

CONTESTED MULTIPLAYER ONLINE GAME

STATIC TESTING

BY APICHL & CHANRUNGRUEANG

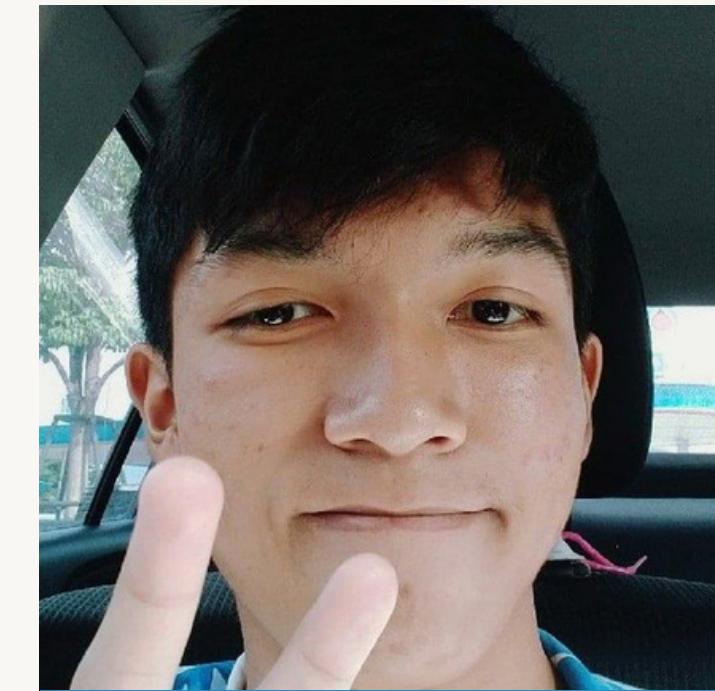


ສມາຟຶກ

Team



ชาณรุ่งเรือง ຈັນກວາຮາ



ອກົບຍ ກັພງູຕາ

CONTESTED MULTIPLAYER ONLINE GAME

เหตุผลในการเลือกโปรเจค

- Size ของโปรเจค
- ภาษาของโปรเจค
- สนใจโปรเจคประเภทไหน





TOPIC

หัวข้อ

- 01 Reusability
- 02 Complexity
- 03 Understandability
- 04 Maintainability

REUSABILITY

การวัดการนำกลับมาใช้ใหม่ของโค้ด

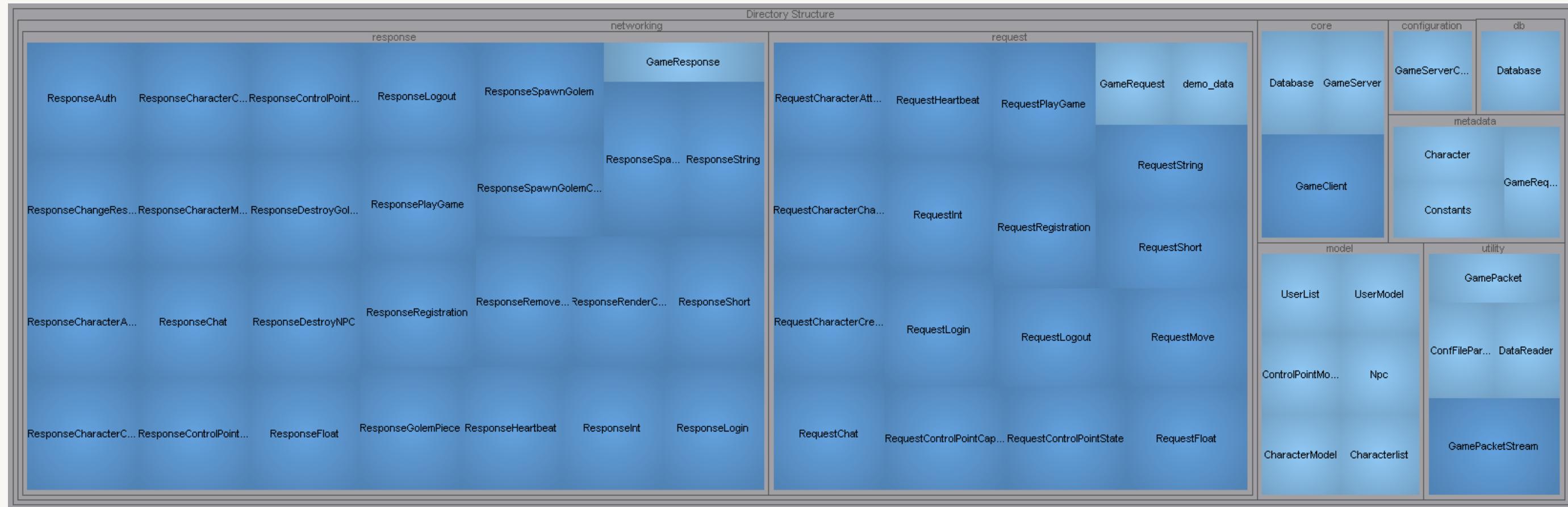
IT

Inheritance Tree

IT จะสามารถวัดได้ว่า [super-class](#) ส่งผลต่อ class อื่นๆ ได้มากน้อยแค่ไหน โดยจะเกี่ยวข้องกับ OOP

Inheritance Tree | Metric



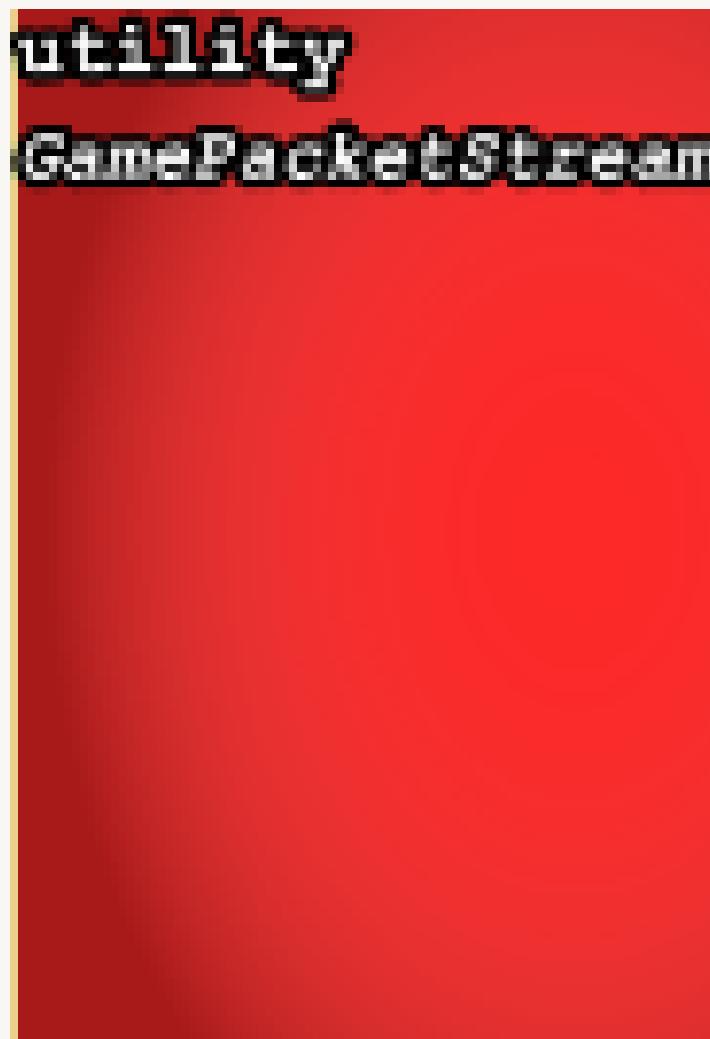
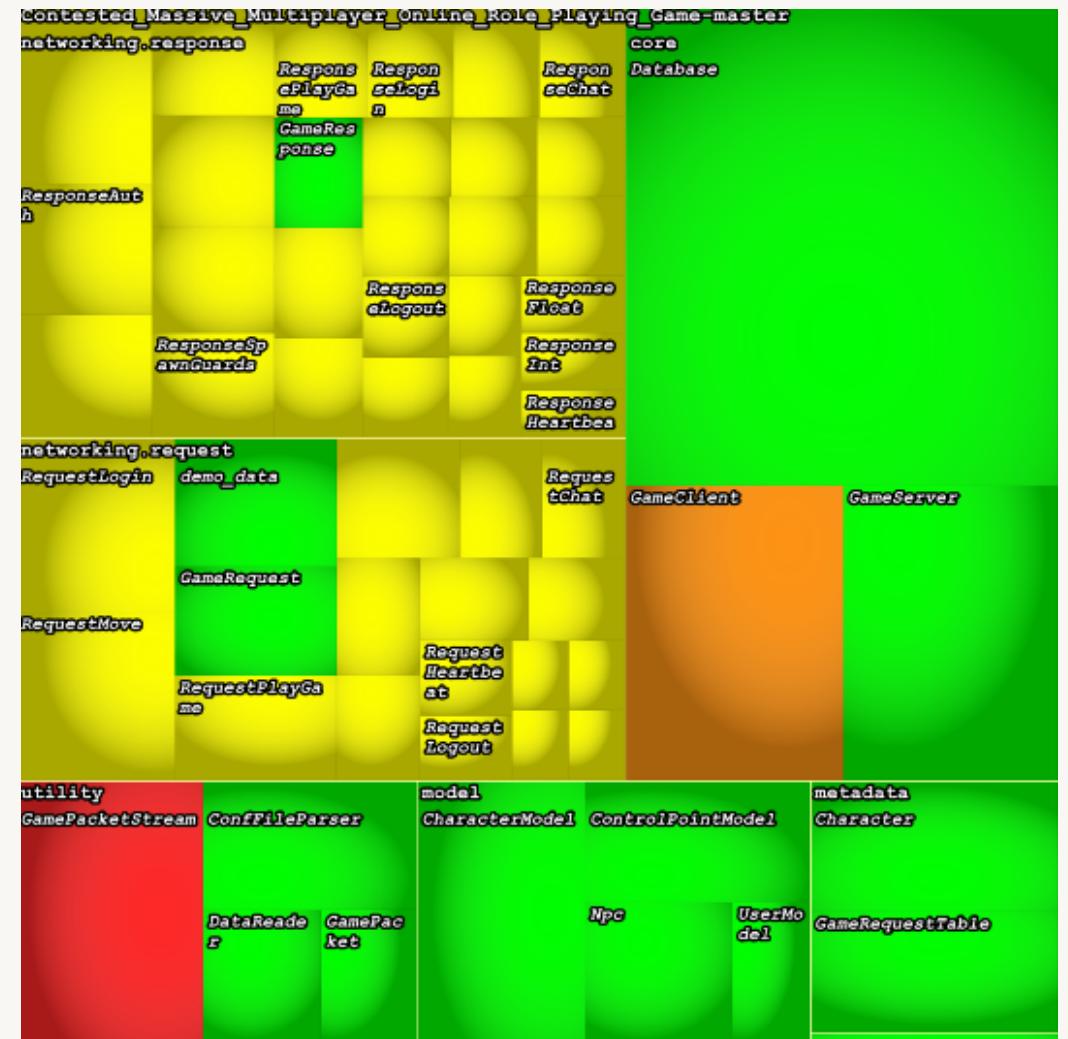


UNDERSTAND

Static Testing

Inheritance Tree | Metric

INHERITANCE TREE



JARCHITECT

Static Testing

Inheritance Tree | Metric

INHERITANCE TREE

จากการพบว่าในส่วนของ GamePacketStream มี 3 unit ซึ่งเป็นค่าสูงสุด

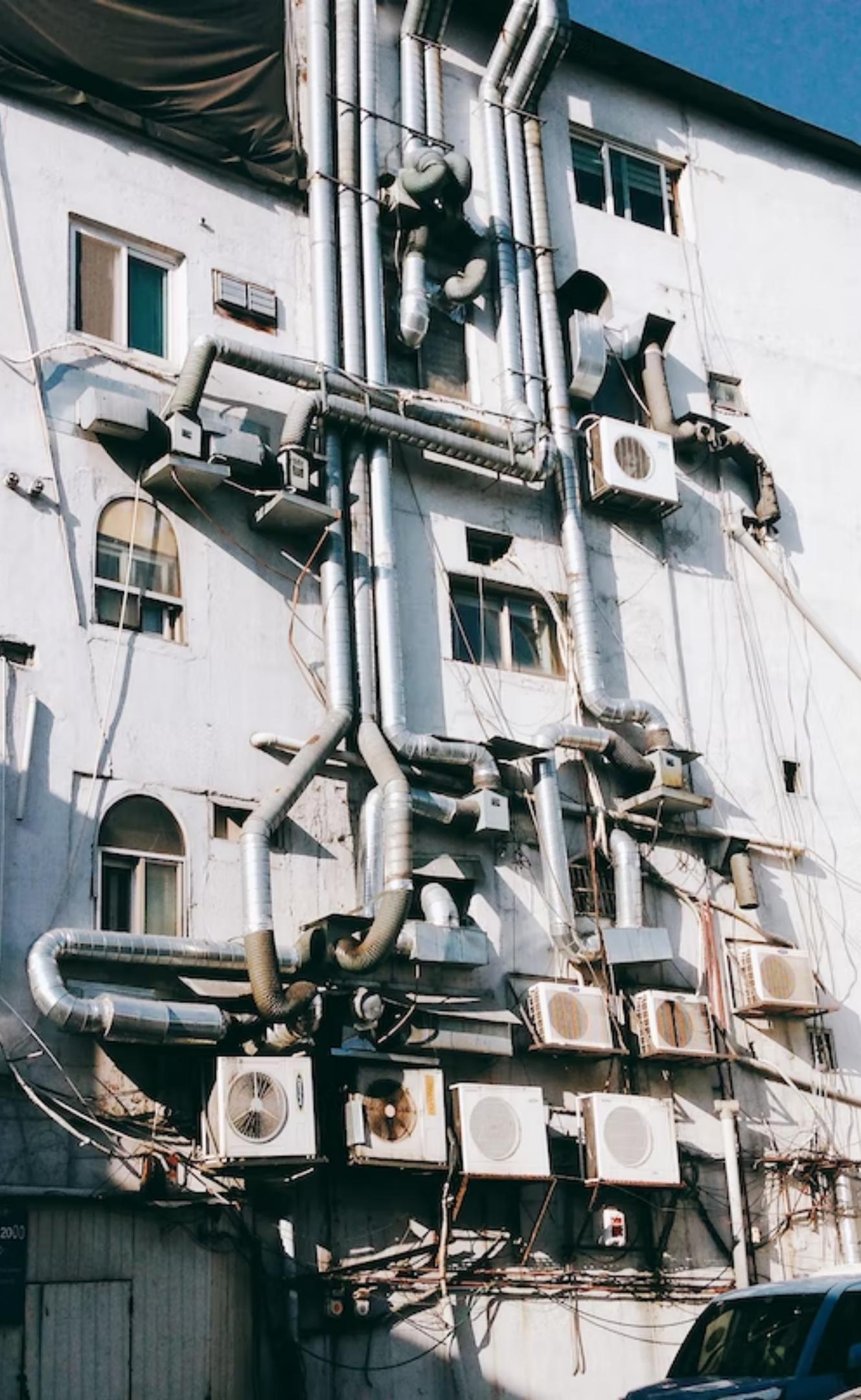
COMPLEXITY

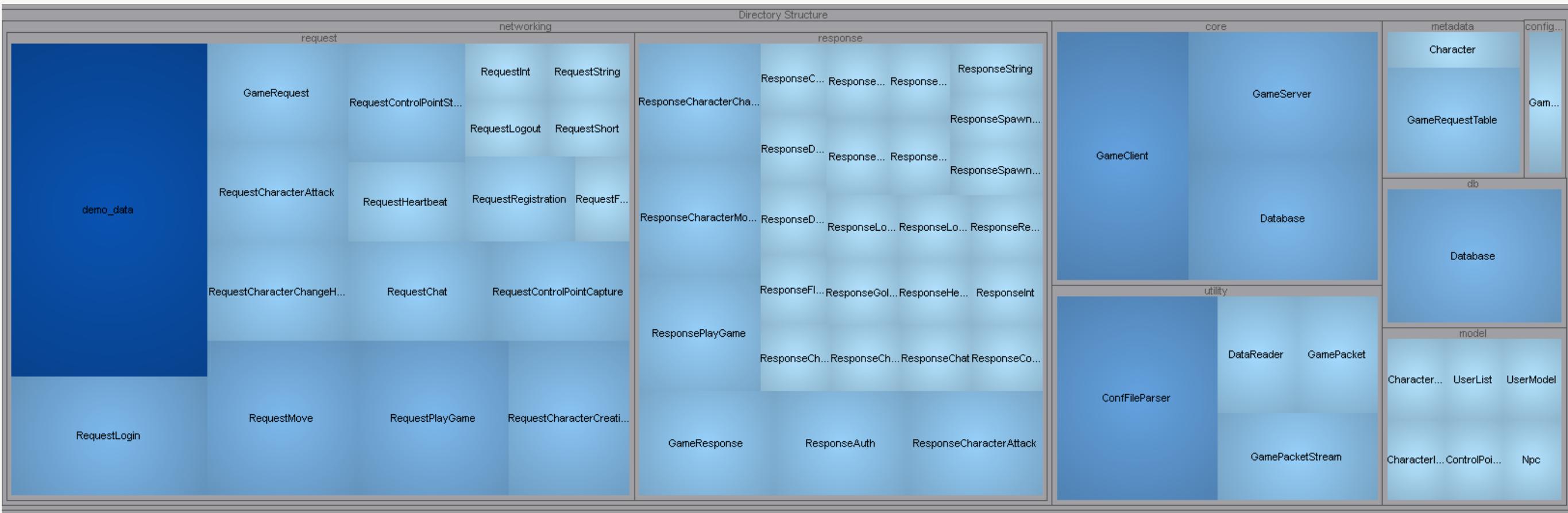
ความซับซ้อนของโค้ด

CC

ใช้สำหรับการวัดค่าความซับซ้อนของ class และ method โดยพื้นฐานจะบันทุกความซับซ้อนของ method **จากเส้นทางกั้งหมุด** กีเป็นไปได้ของการประมวลผล

Cyclomatic Complexity | Metric





UNDERSTAND

Static Testing

Cyclomatic Complexity | Metric

CYCLOMATIC COMPLEXITY

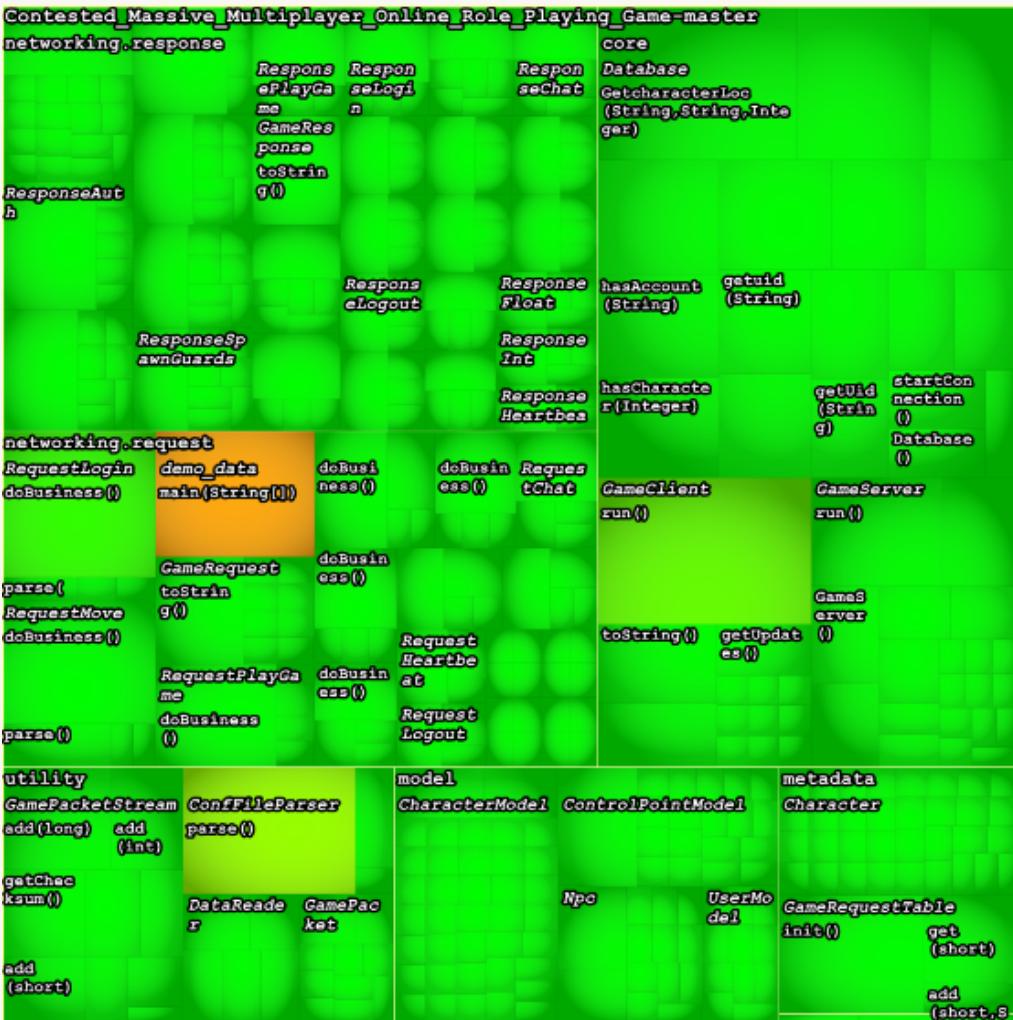
ทดลองในระดับ Class โดยมี

1. demo_data กี่อยู่ในส่วนของ request ได้ 14 Unit
 2. GameClient กี่อยู่ในส่วนของ core ได้ 7 Unit
 3. ConfFileParser กี่อยู่ในส่วนของ utility ได้ 7 Unit

JARCHITECT

Static Testing

Cyclomatic Complexity | Metric



CYCLOMATIC COMPLEXITY

JArchitect สรุปส่วนของ demo_data main(String[]) นั้นจะขึ้นเป็นสีส้มซึ่งสื่อถึงว่ามีความซับซ้อนใน class นี้สูง โดยได้ค่าสูงสุดคือ 14 unit

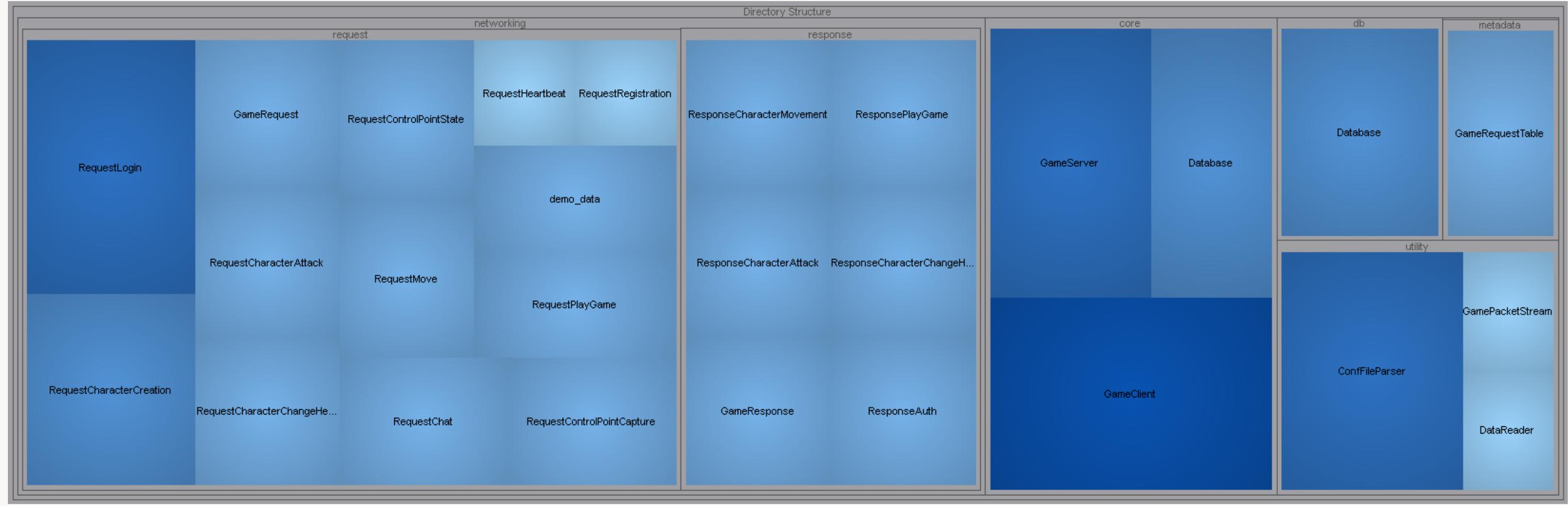
ND

Nesting Depth

จำนวนคำสั่งเงื่อนไข (branches, loops) ที่ถูกซ้อนกันอยู่สำหรับ
ควบคุมโครงสร้างของโปรแกรม

Nesting Depth | Metric





UNDERSTAND

Static Testing

Nesting Depth | Metric

NESTING DEPTH

สามารถทดสอบ Nesting Depth ในระดับ Class

1. GameClient ที่อยู่ในส่วน core มีความกับซ้อนกันที่ 5 unit
2. GameServer ที่อยู่ในส่วนของ core จะมีความกับซ้อนกันที่ 4 unit
3. ConFileParser ที่อยู่ในส่วนของ utility จะมีความกับซ้อนกันที่ 4 unit
4. RequestLogin ที่อยู่ในส่วนของ request จะมีความกับซ้อนกันที่ 4 unit

JARCHITECT

Static Testing

Nesting Depth | Metric

 0 Scopes	Max IL Nesting Depth for Methods
 N/A Scopes	Average IL Nesting Depth for Methods

NESTING DEPTH

JArchitect นั้นมีฟังก์ชันในการตรวจสอบ Nesting Depth ได้แต่ในโปรแกรมที่กดสอบนั้นได้สรุปผลออกมาเป็นค่า max คือ 0 scope และ average คือ N/A

UNDERSTANDABILITY

การวัดความเข้าใจของโค้ด

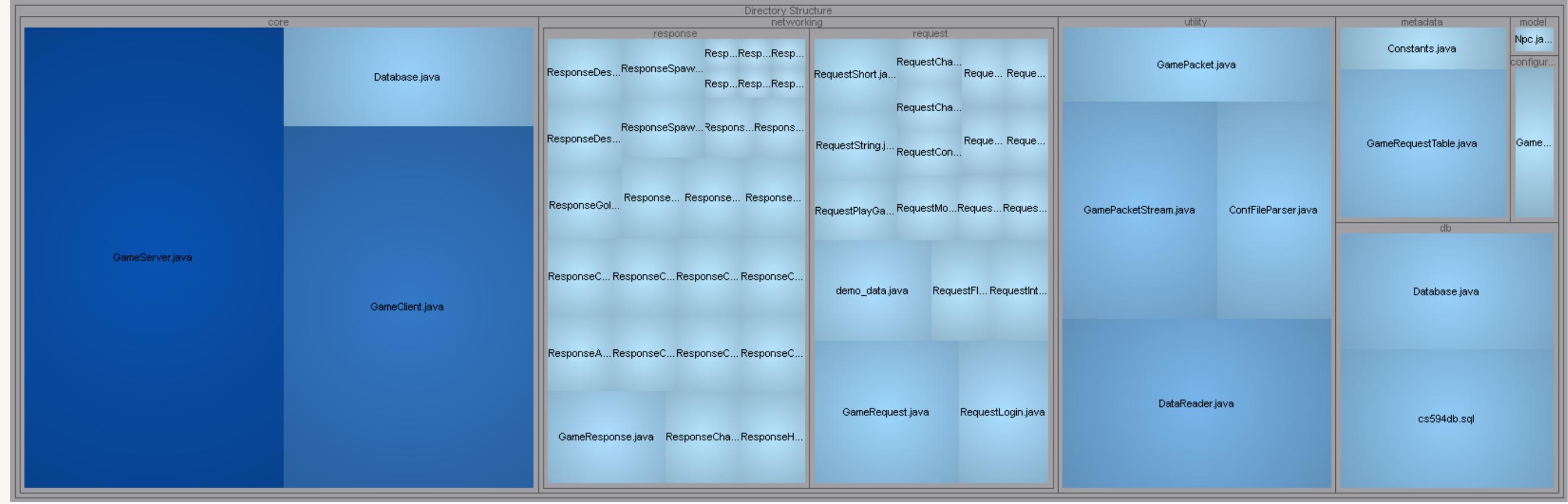
CLOC

Line of Comment

จำนวนของการCommentในแต่ละส่วนของโค้ด เพื่ออธิบายการทำงานของแต่ละส่วนเพื่อให้ง่ายต่อการแก้ไขและพัฒนาต่อไปได้

Percentage of Comment | Metric





UNDERSTAND

Static Testing

Percentage of Comment | Metric

LINE OF COMMENT

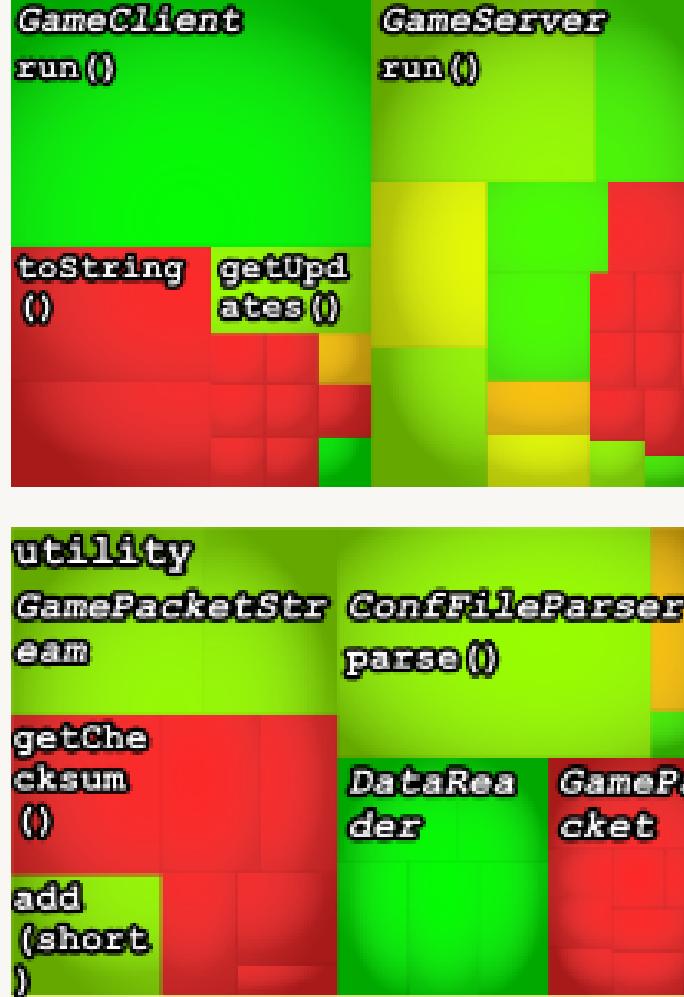
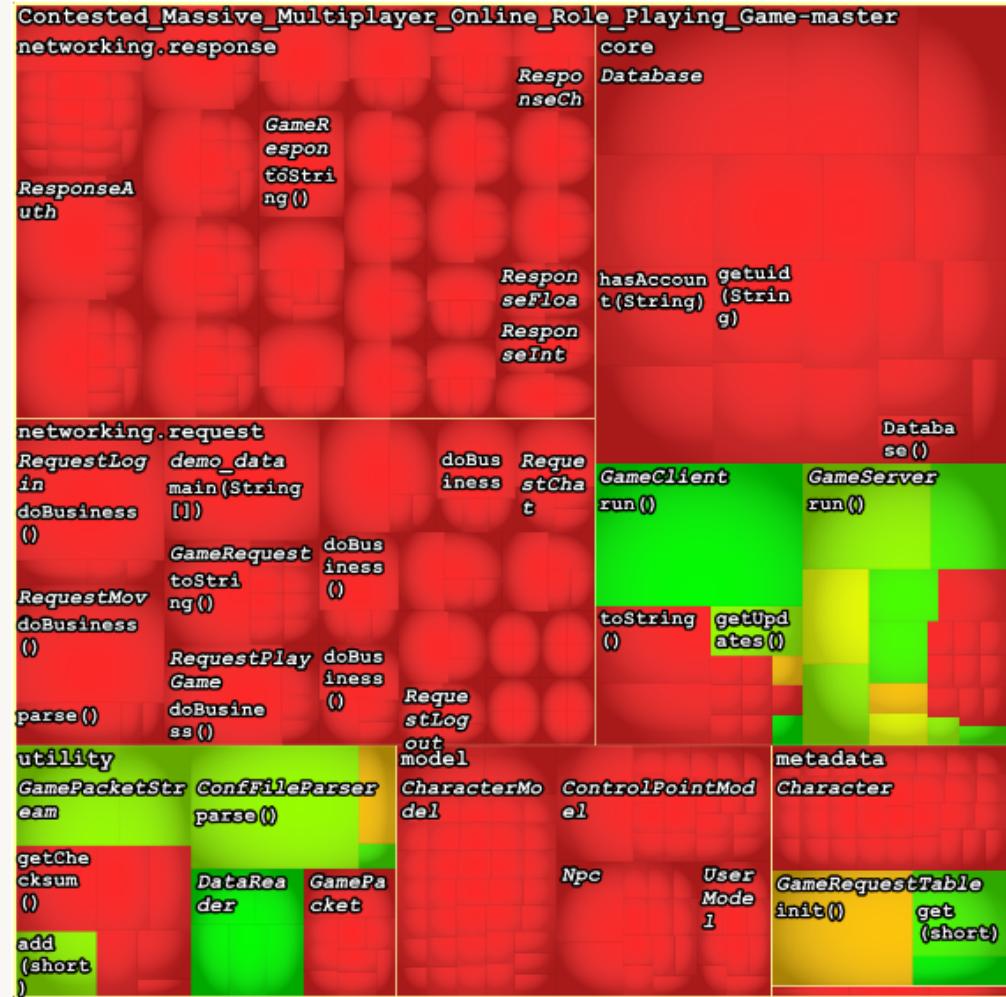
ทดสอบจำนวน Comment ในแต่ละ Class โดยใช้ Count Line Comment ซึ่งได้ค่ามากที่สุดโดยเลือกมา 3 อันดับแรก

1. GameServer จำนวน Comment 115 บรรทัด
2. GameClient จำนวน Comment 87 บรรทัด
3. Data Reader จำนวน Comment 45 บรรทัด

JARCHITECT

Static Testing

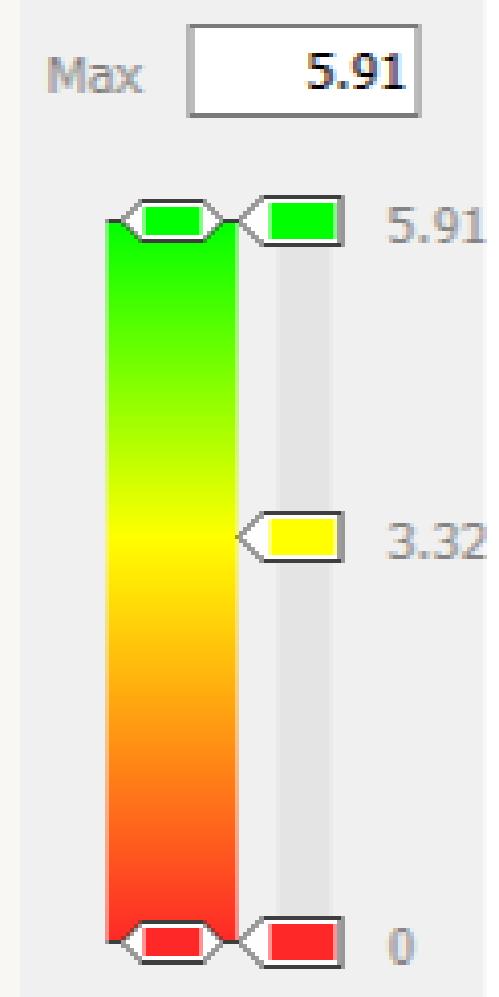
Percentage of Comment | Metric



Comment

4.68%

54 Lines of Comment



CIV

Count Instant Variable

การนับตัวแปรที่กำหนดในคลาสที่เข้าถึงได้ผ่านวัตถุของคลาสนั้น

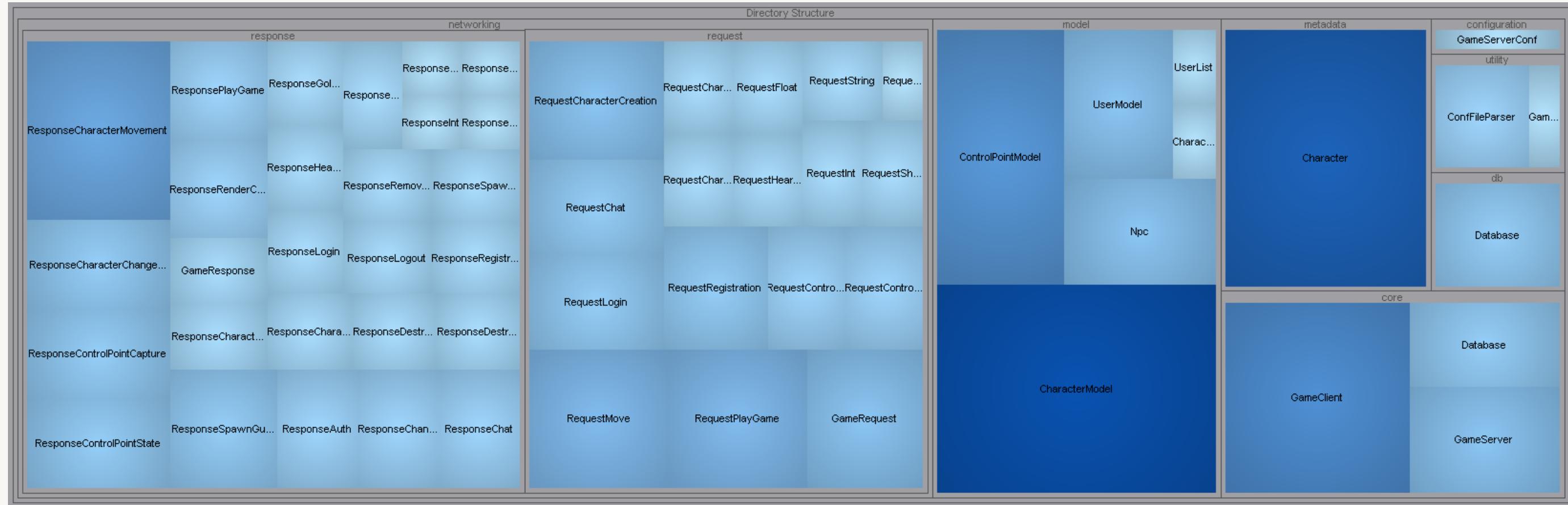
Count Instant Variable | Metric



UNDERSTAND

Static Testing

Count Instant Variable | Metric



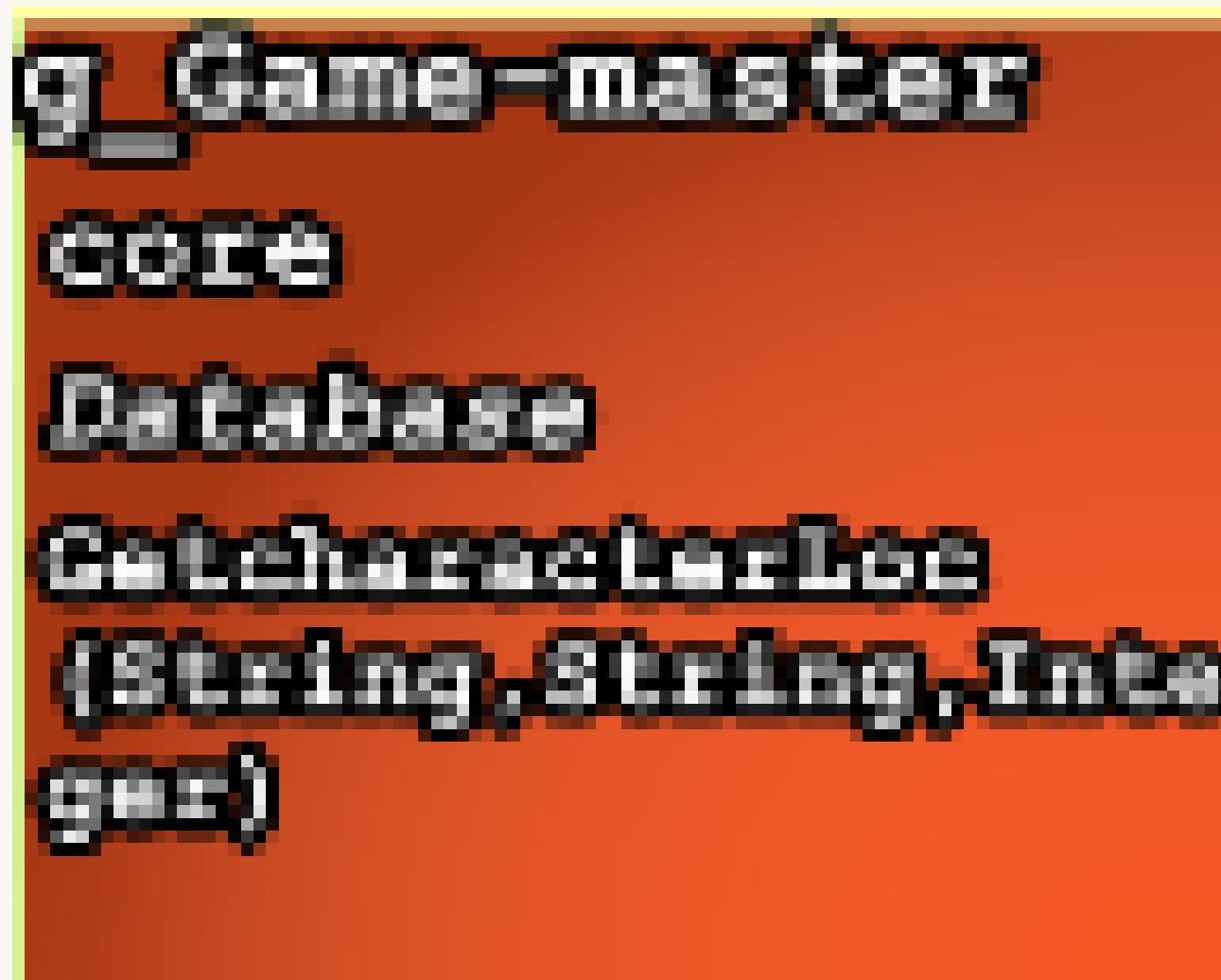
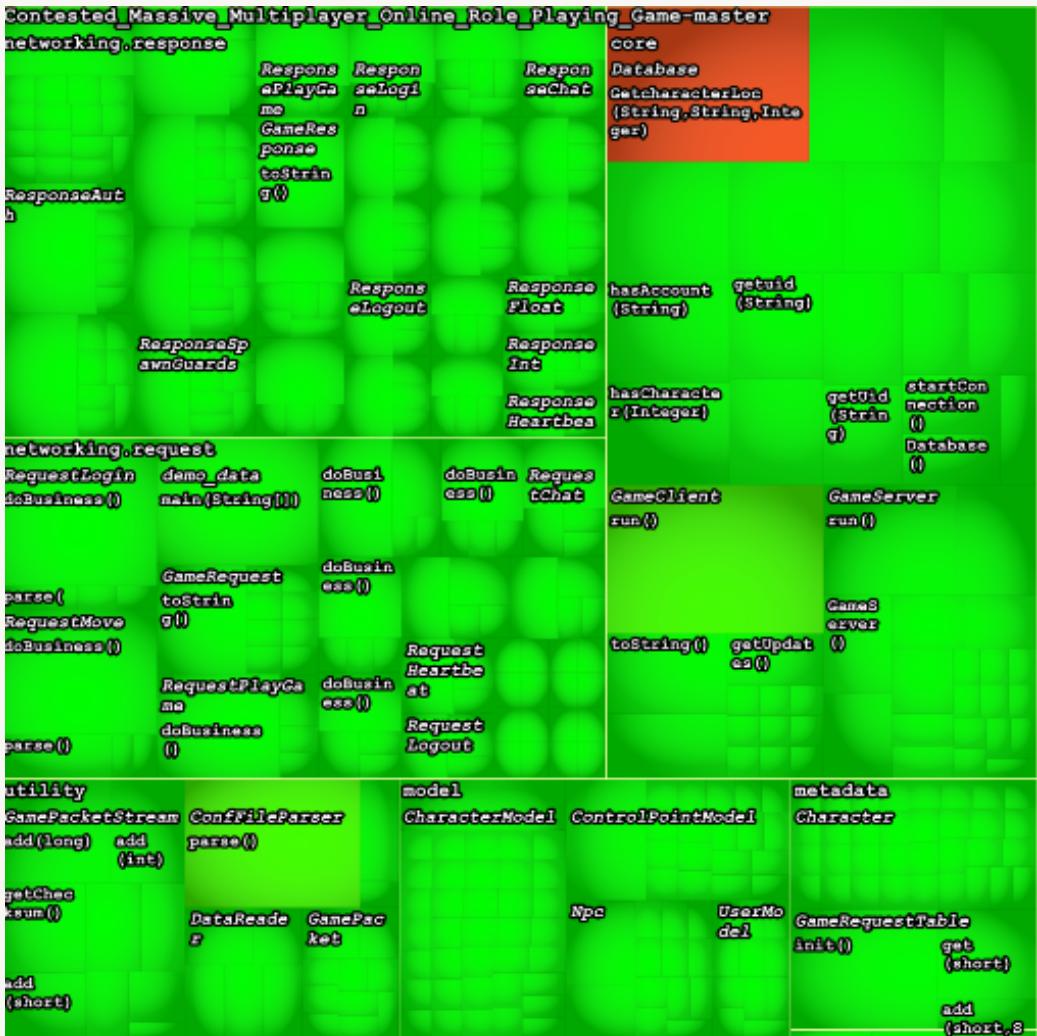
COUNT INSTANT VARIABLE

ทดสอบในระดับของ Class โดย Instant variable มากสุดที่อยู่ในส่วนของ model จะมีตัวแปรกั้งหมด 18 ตัวและ Character ที่อยู่ในส่วนของ metadata จะมีตัวแปรกั้งหมด 16 variable

JARCHITECT

Static Testing

Count Instant Variable | Metric



COUNT INSTANT VARIABLE

จากการพจงเห็นว่าผลลัพธ์ที่ได้มาบันทุกใน Database มีค่า 12 variable

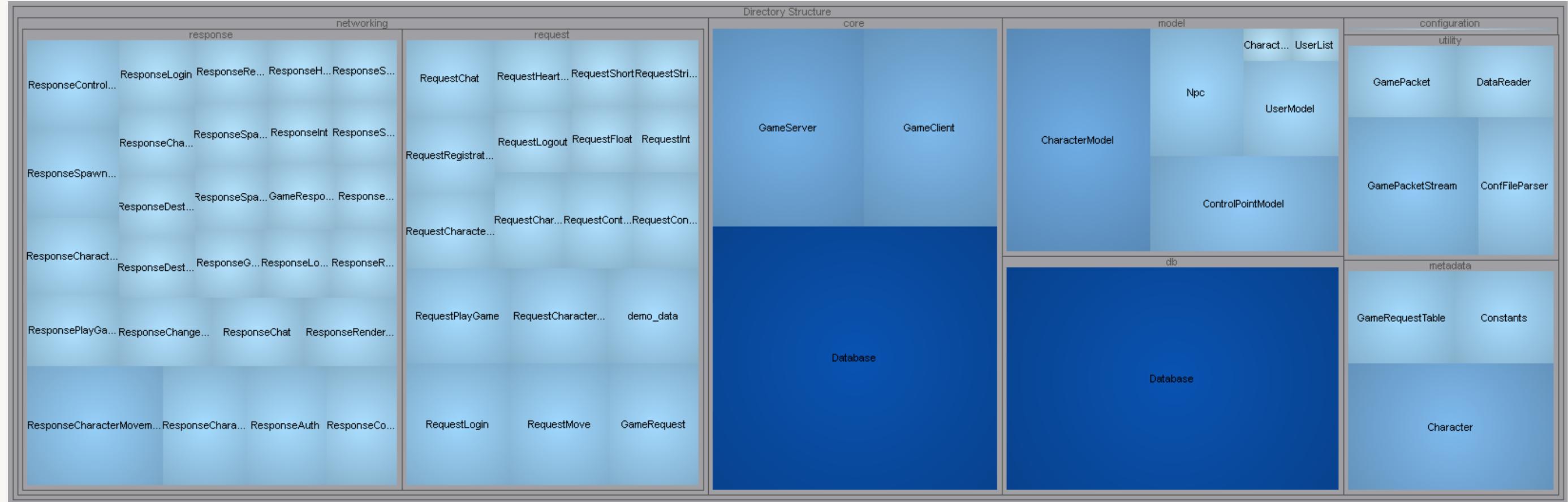
MAINTAINABILITY

การบำรุงรักษาของโค้ด

LOC

Line of Code

จำนวนบรรทัดในข้อความของซอฟต์แวร์สโค้ดของโปรแกรมและ LOC นั้นสามารถกี่จะช่วยประเมินการณ์ได้ เช่น จำนวนชั่วโมงในการเขียน, จำนวนคนที่จำเป็น, รวมถึงประมาณราคางานซอฟต์แวร์



UNDERSTAND

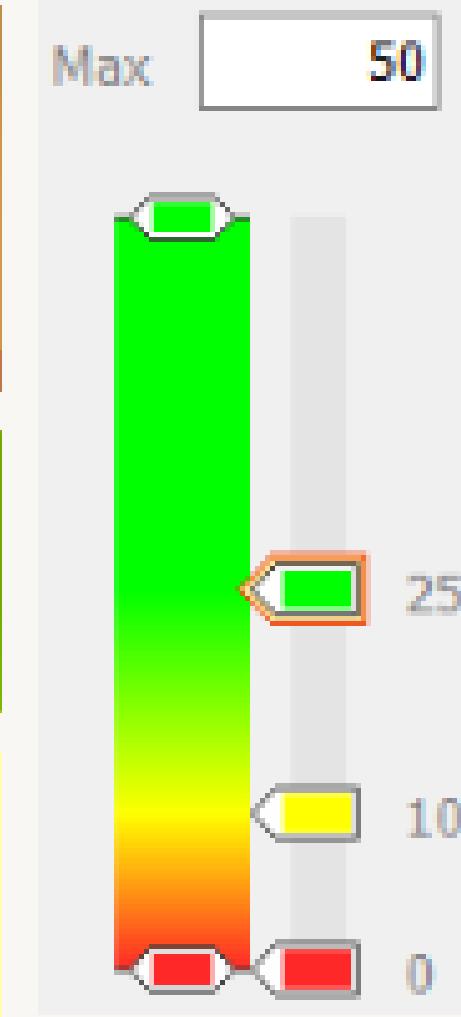
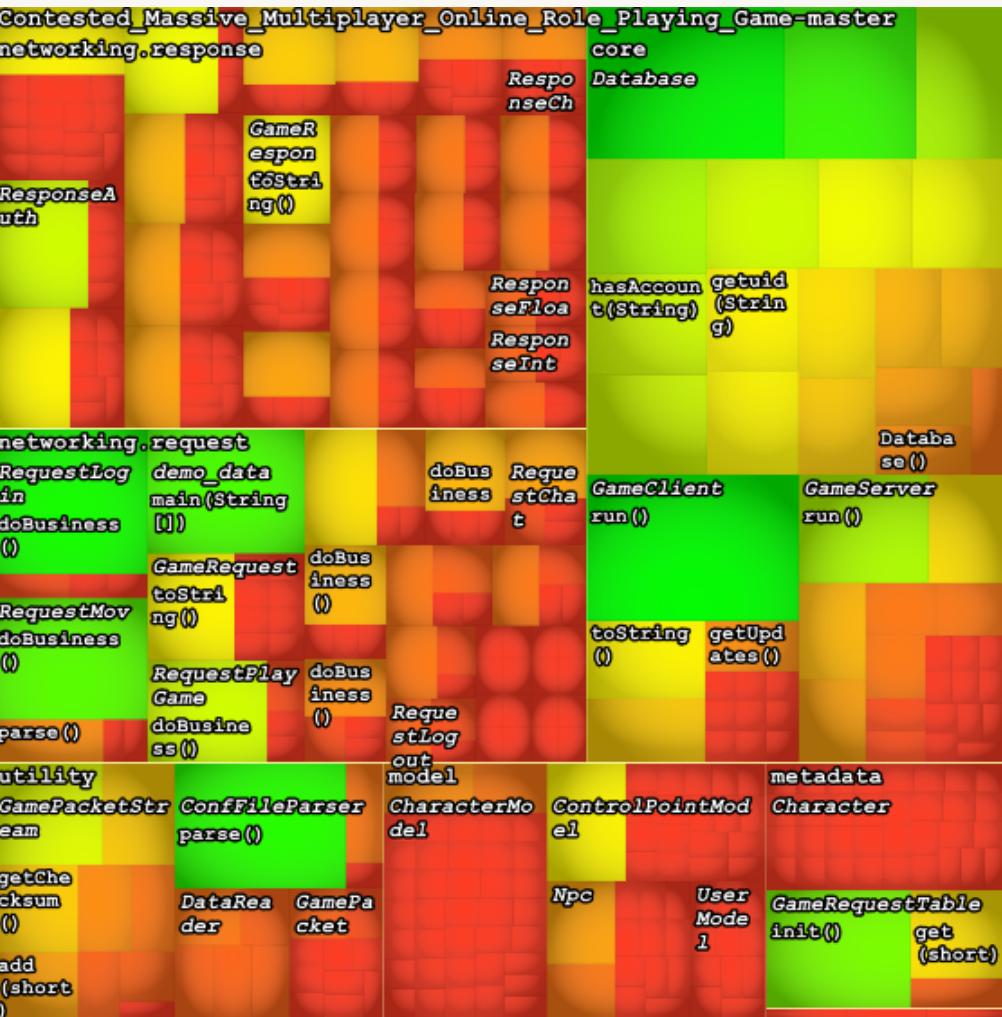
Static Testing

LINE OF CODE

ทดสอบในระดับของ Class โดย Line of Code มากสุดซึ่ง database ในส่วนของ core มี กันหมด 337 บรรทัด และในส่วนของ db มีจำนวน 336 บรรทัด

JARCHITECT

Static Testing



LINE OF CODE

JArchitect นับจำนวน LOC จาก concrete ทำให้จำนวนบรรทัดนั้นน้อยกว่าโปรแกรมที่ใช้ static test ตัวอื่นๆ ทำให้มีค่าสูงสุดคือ 33 บรรทัดใน database

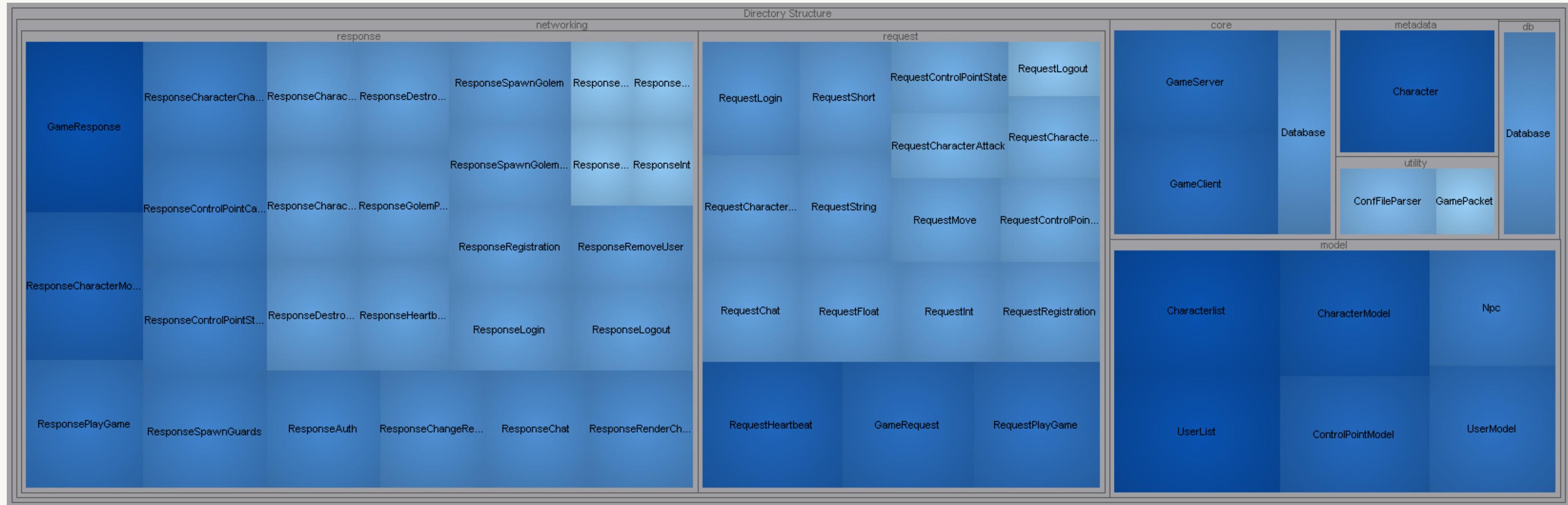
LCOM

Lack of Cohesion

ใช้สำหรับ [วัดค่าความสอดคล้อง](#) กันของ object ซึ่งใน object ควรจะมีหน้าที่รับผิดชอบเดียวกันทำให้สามารถแยก module ของ code ได้โดยหากค่า LCOM มีค่าที่สูงจะแสดงให้เห็นว่า code ส่วนนั้น หรือ module นั้น [สามารถทำงานได้ด้วยตัวเอง](#)

Lack of Cohesion | Metric





UNDERSTAND

Static Testing

Lack of Cohesion | Metric

LACK OF COHESION (LCOM)

การทดสอบในระดับ Class จะแสดงให้เห็นว่า Percent Lack of Cohesion ที่น้อยที่สุด 3 อันดับคือ

1. GamePacket มีเปอร์เซ็นต์ 20 %
2. ResponseString มีเปอร์เซ็นต์ 25 %
3. ResponseShort มีเปอร์เซ็นต์ 25 %
4. ส่วนอื่นๆ มีภาพรวม Percent Lack Of Cohesion ที่ค่อนข้างสูงซึ่งแสดงว่าการทำงานของ Class ในแต่ละ Class มีการทำงานที่สอดคล้องกันค่อนข้างน้อย

	3	Avoid types with poor cohesion
▶	Contested_Massive_Multiplayer_Online_	
▶	core (1 type)	
▶	metadata (1 type)	
▶	model (1 type)	

type poor cohesion ที่เกิดขึ้นใน code smell

	1	Projects with poor cohesion (RelationalCohesion)
	1	Packages with poor cohesion (RelationalCohesion)
▶	Contested_Massive_Multiplayer_Online_	63 types
▶	Contested_Massive_Multiplayer_Online_	
▶	networking.response	27 types

project, package poor cohesion ที่เกิดขึ้นใน architecture

JARCHITECT

Static Testing

Lack of Cohesion | Metric

LACK OF COHESION (LCOM)

ในโปรแกรมไม่ได้สรุปผลในรูปแบบตัวเลข แต่จะบ่งชี้ว่าโค้ดส่วนไหนที่ควรปรับแก้

ទំនាក់ទំសណ៍

Thank you for your attention

