

รายงาน

Battleship Project

โดย

นาย นครินทร์ สุขแพ
รหัสนักศึกษา 6204062610067

เสนอ

อาจารย์ สติต ประสมพันธ์

รายงานนี้เป็นส่วนหนึ่งของวิชา Object Oriented Programming

รหัสวิชา 040613222

บทที่ 1

บทนำ

ที่มาและความสำคัญ

โครงการนี้เกิดจากการที่ผู้จัดอยากสร้างเกมที่สามารถใช้ฝึกทักษะการคิด วิเคราะห์ที่น่าเสนอ ออกมาในรูปแบบของเกมจึงทำให้ได้สืบค้นและพบว่าหลาย ๆ เกมที่มีการจากการปิดการมองเห็นของผู้เล่นจากศัตรูจะทำให้ผู้เล่นได้ใช้ทักษะการวิเคราะห์และคาดเดาว่าอีกฝ่ายกำลังทำอะไรและผู้เล่นควรทำอะไรจึงเป็นเหตุให้ได้คิดและออกแบบรูปแบบของเกมทำให้ได้มาซึ่งกฎ กติกาการเล่น ที่ออกแบบมาด้วย concept ข้างต้นนั่นเอง

ประโยชน์

เพื่อฝึกทักษะการคิดวิเคราะห์ การวางแผน การศึกษาการเดินของคู่ต่อสู้ และเพื่อความบันเทิง

ขอบเขตของโครงการ

สามารถเล่นได้ทุกเพศทุกวัย แต่อายุที่เหมาะสมในการทำความเข้าใจเนื้อหาหลักของเกมจะแนะนำที่อายุ 15 ปีขึ้นไป

ตารางการทำงาน

| ลำดับ | รายการ | 10 - 15 | 16 - 25 | 26 - 30 |
|-------|--|---------|---------|---------|
| 1 | ออกแบบแผนที่และหารูปประกอบ | | | |
| 2 | ศึกษาข้อมูลที่เป็นและเอกสารที่เกี่ยวข้อง | | | |
| 3 | เริ่มการเขียนโปรแกรม | | | |
| 4 | จัดทำเอกสาร | | | |
| 5 | ทำคลิปวิดีโอและตรวจสอบข้อผิดพลาด | | | |

บทที่ 2

ส่วนการพัฒนา

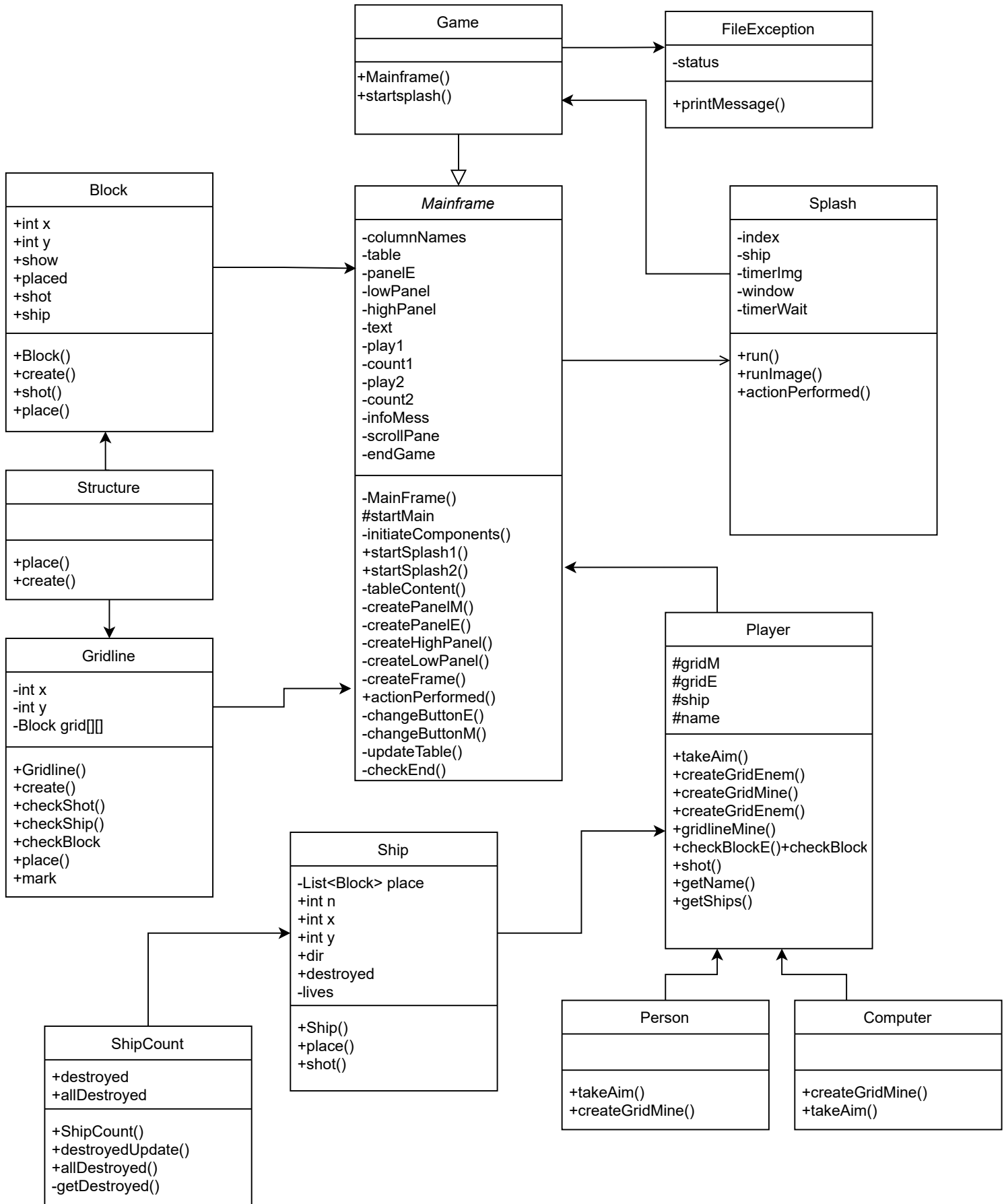
เนื้อเรื่องย่อ

คุณได้รับเลือกให้ค้นหาเรือดำน้ำที่ซ่อนอยู่ของศัตรูโดยที่เรือของคุณและศัตรูต่างใช้เรดาร์ตรวจจับไม่ได้ เพราะโดนคลื่นรบกวนคุณจะต้องหาเรือทั้ง 7 ลำแล้วทำลายทิ้งให้หมดก่อนที่จะศัตรูจะหาเรือที่ซ่อนอยู่และทำลายเรือทุกลำของคุณ

วิธีการเล่น

1. กดจุดบนแผนที่เพื่อเริ่มเกม
2. เรือแบ่งเป็นประเภท 5,4,3,2,1 ช่องประเภทละ 1,1,1,2,2 ลำ ตามลำดับ
3. แต่ละตาผู้เล่นและbotจะได้เลือกแค่1จุดบนแผนที่
4. หากจุดที่เรือซ่อนอยู่ของอีกฝั่งแล้วทำลายทิ้งถ้าเรือถูกทำลายจะเป็นสีเขียว
5. ถ้าเลือกจุดที่ไม่มีเรือซ่อนอยู่จุดบนแผนที่จะเป็นสีแดง
6. ฝั่งที่ทำลายเรือของฝั่งตรงข้ามได้ครบ 7 ลำก่อนจะเป็นผู้ชนะ

Class Diagram



Encapsulation:

```
1. public class Gridline implements Structure
2. {
3.     private int x;
4.     private int y;
5.     private Block grid[][];
6. }
```

จาก class gridline มีการกำหนดการเข้าถึงของตัวแปร

Inheritance:

```
1. public class Computer extends Player
2. {
3.
4.     public Computer(String name)
5.     {
6.         super(name);
7.     }
8. }
```

Class Computer มีการ extends class Player

Abstract

```
1. public abstract class Player
2. {
3.     public abstract boolean takeAim(Player pers, int x, int y);
4.     public abstract void createGridMine() throws BadFileException;
5. }
```

เป็นmethodที่ใช้เพื่อตรวจสอบค่า

Thread

```
1. class Splash extends Thread implements ActionListener
2. {
3.     static volatile boolean ready = true;
4.     private int index;
5.     private volatile ShipCount ship;
```

GUI

```
1. protected boolean startMain(String protag, String antag)
2. {
3.     play1 = new Person(protag);
4.     play2 = new Computer(antag);
5.     try
6.     {
7.         play1.createGridMine();
8.         play2.createGridMine();
9.     }
10.    catch(BadFileException e)
11.    {
12.        e.printStackTrace();
13.        return false;
14.    }
15.    initiateComponents();
16.    frame.setVisible(true);
17.    return true;
18. }
19. private void createFrame()
20. {
21.     frame.setSize(900,700);
22.     frame.setLayout(new GridLayout(2, 1));
23.     frame.add(highPanel);
24.     frame.add(lowPanel);
25.     frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
26.
27.     Dimension screenSize =
Toolkit.getDefaultToolkit().getScreenSize();
28.     Dimension labelSize = frame.getPreferredSize();
29.     frame.setLocation(screenSize.width/2 -
(labelSize.width/2),
30.                       screenSize.height/2 -
(labelSize.height/2));
31. }
```

Event Handling

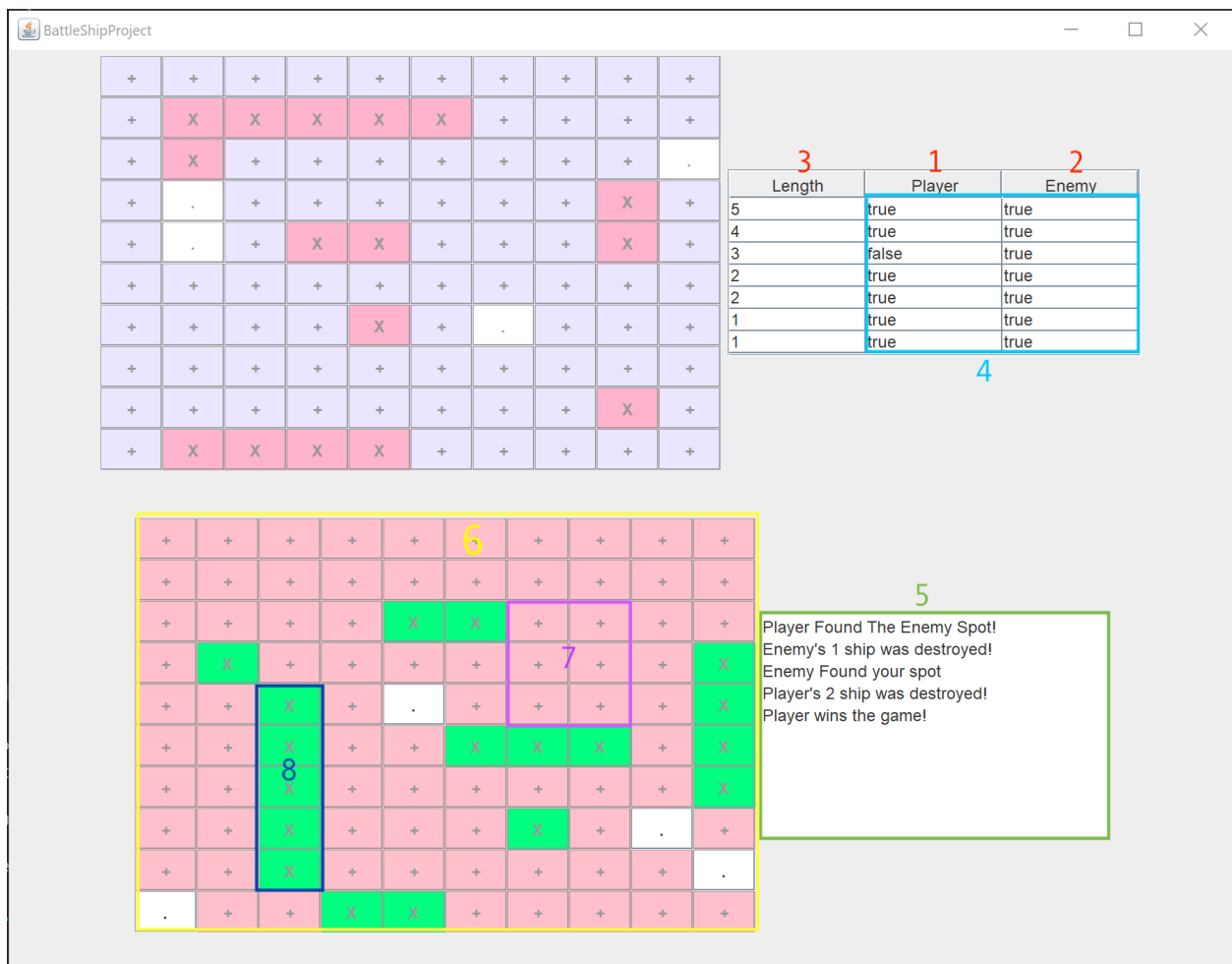
```
1. public void actionPerformed(ActionEvent e) {
2.     if(endGame == true)
3.         return;
4.     String command = (e.getActionCommand());
5. }
```

Algorithm

```
1. private void createPanelE()
2. {
3.     panelE.setLayout(new GridLayout(10, 10));
4.     panelE.setPreferredSize(new Dimension(450, 300));
5.     for(int i = 0; i < 10; i++)
6.     {
7.         for(int j = 0; j < 10; j++)
8.         {
9.             buttonE[j][i] = new JButton();
10.            buttonE[j][i].setText(".");
11.            buttonE[j][i].setBackground(Color.WHITE);
12.            panelE.add(buttonE[j][i]);
13.            buttonE[j][i].setActionCommand(String.valueOf(j)+String.valueOf(i));
14.            buttonE[j][i].addActionListener(this);
15.        }
16.    }
17. }
```

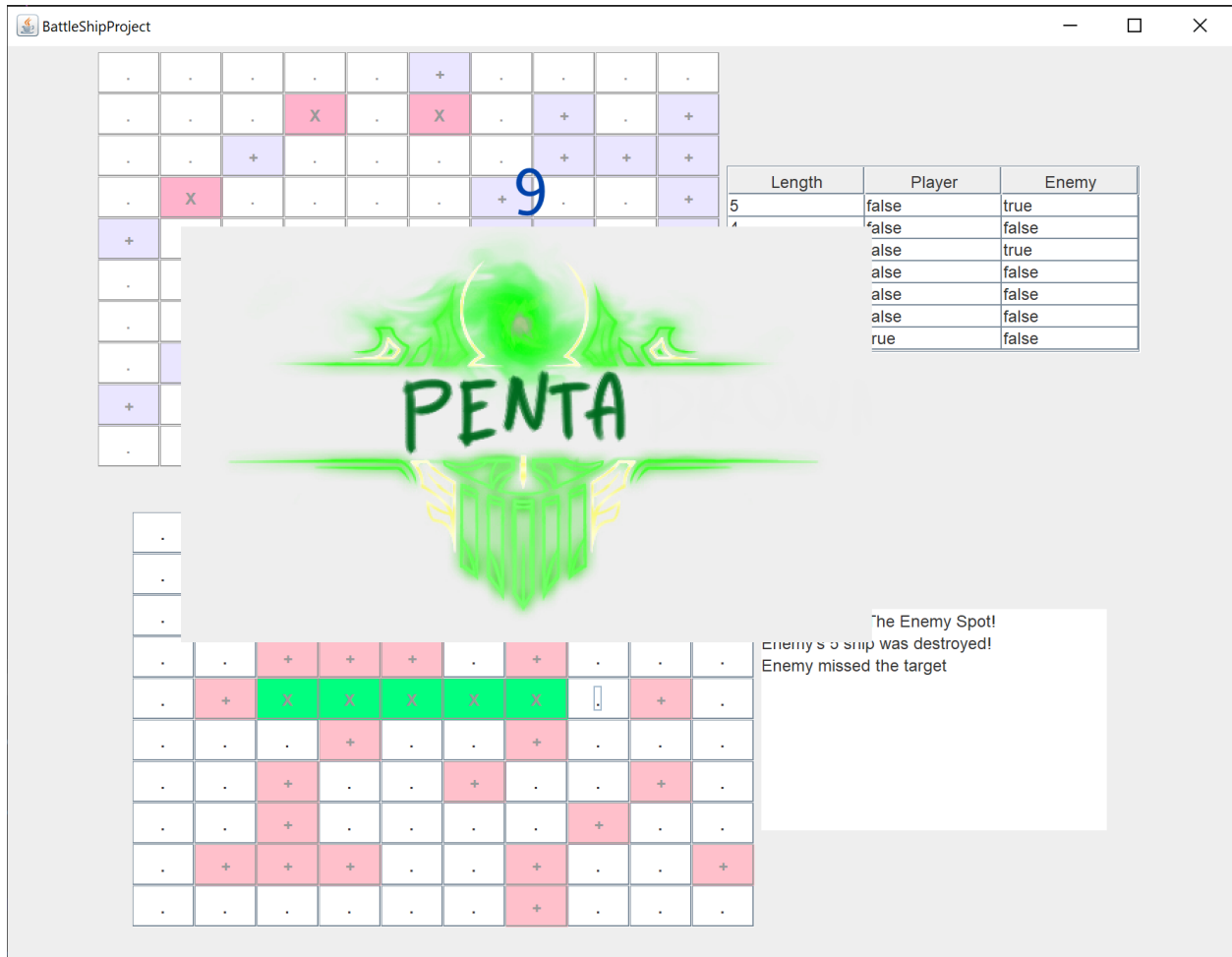
รับค่าและกำหนดตำแหน่งที่Playerเลือก

GUI



1. ชื่อผู้เล่น
2. ชื่อศัตรู
3. ขนาดของเรือ
4. ตารางแสดงบอกเรือของฝั่งใดถูกทำลายไปแล้วบ้าง
5. Action Log
6. แผนที่ของเกม
7. จุดที่เลือกผิด
8. จุดที่เลือกถูก

9. หน้าต่างแสดงการทำลายเรือ



บทที่ 3

สรุปผล

ปัญหาที่พบระหว่างการพัฒนา

1. รูปที่แสดงตอนทำลายเรือได้ขนาดผิดปกติ
2. มีผู้ชนะแล้วแต่ยังสามารถเล่นเกมต่อได้
3. ไม่สามารถทำให้รันด้วย.JAR Fileได้

จุดเด่นที่ไม่เหมือนใคร

- มีการตรวจเช็คไฟล์ถ้าไม่สมบูรณ์จะขึ้นแจ้งเตือนผู้ใช้
- มีการขึ้นภาพeffectการskillเพื่อเพิ่มอรรถรสให้ผู้เล่น

คำแนะนำ

อยากให้มีการมาเฉลยและอธิบายการบ้าน Assignment ของแต่ละสัปดาห์ด้วยครับ