Game Programming Concept

Suphot Sawattiwong
tohpus@hotmail.com

จาก Lab 1.4: การวาด Font

- ให้ทำการสร้างโปรเจค เพื่อวาดตัวหนังสือขึ้นมาเป็นคำต่างๆ ดังต่อไปนี้ “Game Design”, “Game Development”, “Graphic Design”, “Technical Design”, “Game Asset”, “Asset List”
- โดยให้มีการใช้สีแบบต่างๆ โดยการใช้คำสั่ง Random ซึ่งมีตัวอย่างให้ในหน้าถัดไป และ ให้มีการ random ตำแหน่งของตัวหนังสือหลายๆ แบบด้วย และยังต้องหันองศาแบบ random ด้วยเช่นกัน

การสุ่มหรือ Random

- ให้ประกาศตัวแปรแบบ Random ขึ้นมา ใน Class

โดยมีรูปแบบดังนี้

Random <ชื่อตัวแปร>;

เช่น Random random;

- ให้จัดการ new Random(); ที่ Initialize() หรือที่ Construtor

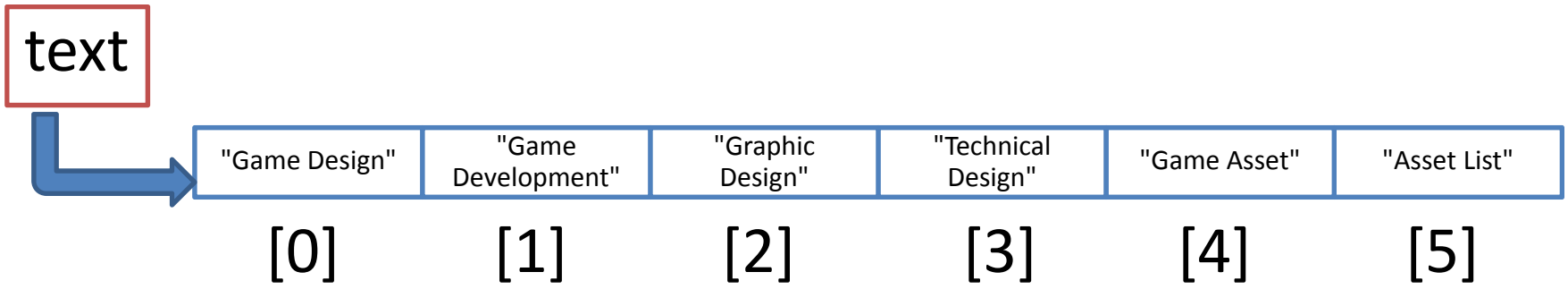
เช่น random = new Random();

- ส่วนวิธีใช้ int a= random.Next(<MaxValue>);
- เช่น

int a= random.Next(9); จะมีค่าสูงสุด แค่ 0-8

ใน Lab 1.4 จากโจทย์เห็นได้ว่า

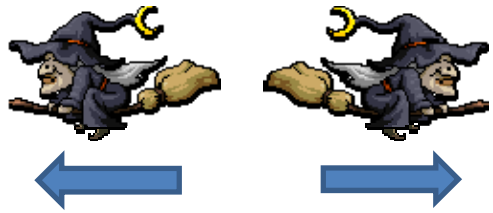
- Text Template ที่กำหนดมา ทำให้กำหนดเป็นชุด Array ได้ดังนี้



- หากต้องการ Random ค่าเหล่านี้ มาอยู่ใน Array ที่มีสมาชิก 10 ตัว โดยจะเก็บค่า Index ของ text ไว้ ไม่ใช่เก็บ ข้อความไว้ ใน message
- หากต้องการเรียกใช้ ค่าในtext ให้ทำตามดังนี้ **text[message[i]]**
- โดย i เป็นค่าของสมาชิกของจำนวนข้อความใน 1 หน้าจอ

จาก Lab 3.3 เกมเก็บของ

- ให้ใช้แม่มดอยู่ด้านล่างสุดของจอ โดยให้แม่มดเป็นตัวที่ใช้เก็บของ
- โดยแม่มดสามารถเลื่อนซ้ายเลื่อนขวาได้



- ให้วาดคะแนนอยู่มุมขวาบน
- ให้หาทำการปล่อยภาพต่อไปนี้ลงมา

- โดย **random** จุดปล่อยลงมาในแนวแกน **x**
- ให้เก็บเฉพาะภาพสีเหลือง ได้คะแนนภาพละ 100
- หากเป็นภาพสีแดงต้องไม่เก็บ และ คะแนนจะลดลง 100 คะแนนถ้าไปเก็บโดนสีแดง
- แต่หากคะแนนน้อยกว่า 0 ให้เท่ากับ 0

Lab 3.3

- ก่อนอื่นต้องคิดก่อนว่าจากโจทย์ต้องมีองค์ประกอบอะไรบ้าง
- ซึ่งหากได้อ่านโจทย์จะแยกเป็น 2 ส่วน คือ
  - ส่วนแม่มด
  - ลูกบอลที่ตกลงมา
- แม่มดต้องเก็บค่าอะไรบ้าง
  - พิกัด
  - กว้าง ยาว
  - การเก็บค่า SpriteEffect
  - ~~Texture2D~~

Lab 3.3

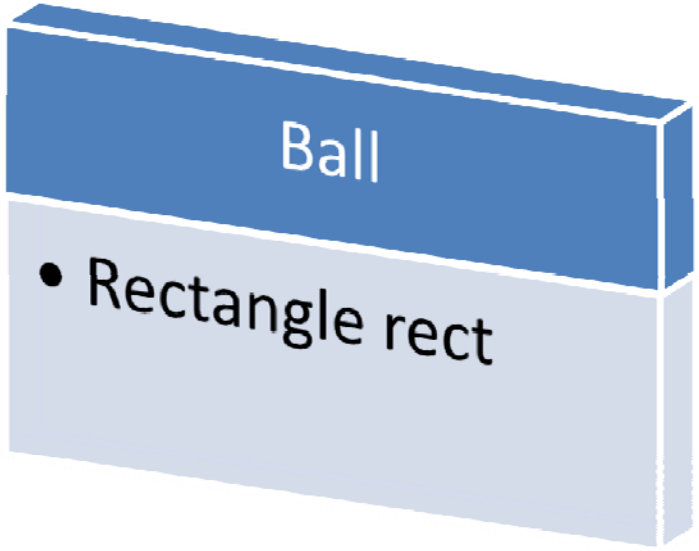
- เนื่องจากการเก็บค่าของแม่มด มีแค่ 2 อย่าง ซึ่งจริงๆ สามารถเก็บได้ใน Rectangle อยู่แล้ว และอาจจะค่า SpriteEffect อีกตัวเท่านั้น
- ทำไมการเก็บไฟล์รูปภาพไม่ถือเป็น data ของแม่มด
  - เพราะหากมีการเก็บแม่มดหลายตัว ซึ่งจริงๆ แล้วก็ใช้แค่ภาพเดียวกัน จึงไม่จำเป็นต้องเก็บหลายตัวแปร จึงไม่จำเป็นที่ต้องทำ Array กับไฟล์ภาพ ไม่ว่าอันไหน เพราะเราสามารถสั่ง draw ได้หลายตัวอยู่แล้ว
- จำเป็นต้องสร้าง Class หรือ struct ขึ้นมารองรับหรือไม่
  - การสร้าง Class สำหรับแม่มด อาจจะเป็นการดีในการออกแบบตามหลัก Object Oriented Concept แต่ในโจทย์ข้อ 3.3 อาจจะไม่จำเป็น
- แล้วเราต้องสร้าง Class สำหรับ แม่มดเมื่อไหร่
  - เมื่อมันมีเก็บค่าที่มากขึ้นเช่น หากความเร็วไม่คงที่
  - แม่มดมีมากกว่า 1 ตัว เป็นต้น

Lab 3.3

- มาดูส่วน ลูกบอล จำเป็นหรือเปล่า ที่จะต้องใช้การเขียนโปรแกรมแบบ Object Oriented มาช่วย
- ก่อนอื่นมาดูตัวแปรที่ใช้กันก่อน
  - พิกัด
  - กว้าง ยาว
  - สีแดง หรือ เหลือง
- เห็นได้ชัดเลยว่า มี ตัวแปร เพียงแค่ 2 ตัว คือ Rectangle กับชนิดของสีเท่านั้น

Lab 3.3

- หากไม่ออกแบบตามโดยใช้ Class แล้วจะทำอย่างไรได้บ้าง
  - ทำตัวแปรทุกตัวของลูกบอลให้เป็น Array ตามจำนวนที่อยากให้ลูกบอลสามารถหล่นลงมาได้พร้อมกัน
- หากคิดเป็นลักษณะ Class ให้สร้าง class ลูกบอลขึ้นมาก่อนดังต่อไปนี้



ตัวอย่าง Class Ball

```
class Ball
  {
      Rectangle rect;
  }
```

จาก Diagram จะเห็นได้ว่า Class เริ่มต้นของ

Lab 3.3

- หากต้องทำชนิดของลูกบอล ควรจะทำอย่างไร
  - Boolean ใช่มั๊ย
  - int ใช่หรือไม่
  - หรือหากจะต้องใช้ enumeration

  Boolean หากมันมีชนิดมากกว่า 2 แบบ ก็คงจะเป็นปัญหาแน่นอน

  หากเก็บเป็นจำนวนเต็ม ก็คงเข้าใจยากอีกว่า 1 หมายถึงอะไร 0 หมายถึงอะไร

  การใช้ enum จึงเป็นคำตอบที่ดีสำหรับ case เหล่านี้ ไม่ว่าจะเป็น ชนิด

การใช้ enum ให้กำหนดดังนี้

```
enum BallType
  {
      RedBall,
      YellowBall
  }
```

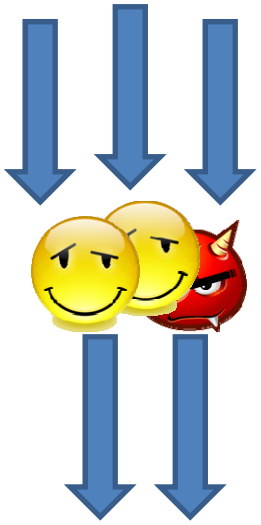
- BallType เป็นชื่อของ enum
- RedBall และ YellowBall เป็นชื่อชนิดของ Ball

Lab 3.3

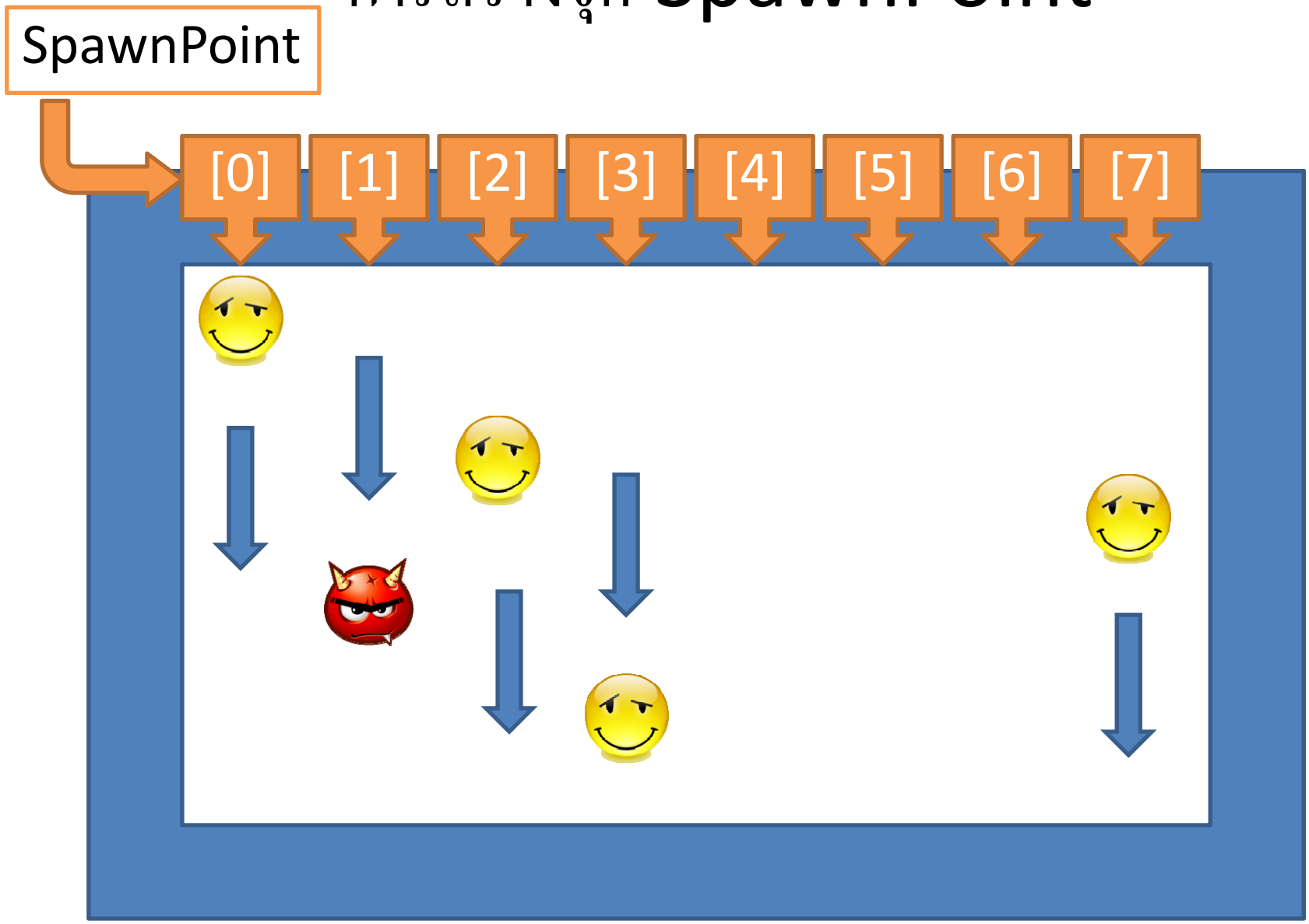
- นอกจากการใช้ enum ในชนิดแล้ว เรายังใช้ enum ในการแทนค่าของสถานะได้อีกด้วย เช่น BallState.Ball_Falling
- การใช้แทนสถานะเพื่อให้รู้ว่าตอนนี้บอลอยู่ในจังหวะที่ควรปล่อยออกมาหรือไม่
- หากเราวาด Ball มากเกินไปจะเกิดอะไรขึ้นได้บ้าง
  - เครื่องกระตุก
  - ควบคุมเกมไม่ได้
  - เกมค้าง

การจำกัดบอลที่หล่น

- สามารถทำได้โดยการกำหนดจำนวนบอลที่มีอยู่ในหน้าจอ
- ทำอย่างไรหากต้องการให้ลูกบอลสลับกันออกมา ไม่ใช่ หล่นอย่างคงที่
  - ทำ State Event ให้กับลูกบอล
  - สร้างระยะเวลาการหล่นของลูกบอล
- การแก้ปัญหาบอลหล่นมาติดกันจนเกินไป
  - สร้างจุด spawn point



การสร้างจุด SpawnPoint



Random จุด SpawnPoint

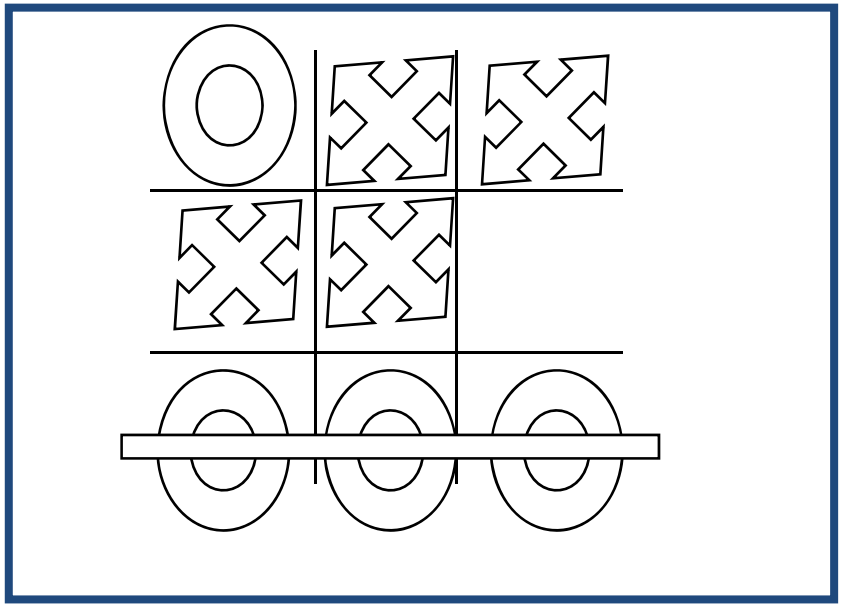
- ส่วนใหญ่ การ Random Position จะทำการ Random พิกัด x และ y โดยตรง แต่ในครั้งนี้ จะทำการ random ค่า index มาใช้แทนเช่น

```
r= random.Next(8);
Vector2 pos=spawnPoint[r];
```

การเช็ค Collision ของลูกบอลในหน้าจอ

- ในเงื่อนไขของเกม ต้องเช็ค Collision ของลูกบอล เช่น
  - ใช้ Array เพื่อเก็บ
  - กำหนดตัวแปร ball1, ball2…. Ball10
  - สร้าง Class BallsManager ไว้คอยดูแลการหล่น ยันเช็ค collision
  - อื่นๆ

1. เกม OX(Tic Tac Toe)



- วิธีเล่นให้ผู้เล่น เลือก ช่องที่จะใส่ O และ ส่วนการเลือก X ให้ Computer เป็นผู้เลือก และเลือกสลับกันไปจนกว่าจะได้ผู้เล่นที่ได้แถวเดียวกัน 3 อัน

1. เกม OX(Tic Tac Toe)

- ใน OX วิธีการเลือกวางของ Computer นั้น สำหรับ 3*3 ช่องนั้น ไม่ถือว่าเป็นการใช้ AI
- ลองคิดหาวิธีที่ให้ computer สามารถวางดักทางเราให้ได้ จะมี Bonus พิเศษให้สำหรับคนที่ทำได้
- ส่วนการควบคุมให้กำหนดเอง และเขียนบอกมาในหน้าจอด้วยว่าบังคับอย่างไร

การเก็บข้อมูลใน OX

- OX นี้มีตารางทั้งหมด 9 ช่อง มีวิธีอะไรบ้าง
  - ตัวแปร 9 ตัว เช่น Block1, …. Block9
  - Array 2 มิติ
  - Array 1 มิติ
  - Int 2 ตัว อันนี้ไม่เทพ อย่าทำนะครับ
- การเก็บตัวแปรจะเก็บแบบไหนได้บ้าง
  - Boolean
  - Int
  - enum

ค่าที่เก็บในแต่ละ Block

- โดยปกติหากเราใช้ enum ในการเก็บจะมี ค่าอะไรบ้าง
  - None
  - ค่าของ O
  - ค่าของ X
- เมื่อระบุค่าของ Block เข้าไปในแต่ละ Block ก็สามารถทำ condition ในการวาดได้ โดยเช็คว่าเป็นค่าแบบไหน ก็ให้วาดแบบนั้น

การเช็คเงื่อนไข

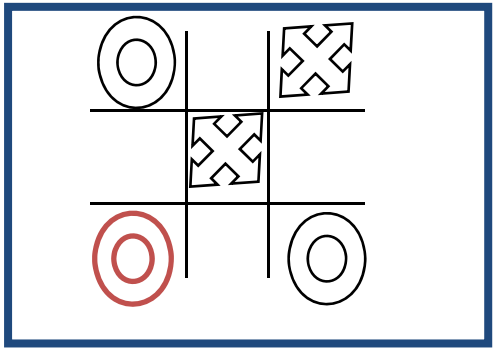
- การเช็คเงื่อนไขแพ้ชนะ
  - If else ไปเรื่อยๆ มีทั้งหมด 8 case ที่จะชนะได้
  - เช็คจากตำแหน่งที่ลงครั้งล่าสุดว่ามีเข้าข่ายชนะหรือไม่
  - อื่นๆ

Computer Player

- การวาง Block อัตโนมัติของ Com แบ่งเป็น 2 ชนิด
  - Com เล่นก่อน
  - Com เล่นเป็นคนที่ 2

Com เล่นก่อน

1. เช็คว่าฝั่งตรงข้ามมี case ไหนที่ชนะบ้าง ให้วางดัก
2. หาก Com เป็นผู้เล่นก่อน ต้องเลือกวางในจุดที่ได้เปรียบ ซึ่งต้องเลือกมุมและตรงกลางก่อน
3. หาทางวางให้เป็นรูปตัว L เช่น
4. ถ้าไม่มีตัวเลือกให้หาจุดที่มีโอกาสชนะ
5. กลับไปเริ่ม ที่ 1 ใหม่



Computer Player

Com เล่นเป็นผู้เล่นที่ 2

1. เช็คว่าฝั่งตรงข้ามมี case ไหนที่ชนะบ้าง ให้วางดักโดยให้ระวังอย่าให้เกิดรูปตัว L ขึ้น
2. วางดักหากตรงกลางยังว่าง
3. หาทางวางให้ชนะ แต่ใน 3X3 ฝ่ายที่เล่นหลัง มีโอกาสชนะน้อยมาก ส่วนใหญ่ได้แต่วางให้เสมอ