



Mini Project Game

Object Oriented Programming Subject

จัดทำโดย

6404062610464 นางสาวชุติกานจน์ ชีคำม

อาจารย์ผู้สอน

ผู้ช่วยศาสตราจารย์สถิต ประสมพันธ์

วิชา Object Oriented Programming

ภาคเรียนที่ 1/2565

บทนำ

เกี่ยวกับโครงการ

ชื่อเกม : Hunter's x Monster

ผู้จัดทำ : นางสาวชุตिकाญจน์ ชีคำม

อาจารย์ผู้สอน : ผู้ช่วยศาสตราจารย์สถิต ประสมพันธ์

บทที่ 1 : ที่มาและความสำคัญ

โครงการนี้จัดทำขึ้นเพื่อชี้วัดความรู้ในวิชา Object Oriented Programming เพื่อนำความรู้ที่ได้เรียนมาเป็นระยะเวลาหนึ่งเทอมการศึกษานำมาใช้ประโยชน์จึงได้สร้างผลงานเป็นชิ้นงาน โดยชิ้นงานนี้เป็นการนำความรู้ทั้งในและนอกห้องเรียน การหาความรู้ค้นคว้าด้วยตัวเองจนผลงานเสร็จสมบูรณ์

ประโยชน์

- 1.เพื่อความบันเทิง
- 2.นำความรู้ที่ได้เรียนรู้และฝึกฝนการค้นคว้าด้วยตนเองนำไปใช้งานจริง
- 3.เพื่อนำโครงการไปต่อยอดในอนาคต

ขอบเขตของโครงการ (Proposal)

Story:

เรื่องราวของนักผจญภัยหนุ่มที่มีภารกิจจากชายชราให้ไปตามหาสมบัติที่ล้ำค่าจากถ้ำศักดิ์สิทธิ์ที่ขึ้นชื่อว่าเป็นที่อยู่ของเหล่าสัตว์ประหลาดมากมาย แต่ด้วยความมั่นใจและความหยิ่งผยองของชายหนุ่ม เรื่องเล่าตำนานของสัตว์ประหลาดไม่ได้ทำให้ความเกรงกลัวของเขาลดน้อยลงสักนิด เขาจะพิชิตถ้ำที่มืดแสนอันตรายได้หรือไม่ !!

How to play:

w a s d จากคีย์บอร์ดเพื่อควบคุมตัวละคร บน ซ้าย ล่าง ขวา ตามลำดับ

e เพื่อทำกริยากับobject เช่น โจมตี หรือ เก็บ สิ่งของ

i เพื่อสำรวจช่องเก็บของ

story board:



นักผจญภัย สัตว์ประหลาดตัวเล็ก สัตว์ประหลาดตัวใหญ่



ประโยชน์ที่ได้จากเกม

-ฝึกการจัดสรรทรัพยากรที่มีให้เกิดประโยชน์สูงสุด

-ฝึกความอดทน

-ฝึกการคิดในการแก้ปัญหา

การแบ่งเวลาในการทำงาน (เดือน ตุลาคม)

-วาดรูป ออกแบบฉาก วางแผนระบบ 1-7 ต.ค.

-coding 8 - 15

-ทดสอบ ปรับปรุงระบบ 16 - 18

-ทำเอกสาร 19-20

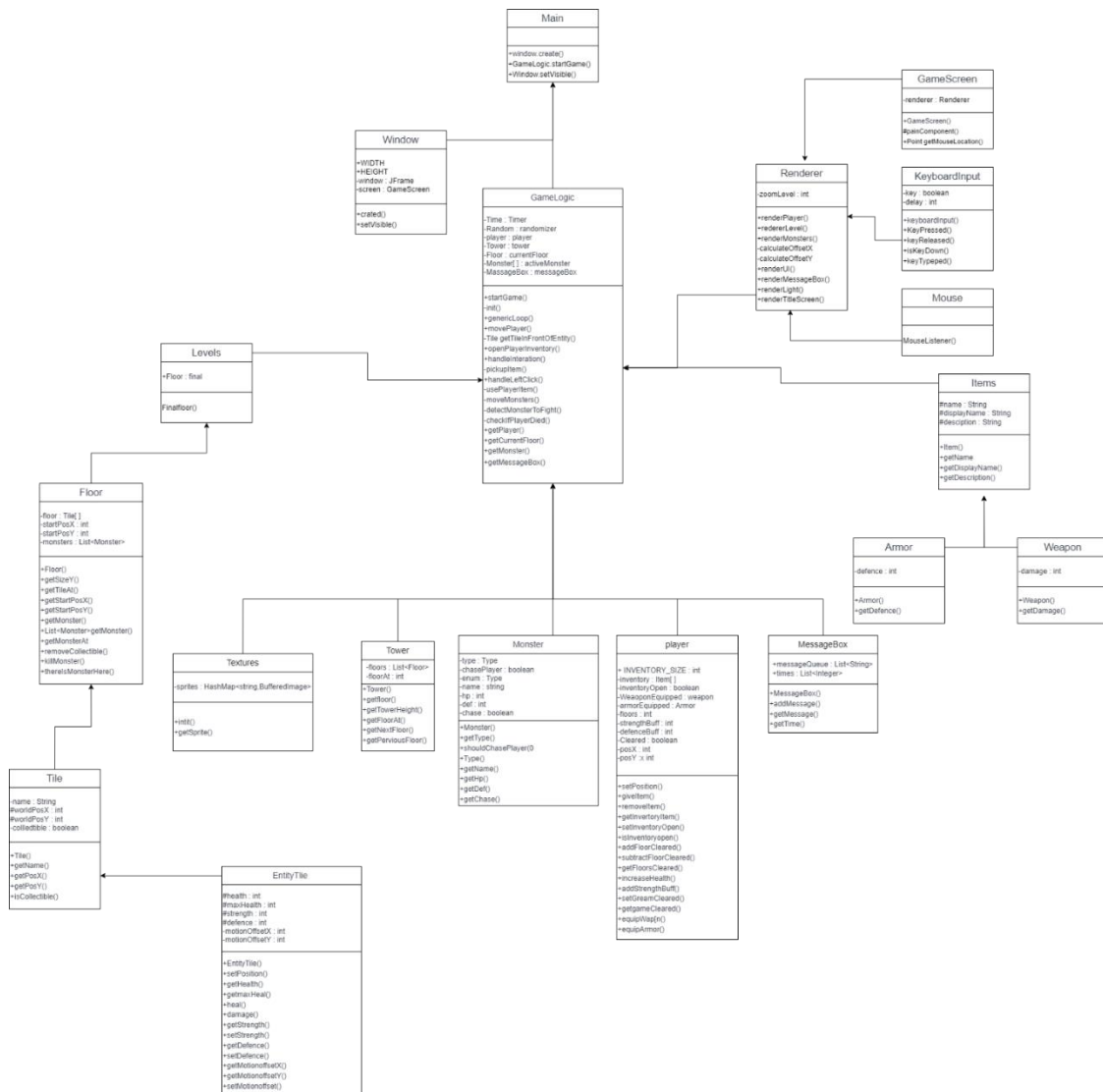
บทที่ 2

ส่วนของการพัฒนา

Story:

เรื่องราวของนักผจญภัยหนุ่มที่มีการฝึกจากชายชราให้ไปตามหาสมบัติที่ล้ำค่าจากถ้ำศักดิ์สิทธิ์ที่ขึ้นชื่อว่าเป็นที่อยู่ของเหล่าสัตว์ประหลาดมากมาย แต่ด้วยความมั่นใจและความหยิ่งผยองของชายหนุ่ม เรื่องเล่าตำนานของสัตว์ประหลาดไม่ได้ทำให้ความเกรงกลัวของเขาลดน้อยลงสักนิด เขาจะพิชิตถ้ำที่สุดแสนอันตรายได้หรือไม่ !!

class diagram



อธิบายส่วนของโปรแกรม

constructor

```
19 public class GameLogic {  
20  
21     private static Timer timer;  
22  
23     private static Random randomizer;  
24  
25     private static Player player;  
26     private static Tower tower;  
27     private static Floor currentFloor;  
28     private static Monster[] activeMonsters;  
29     private static MessageBox messageBox;  
30  
31     private static boolean onTitleScreen;
```

ประกาศ constructor ใน class GameLogic

```
6 public class EntityTile extends Tile {  
7  
8     protected int health;  
9     protected int maxHealth;  
10    protected int strength;  
11    protected int defence;  
12  
13    private int motionOffsetX;  
14    private int motionOffsetY;  
15
```

ประกาศ constructor ใน class EntityTile

```
4 public class Monster extends EntityTile {  
5  
6     private Type type;  
7     private boolean chasePlayer;  
8  
9     public Monster(Type type, int posX, int posY) {  
10         super(type.getName(), posX, posY, type.getHp());  
11         this.strength = type.getStr();  
12         this.defence = type.getDef();  
13         this.type = type;  
14         this.chasePlayer = type.getChase();  
15     }
```

ประกาศ constructor ใน class Monster

```

8 public class Player extends EntityTile {
9
10     public static final int INVENTORY_SIZE = 3;
11
12     private Item[] inventory;
13     private boolean inventoryOpen;
14
15     private Weapon weaponEquipped;
16     private Armor armorEquipped;
17
18     private int floors;
19
20     private int strengthBuff;
21     private int defenceBuff;
22     private boolean Cleared;
23     public static int posX = 0;
24     public static int posY = 0;
25

```

ประกาศ constructor ใน class Player

```

10 public class Floor {
11
12     private Tile[][] floor;
13
14     private int startPosX;
15     private int startPosY;
16
17     private List<Monster> monsters;
18

```

ประกาศ constructor ใน class Floor

```

3 public class Tile {
4
5     private String name;
6
7     protected int worldPosX;
8     protected int worldPosY;
9
10    private boolean collectible;
11

```

ประกาศ constructor ใน class Tile

```

8 public class Items {
9
10     public static final Item HP_POTION = new Item("hp_potion", "Health potion", "Restores 10 HP");
11     public static final Item DEF_POTION = new Item("def_potion", "Mysterious potion", "Its effect is unknown");
12
13     public static final Weapon SECRET_SWORD = new Weapon("secret-sword", "secret-sword", 1000);
14     public static final Weapon RUSTY_SWORD = new Weapon("rusty_sword", "rusty Sword", 5);
15     public static final Weapon AXE = new Weapon("axe", "Axe", 7);
16     public static final Weapon GREAT_SWORD = new Weapon("great_sword", "Great Sword", 9);
17
18     public static final Armor LEATHER_ARMOR = new Armor("leather_armor", "leather Armor", 4);
19     public static final Armor MEDIEVAL_ARMOR = new Armor("medieval_armor", "Medieval Armor", 7);
20     public static final Armor MISTERIOUS_ARMOR = new Armor("mysterious_armor", "Mysterious Armor", 9);
21 }

```

ประกาศ constructor ใน class Item

Encapsulation

class Gamellogic

```
134 private static Tile getTileInFrontOfEntity(EntityTile entity, int dirX, int dirY) {
135     return currentFloor.getTileAt(entity.getPosX()+dirX, entity.getPosY()+dirY);
136 }
137

154 private static void pickupItem(int itemPosX, int itemPosY) {
155     switch(currentFloor.getTileAt(itemPosX, itemPosY).getName()) {
156         case "secret-sword":
157             if(player.giveItem(Items.SECRET_SWORD)) {
158                 player.equipWeapon(Items.SECRET_SWORD);
159                 currentFloor.removeCollectible(itemPosX, itemPosY);
160                 messageBox.addMessage("You picked up a SECRET SWORD!", 20);
161             }
162             else {
163                 messageBox.addMessage("Your inventory is full!", 20);
164             }
165             break;

238 private static void usePlayerItem(int index) {
239     Item item = player.getInventoryItem(index);
240
241     if(item == null) return;
242
243     if(item == Items.HP_POTION) {
244         player.heal(10);
245         messageBox.addMessage("You drank a red potion and you recovered health!", 30);
246     }
247     else if(item == Items.DEF_POTION) {
248         player.addDefenceBuff();
249         messageBox.addMessage("You drank a defence potion! Defence temporary increased!", 30);
250     }
251     player.removeItem(index);
252 }

254 private static void moveMonsters() {
255     for(Monster monster : activeMonsters) {
256         if(monster.getHealth() <= 0)
257             continue;
258
259         if(!monster.shouldChasePlayer()) {
260             switch(randomizer.nextInt(4)) {
261                 case 0:
262                     if(currentFloor.thereIsMonsterHere(monster.getPosX()+1, monster.getPosY())) {
263                         return;
264                     }
265                     else if(monster.getPosX()+1 == player.getPosX() && monster.getPosY() == player.getPosY()) {
266                         messageBox.addMessage("Monster deal "+monster.getStrength()+" to you. You have "+player.getHealth()+" left.", 30);
267                         player.damage(monster.getStrength() - player.getDefence()/3);
268                         break;
269                     }
270                     if(getTileInFrontOfEntity(monster, 1, 0).getName() == "floor2-1" ||
271                        getTileInFrontOfEntity(monster, 1, 0).getName() == "floor3-1" ||
272                        getTileInFrontOfEntity(monster, 1, 0).getName() == "floor3-2" ||
273                        getTileInFrontOfEntity(monster, 1, 0).getName() == "floor4") {
274                         monster.setPosition(monster.getPosX()+1, monster.getPosY(), true);
275                         break;

```



```

387 private static boolean detectMonsterToFight(int dirX, int dirY) {
388     if(currentFloor.getMonsterAt(player.getPosX()+dirX, player.getPosY()+dirY) != null) {
389
390         Monster fight = currentFloor.getMonsterAt(player.getPosX()+dirX, player.getPosY()+dirY);
391         fight.damage(player.getStrength() - fight.getDefence()/3);
392
393
394         if(fight.getHealth() <= 0) { //Monster is dead
395             if(currentFloor.getMonsterAt(player.getPosX()+dirX, player.getPosY()+dirY).getName() == "GUNTHER") {
396                 messageBox.addMessage("You've killed GUNTHER!!!", 30);
397             }
398
399             if(currentFloor.getMonsterAt(player.getPosX()+dirX, player.getPosY()+dirY).getName() == "PEPPERMINT_BUTLER") {
400                 messageBox.addMessage("You've killed PEPPERMINT BUTLER!!!", 30);
401             }
402
403             if(currentFloor.getMonsterAt(player.getPosX()+dirX, player.getPosY()+dirY).getName() == "ICEKING") {
404                 messageBox.addMessage("You've killed ICEKING!!!", 30);
405             }
406             currentFloor.killMonster(player, fight.getPosX(), fight.getPosY());
407         }
408         else if(fight.getHealth() > 0){ //Monster is still alive after attack
409             if(currentFloor.getMonsterAt(player.getPosX()+dirX, player.getPosY()+dirY).getName() == "GUNTHER") {
410                 messageBox.addMessage("You've deal " + (player.getStrength() - fight.getDefence()/3) + "/" +
411                                     fight.getMaxHealth() + " damage to GUNTHER!!!", 30);
412             }
413
414             if(currentFloor.getMonsterAt(player.getPosX()+dirX, player.getPosY()+dirY).getName() == "PEPPERMINT_BUTLER") {
415                 messageBox.addMessage("You've deal " + (player.getStrength() - fight.getDefence()/3) + "/" +
416                                     fight.getMaxHealth() + " damage to PEPPERMINT BUTLER!!!", 30);
417             }
418         }
419     }
420 }

```

```

447 private static boolean checkIfPlayerDied() {
448     if(player.getHealth() <= 0) {
449         return true;
450     }
451     return false;
452 }

```

Inheritance

```

13 public class GameScreen extends JPanel {
14
15     private Renderer renderer;
16
17     /**Creates a new GameScreen object<br>
18      * Extension of JPanel*/
19     public GameScreen() {
20         super();
21         this.setFocusable(true);
22         this.addKeyListener(new KeyboardInput());
23         this.addMouseListener(new Mouse());
24
25         this.renderer = new Renderer();
26
27     }

```

```

6 public class EntityTile extends Tile {
7
8     protected int health;
9     protected int maxHealth;
10    protected int strength;
11    protected int defence;
12
13    private int motionOffsetX;
14    private int motionOffsetY;
15
16    public EntityTile(String name, int posX, int posY, int health) {
17        super(name, posX, posY);
18        this.maxHealth = health;
19        this.health = health;
20    }

```

```

4 public class Monster extends EntityTile {
5
6     private Type type;
7     private boolean chasePlayer;
8
9     public Monster(Type type, int posX, int posY) {
10         super(type.getName(), posX, posY, type.getHp());
11         this.strength = type.getStr();
12         this.defence = type.getDef();
13         this.type = type;
14         this.chasePlayer = type.getChase();
15     }

8 public class Player extends EntityTile {
9
10     public static final int INVENTORY_SIZE = 3;
11
12     private Item[] inventory;
13     private boolean inventoryOpen;
14
15     private Weapon weaponEquipped;
16     private Armor armorEquipped;
17
18     private int floors;
19
20     private int strengthBuff;
21     private int defenceBuff;
22     private boolean Cleared;
23     public static int posX = 0;
24     public static int posY = 0;

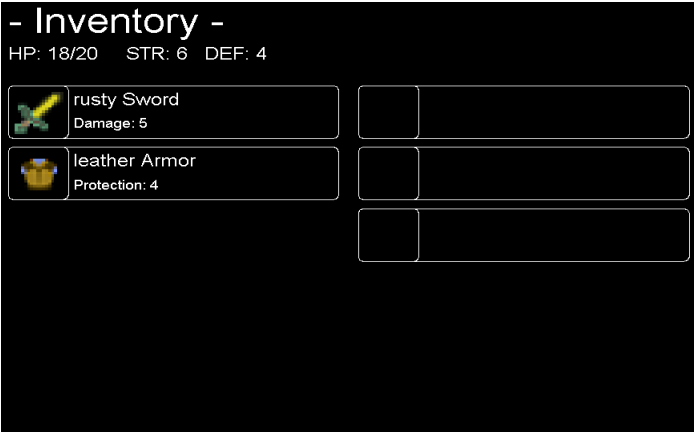
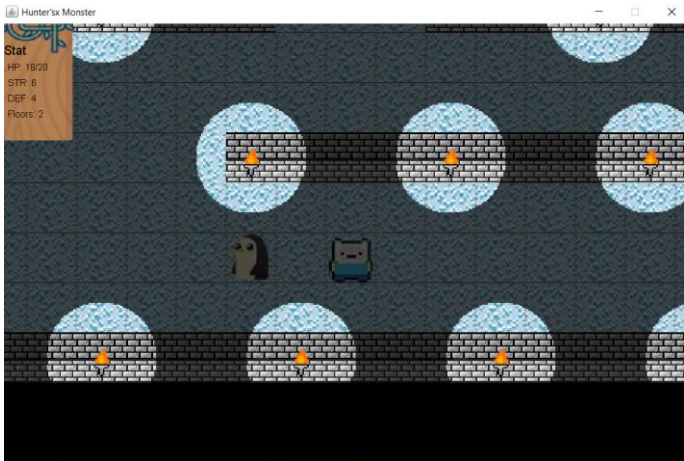
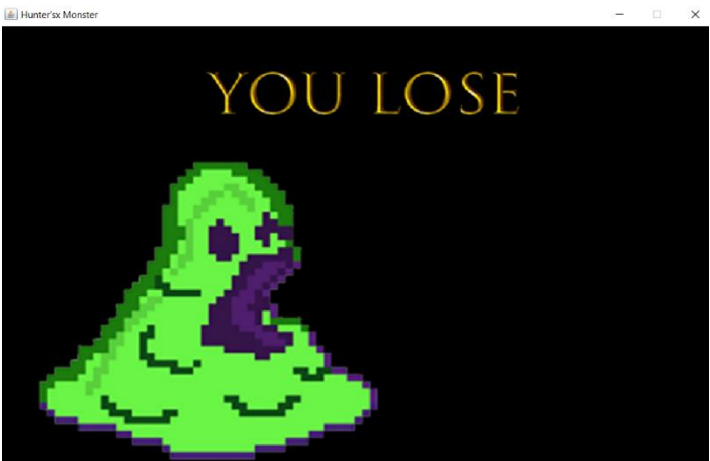
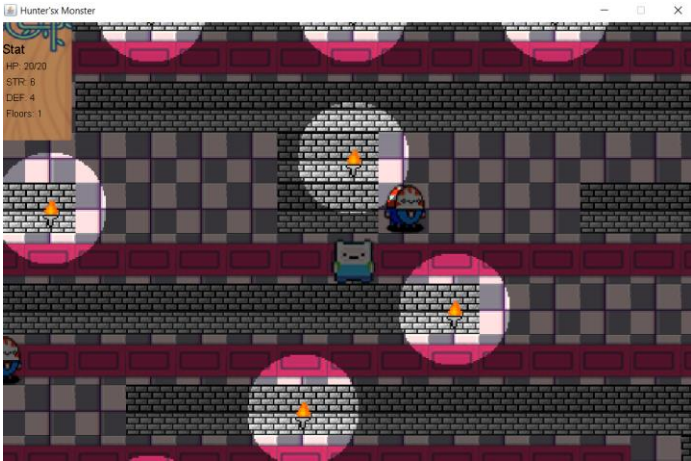
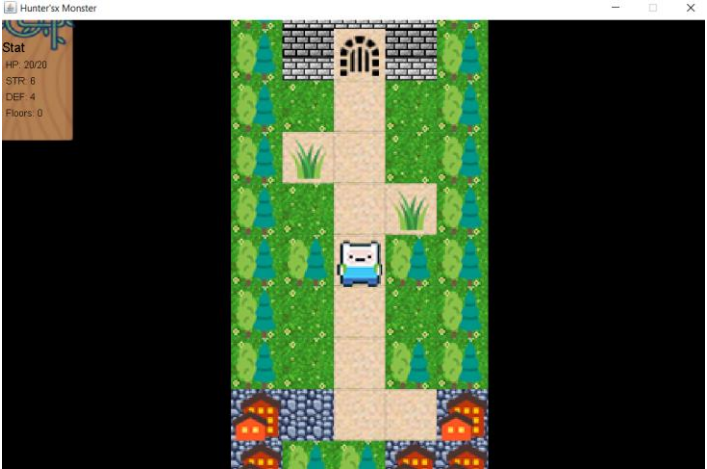
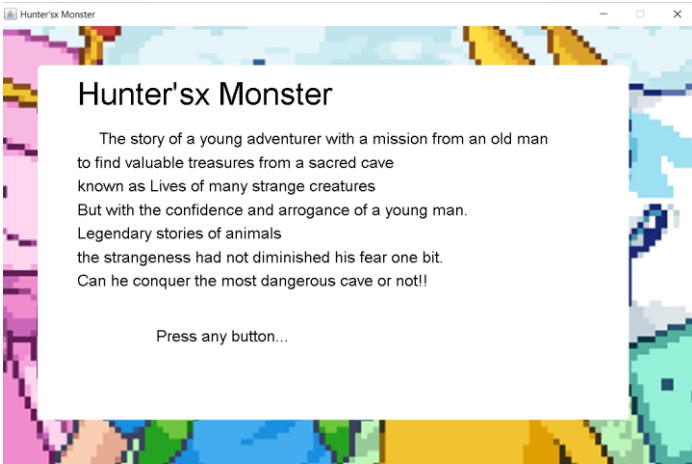
41     public void setPosition(int dirX, int dirY, boolean animated) {
42         super.setPosition(dirX, dirY, animated);
43         if(animated) {
44             this.strengthBuff--;
45             this.defenceBuff--;
46         }
47     }
48

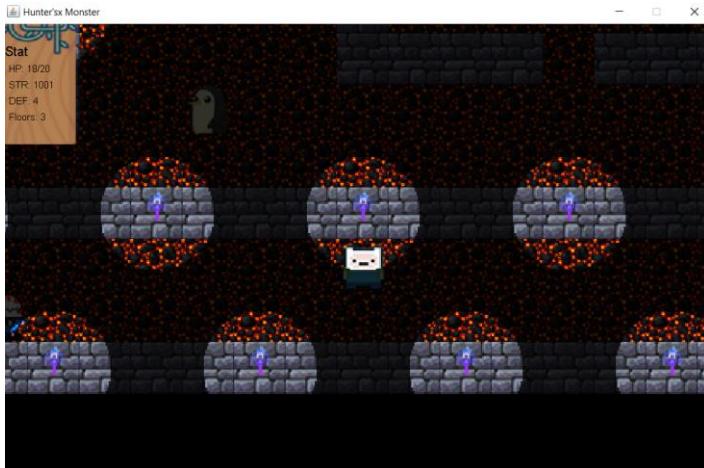
~
3 public class Armor extends Item {
4
5     private int defence;
6
7     public Armor(String name, String displayName, int defence) {
8         super(name, displayName, "Protection: "+defence);
9         this.defence = defence;
10    }

3 public class Weapon extends Item {
4
5     private int damage;
6
7     public Weapon(String name, String displayName, int damage) {
8         super(name, displayName, "Damage: "+damage);
9         this.damage = damage;
10    }
11
12     public int getDamage() {
13         return damage;
14    }
15
16 }

```

GUI





setting GUI Width , Height

```

1 package Project.gui;
2
3
4 import javax.swing.JFrame;
5
6 public class Window {
7
8     public static final int WIDTH = 900;
9     public static final int HEIGHT = 600;
10
11     private static JFrame window;
12     private static GameScreen screen;
13
14 // Create Windows which Title is "Hunter'sx Monster" and call GameScreen to load every object and logic
15 public static void create() {
16     window = new JFrame("Hunter'sx Monster");
17     window.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
18     window.setBounds(20, 20, WIDTH, HEIGHT);
19     window.setResizable(false);
20
21     screen = new GameScreen();
22     window.add(screen);
23 }
24
25 // Set windows to be visible when everything is ready
26 public static void setVisible() {
27     if(window!=null) window.setVisible(true);
28 }
29 }

```

class Renderer

```

1 package Project.gui;
2
3 import java.awt.AlphaComposite;
4
5 public class Renderer {
6
7     private int zoomLevel;
8
9     public static final Rectangle inventory = new Rectangle(150, 50, 700, 500);
10    public static BufferedImage sprite = null;
11    public static final Rectangle inventorySlot1 = new Rectangle(510, 150, 330, 60);
12    public static final Rectangle inventorySlot2 = new Rectangle(510, 220, 330, 60);
13    public static final Rectangle inventorySlot3 = new Rectangle(510, 290, 330, 60);
14
15    public static final Rectangle weaponSlot = new Rectangle(160, 150, 330, 60);
16    public static final Rectangle armorSlot = new Rectangle(160, 220, 330, 60);
17
18    public static final Rectangle messageBox = new Rectangle(200, 480, 600, 50);
19
20    public Renderer() {
21        this.zoomLevel = 2;
22    }
23
24 // Render player at position on middle of the screen
25 public void renderPlayer(EntityTile playerData, Graphics graphics) {
26     BufferedImage sprite = Textures.getSprite("player");
27
28     int drawPosX = (Window.WIDTH/2)-(sprite.getWidth()/2)*zoomLevel;
29     int drawPosY = (Window.HEIGHT/2)-(sprite.getHeight()/2)*zoomLevel;
30
31     graphics.drawImage(sprite, drawPosX, drawPosY, sprite.getWidth()*zoomLevel, sprite.getHeight()*zoomLevel, null);
32 }
33
34 // Render object in map by get data from floorData by using Player position X, Y
35 public void renderLevel(Floor floorData, Player player, Graphics graphics) {
36     for(int y = 0; y < floorData.getSizeY(); y++) {
37         for(int x = 0; x < floorData.getSizeX(); x++) {
38             BufferedImage sprite = Textures.getSprite(floorData.getTileAt(x, y).getName());
39
40             if(floorData.getTileAt(x, y).getName() == "wall" && y == floorData.getSizeY()-1)

```

Event handling

Keyboard

```
3*import java.awt.event.KeyEvent;
5
6 public class KeyboardInput implements KeyListener {
7
8     private static boolean[] keys;
9
10    private static int delay;
11
12    public KeyboardInput() {
13        // Set size array of boolean to store every keyboard button
14        keys = new boolean[1000];
15        // Set Delay timing for every Keyboard pressed to activated function which each key in 96 millisecond
16        delay = 96;
17    }
18
19    @Override
20    public void keyPressed(KeyEvent arg0) {
21        keys[arg0.getKeyCode()] = true;
22    }
23
24    @Override
25    public void keyReleased(KeyEvent arg0) {
26        keys[arg0.getKeyCode()] = false;
27    }
28
29    // Check if any is pressing on keyboard and activated that key when delay is less than 0 or equal
30    public static boolean isKeyDown(int key) {
31        if(keys[key] == true && delay <= 0) {
32            delay = 96;
33            return true;
34        }
35        else {
36            delay--;
37            return false;
38        }
39    }
40
41    @Override
42    public void keyTyped(KeyEvent arg0) {}
43 }
```

Mouse

```
1 package Project.gui;
2
3*import java.awt.event.MouseEvent;
7
8 public class Mouse implements MouseListener {
9
10    @Override
11    public void mouseClicked(MouseEvent arg0) {}
12
13    @Override
14    public void mouseEntered(MouseEvent arg0) {}
15
16    @Override
17    public void mouseExited(MouseEvent arg0) {}
18
19    @Override
20    public void mousePressed(MouseEvent arg0) {}
21
22    @Override
23    public void mouseReleased(MouseEvent arg0) {
24        GameLogic.handleLeftClick(arg0.getX(), arg0.getY());
25    }
26
27 }
```

Algorithm

Monster movement

เชื่อว่าม็ศัตรูใน map ใหม่ และศัตรูตามผู้เล่นอยู่หรือเปล่า ถ้าไม่ให้สุ่มทิศทางการเดิน 4 วิธี

```
269* private static void moveMonsters() {
270     for(Monster monster : activeMonsters) {
271         if(monster.getHealth() <= 0)
272             continue;
273
274         if(!monster.shouldChasePlayer()) {
275             switch(randomizer.nextInt(4)) {
276 // Monster stay at right next to player
277             case 0:
278                 if(currentFloor.thereIsMonsterHere(monster.getPosX()+1, monster.getPosY())) {
279                     return;
280                 }
281                 else if(monster.getPosX()+1 == player.getPosX() && monster.getPosY() == player.getPosY()) {
282                     messageBox.addMessage("Monster deal "+monster.getStrength()+" to you. You have "+player.getHealth()+" left.", 30);
283                     player.damage(monster.getStrength() - player.getDefence()/3);
284                     break;
285                 }
286                 if(getTileInFrontOfEntity(monster, 1, 0).getName() == "floor2-1" ||
287                    getTileInFrontOfEntity(monster, 1, 0).getName() == "floor3-1" ||
288                    getTileInFrontOfEntity(monster, 1, 0).getName() == "floor3-2" ||
289                    getTileInFrontOfEntity(monster, 1, 0).getName() == "floor4") {
290                     monster.setPosition(monster.getPosX()+1, monster.getPosY(), true);
291                     break;
292                 }
293 // Monster stay at left next to player
294             case 1:
295                 if(currentFloor.thereIsMonsterHere(monster.getPosX()-1, monster.getPosY())) {
296                     return;
297                 }
298                 else if(monster.getPosX()-1 == player.getPosX() && monster.getPosY() == player.getPosY()) {
299                     messageBox.addMessage("Monster deal "+monster.getStrength()+" to you. You have "+player.getHealth()+" left.", 30);
300                     player.damage(monster.getStrength() - player.getDefence()/3);
301                     break;
302                 }
303                 if(getTileInFrontOfEntity(monster, -1, 0).getName() == "floor2-1" ||
304                    getTileInFrontOfEntity(monster, -1, 0).getName() == "floor3-1" ||
305                    getTileInFrontOfEntity(monster, -1, 0).getName() == "floor3-2" ||
306                    getTileInFrontOfEntity(monster, -1, 0).getName() == "floor4") {
307                     monster.setPosition(monster.getPosX()-1, monster.getPosY(), true);
308                     break;
309                 }
310 // Monster stay at above next to player
311             case 2:
312                 if(currentFloor.thereIsMonsterHere(monster.getPosX(), monster.getPosY()+1)) {
313                     return;
314                 }
315                 else if(monster.getPosY()+1 == player.getPosY() && monster.getPosX() == player.getPosX()) {
316                     messageBox.addMessage("Monster deal "+monster.getStrength()+" to you. You have "+player.getHealth()+" left.", 30);
317                     player.damage(monster.getStrength() - player.getDefence()/3);
318                     break;
319                 }
320                 if(getTileInFrontOfEntity(monster, 0, 1).getName() == "floor2-1" ||
321                    getTileInFrontOfEntity(monster, 0, 1).getName() == "floor3-1" ||
322                    getTileInFrontOfEntity(monster, 0, 1).getName() == "floor3-2" ||
323                    getTileInFrontOfEntity(monster, 0, 1).getName() == "floor4") {
324                     monster.setPosition(monster.getPosX(), monster.getPosY()+1, true);
325                     break;
326                 }
327             }
328         }
329     }
330 }
```



```

327 // Monster stay at below next to player
328     case 3:
329         if(currentFloor.thereIsMonsterHere(monster.getPosX(), monster.getPosY()-1)) {
330             return;
331         }
332         else if(monster.getPosX() == player.getPosX() && monster.getPosY()-1 == player.getPosY()) {
333             player.damage(monster.getStrength() - player.getDefence()/3);
334             messageBox.addMessage("Monster deal "+monster.getStrength()+" to you. You have "+player.getHealth()+" left.", 30);
335             break;
336         }
337         if(getTileInFrontOfEntity(monster, 0, -1).getName() == "floor2-1" ||
338            getTileInFrontOfEntity(monster, 0, -1).getName() == "floor3-1" ||
339            getTileInFrontOfEntity(monster, 0, -1).getName() == "floor3-2" ||
340            getTileInFrontOfEntity(monster, 0, -1).getName() == "floor4") {
341             monster.setPosition(monster.getPosX(), monster.getPosY()-1, true);
342             break;
343         }
344     }
345 } else {
346 // Monster stay at right next to player
347     if(player.getPosX() > monster.getPosX()) {
348         if(monster.getPosX()+1 == player.getPosX() && monster.getPosY() == player.getPosY()) {
349             player.damage(monster.getStrength() - player.getDefence()/3);
350             messageBox.addMessage("Monster deal "+monster.getStrength()+" to you. You have "+player.getHealth()+" left.", 30);
351         }
352         else if(getTileInFrontOfEntity(monster, 1, 0).getName() == "floor2-1" ||
353            getTileInFrontOfEntity(monster, 1, 0).getName() == "floor3-1" ||
354            getTileInFrontOfEntity(monster, 1, 0).getName() == "floor3-2" ||
355            getTileInFrontOfEntity(monster, 1, 0).getName() == "floor4") {
356             monster.setPosition(monster.getPosX()+1, monster.getPosY(), true);
357         }
358     }
359 // Monster stay at left next to player
360     else if(player.getPosX() < monster.getPosX()) {
361         if(monster.getPosX()-1 == player.getPosX() && monster.getPosY() == player.getPosY()) {
362             player.damage(monster.getStrength() - player.getDefence()/3);
363             messageBox.addMessage("Monster deal "+monster.getStrength()+" to you. You have "+player.getHealth()+" left.", 30);
364         }
365         else if(getTileInFrontOfEntity(monster, -1, 0).getName() == "floor2-1" ||
366            getTileInFrontOfEntity(monster, -1, 0).getName() == "floor3-1" ||
367            getTileInFrontOfEntity(monster, -1, 0).getName() == "floor3-2" ||
368            getTileInFrontOfEntity(monster, -1, 0).getName() == "floor4") {
369             monster.setPosition(monster.getPosX()-1, monster.getPosY(), true);
370         }
371     }
372 // Monster stay at above next to player
373     else if(player.getPosY() > monster.getPosY()) {
374         if(monster.getPosX() == player.getPosX() && monster.getPosY()+1 == player.getPosY()) {
375             player.damage(monster.getStrength() - player.getDefence()/3);
376             messageBox.addMessage("Monster deal "+monster.getStrength()+" to you. You have "+player.getHealth()+" left.", 30);
377         }
378         else if(getTileInFrontOfEntity(monster, 0, 1).getName() == "floor2-1" ||
379            getTileInFrontOfEntity(monster, 0, 1).getName() == "floor3-1" ||
380            getTileInFrontOfEntity(monster, 0, 1).getName() == "floor3-2" ||
381            getTileInFrontOfEntity(monster, 0, 1).getName() == "floor4") {
382             monster.setPosition(monster.getPosX(), monster.getPosY()+1, true);
383         }
384     }
385 // Monster stay at below next to player
386     else if(player.getPosY() < monster.getPosY()) {
387         if(monster.getPosX() == player.getPosX() && monster.getPosY()-1 == player.getPosY()) {
388             player.damage(monster.getStrength() - player.getDefence()/3);
389             messageBox.addMessage("Monster deal "+monster.getStrength()+" to you. You have "+player.getHealth()+" left.", 30);
390         }
391         else if(getTileInFrontOfEntity(monster, 0, -1).getName() == "floor2-1" ||
392            getTileInFrontOfEntity(monster, 0, -1).getName() == "floor3-1" ||
393            getTileInFrontOfEntity(monster, 0, -1).getName() == "floor3-2" ||
394            getTileInFrontOfEntity(monster, 0, -1).getName() == "floor4") {
395             monster.setPosition(monster.getPosX(), monster.getPosY()-1, true);
396         }
397     }
398 }
399 }
400 }
401

```

บทที่ 3

สรุป

ปัญหาที่พบระหว่างการทำงาน

1. บอทศัตรูบางครั้งไม่ไล่ตามหรือเกิดข้อผิดพลาด
2. เกิด error บ่อยครั้ง โดยไม่รู้สาเหตุ
3. ใช้เวลาในการหา กราฟฟิคค่อนข้างนาน

จุดเด่นของโปรแกรมที่ไม่เหมือนใคร

- มีการกำหนดอัลกอริทึมของศัตรูที่สอดคล้องกับการเดินของผู้เล่นมีการเดินตาม/ไม่ตาม
- ความมืดของด่านจะไม่ถูกกำหนดตายตัว
- มีไอเทมพิเศษสำหรับผู้เล่นเพื่อการทำลาย

คำแนะนำสำหรับผู้สอน

อาจารย์เป็นคนกันเอง ใจดี เข้าถึงง่าย ช่วยเหลือนักศึกษาเต็มທີ່มากๆ แบบฝึกหัดเยอะมาก เช่นกัน ส่วนตัวอยากให้อาจารย์ก่อนทำแลบอยากให้อาจารย์ยกตัวอย่างแบบฝึกหัดที่ความยากพอๆกับแบบฝึกหัดเขียนไปพร้อมๆกับนักศึกษาก่อนที่จะปล่อยให้ลองทำเอง สำหรับตัวผู้เรียนเองในตอนแรกคิดว่ายับยั้งตันว่าจะเริ่มจุดไหนก่อนเลยไม่มีแรงจูงใจให้เรียนได้สนุกเท่าที่ควร เพราะพื้นฐานแต่ละคนไม่เท่ากัน หากอาจารย์รับพิจารณาจะเป็นประโยชน์ในรุ่นต่อไปมากๆ ขอขอบคุณค่ะ

คำแนะนำให้รุ่นน้อง

ตั้งใจเรียน เรียนรู้ syntax มาก่อนเลยเป็นไปได้นั่นเรื่องพื้นฐานมาจะง่ายสุด ภาษาจาวา
ค่อนข้างเรื่องมากและเรื่องที่เราเรียนก็ค่อนข้างเรื่องมาเช่นกัน ค่อยๆเก็บไปที่ละบทสะสมไปเรื่อยๆ
ทุกวัน พยายามทำเองส่งสัปดาห์ตามอาจารย์ ทำโปรเจกต์แต่เนิ่นๆ อย่าหักโหม