Game Performance

Suphot Sawattiwong
tohpus@hotmail.com

Performance

- ประสิทธิภาพในการทำงาน
- เราจะรู้ได้อย่างไรว่าเกมที่เราทำมีประสิทธิภาพที่ดี
  - เกมมี graphic ที่สวยงาม ? เป็นเรื่องคุณภาพ
  - ใช้เนื้อที่น้อย ? เป็นเรื่องประสิทธิภาพ
  - เกมเล่นแล้วภาพกระตุก เป็นเรื่องประสิทธิภาพ
  - ใช้ Memory น้อย? เป็นเรื่องประสิทธิภาพ
  - เกมเล่นแล้วสนุกสุดยอด ? เป็นเรื่องคุณภาพ
  - โหลดเกม โหลดฉากได้เร็ว เป็นเรื่องประสิทธิภาพ

Measurement

- คือการวัดค่าประสิทธิภาพต่างๆ
- ในเกมเราวัดได้จาก.....
  - ความเร็วการวาดใน 1 หน้าจอ
  - จำนวนหน้าจอที่ทำการวาดใน 1 วินาที
  - Running Time
  - การใช้งาน Memory
  - จำนวนความจุของ HDDที่ใช้ในเกม
  - LOD (Lines of Code)
  - อื่นๆ

Benchmark

- เป็นกระบวนการวัดและเปรียบเทียบการทำงานในเกม กับเกมหรือกลุ่มตัวอย่างอื่นๆ เพื่อนำผลของการเปรียบเทียบมาใช้ในการปรับปรุงประสิทธิภาพในเกม
- ตัวอย่างเช่น
  - เกมที่ทำ เปรียบเทียบกับเครื่องคอมพิวเตอร์ที่ spec ต่างกันความเร็วการวาดเท่ากันหรือไม่
  - เกมที่เราทำ ในเครื่องspec ต่ำ สามารถเล่นเกมเราในความเร็วที่เรากำหนดได้หรือไม่

Frames Per Second(fps)

- Frames Per Second คือจำนวนภาพที่วาดต่อ 1 วินาที
- การวัดค่าจาก FPS นั้นสามารถวัดอะไรได้บ้าง
  1. วัดความเร็วในการวาดภาพใน 1 วินาที โดยเกมโดยทั่วไป หากเป็นเกม 2 มิติ FPS ควรจะวิ่งอยู่ที่ 60 FPS
  2. สามารถรู้ได้ว่าในหน้าจอในแต่ละเหตุการณ์กินประสิทธภาพเครื่องต่างกันมากน้อยเพียงใด
  3. สาเหตุการกระตุกเกิดจากการวาด หรือการคำนวณจากจุดอื่นๆ
  4. เป็นต้น

โดย FPS นั้นสามารถดูตัวอย่างการเขียนได้ที่ FPSTest.zip ในเวป

Algorithm ในการคำนวณ FPS

1. ทำการวัดช่วงเวลาในตอนเข้าทำงาน (elapsed) ในส่วนของ Draw Method
2. ให้ทำการเพิ่มค่าจำนวนภาพ(framecount) ทุกๆ ครั้งที่มีการวาด หรือการใช้ Draw Method
3. ทำการบวกช่วงเวลาที่ได้ (elapsed) เข้าไปในเวลาในการเปรียบเทียบครั้งล่าสุด (timeSinceLastUpdate)
4. นำระยะเวลาในการเปรียบเทียบครั้งล่าสุด (timeSinceLastUpdate) มาทำการเปรียบเทียบกับเวลาในเปรียบเทียบ (updateInterval) หรือ 1 วินาที หากเวลาในการเปรียบเทียบครั้งล่าสุดมากกว่าให้กระทำต่อในขั้นที่ 5 หากน้อยกว่าให้ทำการวาดภาพตามปกติใน Draw Method

Algorithm ในการคำนวณ FPS (ต่อ)

5. นำค่าที่ได้มาทำการคำนวณ FPS ดังสูตรต่อไปนี้ fps = framecount / timeSinceLastUpdate
6. ทำการreset ค่าจำนวนภาพ (framecount) ให้มีค่าเป็น 0 เพื่อเริ่มนับใหม่
7. ทำการ reset ค่าเวลาในการเปรียบเทียบครั้งล่าสุด (timeSinceLastUpdate) โดยการนำเวลาที่ใช้เปรียบเทียบ (updateInterval) ลบด้วยเวลาในการเปรียบเทียบครั้งล่าสุด

คำสั่งที่ช่วยในการคิด FPS ใน XNA

- graphics.SynchronizeWithVerticalRetrace = true หรือ false
  - ยอมให้มีการ Sync Draw Method กับการทำ Vertical Retrace ในกระบวนการ refresh rate ของ Monitor หรือไม่
- IsFixedTimeStep = true หรือ false
  - ยอมให้มีการfixed rate ให้อยู่ที่ 60 fps หรือ ทำให้ fps เป็นอิสระ
- gameTime.ElapsedRealTime เป็น ช่วงเวลาของการ Draw Method
- gameTime.ElapsedGameTime เป็น ช่วงเวลาของการ UpdateMethod

จากตัวอย่างเรื่องการ Check Performance

All Method Time – Avg: 0.53 Total: 0.0053
Initialize TransformWith Calculation – Avg: 400.83 Total: 4.0083
Constant – Avg: 390.16 Total: 3.9016
Division – Avg: 388.14 Total: 3.8814
ConstantReferenceOut – Avg: 320.48 Total: 3.2048
AspectRatio – Avg: 340.90 Total: 3.409

——

TransformVector Reference – Avg: 178.79 Total: 1.7879
Value – Avg: 191.28 Total: 1.9128
ReferenceAndOut – Avg: 180.97 Total: 1.8097
RefOutVectorAdd – Avg: 142.83 Total: 1.4283

——

CreateCameraReferenceWith Property – Avg: 162.17 Total: 1.6217
Value – Avg: 160.52 Total: 1.6052

——

RotateWithMod – Avg: 28.07 Total: 0.2807
RotateWithoutMod – Avg: 24.69 Total: 0.2469
RotateElseIf – Avg: 24.64 Total: 0.2464

จากตัวอย่างเรื่องการ Check Performance ครั้งที่ 3

All Method Time – Avg: 0.55 Total: 0.0055
Initialize TransformWith Calculation – Avg: 392.75 Total: 3.9275
Constant – Avg: 386.44 Total: 3.8644
Division – Avg: 420.50 Total: 4.205
ConstantReferenceOut – Avg: 320.76 Total: 3.2076
AspectRatio – Avg: 332.41 Total: 3.3241

——

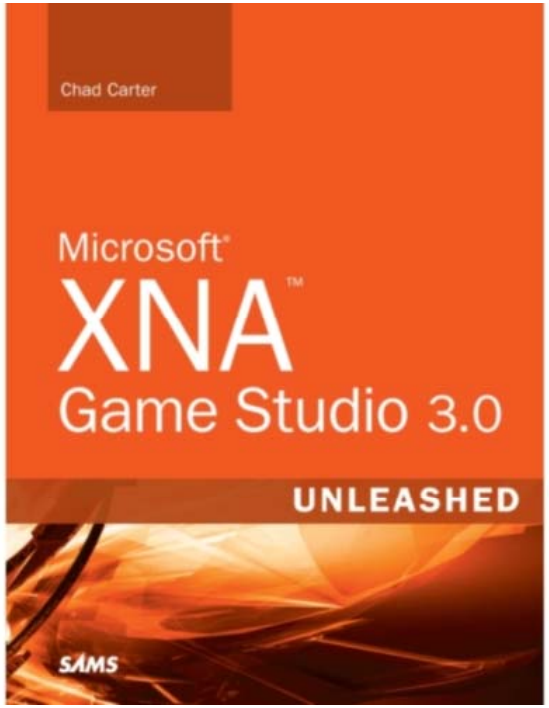
TransformVector Reference – Avg: 178.79 Total: 1.7879
Value – Avg: 191.25 Total: 1.9125
ReferenceAndOut – Avg: 209.30 Total: 2.093
RefOutVectorAdd – Avg: 144.23 Total: 1.4423

——

CreateCameraReferenceWith Property – Avg: 160.49 Total: 1.6049
Value – Avg: 159.23 Total: 1.5923

——

RotateWithMod – Avg: 28.04 Total: 0.2804
RotateWithoutMod – Avg: 24.64 Total: 0.2464
RotateElseIf – Avg: 24.64 Total: 0.2464

Reference

- Microsoft XNA Game Studio 3.0 Unleashed, Chad Carter, SAMS