



Starving cat Game

BY COOLCAT GROUP



1



โจทย์และการ วิเคราะห์โจทย์

Problem and Problem Analysis



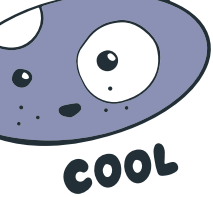
Problem and Problem Analysis

โจทย์ปัญหา

Problem

กลุ่มของเราได้แรงบันดาลใจมาจากเกม Pacman ที่เคยเล่นในสมัยเด็กๆ โดยนำเกมมาปรับปรุงใหม่ให้มีความน่าสนใจมากขึ้น อีกทั้งมีการปรับตัวละครและฉากของเกมให้มีความน่าดึงดูดใจเพโดยใช้โปรแกรมเชิงวัตถุ (Object-oriented Programming) ในการทำเกม





Problem and Problem Analysis

วิเคราะห์โจทย์

1

นำความรู้เรื่องการเขียนโปรแกรมเชิงวัตถุ (Object-oriented Programming) และ ภาษา JAVA มาใช้ให้เกิดประโยชน์

2

ออกแบบตัวละคร และฉากเกมให้มีความน่าดึงดูดใจเพิ่มมากขึ้น

3

ออกแบบเกมที่มีวิธีเล่นที่เข้าใจง่าย สามารถเล่นได้ทุกเพศทุกวัย

แนวทางการแก้ปัญหา

Problem Analysis

1

ศึกษารายละเอียด
และโครงสร้างของ
การทำเกม

2

แบ่งหน้าที่และ
วางแผนการทำงาน
ของสมาชิก
ในกลุ่ม

3

ออกแบบ
ระบบเกมว่า
ภายในตัวเกม
จะมีอะไรบ้าง

4

ออกแบบตัวละคร
และฉากให้มีความ
น่าสนใจ

5

สร้างโปรแกรมเชิง
วัตถุอย่างเป็น
ระบบ

6

ทดสอบการทำงาน
ของโปรแกรม แก้ไข
และปรับปรุงข้อผิดพลาดที่เกิดขึ้น

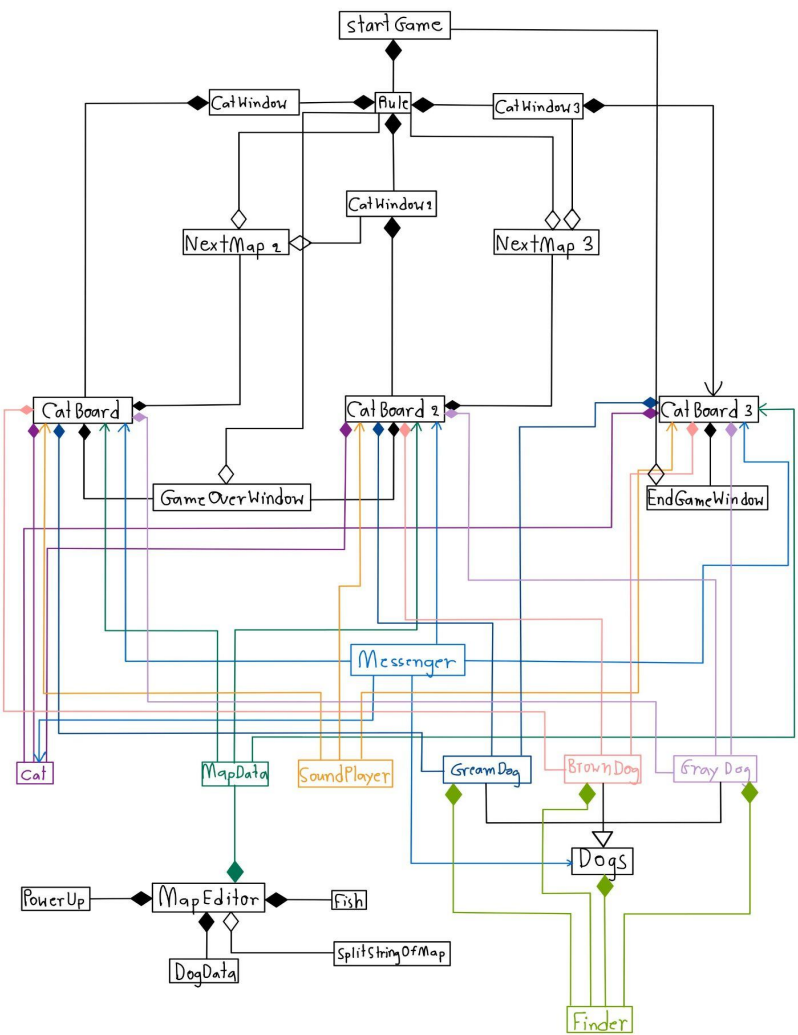
2



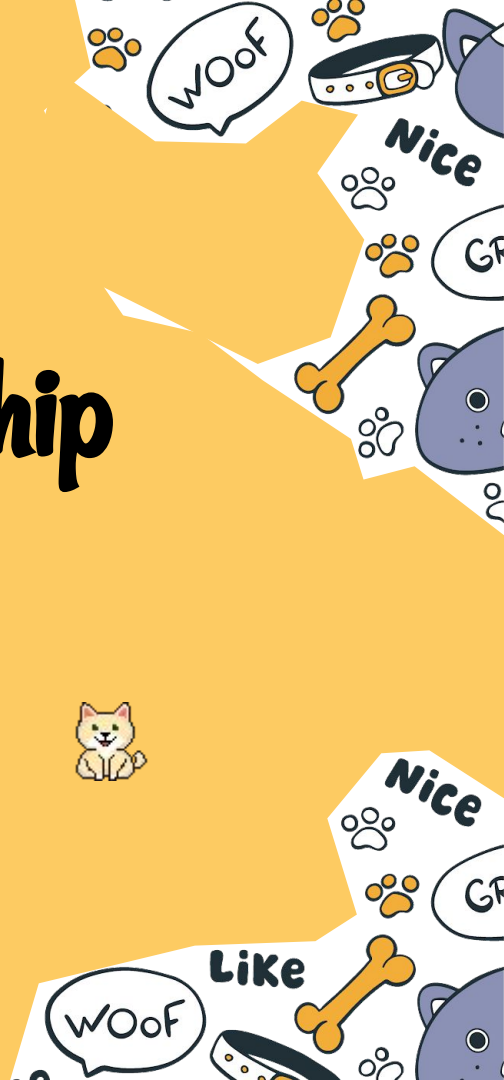
การออกแบบคลาส

Class design





Relationship



PACKAGE: StarvingCat

StartGame

- frame : JFrame
- img : ImageIcon
- label : JLabel
- startButton : JButton
- exitButton : JButton

- + main()
- + StartGame()
- + keyPressed(KeyEvent e) : void
- + keyReleased(KeyEvent e) : void
- + keyTyped(KeyEvent e) : void

MapEditor

- + x : int
- + y : int
- + map : int[][]
- + i : int = 0
- + j : int = 0
- + mapstr : char[]

- + compileMap(String mapStr) : MapData

SplitStringofMap

- + strSplit : String[]
- + SplitString(String mapStr) : int

CatWindow2

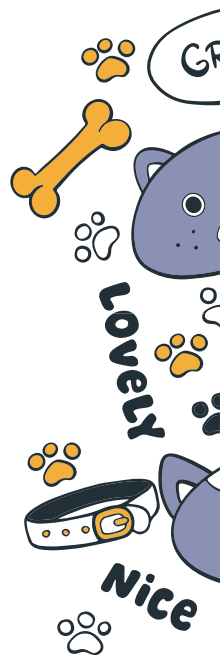
- + CatWindow2()
- + loadMap(int mx, int my, String mapA) : int[][]
- + getMapFromResource(String mapA) : MapData
- + adjustMap(MapData mapA) : void

CatWindow

- + CatWindow
- + loadMap(int mx,int my,String mapA) : int[][]
- + getMapFromResource(String mapA) : MapData
- + adjustMap(MapData mapA) : void

CatWindow3

- + CatWindow3()
- + loadMap(int mx, int my, String mapA) : int[][]
- + getMapFromResource(String mapA) : MapData
- + adjustMap(MapData mapA) : void

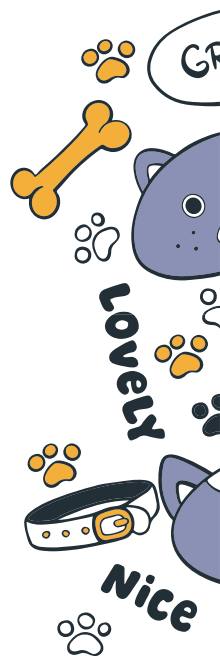


Class MapData



MapData
<ul style="list-style-type: none">- x : int- y : int- map : int[][]- CatPosition : Point- DogBasePosition : Point- isCustom : Boolean- fishPosition : ArrayList<Fish>- powUpPosition : ArrayList<PowerUp>- move : ArrayList<Move>- dogData : ArrayList<DogData>
<ul style="list-style-type: none">+ MapData()+ MapData(int x, int y)+ MapData(int x, int y, int[][] map, Point catPosition)

- + getX() : int
- + setX(int x) : void
- + getY() : int
- + setY(int y) : void
- + getMap() : int[][]
- + setMap(int[][] map) : void
- + getCatPosition() : Point
- + setCatPosition(Point catPosition) : void
- + getDogBasePosition() : Point
- + setDogBasePosition(Point dogBassePosition) : void
- + getFishPositions() : ArrayList<Fish>
- + getPowerUpPositions() : ArrayList<PowerUp>
- + getTeleports() : ArrayList<Move>
- + getDogData() : ArrayList<DogData>
- + isCustom() : boolean
- + setCustom(Boolean custom) : void

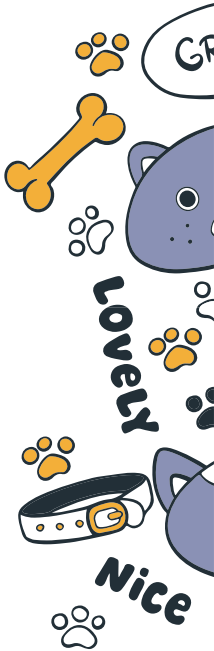


Class Cat



Cat
+ moveTimer : Timer
+ moveAL : ActionListener
+ activeMove : MoveType
+ todoMove : MoveType
+ isStuck : Boolean = true
+ animTimer : Timer
+ animal : ActionListener
+ cat : Image
+ activeImage : int = 0
+ addFactor : int = 1
+ pixelPosition : Point
+ logicalPosition : Point

- parentBoard : CatBoard
- parentBoard2 : CatBoard2
- parentBoard3 : CatBoard3
+ Cat(int x, int y, CatBoard catBoard)
+ Cat(int x, int y, CatBoard2 catBoard2)
+ Cat(int x, int y, CatBoard3 catBoard3)
+ isPossibleMove1(MoveType todoMove2) : Boolean
+ isPossibleMove2(MoveType todoMove2) : Boolean
+ isPossibleMove3(MoveType todoMove2) : Boolean
+ getCatImage() : Image
+ keyTyped(KeyEvent e) : void
+ keyReleased(KeyEvent e) : void
+ keyPressed(KeyEvent ke) : void



Class CatBoard

CatBoard

```
+ redrawTimer : Timer
+ redrawAL : ActionListener
+ map : int[]
+ mapSegments : Image[]
+ fishImage : Image
+ cherry: Image
+ cat : Cat
+ fish : ArrayList<Fish>
+ cherry: ArrayList<PowerUp>
+ dog : ArrayList<Dog>
+ move : ArrayList<Move>
+ isCustom : boolean = false
+ isGameOver : boolean = false
+ isWin : boolean = false
+ drawScore : boolean = false
+ clearScore : boolean = false
```

```
+ scorePower : int = 0
+ score : int
+ scoreboard : JLabel
+ dogBase : Point
+ move_x : int
+ move_y : int
+ md_backup : MapData
+ windowBase : CatWindow
+ clickSoundURL : URL

+ CatBoard(JLabel scoreboard, MapData md, CatWindow cw)
- collisionTest() : void
- update() : void
+ paintComponent(Graphics g) : void
+ processEvent(AWTEvent ae) : void
```



Class CatBoard2

CatBoard2

```
+ redrawTimer : Timer
+ redrawAL : ActionListener
+ map : int[]
+ mapSegments : Image[]
+ fishImage : Image
+ water : Image
+ cat : Cat
+ fish : ArrayList<Fish>
+ water : ArrayList<PowerUp>
+ dog : ArrayList<Dog>
+ move : ArrayList<Move>
+ isCustom : boolean = false
+ isGameOver : boolean = false
+ isWin : boolean = false
+ drawScore : boolean = false
+ clearScore : boolean = false
```

```
+ scorePower : int = 0
+ score : int
+ scoreboard : JLabel
+ dogBase : Point
+ move_x : int
+ move_y : int
+ md_backup : MapData
+ windowBase : CatWindow2
+ clickSoundURL : URL

+ CatBoard2(JLabel scoreboard, MapData md, CatWindow2 catWindow2)
-collisionTest() : void
-update() : void
+ paintComponent(Graphics g) : void
+ processEvent(AWTEvent ae) : void
```





Class CatBoard3

CatBoard3

```
+ redrawTimer : Timer
+ redrawAL : ActionListener
+ map : int[]
+ mapSegments : Image[]
+ fishImage : Image
+ strawberryImage : Image
+ cat : Cat
+ fish : ArrayList<Fish>
+ strawberry : ArrayList<PowerUp>
+ dog : ArrayList<Dog>
+ move : ArrayList<Move>
+ isCustom : boolean = false
+ isGameOver : boolean = false
+ isWin : boolean = false
```

```
+ drawScore : boolean = false
+ clearScore : boolean = false
+ scorePower : int = 0
+ score : int
+ scoreboard : JLabel
+ dogBase : Point
+ move_x : int
+ move_y : int
+ md_backup : MapData
+ windowBase : CatWindow3

+ CatBoard3(JLabel scoreboard, MapData md, CatWindow3 catWindow3)
-collisionTest() : void
-update() : void
+ paintComponent(Graphics g) : void
+ processEvent(AWTEvent ae) : void
```



Class Dogs

Dogs

```
+ animationTime : Timer
+ animationAL : ActionListener
+ pendingTimer : Timer
+ pendingAL : ActionListener
+ moveTimer : Timer
+ moveAL : ActionListener
+ activeMove : MoveType
# isStuck : Boolean = true
+ isPending : Boolean = false
# isDead : boolean = false
+ dogImage : Image
+ activeImage : int = 0
+ pixelPosition : Point
+ logicalPosition : Point
+ dog : Image
+ dogNormal : int
```

```
+ dogDead : int = 5
+ baseReturner : Finder
# baseboard : CatBoard
# baseboard2 : CatBoard2
# baseboard3 : CatBoard3
```

```
+ abstract class Dogs
+ isDead() : Boolean
+ Dogs(int x, int y, CatBoard cb, int dogDelay)
+ Dogs(int x, int y, CatBoard2 cb2, int dogDelay)
+ Dogs(int x, int y, CatBoard3 cb3, int dogDelay)
+ actionPerformed(ActionEvent event) : void
+ loadImages() : void
+ getMove1() : MoveType
+ getMove2() : MoveType
+ getMove3() : MoveType
+ getDogImage() : Image
+ dead() : void
+ undie : void
```



Fish
+ position : Point
+ Fish(int x, int y)

DogData
- x : int
- y : int
- color : DogType
+ DogData(int x, int y, DogType color)
+ setX(int x) : void
+ getX() : int
+ setY(int y) : void
+ getY() : int
+ getType() : DogType

MoveType
+ MoveType : enum
DogType
+ DogType : enum





PowerUp
+ position : Point
+ PowerUp(int x, int y)

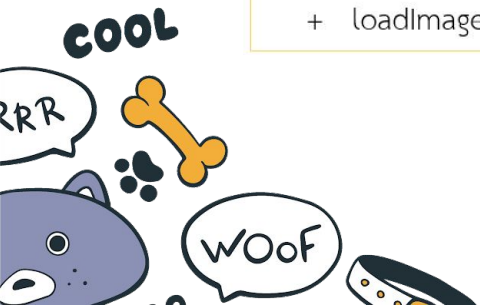


BrownDog
+ move : MoveType = MoveType.UP
+ bfs : Finder
+ BrownDog(int x,int y, CatBoard2 catBoard2)
+ BrownDog(int x,int y, CatBoard catBoard)
+ BrownDog(int x,int y, CatBoard3 catBoard3)
+ getMove1() : MoveType
+ getMove2() : MoveType
+ getMove3() : MoveType
+ loadImage() : void

Finder
+ map : int[]
+ moveX : int
+ moveY : int
+ i : int
+ j : int
+ Finder(CatBoard cb)
+ Finder(CatBoard2 cb2)
+ Finder(CatBoard3 cb3)
- MazeCell
+ MazeCell(int x, int y)
+ isUsable(int l, int j, Boolean[][] mark) : Boolean
+ getMove(int x, int y, int topX, int topY) : MoveType



CreamDog	GrayDog
+ move : MoveType = MoveType.UP	+ move : MoveType = MoveType.UP
+ bfs : Finder	+ bfs : Finder
+ CreamDog(int x,int y, CatBoard catBoard)	+ GrayDog(int x,int y, CatBoard catBoard)
+ CreamDog(int x,int y, CatBoard2 catBoard2)	+ GrayDog(int x,int y, CatBoard2 catBoard2)
+ CreamDog(int x,int y, CatBoard3 catBoard3)	+ GrayDog(int x,int y, CatBoard3 catBoard3)
+ getMove1() : MoveType	+ getMove1() : MoveType
+ getMove2() : MoveType	+ getMove2() : MoveType
+ getMove3() : MoveType	+ getMove3() : MoveType
+ loadImage() : void	+ loadImage() : void



Move

- From : Point
 - To : Point
 - reqMove : MoveType
-
- + getTo() : Point
 - + setTo(Point to) : void
 - + getFrom() : Point
 - + setFrom(Point from) : void
 - + getReqMove() : MoveType
 - + setReqMove(MoveType reqMove) : void
 - + Move(int x1, int y1, int x2, int y2, MoveType reqMove)

NextMap2

- frame : JFrame
 - startButton : JButton
 - exitButton : JButton
 - label : JLabel
-
- + vicImage : ImageIcon
-
- + NextLevel2()

Messenger

- + UPDATE : int = AWTEvent.RESERVED_ID_MAX + 1
- + COLTEST : int = AWTEvent.RESERVED_ID_MAX + 2

NextMap3

- frame : JFrame
 - startButton : JButton
 - exitButton : JButton
 - label : JLabel
-
- + vicImage : ImageIcon
-
- + NextLevel3()



SoundPlayer

+ play(final URL name) : void

GameOverWindow

- frame : JFrame
- + failImage : ImageIcon
- + GameOverWindow()

EndGameWindow

- frame : JFrame
- exitButton : JButton
- label : JLabel
- + vicImage : ImageIcon
- + failImage : ImageIcon
- + win() : void
- + lose() : void

Rule

- frame : JFrame
- startButton : JButton
- exitButton : JButton
- label : JLabel
- img : ImageIcon
- + Rule()

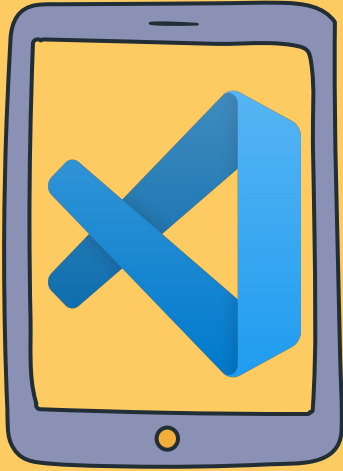




3



เครื่องมือและ
เทคนิคที่ใช้
(Tool and Technique)



Visual Studio Code

ในการทำเกม Starving Cat เราได้ใช้โปรแกรม Visual Studio Code เขียนทั้งหมด เพราะมีความคุ้นชินกับการใช้โปรแกรมนี้จากวิชา 204111 อีกทั้งตัวโปรแกรมยังสามารถแชร์กันทำงานพร้อมกันได้ และมีประสิทธิภาพในการทำงานเป็นอย่างมาก

Piskel

การออกแบบกราฟิก เราได้ใช้เว็บ Piskel ในการทำเนื่องจากใช้งานง่าย และได้รูปภาพตามที่ต้องการ





I love img

เว็บไซต์ใช้ในการปรับ
ขนาดรูปภาพ

I ♥ IMG

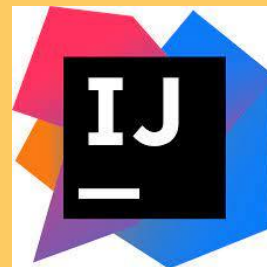


PhotoScape X

ใช้โปรแกรม PhotoScape X
ในการครอบตัดรูปภาพ

IntelliJ

ใช้โปรแกรม IntelliJ
ในการexport ไฟล์



4

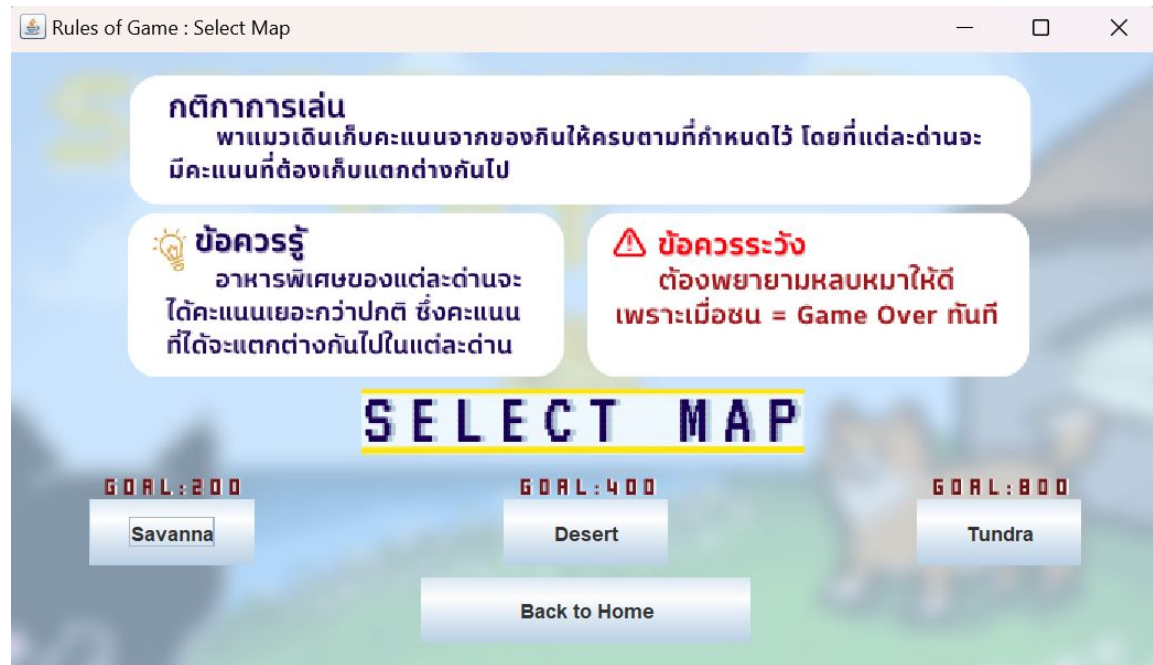


ภาพการทำงานของโปรแกรม
(Screen Capture)





กดปุ่ม START GAME เพื่อเข้าเล่นเกม หรือปุ่ม EXIT เพื่อ
ออกจากเกม



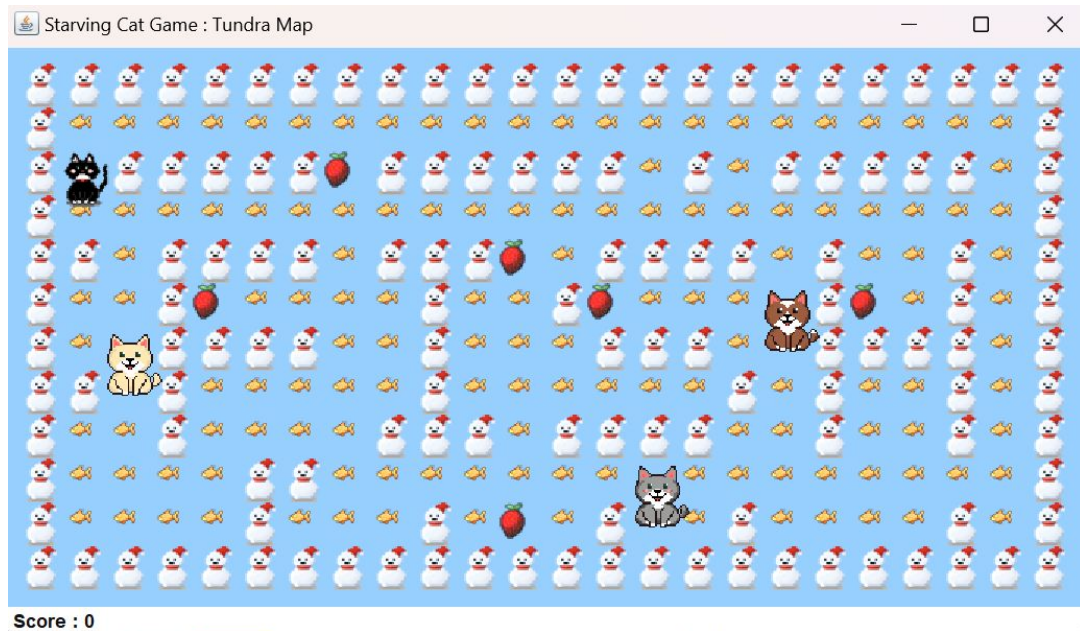
ให้ผู้เล่นอ่านกติกาการเล่นเกม โดยผู้เล่นสามารถเลือก map ก่อนเข้าเล่นได้ 3 map นั่นคือ Savanna, Desert และ Tundra หากผู้เล่นต้องการกลับสู่หน้าหลักให้กดปุ่ม Back to Home



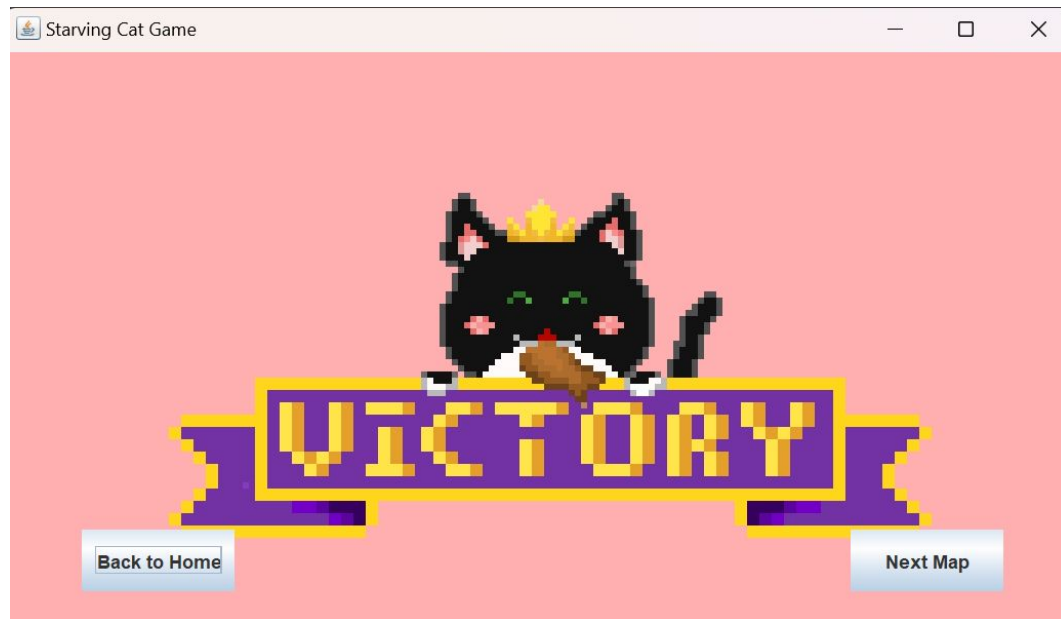
ตัวอย่างหน้าต่างเกม (เกมด่าน Savanna) หากผู้เล่นเลือกด่านนี้ ผู้
เล่น(Black cat)ต้องกดปุ่มลูกศรขึ้นลงซ้ายขวา บนคีย์บอร์ดเพื่อ
เคลื่อนย้ายแมว และเก็บคะแนน(Fish)ให้ได้ครบ **200 คะแนน** ตามที่
กติกาได้กำหนดไว้จึงจะชนะ หากผู้เล่นชนกับศัตรู(Dog)ถือว่าแพ้ใน
เกมนั้น



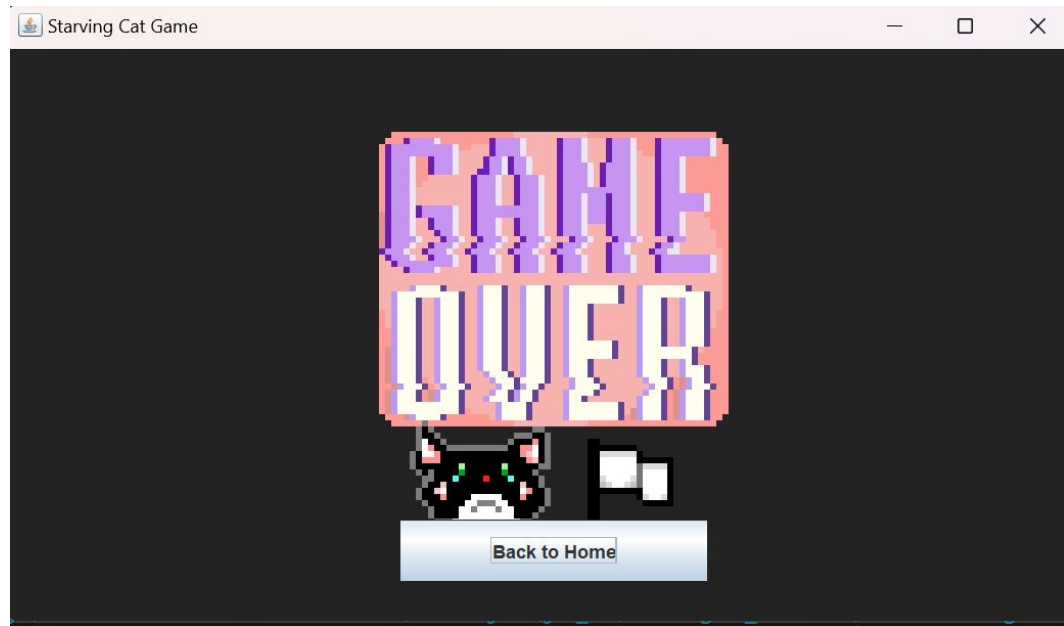
ตัวอย่างหน้าต่างเกม (เกมด่าน Desert) หากผู้เล่นเลือกด่านนี้ ผู้
เล่น(Black cat)ต้องกดปุ่มลูกศรขึ้นลงซ้ายขวา บนคีย์บอร์ดเพื่อ
เคลื่อนย้ายแมว และเก็บคะแนน(Fish)ให้ได้ครบ **400 คะแนน** ตามที่
กติกาได้กำหนดไว้จึงจะชนะ หากผู้เล่นชนกับศัตรู(Dog)ถือว่าแพ้ใน
เกมนั้น



ตัวอย่างหน้าต่างเกม (เกมด่าน Tundra) หากผู้เล่นเลือกด่านนี้ ผู้
เล่น(Black cat)ต้องกดปุ่มลูกศรขึ้นลงซ้ายขวา บนคีย์บอร์ดเพื่อ
เคลื่อนย้ายแมว และเก็บคะแนน(Fish)ให้ได้ครบ **800 คะแนน** ตาม
ที่กติกาได้กำหนดไว้จึงจะชนะ หากผู้เล่นชนกับศัตรู(Dog)ถือว่าแพ้
ในเกมนั้น



ตัวอย่างหน้าต่างเกมหาคชนะในด้าน Savanna, Desert, Tundra และ
เมื่อผู้เล่นกดปุ่ม Back to Home จะกลับสู่หน้ากติกา



ตัวอย่างหน้าต่างเกมหากแพ้ และเมื่อผู้เล่นกดปุ่ม Back to Home จะกลับสู่หน้ากติกา

5



การติดตั้งโปรแกรม (Program Installation)



1. ติดตั้งโปรแกรม JAVA รุ่นที่ติดตั้งเกมได้ รุ่นที่รองรับตามภาพด้านล่าง

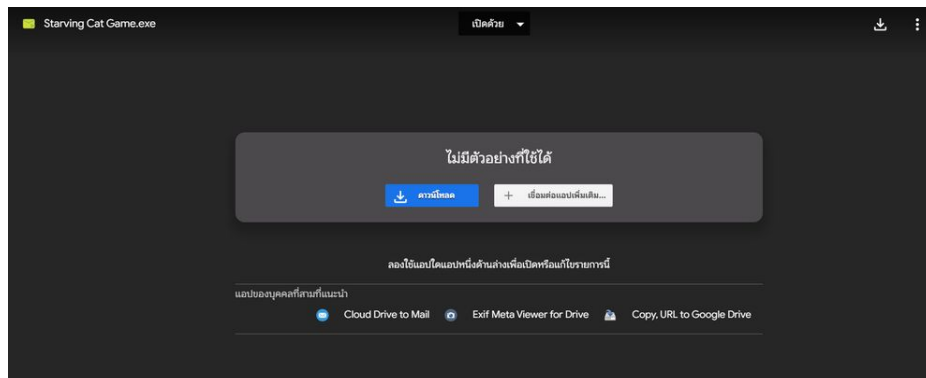
Command 'java' not found, but can be installed with:

```
sudo apt install openjdk-11-jre-headless # version 11.0.13+8-0ubuntu1~20.04, or
sudo apt install default-jre # version 2:1.11-72
sudo apt install openjdk-16-jre-headless # version 16.0.1+9-1~20.04
sudo apt install openjdk-17-jre-headless # version 17.0.1+12-1~20.04
sudo apt install openjdk-8-jre-headless # version 8u312-b07-0ubuntu1~20.04
sudo apt install openjdk-13-jre-headless # version 13.0.7+5-0ubuntu1~20.04
```

2. ดาวน์โหลดเกมโดยกดที่ลิงก์

https://drive.google.com/file/d/1686zkS1y33LcDTZeOrLocJU0NRopn2sf/view?usp=share_link

3. เมื่อขึ้นหน้าต่างกด “ดาวน์โหลด”



4. เมื่อขึ้นหน้าต่างถัดไปให้กด “ดาวน์โหลดต่อ

Google ไดรฟ์ไม่สามารถสแกนไฟล์นี้เพื่อตรวจหาไวรัส

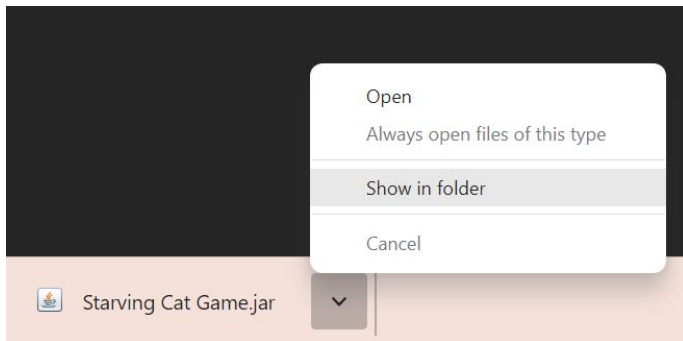
ไฟล์นี้เป็นไฟล์ปฏิบัติการและอาจเป็นอันตรายต่อคอมพิวเตอร์

[Starving Cat Game.exe \(2.8M\)](#)

ดาวน์โหลดต่อไป

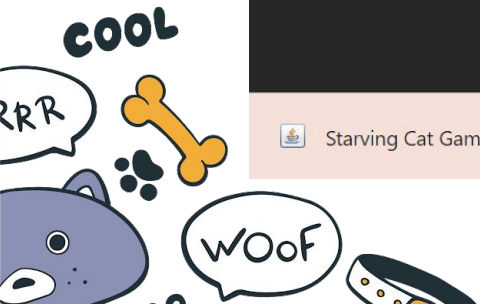


5. เมื่อแท็บดาวน์โหลดขึ้นดาวน์โหลดเสร็จเรียบร้อยแล้ว กด “show in folder” หรือ “เปิดในโฟลเดอร์”







Tips







เมื่อดาวน์โหลดเสร็จสิ้นสามารถเปิดโปรแกรม file explorer และกดตรงประวัติการดาวน์โหลดได้เช่นเดียวกัน



5. หลังจากพบที่อยู่ไฟล์ที่ดาวน์โหลดมา Click Open เพื่อเปิดไฟล์

▼ Today

 Starving Cat Game	6/3/2566 22:06	Executable Jar File	2,828 KB
 Starving Cat Game	6/3/2566 22:06		2,890 KB
 launch4j-3.50-win32 (1)	6/3/2566 22:06		8,618 KB
 Unconfirmed 295520 ordownload	6/3/2566 22:06		170 101 KB

    
 Open Enter

เท่านี้ก็ขึ้นหน้าต่าง
เริ่มเกมก็พร้อมให้เล่นแล้ว





สมาชิกภายในกลุ่ม

นางสาว กนกวรรณ มิ่งขวัญ 650510652 ตอน 001

นางสาว ชัญญชิดา ประทุม 650510659 ตอน 002

นางสาว เมษนี ลายเอื้อง 650510676 ตอน 002

นางสาว วาริศา นาระทะ 650510683 ตอน 001

