรรถนะ Unit of Competence : เขียนโปรแกรมเกมโดยใช้ไลบรารีหรือเครื่องมือ (Libraries/Tools) ของเกมเอนจิน (Game Engine)

ชื่อหัวข้อวิชา (Content Title)	เกณฑ์ในการปฏิบัติงาน (Performance Criteria)	รายละเอียดเนื้อหาวิชา (Content)
1. การวิเคราะห์ความต้องการจากเอกสารออกแบบ (Design Document) และการจัดเตรียม Assets สำหรับสร้างเกม	<ol style="list-style-type: none">ระบุและรวบรวม Assets ทั้งหมด ที่ต้องใช้ในตัวเองงานตรวจสอบความถูกต้องของ Assets และ จัดเก็บ Assets	<ol style="list-style-type: none">วิเคราะห์เอกสารการออกแบบ (Design document)ระบุและรวบรวม Assets ทั้งหมด ที่ต้องใช้ในการพัฒนาเกมได้ตรวจสอบความถูกต้องของ Assets และ จัดเก็บ Assets ได้ (Texture, Mip mapping, Bump Mapping, Shader, Anisotropic filtering, Trilinear Filtering, Color in Game Engine, Frame Rate, Character Modeling and Animation, Buffering, Rendering, Audio, Effect)
2. การเขียนโปรแกรมตามเอกสารออกแบบ (Design Document)	<ol style="list-style-type: none">เลือกใช้ไลบรารีหรือเครื่องมือ (Libraries/Tools) ใน Game Engineใช้ไลบรารีหรือเครื่องมือ (Libraries/Tools) และ Assets โดยกำหนดค่าได้ตรงตามเอกสารออกแบบเขียน Game Logic เพิ่มเติมได้ เอกสารออกแบบทางเทคนิค (Technical Document)จัดทำเอกสาร (Programming Document)	<ol style="list-style-type: none">1. เลือกใช้ไลบรารีหรือเครื่องมือ (Libraries/Tools) ในเกมเอนจินได้2. กำหนดคุณสมบัติให้กับ Game Object ได้3. เขียนโปรแกรมเชิงตรรกะได้4. อธิบายเอกสารที่เกี่ยวข้องกับ Programming Document ได้ (Properties: Collider, Rigidbody, Audio Source, Material, XR, Visual Effect, Vector3 Velocity, AddForce, Random, Logic programming, Flowchart, Programming Document)

หน่วยสมรรถนะ Unit of Competence : ทดสอบ และแก้ไข ปัญหาที่เกิดขึ้นจากการเขียนโปรแกรมเกม

ชื่อหัวข้อวิชา (Content Title)	เกณฑ์ในการปฏิบัติงาน (Performance Criteria)	รายละเอียดเนื้อหาวิชา (Content)
1. การทดสอบหาข้อผิดพลาดของโปรแกรม	<ol style="list-style-type: none"> ระบุวิธีการและเลือกใช้เครื่องมือในการทดสอบโปรแกรม ค้นหาปัญหาและข้อผิดพลาดของโปรแกรม 	<ol style="list-style-type: none"> อธิบายการเลือกใช้เครื่องมือในการทดสอบโปรแกรม อธิบายความรู้เกี่ยวกับการกำหนดวิธีการ ทฤษฎี และเครื่องมือที่ใช้ในการทดสอบโปรแกรม อธิบายวิธีการค้นหาปัญหาและข้อผิดพลาดของโปรแกรม อธิบายความรู้เกี่ยวกับการใช้เทคนิคการทดสอบหาข้อผิดพลาดของโปรแกรม (Syntax Error, Logic Error, Operators (Arithmetic, Relational, Logical, Bitwise, Assignment, Conditional), Automatic Type Conversion, Explicit Type Conversion)
2. การแก้ไขข้อผิดพลาดของโปรแกรมที่ตรวจพบ	<ol style="list-style-type: none"> ระบุสาเหตุของปัญหาที่เกิดขึ้น แก้ไขปัญหาโปรแกรมที่เกิดขึ้น 	<ol style="list-style-type: none"> อธิบายการระบุสาเหตุของปัญหาที่เกิดขึ้น ความรู้ด้านเทคนิคเกี่ยวกับการทดสอบหาข้อผิดพลาดของโปรแกรม อธิบายวิธีการแก้ไขปัญหาโปรแกรมที่เกิดขึ้น (NullReferenceException, Exception, Overloading)

หน่วยสมรรถนะ Unit of Competence : เขียนโปรแกรมเพื่อเพิ่มประสิทธิภาพของเกม (Optimization)

ชื่อหัวข้อวิชา (Content Title)	เกณฑ์ในการปฏิบัติงาน (Performance Criteria)	รายละเอียดเนื้อหาวิชา (Content)
1. การเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming) ได้อย่างมีประสิทธิภาพ	<ol style="list-style-type: none"> ระบุทฤษฎีการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming) ประยุกต์ใช้ทฤษฎีการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming) 	<ol style="list-style-type: none"> อธิบายการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming) สามารถประยุกต์ใช้ทฤษฎีการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming) (Encapsulation, Reuse the Object, Polymorphism, Method, Class, Object, Namespaces, Property, Delegate)
2. การใช้ Data Structures และ Variables ได้อย่างมีประสิทธิภาพ	<ol style="list-style-type: none"> ระบุทฤษฎีการใช้และข้อจำกัดของ Data Structures และ Variables ใช้ Data Structures และ Variables ได้ 	<ol style="list-style-type: none"> การเลือกใช้ Data Structures ได้อย่างมีประสิทธิภาพ การประกาศตัวแปรได้อย่างมีประสิทธิภาพ (Data Structure, Variables, Datatype, Loop)
3. การใช้ Statements และ ประยุกต์ใช้ Algorithms ได้อย่างมีประสิทธิภาพ	<ol style="list-style-type: none"> ระบุทฤษฎีการใช้และข้อจำกัดของ Statements และ Algorithms ใช้ Statements และ Algorithms ได้ 	<ol style="list-style-type: none"> อธิบายความรู้เกี่ยวกับ Statements และ Algorithms ได้ สามารถใช้ Statements และ Algorithms ได้ (Statement, Algorithms, Enum, Collections, List, Decision Making Statements (if, if-else, if-else-if), Switch Statement, Jump Statements)

หน่วยสมรรถนะ Unit of Competence : ใช้ระบบจัดการเก็บการเปลี่ยนแปลงที่เกิดกับไฟล์ในการพัฒนาเกม (Version Control)

ชื่อหัวข้อวิชา (Content Title)	เกณฑ์ในการปฏิบัติงาน (Performance Criteria)	รายละเอียดเนื้อหาวิชา (Content)
1. การใช้งานระบบจัดการเก็บการเปลี่ยนแปลงที่เกิดกับไฟล์ได้ (Version Control Management)	<ol style="list-style-type: none">1. ใช้งานระบบจัดเก็บภายในของตนเอง2. ใช้งานระบบจัดการเก็บการเปลี่ยนแปลงภายนอก (Remote Version Control)	<ol style="list-style-type: none">1. อธิบายถึงวิธีการจัดเก็บไฟล์ได้อย่างถูกต้อง2. สามารถใช้งานระบบจัดการเก็บการเปลี่ยนแปลงภายนอก (Remote Version Control) (Version Control Systems, Local vcs, Central vcs, Distributed vcs, Git, Command for Git)
2. การใช้งานการแยกเป็นสาขาของระบบจัดการเก็บการเปลี่ยนแปลงที่เกิดกับไฟล์ (Version Control Branching Management)	<ol style="list-style-type: none">1. ใช้งานการแยกเป็นสาขา (Branching) ของระบบจัดเก็บภายในของตนเอง2. ใช้งานการแยกเป็นสาขา (Branching) ของระบบจัดเก็บภายนอกได้ (Remote Version Control)	<ol style="list-style-type: none">1. อธิบายการแยกเป็นสาขา (Branching) ของการจัดเก็บภายในของตนเองได้2. อธิบายถึงการแยกเป็นสาขา (Branching) ของระบบจัดเก็บภายนอกได้ (Remote Version Control)