

แผนบริหารการสอนประจำบทที่ 3

หัวข้อเนื้อหา

การออกแบบโปรแกรมเกมคอมพิวเตอร์ในอดีต
การออกแบบโปรแกรมเกมคอมพิวเตอร์แบบภาษาเชิงวัตถุ
ประโยชน์ของการออกแบบโปรแกรมเกมคอมพิวเตอร์แบบภาษาเชิงวัตถุ

วัตถุประสงค์เชิงพฤติกรรม

หลังจากศึกษาบทที่ 3 แล้วนักเรียนมีความสามารถดังต่อไปนี้

1. มีความเข้าใจการออกแบบโปรแกรมเกมคอมพิวเตอร์
2. สามารถบอกประโยชน์ของการออกแบบโปรแกรมเกมคอมพิวเตอร์แบบภาษาเชิงวัตถุ

วิธีการสอน และกิจกรรม

1. บรรยาย
2. ฝึกปฏิบัติ
3. มอบหมายงานให้ไปปฏิบัติ

สื่อการเรียนการสอน

1. ไฟล์นำเสนอบทที่ 3
2. วิดีโอ
3. คู่มือ

การวัดและประเมินผล**การวัดผล**

1. ตอบคำถามในชั้นเรียน 50 %
2. แบบฝึกหัด 50 %

การประเมินผล

1. การสังเกตพฤติกรรมการเรียนรู้
2. การประเมินผลจากความสนใจและการอภิปราย
3. การทำแบบฝึกหัด

บทที่ 3

การออกแบบโปรแกรมเกมคอมพิวเตอร์

การออกแบบโปรแกรมเกมคอมพิวเตอร์ มีลักษณะคล้ายกับการออกแบบโปรแกรมคอมพิวเตอร์ ขั้นตอนการออกแบบสามารถนำมาประยุกต์ใช้ร่วมกันได้ ใช้หลักการในการพัฒนาโปรแกรมคอมพิวเตอร์ ขั้นตอนการพัฒนาโปรแกรมคอมพิวเตอร์ซอฟต์แวร์แอนิเมชันจริง โดยเนื้อหาในบทนี้ประกอบด้วย

1. การออกแบบโปรแกรมเกมคอมพิวเตอร์ในอดีต
2. การออกแบบโปรแกรมเกมคอมพิวเตอร์แบบภาษาเชิงวัตถุ
3. ประโยชน์ของการออกแบบโปรแกรมเกมคอมพิวเตอร์แบบภาษาเชิงวัตถุ

3.1 การออกแบบโปรแกรมเกมคอมพิวเตอร์ในอดีต

ประวัติการออกแบบโปรแกรมเกมคอมพิวเตอร์ในอดีต เริ่มขึ้นประมาณปี ค.ศ. 1993 มีการพัฒนาคอมพิวเตอร์ต่ำกว่ารุ่น 386 ส่วนประกอบภายในคอมพิวเตอร์มีการใช้ทรัพยากรที่ค่อนข้างจำกัดและมีราคาสูง การพัฒนาตั้งแต่รุ่นที่เป็นระบบปฏิบัติการเป็นเวอชันดอส 5.0 มีการใช้เมาส์และการออกแบบโปรแกรมเกมคอมพิวเตอร์ให้ใช้งานร่วมกัน การออกแบบโปรแกรมเกมคอมพิวเตอร์ เป็นส่วนหนึ่งของการพัฒนาเกมคอมพิวเตอร์ ดังนั้นเพื่อเป็นการทำความเข้าใจ กระบวนการขั้นตอนการพัฒนาเกมคอมพิวเตอร์ มีกระบวนการดังนี้

- 1) การออกแบบเกมคอมพิวเตอร์
- 2) การวางแผนและเตรียมการพัฒนาเกมคอมพิวเตอร์
- 3) การเตรียมและจัดการวัตถุดิบในการทำเกมเครื่องมือที่ใช้ในการพัฒนาเกม
- 4) การพัฒนาเกม มีFrameworkการจัดการ Resource ต่างๆ ที่ใช้ในเกมเข้าใจพื้นฐานการวาดภาพและตัวหนังสือใน Computerการเคลื่อนที่ของตัวละครในเกม
- 5) Game Performance และการวัดค่ากระบวนการวัดประสิทธิภาพในเกมหลักการและประโยชน์ของ FPS (Frames per Second) การส่งค่า ข้อความนำมาประยุกต์ใช้ในเกม และการจัดการหน่วยความจำในเกม
- 6) อุปกรณ์นำเข้า อุปกรณ์ควบคุม หลักการควบคุมเครื่องมือควบคุม Mouse และ Keyboardการใช้Game Stateการทำตรรกะในเกมเบื้องต้น
- 7) ส่วนประกอบภายในเกม Game Componentและการออกแบบ Object Oriented Concept ในเกม

8) Clip และ Animation การใช้ Clip รูปภาพ เพื่อนำไปใช้ในเกมการวาดภาพในรูปแบบต่างๆได้เข้าใจการทำภาพเคลื่อนไหว

9) เสียงในเกมกระบวนการใช้งานเกี่ยวกับ Sound ในเกมการใช้งาน Audio Engine, SoundBank และ WaveBank กระบวนการจัดการเสียงในเกม

10) AI Concept การใช้งาน AI ในเกมวิธีเลือกใช้ AI สำหรับเกม Turn Base เกม Real Time หลักการ Spawn Enemy หรือ Enemy Management การทำ PathFinding ในแบบต่างๆ เช่น Way Point และอื่นๆ

11) การทดสอบเกมและการทำระบบติดตั้งเกมหลักการในการทดสอบ Game จริยธรรมในการทำการทดสอบเกมการทำ Check List จาก Game Design ขั้นตอนการใช้ Unit Test และ Integration Test การทำระบบติดตั้งเกม

12) การเลือกใช้ Game Engine การประยุกต์กับการพัฒนาเกม

จากกระบวนการพัฒนาเกมคอมพิวเตอร์ โปรแกรมเกมคอมพิวเตอร์จะเป็นส่วนหนึ่งของการพัฒนาเกม โดยต้องทำตามขั้นตอนมาจากการออกแบบเกมสอดคล้องกับรูปแบบ(Frame work) ที่เลือกและการใช้เครื่องมือในการพัฒนาเกม การออกแบบตัวละคร เมื่อทราบกระบวนการดังที่ได้กล่าวมา ขั้นตอนต่อไปจะเป็นขั้นตอนการพัฒนาโปรแกรมเกมคอมพิวเตอร์ ซึ่งประกอบด้วยกระบวนการต่างๆ แสดงดังภาพที่ 3.1 สรุปรายละเอียดได้ดังนี้

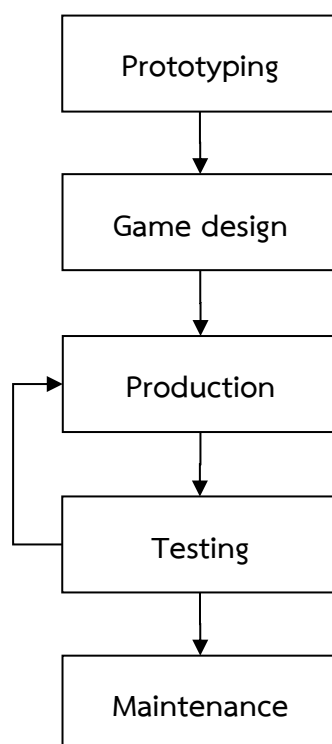
ขั้นตอนที่ 1 ขั้น Prototyping เป็นขั้นตอนแรกในการออกแบบจะเป็นการรวบรวมแนวความคิด (idea) รวบรวมคุณลักษณะต่างๆ ในเกม มีการสำรวจก่อนที่จะพัฒนาเกม มีการสร้างต้นแบบขึ้นอย่างรวดเร็ว

ขั้นตอนที่ 2 ขั้น Game design ขั้นตอนการออกแบบเกม ผู้พัฒนาจะทำการออกแบบทางด้านศิลปะ (Art design) ร่วมกับโปรแกรมเกมคอมพิวเตอร์ เพื่อสร้างให้เกมมีคุณลักษณะที่โดดเด่น และตรงกับความต้องการของผู้ร่วมผลิตหรือผู้สนับสนุน และสร้างเป็นเอกสารในการออกแบบขึ้นมา เพื่อใช้ในการพัฒนาในขั้นต่อไป

ขั้นตอนที่ 3 ขั้น Production ขั้นตอนการลงมือผลิต ผู้พัฒนาทำการสร้างโค้ดโปรแกรม ในการพัฒนาเกมและเอกสารประกอบ ให้ตรงตามวัตถุประสงค์ เอกสารการออกแบบ และระยะเวลาที่ได้รับมอบหมาย พร้อมทั้งแสดงสถานะของการพัฒนา รวมถึงข้อจำกัด ในคุณสมบัติของเกม

ขั้นตอนที่ 4 ขั้น Testing ขั้นตอนการทดสอบ เป็นขั้นตอนการทดสอบเกมหลังจากพัฒนาเกมเสร็จ เป็นการหาข้อผิดพลาด พร้อมรายงานไปยังผู้พัฒนาเพื่อแก้ไขข้อผิดพลาดต่อไป

ขั้นตอนที่ 5 ขั้น Maintenance ขั้นตอนการดูแลรักษา เป็นขั้นตอนหลังจากมีการใช้งานจริงแล้วเกิดการดำเนินงานของเกมที่ใช้ระยะเวลายาวนานขึ้น หรือมีปริมาณไฟล์ที่เพิ่มมากขึ้น ทำให้ต้องมีการลบออก หรือปรับปรุงโปรแกรมบางส่วน



ภาพที่ 3.1 ขั้นตอนการพัฒนาโปรแกรมเกมคอมพิวเตอร์

(แหล่งที่มา: https://en.wikipedia.org/wiki/Game_programming)

3.2 การออกแบบโปรแกรมเกมภาษาเชิงวัตถุ

การออกแบบโปรแกรมเกมภาษาเชิงวัตถุ มีลักษณะคล้ายกับการออกแบบโปรแกรมคอมพิวเตอร์ภาษาเชิงวัตถุ คือ หลักการในการออกแบบในการสร้างวัตถุในโปรแกรมเกมคอมพิวเตอร์ให้สามารถนำมาประกอบกันและทำงานร่วมกัน รวมถึงมีการประมวลผลและส่งข่าวสารไปยังวัตถุอื่นๆที่เกี่ยวข้องภายในเกมคอมพิวเตอร์

ตัวอย่างการออกแบบเกมแบบภาษาเชิงวัตถุ การออกแบบเกม Tetris ที่เป็นเกมสุ่มของวัตถุที่มีรูปร่างหลายๆ แบบ (tetrominoes) เคลื่อนมาจากด้านบนลงสู่ด้านล่าง ผู้เล่นจะต้องนำมาเรียงต่อกันโดยการขยับตำแหน่ง เมื่อในแต่ละชั้นไม่มีช่องว่างวัตถุจะหายไป แต่ถ้ามีวัตถุเรียงสูงจนวัตถุชิ้นใหม่ไม่สามารถหล่นได้ จะเป็นการจบเกมสามารถออกแบบโปรแกรมให้อยู่ในรูปแบบภาษาเชิงวัตถุ ดังนี้

- กำหนดให้วัตถุมีลักษณะมรูปร่าง (shape) สี (color)
- วัตถุมีพฤติกรรม คือ ตก (falling) เคลื่อนที่เฉพาะด้านข้าง (moving sideways) หมุน (rotation)



ภาพที่ 3.2 ตัวอย่างเกมคอมพิวเตอร์ เกม Tetris
(แหล่งที่มา <http://www.gamingdose.com/>)

การออกแบบโปรแกรมคอมพิวเตอร์ภาษาเชิงวัตถุ จะสามารถใช้คุณสมบัติได้ดังต่อไปนี้

1. การห่อหุ้ม (Encapsulation) หมายถึงการซ่อนวิธีการทำงาน (Information Hiding) เป็นการซ่อนรายละเอียดการทำงานและข้อมูลไว้ภายใน ไม่ให้ภายนอกสามารถเห็นได้

หลักการห่อหุ้มโดยกำหนดให้ access modifier ของ attribute เป็น private แล้วสร้าง accessor method และ mutator method ขึ้นมา หากผู้ใช้จะต้องการเข้าถึง attribute ก็จะต้องเรียกใช้ accessor method สำหรับ attribute นั้นๆ หรือถ้าผู้ใช้ต้องการเปลี่ยนแปลงค่าของ attribute ก็จะต้องทำผ่าน mutator method เท่านั้น ข้อดีของการใช้หลักการห่อหุ้มในการสร้างคลาสก็คือ

1) ผู้ใช้สามารถใช้งานได้โดยไม่ต้องสนใจรายละเอียดว่าคลาสนั้นมีรายละเอียดต่างๆ (เช่นชื่อ attribute) อย่างไร แต่สนใจแค่ว่าจะใช้งานคลาสนั้นได้อย่างไร

2) ผู้ใช้ที่ต้องการเปลี่ยนค่าของ attribute ต้องกระทำผ่าน mutator method เท่านั้น ทำให้สามารถป้องกันไม่ให้ผู้ใช้เปลี่ยนค่า attribute เป็นค่าที่ไม่ถูกต้องได้เนื่องจากเราสามารถเช็คเงื่อนไขได้ที่ mutator method ก่อน หากค่าที่ต้องการกำหนดไม่ผิดเงื่อนไขจึงจะกำหนดค่าให้กับ attribute นั้น

3) หากมีการแก้ไขคลาสนั้นๆ ในภายหลัง (เช่นการเปลี่ยนชื่อตัวแปรใหม่) เราไม่จำเป็นต้องปรับแก้โค้ดในส่วนของการใช้งานคลาสถ้าหากผู้ใช้ใช้งานผ่าน accessor และ mutator method ในการสืบทอดนั้น หาก attribute ที่อยู่ในคลาสแม่มี access modifier เป็น private คลาสลูกจะไม่ได้สืบทอด attribute นั้นๆ มาโดยตรง เราจึงไม่สามารถใช้งาน attribute นั้นผ่านชื่อของ attribute ได้ ถ้าต้องการจะใช้งาน attribute นั้น จะต้องใช้งานผ่าน public method ที่สืบทอดมาจากคลาสแม่ โดยถ้าต้องการเข้าถึงค่าของ attribute ก็ให้ใช้ accessor method และถ้าต้องการเปลี่ยนแปลงค่าก็ให้ใช้ mutator method สำหรับ attribute นั้นๆ

2. การสืบทอด (Inheritance) คือการทำการสร้างสิ่งใหม่ขึ้นด้วยการสืบทอดหรือรับเอาคุณสมบัติบางอย่างมาจากสิ่งเดิมที่มีอยู่แล้ว และสร้างเพิ่มเติมได้สามารถนำสิ่งที่เคยสร้างขึ้นแล้วนำกลับมาใช้ใหม่ (re-use) ได้ เป็นการประหยัดเวลา

การออกแบบโปรแกรมที่ดีนั้น ควรจะมีการนำกลับมาใช้ใหม่ให้ได้มากที่สุดเท่าที่จะทำได้เพื่อลดความซ้ำซ้อนของตัวโค้ด การสืบทอดก็เป็นการนำโค้ดกลับมาใช้ใหม่รูปแบบหนึ่ง คือ การนำคลาสที่ถูกประกาศไว้แล้วมาปรับปรุงแก้ไขให้เกิดเป็นคลาสใหม่ได้โดยไม่ต้องเริ่มต้นเขียนใหม่ทั้งหมด คลาสที่ถูกนำมาใช้เป็นต้นแบบเรียกว่าคลาสแม่ (superclass) และคลาสที่ปรับปรุงมาจากคลาสแม่เรียกว่า คลาสลูก (subclass) โดยปกติแล้วคลาสลูกจะมี attribute และ method ที่เพิ่มเติมขึ้นมาจากคลาสแม่ จึงเหมาะสำหรับใช้สร้างวัตถุที่มีความเฉพาะเจาะจงมากกว่าคลาสแม่ยังสามารถเพิ่ม attribute หรือ method ใหม่ๆ ที่เฉพาะเจาะจงของคลาสนั้นๆ เข้าไปได้ด้วย นอกจากนี้เรายังสามารถนำคลาสนี้ไปสร้างคลาสลูกต่อไปได้เรื่อยๆ

3. การพ้องรูป (Polymorphism) เป็นคุณสมบัติที่ชนิดของวัตถุที่เกิดจากชนิดของวัตถุแม่เดียวกัน มีความสามารถเหมือนแม่แต่มีวิธีการดำเนินงานไม่เหมือนกัน คือมีลักษณะเฉพาะตัวการทำให้สิ่งหนึ่งสามารถทำงานได้หลากหลายรูปแบบตามความต้องการที่เกิดขึ้นในขณะหนึ่ง ๆ

สามารถเขียนเมธอดชื่อเดียวกันให้สามารถรับพารามิเตอร์ได้หลายชนิด และการเขียนเมธอดชื่อเดียวกับคลาสที่สืบทอดมา แต่ทำงานต่างกันได้

3.3 คุณลักษณะของการออกแบบโปรแกรมคอมพิวเตอร์

1) การเรียกใช้ได้หลายครั้ง ออบเจกต์ในโปรแกรมได้ถูกออกแบบตามหลักการที่ว่าสามารถเรียกใช้งานได้หลาย ๆ ครั้ง การออกแบบตัวแรกอาจจะทำได้ยาก แต่ว่าโปรแกรมเกมที่เขียนภายหลังจะสามารถสร้างง่ายเพราะสามารถเรียกใช้ออบเจกต์ที่ถูกสร้างไว้ตั้งแต่โปรแกรมแรกได้

2) ความเชื่อถือได้ โปรแกรมจะมีความเชื่อถือได้สูงเพราะจะรวมเอาส่วนย่อยที่ทดสอบจนได้มาตรฐานแล้วมารวมเข้าไว้ด้วยกัน รหัส (Code) ที่เขียนขึ้นมาใหม่ในแต่ละเกมจะมีไม่มากนัก เนื่องจากรหัสส่วนใหญ่จะถูกดึงมาจากไลบรารีที่มีความเชื่อถือได้สูงอยู่แล้ว

3) ความต่อเนื่องกัน การพัฒนาซอฟต์แวร์แบบเชิงวัตถุ จะเปลี่ยนไปตามฝีมือและจำนวนนักเขียนโปรแกรมภาษา เช่น นักโปรแกรมภาษา C ที่ชำนาญสามารถเรียนรู้หลักการของเชิงวัตถุได้ภายในเวลาไม่นาน และสามารถเข้าใจเนื้อหาได้ไม่ยาก และสามารถแปลงโปรแกรมของ C เป็นโปรแกรมภาษาเชิงวัตถุได้

บทสรุป

เนื้อหาในบทนี้ผู้เรียนการออกแบบโปรแกรมเกมคอมพิวเตอร์ ทราบถึงขั้นตอนการพัฒนาเกมคอมพิวเตอร์ การออกแบบโปรแกรมเกม คุณลักษณะการออกแบบเกมคอมพิวเตอร์ เพื่อใช้ในการพัฒนาโปรแกรมเกมคอมพิวเตอร์ต่อไป

แบบฝึกหัดท้ายบท

- 1) จงอธิบายขั้นตอนการออกแบบโปรแกรมเกมคอมพิวเตอร์
- 2) การออกแบบโปรแกรมเกมคอมพิวเตอร์แบบภาษาเชิงวัตถุ มีหลักการออกแบบอย่างไร
- 3) คุณลักษณะการแบบโปรแกรม ประกอบด้วยอะไรบ้าง

เอกสารอ้างอิง

Curtis Bennett. (2009). **A Simple Introduction to Game Programming With C# and XNA 3.1**. United States of America :xnagamemaking.com.

มนตรี พจนารถลาวัฒน์. (2535). **การเขียนโปรแกรมคอมพิวเตอร์ด้วยเทอร์โบซี**. กรุงเทพมหานคร, ซีเอ็ดดูเคชั่น.

_____.(2559). **Game_programming**. [ออนไลน์]. เข้าถึงข้อมูลวันที่ 2 พฤษภาคม 2559. จาก https://en.wikipedia.org/wiki/Game_programming

_____.(2559). **Game_programming**. [ออนไลน์]. เข้าถึงข้อมูลวันที่ 2 พฤษภาคม 2559. จาก <http://gamedevelopment.tutsplus.com/tutorials/quick-tip-intro-to-object-oriented-programming-for-game-development--gamedev-1805>

_____.(2559). **91-basic-good-game**. [ออนไลน์]. เข้าถึงข้อมูลวันที่ 2 พฤษภาคม 2559. จาก <http://thaicryengine.konlengame.com/index.php/press-thaicry/91-basic-good-game?>