

CONTESTED MULTIPLAYER ONLINE GAME

STATIC TESTING

BY APICHL & CHANRUNGRUEANG

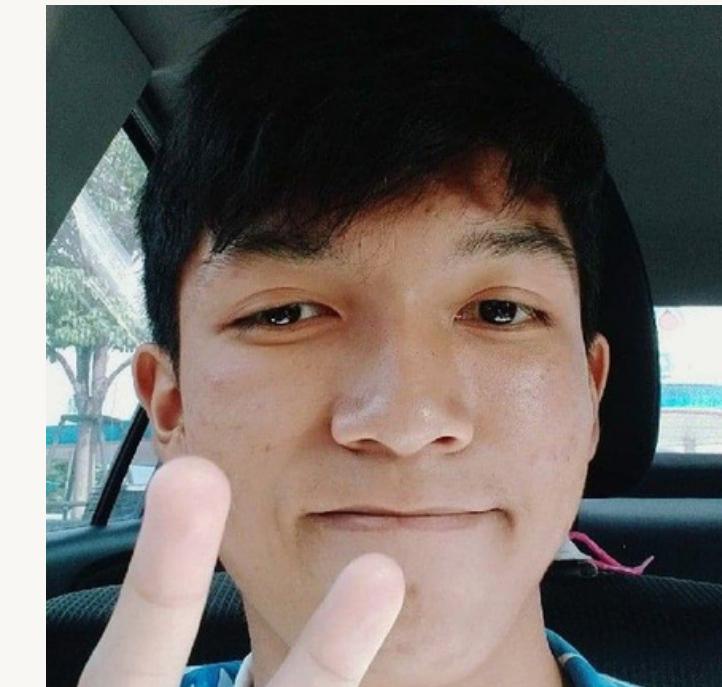


ສມາັືກ

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

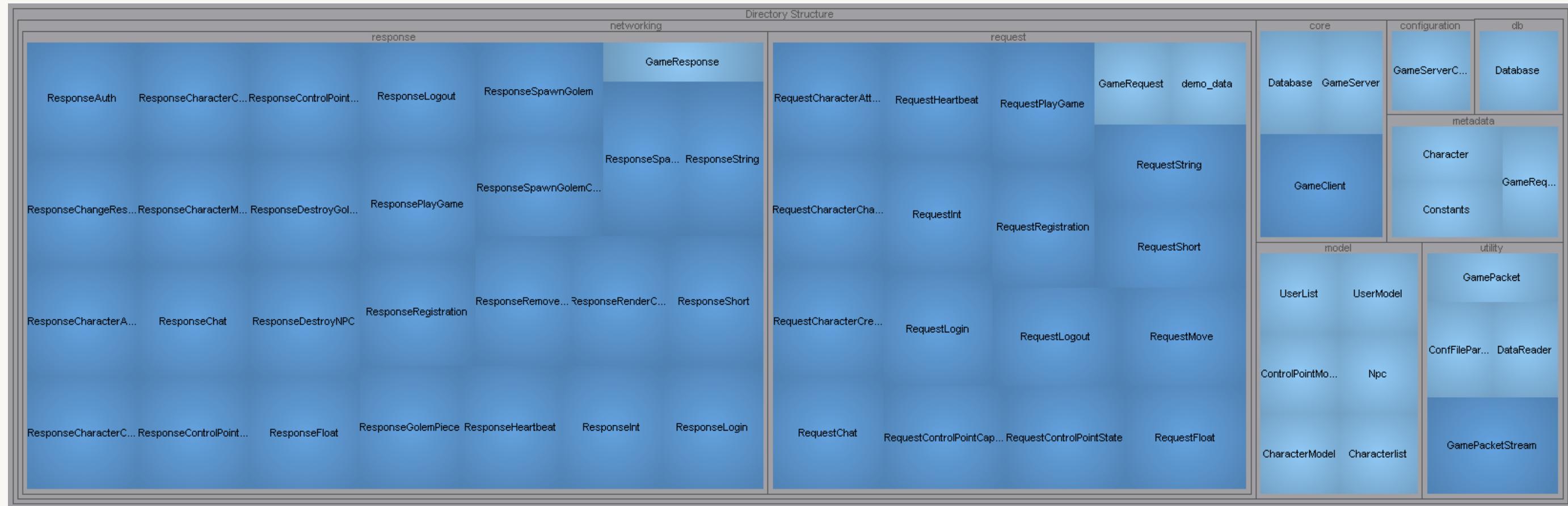
IT

Inheritance Tree

IT ຈະສາມາດວັດໄດ້ວ່າ [super-class](#) ສ່າງຜູດຕ່ວ class ອົບໆ ໄດ້ມາກນ້ອຍແກ່
ໄහນ ໂດຍຈະເກື່ອງຂ້ອງກັບ OOP

Inheritance Tree | Metric





UNDERSTAND

Static Testing

Inheritance Tree | Metric

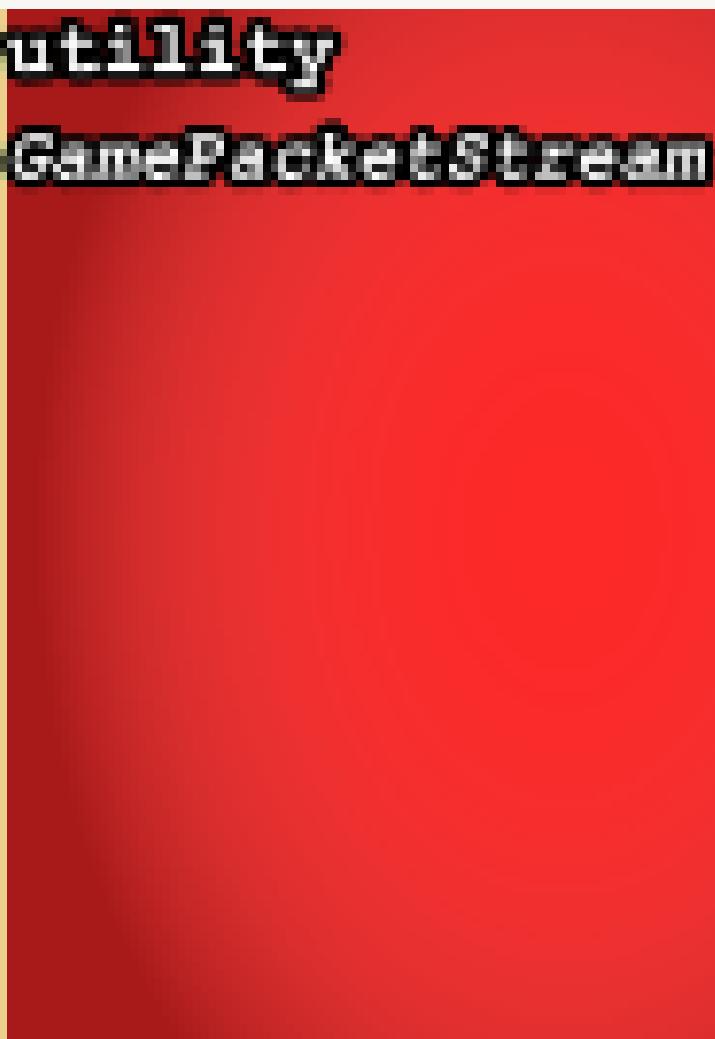
INHERITANCE TREE

ทดสอบในระดับ Class ซึ่งภาพรวมของโปรเจกจะมีค่า Max Inheritance เท่ากับ 2 unit

JARCHITECT

Static Testing

Inheritance Tree | Metric



INHERITANCE TREE

จากการพูดว่าในส่วนของ GamePacketStream นั้นมี 3 unit ซึ่งเป็นค่าสูงสุด

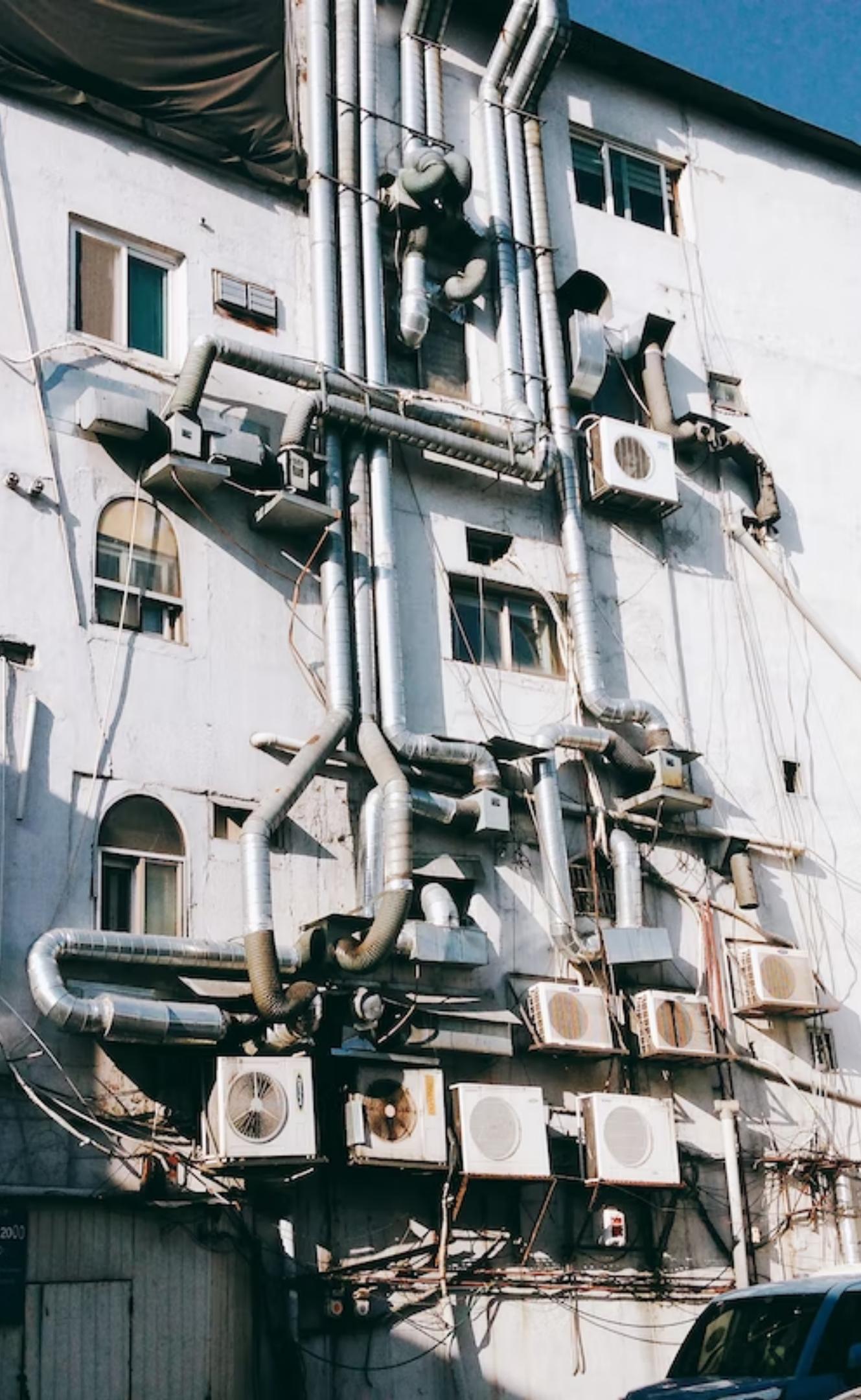
COMPLEXITY

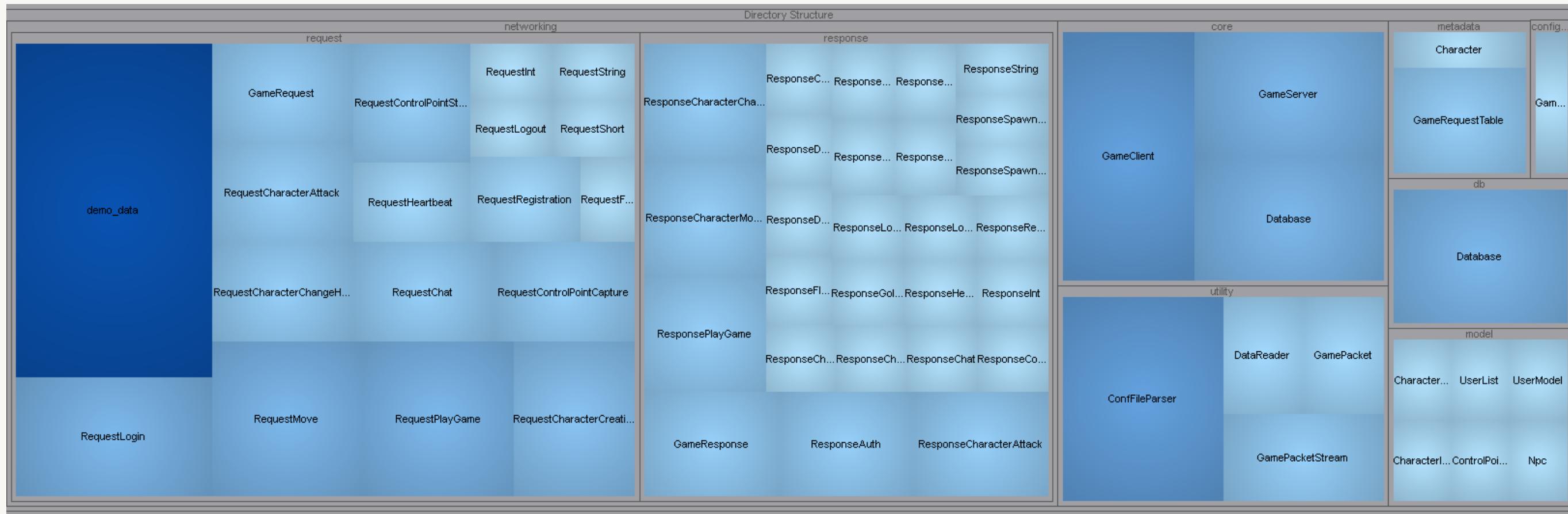
ความซับซ้อนของโค้ด

CC

ใช้สำหรับการวัดค่าความซับซ้อนของ class และ method โดยพื้นฐานจะบันบความซับซ้อนของ method **จากเส้นทางกั้งหมุด** กีเป็นไปได้ของการประมวลผล

Cyclomatic Complexity | Metric





UNDERSTAND

Static Testing

Cyclomatic Complexity | Metric

CYCLOMATIC COMPLEXITY

ทดลองในระดับ Class โดยมี

1. demo_data ที่อยู่ในส่วนของ request ได้ 14 Unit
 2. GameClient ที่อยู่ในส่วนของ core ได้ 7 Unit
 3. ConfFileParser ที่อยู่ในส่วนของ utilility ได้ 7 Unit



JARCHITECT

Static Testing

Cyclomatic Complexity | Metric

CYCLOMATIC COMPLEXITY

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

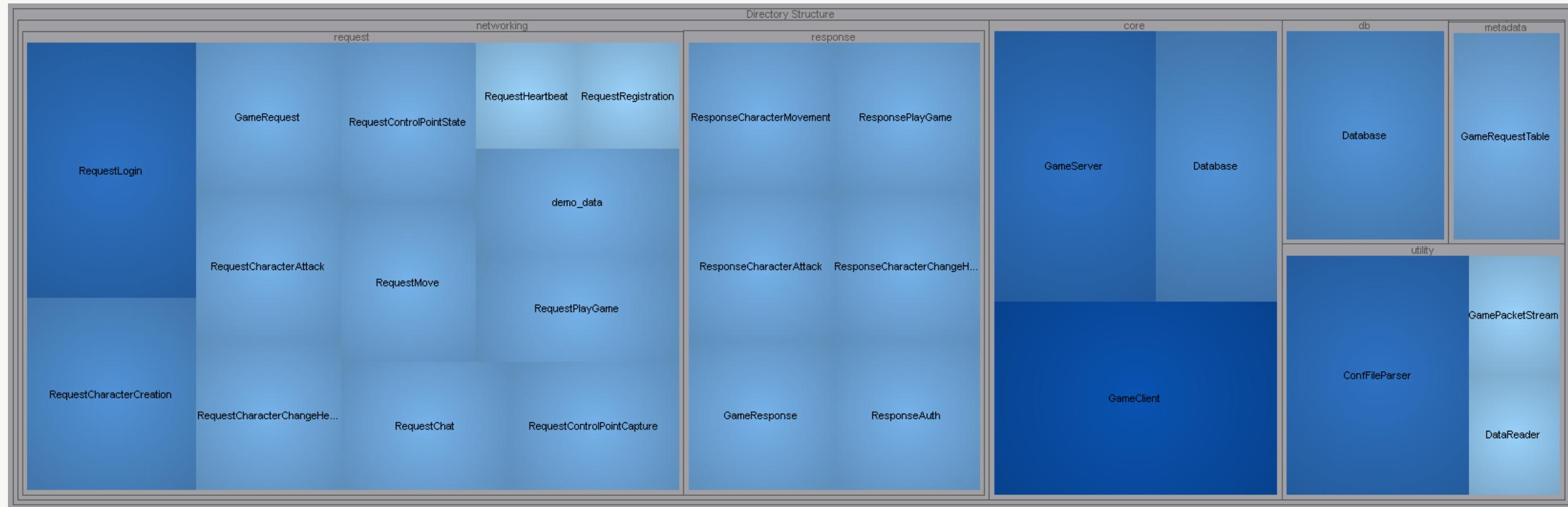
ND

Nesting Depth

จำนวนคำสั่งเงื่อนไข (branches, loops) ที่ถูกซ้อนกันอยู่สำหรับ
ควบคุมโครงสร้างของโปรแกรม

Nesting Depth | Metric





UNDERSTAND

Static Testing

Nesting Depth | Metric

NESTING DEPTH

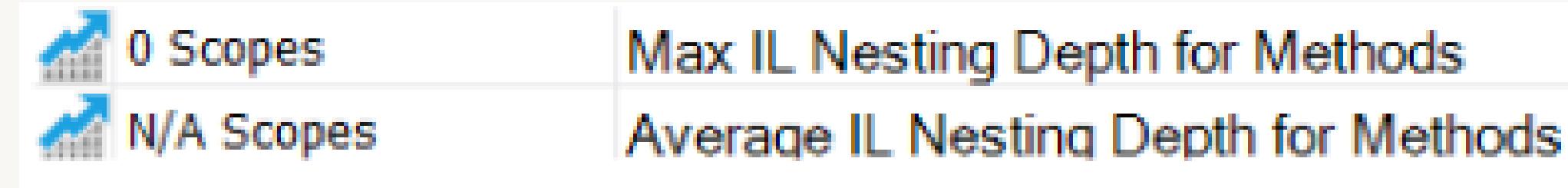
สามารถทดสอบ Nesting Depth ในระดับ Class

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

JARCHITECT

Static Testing

Nesting Depth | Metric



NESTING DEPTH

JArchitect นั้นเป็นกิชันในการตรวจสอบ Nesting Depth ได้แต่ในโปรแกรมที่ทดสอบนั้นได้สรุปผลออกมากดเป็นค่า max คือ 0 scope และ average คือ N/A

UNDERSTANDABILITY

การวัดความเข้าใจของโค้ด

CLOC

Line of Comment

จำนวนของการ Comment ในแต่ละส่วนของโค้ด เพื่ออธิบายการทำงานของแต่ละส่วนเพื่อให้ง่ายต่อการแก้ไขและพัฒนาต่อไปได้

Percentage of Comment | Metric





UNDERSTAND

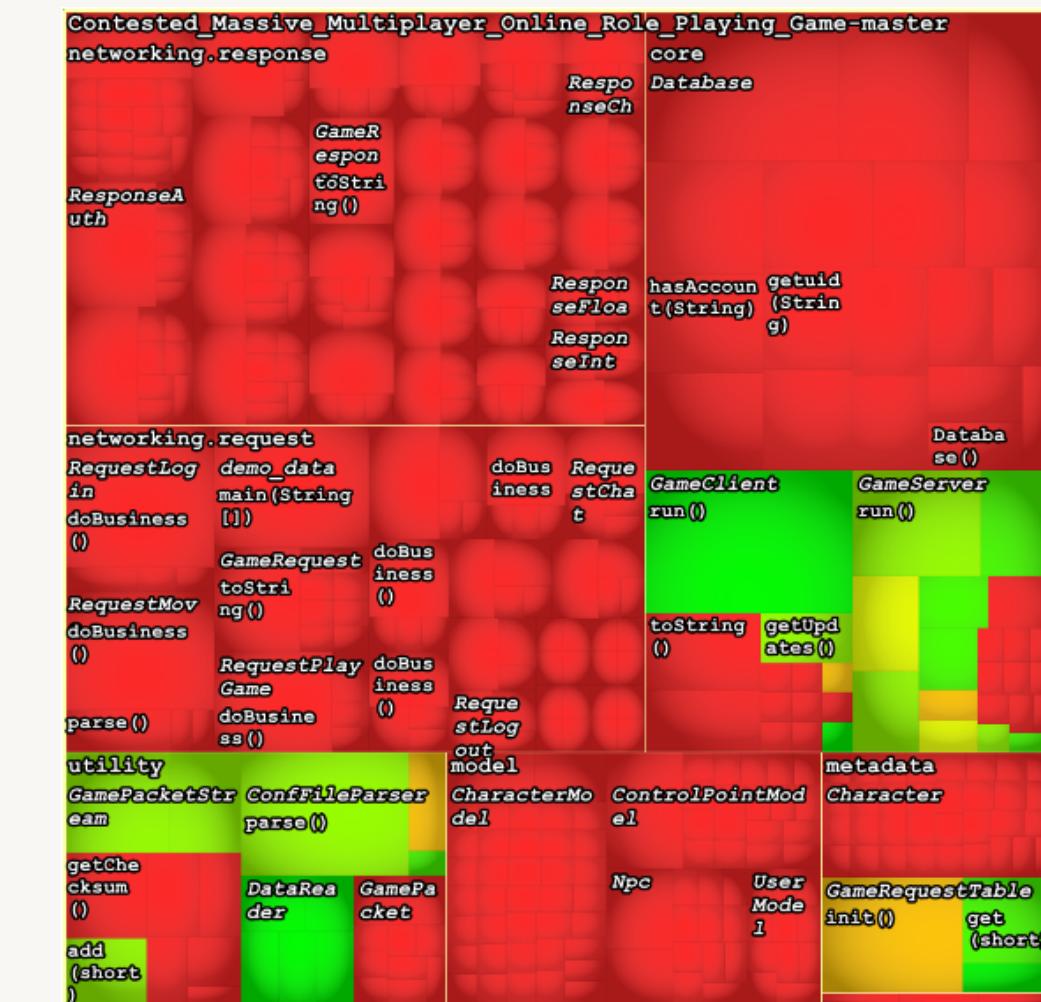
Static Testing

Percentage of Comment | Metric

LINE OF COMMENT

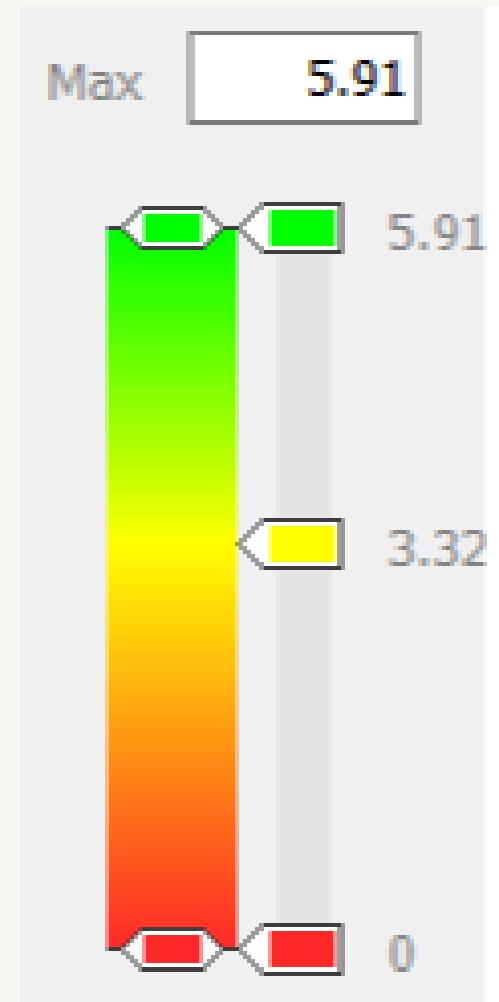
ทดสอบจำนวน Comment ในแต่ละ Class โดยใช้ Count Line Comment ซึ่งได้ค่ามากที่สุดโดยเลือกมา 3 อันดับแรก

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



Comment
4.68%

54 Lines of Comment



JARCHITECT

Static Testing

Percentage of Comment | Metric

LINE OF COMMENT

โดยจากการทดสอบของ JArchitect นั้นได้ค่าดังนี้

- ค่า minimum ของค่า POC จะอยู่ที่ 0% ของคลาส
- ค่า maximum มากที่สุดอยู่ที่ 70% ของคลาส
- ค่า percentage อยู่ที่ 4.68% ของไฟล์โปรเจค

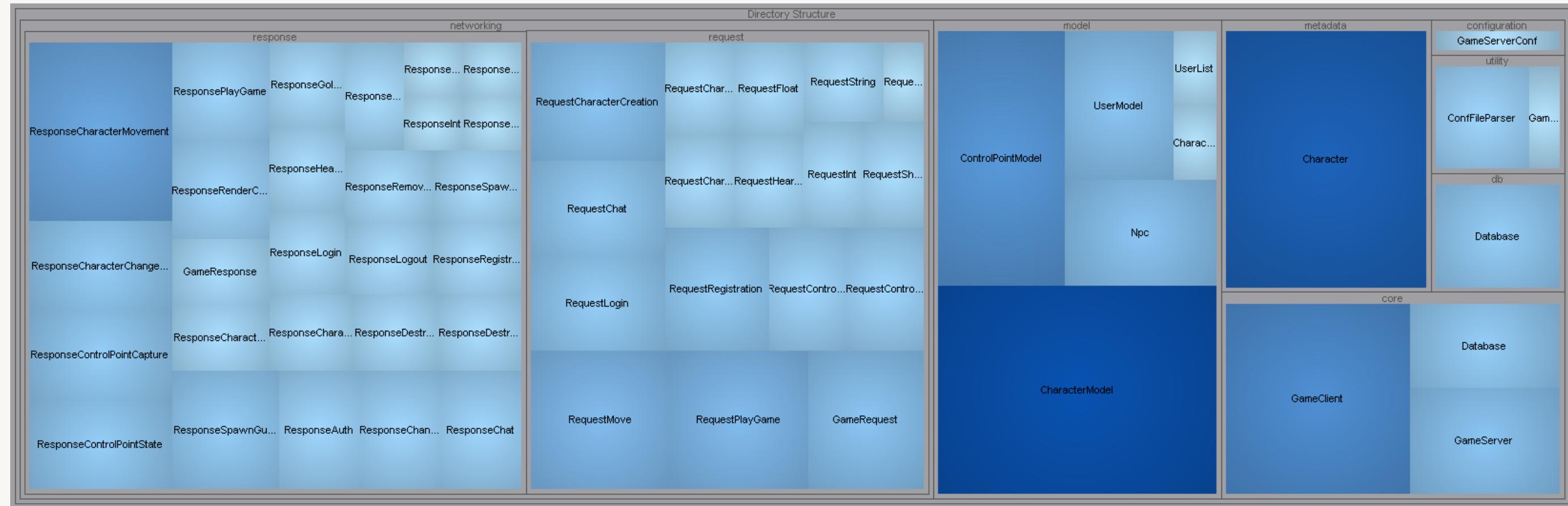
CIV

Count Instant Variable

การนับตัวแปรที่กำหนดในคลาสที่เข้าถึงได้ผ่านวัตถุของคลาสนั้น

Count Instant Variable | Metric





UNDERSTAND

Static Testing

Count Instant Variable | Metric

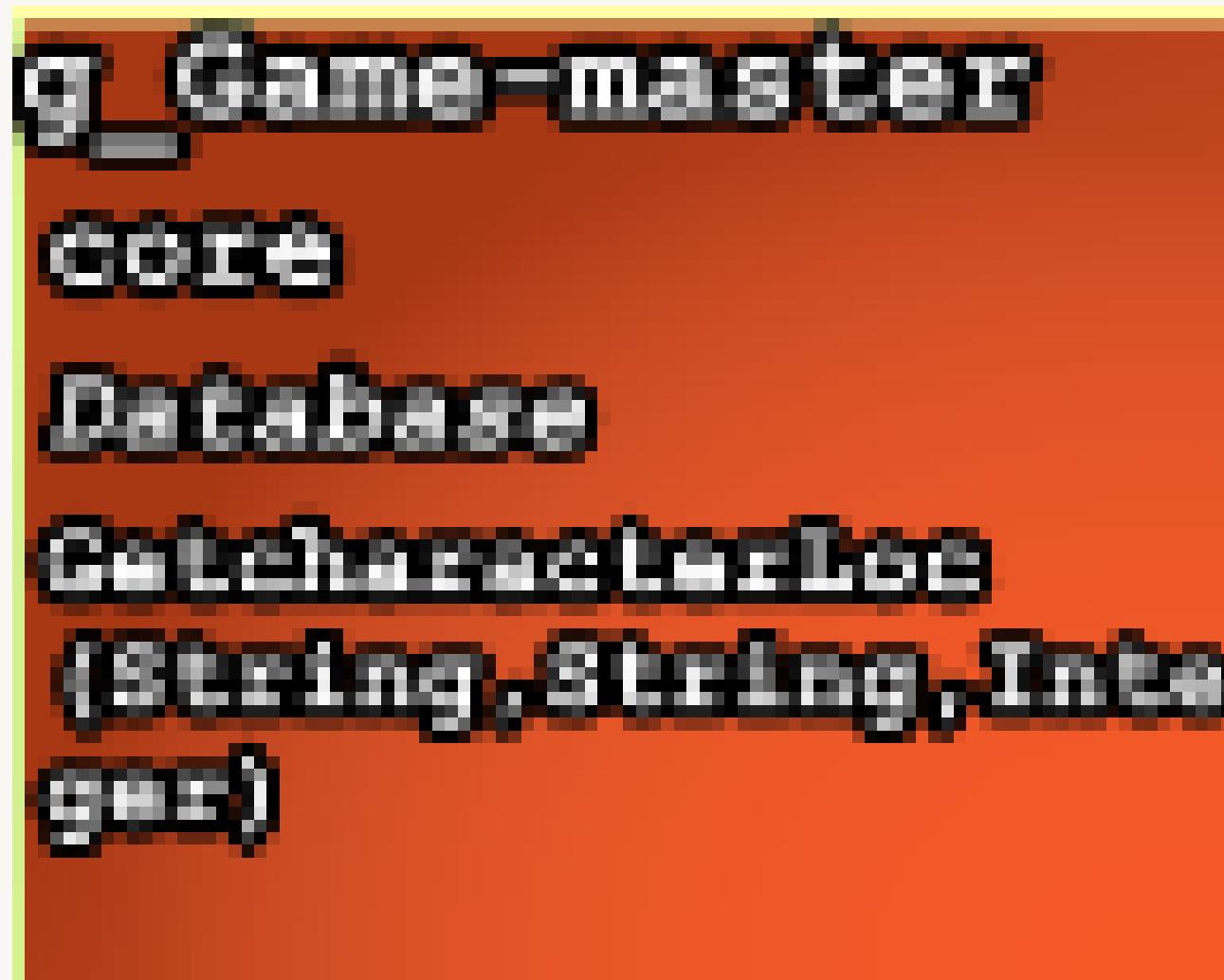
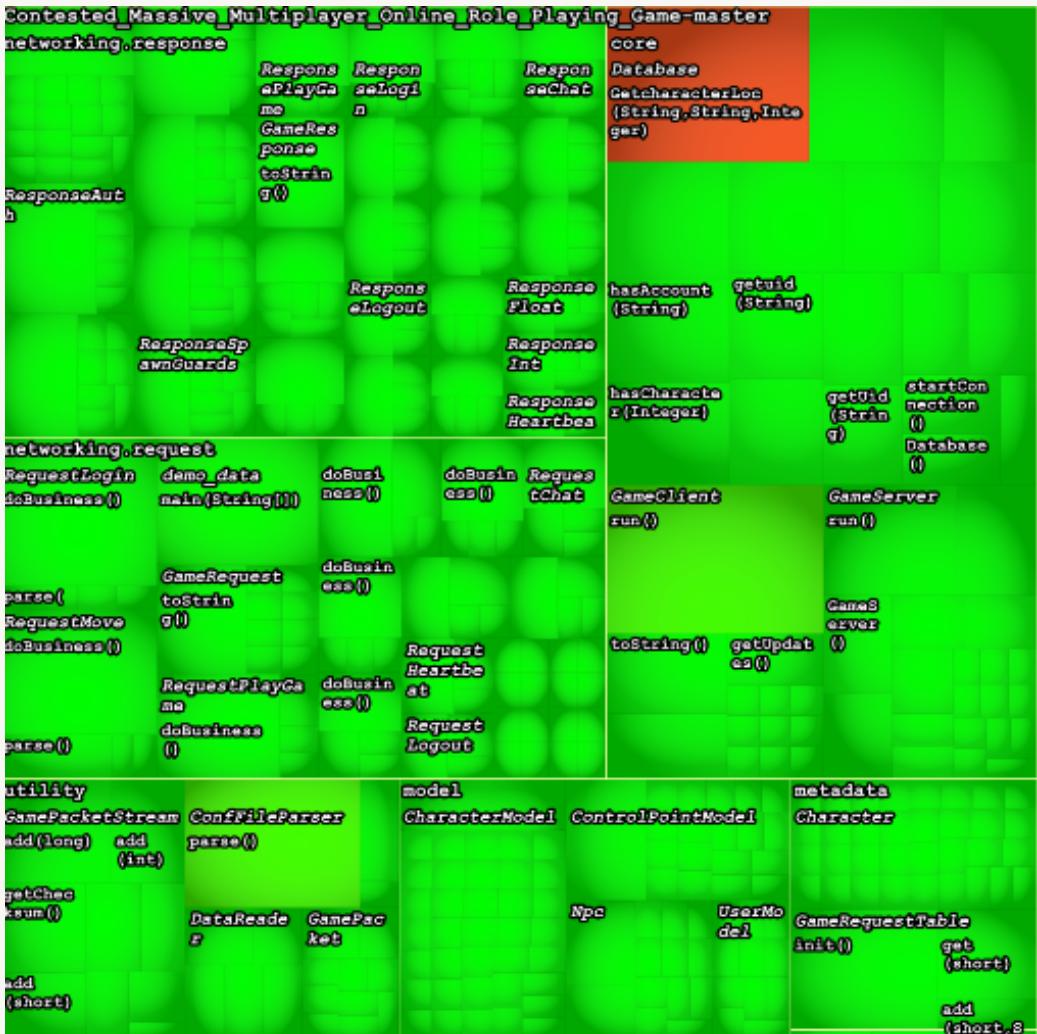
COUNT INSTANT VARIABLE

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

JARCHITECT

Static Testing

Count Instant Variable | Metric



COUNT INSTANT VARIABLE

จากภาพจะเห็นว่าผลลัพธ์ที่ได้บานั้นภายใน Database มีค่า 12 variable

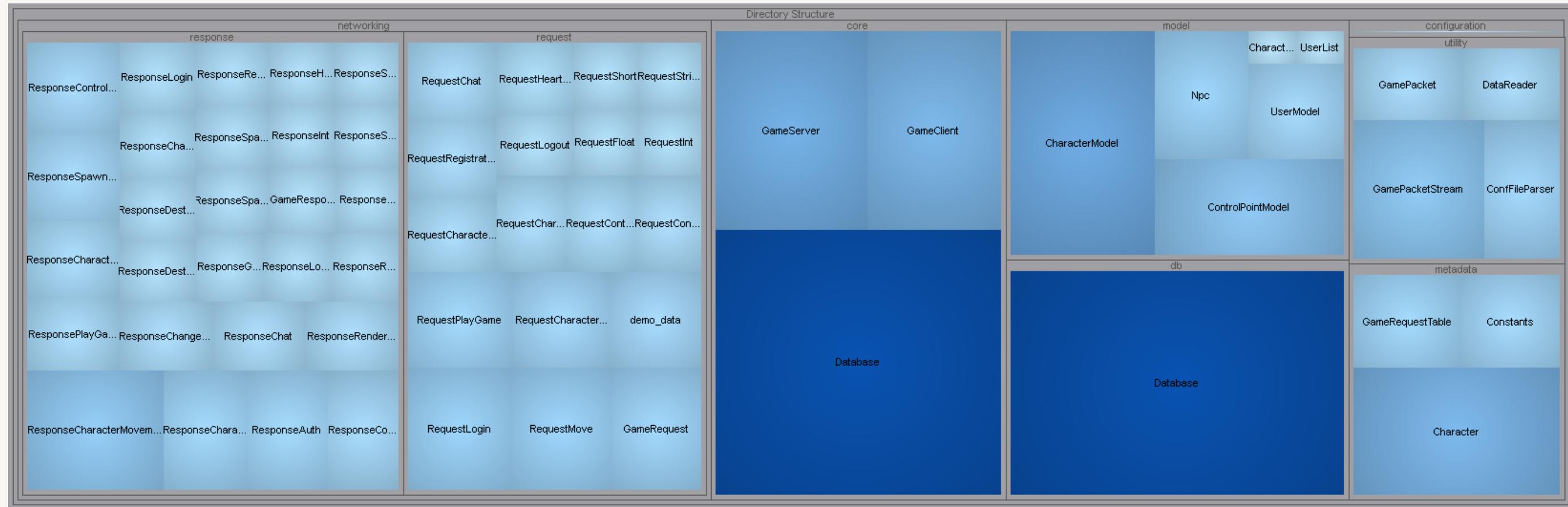
MAINTAINABILITY

การบำรุงรักษาของโค้ด

LOC

Line of Code

จำนวนบรรทัดในข้อความของซอฟต์แวร์สโค้ดของโปรแกรมและ LOC นั้นสามารถที่จะช่วยประเมินการณ์ได้ เช่น จำนวนชั่วโมงในการเขียน, จำนวนคนที่จำเป็น, รวมถึงประเมินราคากลางของซอฟต์แวร์

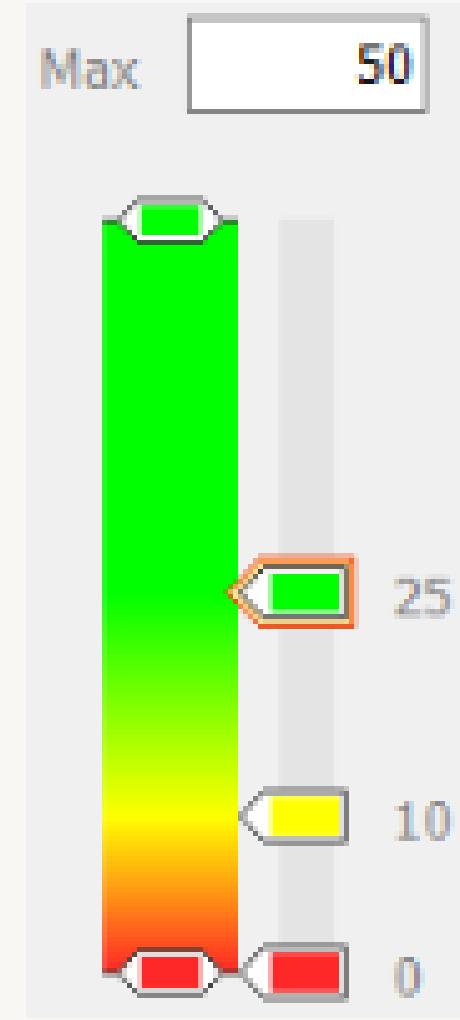
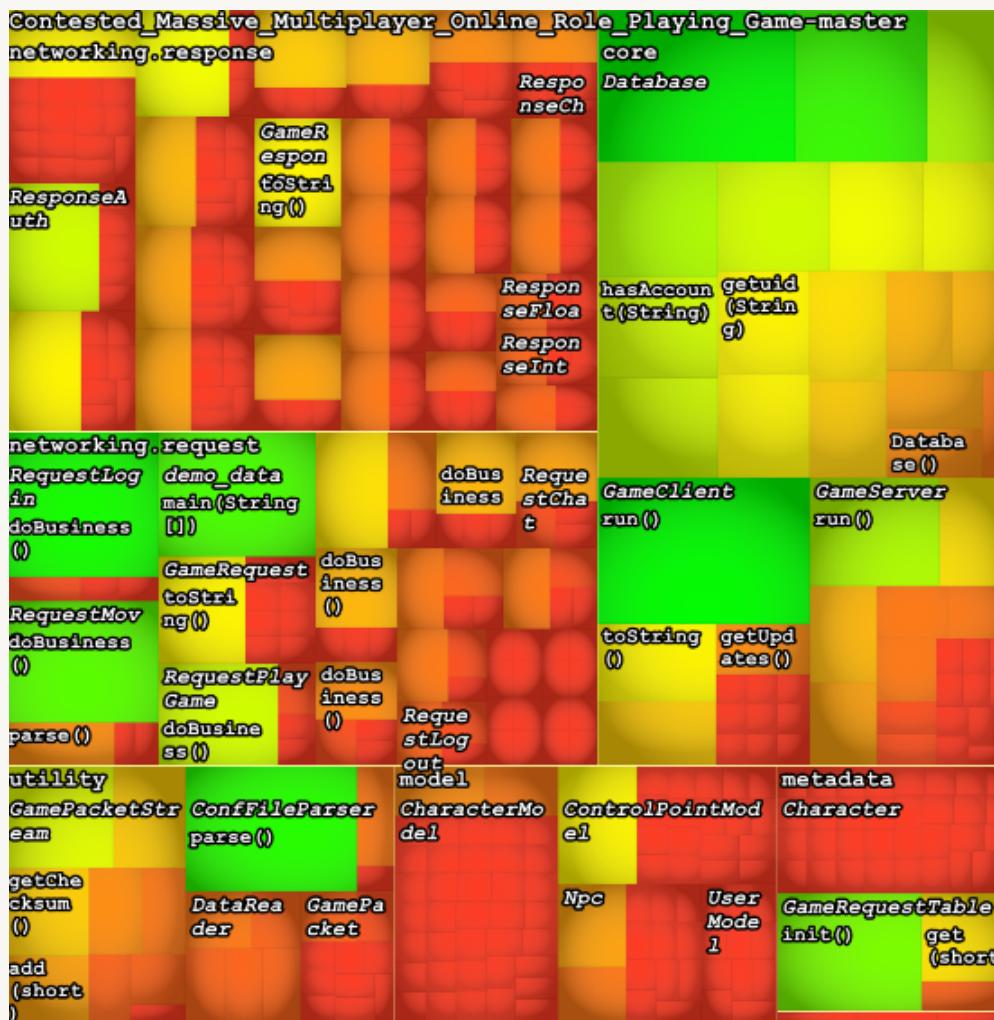


UNDERSTAND

Static Testing

LINE OF CODE

ทดสอบในระดับของ Class โดย Line of Code มากสุดซึ่ง database ในส่วนของ core เป็นทั้งหมด 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

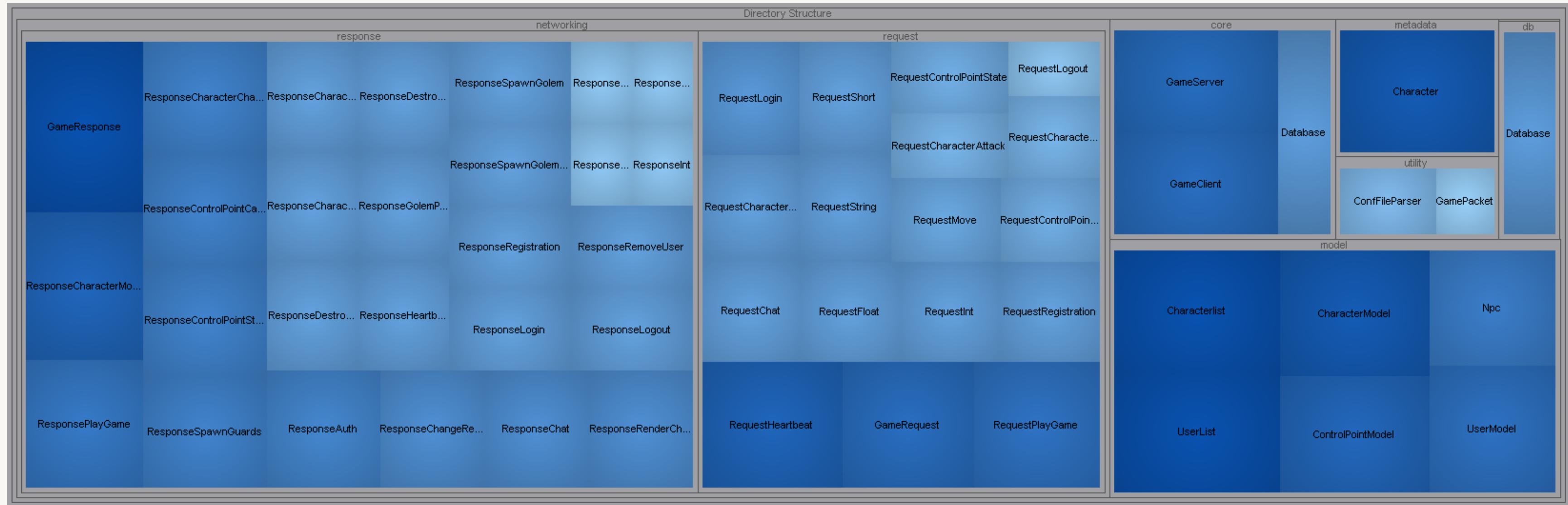
LCOM

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

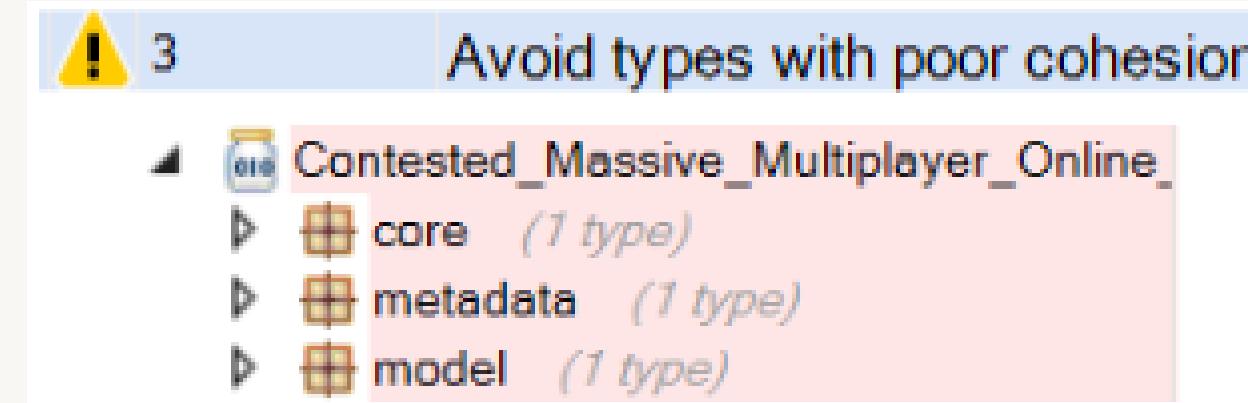
Static Testing

Lack of Cohesion | Metric

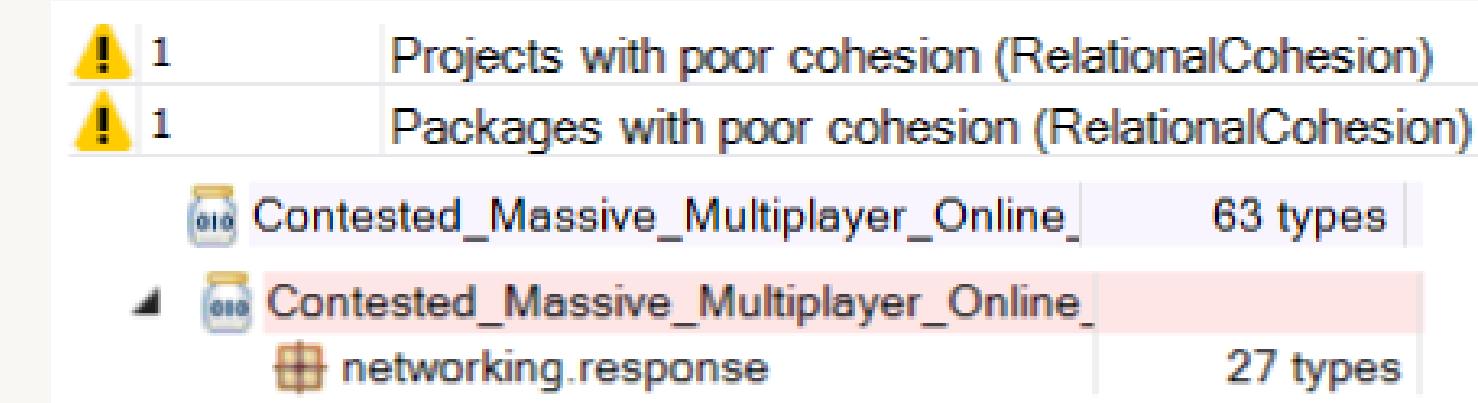
LACK OF COHESION (LCOM)

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 ไม่ได้ทำงานที่สอดคล้องกันค่อนข้างน้อย



type poor cohesion ที่เกิดขึ้นใน code smell



project, package poor cohesion ที่เกิดขึ้นใน architecture

JARCHITECT

Static Testing

Lack of Cohesion | Metric

LACK OF COHESION (LCOM)

ในโปรแกรมไม่ได้สรุปผลในรูปแบบตัวเลข แต่จะบ่งชี้ว่าโค้ดส่วนไหนที่ควรปรับแก้

ចុះការណាំសែនខេត្ត

Thank you for your attention

