

Starving Cat



คณะผู้จัดทำ

นางสาวกนกวรรณ มิ่งขวัญ 650510652 ตอน 001

นางสาวชนัญชิตา ประทุม 650510659 ตอน 002

นางสาวเมษนี ลายเอื้อง 650510676 ตอน 002

นางสาววาริตา นาระทะ 650510683 ตอน 001

อาจารย์ที่ปรึกษา

ผศ.เบญจมาศ ปัญญางาม

ผศ.ดร.เมทินี เขียวกันยะ

โครงการนี้เป็นส่วนหนึ่งของวิชา 204114 การเขียนโปรแกรมเชิงวัตถุเบื้องต้น

ภาควิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยเชียงใหม่

บทนำ

ชื่อกลุ่ม CoolCat

หัวข้อโครงการ Starving Cat

จัดทำโดย

นางสาวกนกวรรณ มิ่งขวัญ 650510652 ตอน 001

นางสาวชนัญชิตา ประทุม 650510659 ตอน 002

นางสาวเมษนี ลายเอื้อง 650510676 ตอน 002

นางสาววาริศา นาระทะ 650510683 ตอน 001

หน้าที่ในกลุ่ม

Coder : ชนัญชิตา ประทุม, เมษนี ลายเอื้อง

Graphic Designer : กนกวรรณ มิ่งขวัญ, วาริศา นาระทะ

Documentation : วาริศา นาระทะ, กนกวรรณ มิ่งขวัญ, ชนัญชิตา ประทุม, เมษนี ลายเอื้อง

Media presentation : กนกวรรณ มิ่งขวัญ, ชนัญชิตา ประทุม, เมษนี ลายเอื้อง, วาริศา นาระทะ

สารบัญ

เรื่อง	หน้า
Problem and Problem Analysis	1
Class design	2
Class relationships	8
Tool and Technique	9
Screen Capture	11
Program Installation	15
Reference	17

โจทย์และการวิเคราะห์โจทย์

(Problem and Problem Analysis)

โจทย์ปัญหา

กลุ่มของเราได้แรงบันดาลใจมาจากเกม Pacman ที่เคยเล่นในสมัยเด็กๆ เกม Pacman ที่เราเคยเห็นกันจะเป็นตัวละครวงกลมสีเหลือง วิ่งหนีเอเลี่ยน ต้องกินจุดให้หมดถึงจะชนะ และมีให้เล่นแมพเดียว แต่ของกลุ่มเรานำเกมมาปรับปรุงให้มีความแปลกใหม่คือ มีหลายแมพให้เลือกเล่น มีการปรับตัวละครและฉากของเกมให้มีความน่าดึงดูดใจเพิ่มขึ้น อีกทั้งแต่ละแมพจะมีกติกาในการเล่นที่แตกต่างกันออกไป ซึ่งกลุ่มของเราจะใช้โปรแกรมเชิงวัตถุ (Object-oriented Programming) ในการทำเกม

วิเคราะห์โจทย์

- 👉 นำความรู้เรื่องการเขียนโปรแกรมเชิงวัตถุ (Object-oriented Programming) และภาษา JAVA มาใช้ให้เกิดประโยชน์
- 👉 ออกแบบตัวละครและฉากเกมให้มีความน่าดึงดูดใจเพิ่มมากขึ้น
- 👉 ออกแบบเกมที่มีวิธีเล่นที่เข้าใจง่าย สามารถเล่นได้ทุกเพศทุกวัย

แนวทางการแก้ไขปัญหา

- 👉 ศึกษารายละเอียดและโครงสร้างของการทำเกม
- 👉 แบ่งหน้าที่และวางแผนการทำงานของสมาชิกภาคในกลุ่ม
- 👉 ออกแบบระบบเกมว่าภายในตัวเกมจะมีอะไรบ้าง
- 👉 ออกแบบตัวละครและฉากให้มีความน่าสนใจ
- 👉 สร้างโปรแกรมเชิงวัตถุอย่างเป็นระบบ
- 👉 ทดสอบการทำงานของโปรแกรม แก้ไข และปรับปรุงข้อผิดพลาดที่เกิดขึ้น

การออกแบบคลาส

(Class design)

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[]
 - + l : int = 0
 - + j : int = 0
 - + mapstr : char[]
-
- + compileMap(String mapStr) : MapData

SplitStringofMap

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

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

CatWindow2

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

Fish

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

MapData

- x : int
- y : int
- map : int[][]
- CatPosition : Point
- DogBasePosition : Point
- isCustom : Boolean
- fishPosition : ArrayList<Fish>
- powUpPosition : ArrayList<PowerUp>
- move : ArrayList<Move>
- dogData : ArrayList<DogData>

- + 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

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

PowerUp

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

CatBoard
<ul style="list-style-type: none"> + 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
<ul style="list-style-type: none"> + CatBoard(JLabel scoreboard, MapData md, CatWindow cw) - collisionTest() : void - update() : void + paintComponent(Graphics g) : void + processEvent(AWTEvent ae) : void

CatBoard2
<ul style="list-style-type: none"> + 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
<ul style="list-style-type: none"> + CatBoard2(JLabel scoreboard, MapData md, CatWindow2 catWindow2) -collisionTest() : void -update() : void + paintComponent(Graphics g) : void + processEvent(AWTEvent ae) : void

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

```

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

```

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

```


CreamDog
+ move : MoveType = MoveType.UP
+ bfs : Finder
+ CreamDog(int x,int y, CatBoard catBoard)
+ CreamDog(int x,int y, CatBoard2 catBoard2)
+ CreamDog(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)
- static MazeCell
+ MazeCell(int x, int y)
+ isUsable(int l, int j, Boolean[][] mark) : Boolean
+ getMove(int x, int y, int topX, int topY) : MoveType

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

MoveType
+ MoveType : enum
DogType
+ DogType : enum

Rule
<ul style="list-style-type: none"> - frame : JFrame - startButton : JButton - exitButton : JButton - label : JLabel - img : ImageIcon
+ Rule()

NextMap3
<ul style="list-style-type: none"> - frame : JFrame - startButton : JButton - exitButton : JButton - label : JLabel + vicImage : ImageIcon
+ NextLevel3()

NextMap2
<ul style="list-style-type: none"> - frame : JFrame - startButton : JButton - exitButton : JButton - label : JLabel + vicImage : ImageIcon
+ NextLevel2()

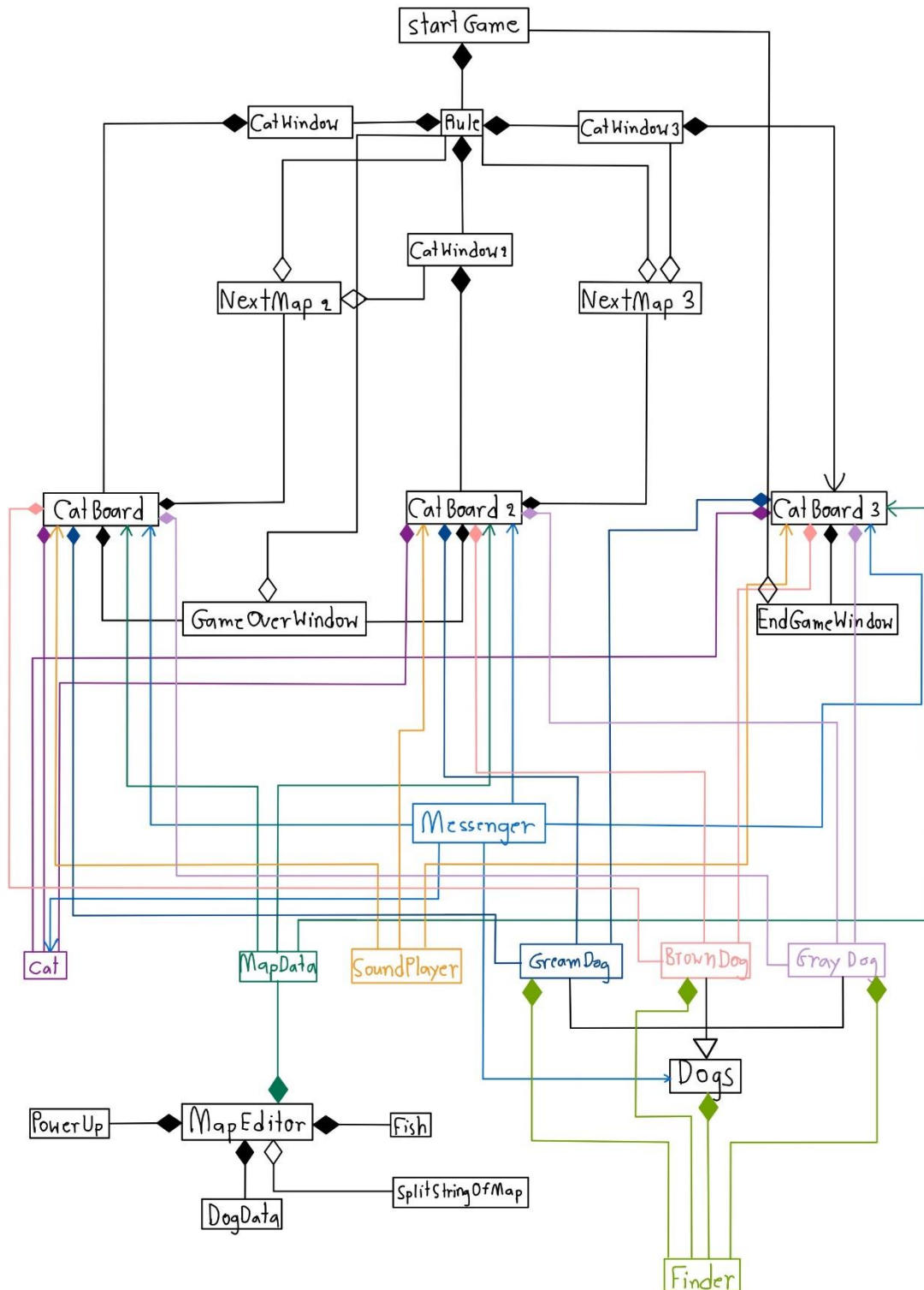
EndGameWindow
<ul style="list-style-type: none"> - frame : JFrame - exitButton : JButton - label : JLabel + vicImage : ImageIcon + failImage : ImageIcon
+ win() : void
+ lose() : void

Messenger
<ul style="list-style-type: none"> + UPDATE : int = AWTEvent.RESERVED_ID_MAX + 1 + COLTEST : int = AWTEvent.RESERVED_ID_MAX + 2

SoundPlayer
+ play(final URL name) : void

GameOverWindow
<ul style="list-style-type: none"> - frame : JFrame + failImage : ImageIcon
+ GameOverWindow()

Class relationships



เครื่องมือและเทคนิคที่ใช้

(Tool and Technique)



Visual Studio Code

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



Piskel

การออกแบบกราฟิก เราได้ใช้เว็บ Piskel ในการทำเนื่องจากใช้งานง่าย และได้รูปภาพตามที่ต้องการ



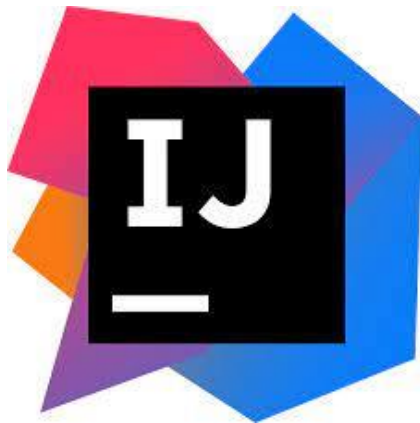
I love img

เราใช้เว็บ I love img ในการปรับขนาดรูปภาพ



PhotoScape X

เราใช้โปรแกรม PhotoScape X ในการตัดบางส่วนของภาพที่ไม่ใช้ออกไป



IntelliJ

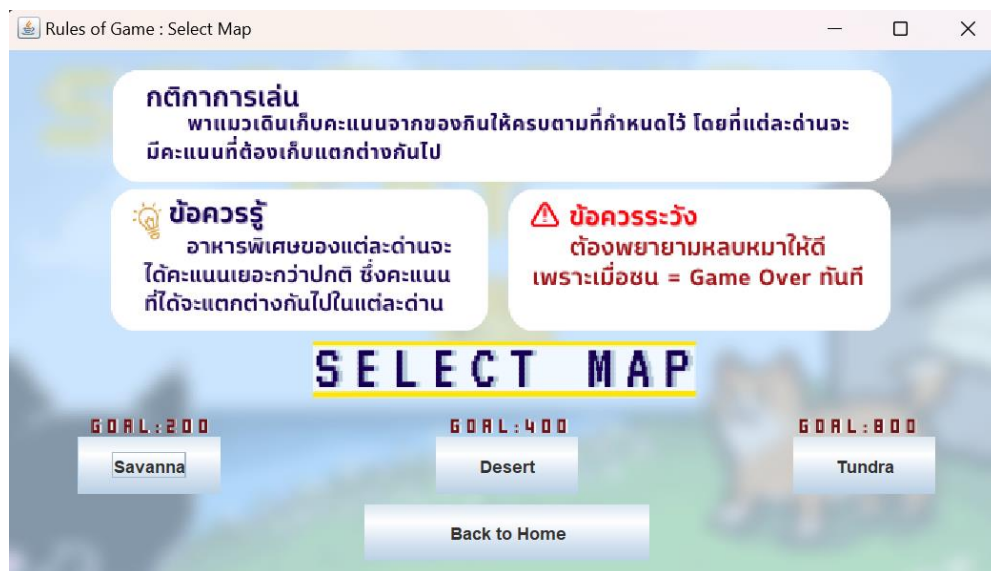
เราใช้โปรแกรม IntelliJ ในการ export ไฟล์

ภาพการทำงานของโปรแกรม

(Screen Capture)



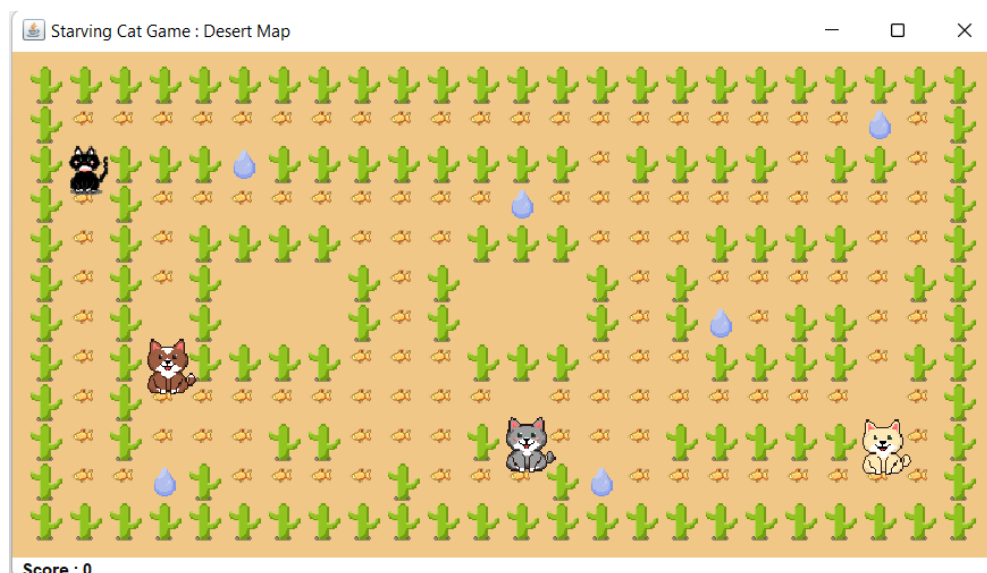
กดปุ่ม START GAME เพื่อเล่นเกม หรือปุ่ม EXIT เพื่อออกจากเกม



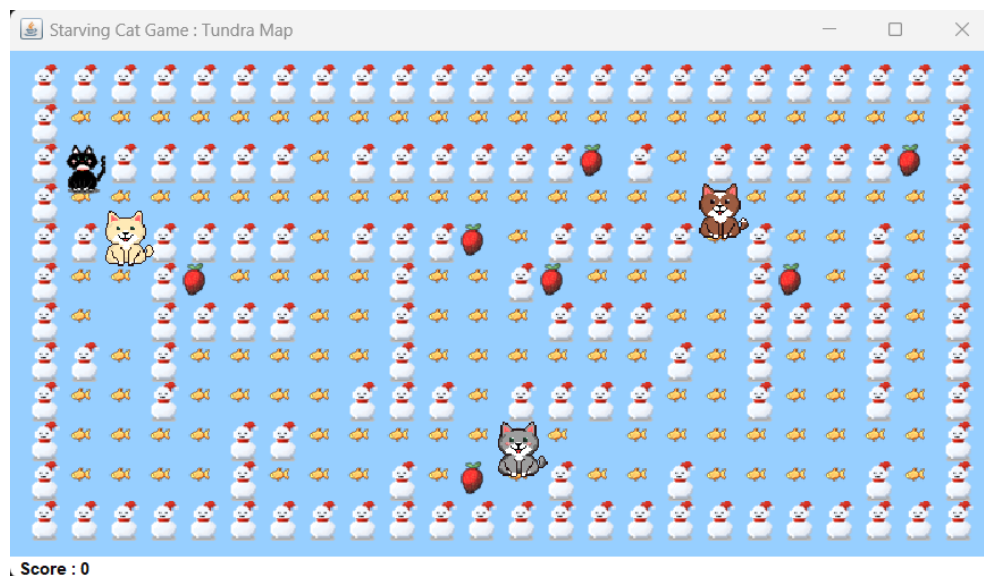
ให้ผู้เล่นอ่านกติกาการเล่น โดยผู้เล่นสามารถเลือก map ก่อนเล่นได้ 3 map นั่นคือ Savanna, Desert และ Tundra หากผู้เล่นต้องการกลับสู่หน้าจอหลักให้กดปุ่ม Back to Home



ตัวอย่างหน้าต่างเกม (Savanna Map) หากผู้เล่นเลือกด่านนี้ ผู้เล่น(Black cat)ต้องกดปุ่มลูกศรขึ้นลง ซ้ายขวา บนคีย์บอร์ดเพื่อเคลื่อนย้ายแมว และเก็บคะแนน(Fish)ให้ได้ครบ 200 คะแนน ตามที่กติกา ได้กำหนดไว้จึงจะชนะ หากผู้เล่นชนกับศัตรู(Dog)ถือว่าแพ้ในเกมนี้



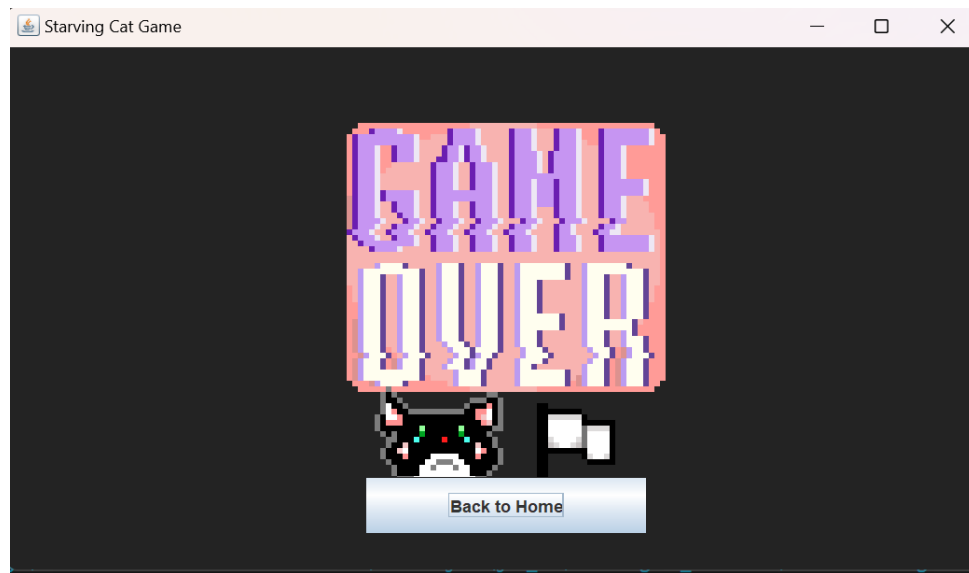
ตัวอย่างหน้าต่างเกม (Desert Map) หากผู้เล่นเลือกด่านนี้ ผู้เล่น(Black cat)ต้องกดปุ่มลูกศรขึ้นลง ซ้ายขวา บนคีย์บอร์ดเพื่อเคลื่อนย้ายแมว และเก็บคะแนน(Fish)ให้ได้ครบ 400 คะแนน ตามที่กติกา ได้กำหนดไว้จึงจะชนะ หากผู้เล่นชนกับศัตรู(Dog)ถือว่าแพ้ในเกมนี้



ตัวอย่างหน้าต่างเกม (เกมด่าน Tundra) หากผู้เล่นเลือกด่านนี้ ผู้เล่น(Black cat)ต้องกดปุ่มลูกศรขึ้นลงซ้ายขวา บนคีย์บอร์ดเพื่อเคลื่อนย้ายแมว และเก็บคะแนน(Fish)ให้ได้ครบ 800 คะแนน ตามที่กติกาได้กำหนดไว้จึงจะชนะ หากผู้เล่นชนกับศัตรู(Dog)ถือว่าแพ้ในเกมนั้น



ตัวอย่างหน้าต่างเกมหากชนะในด่าน Savanna, Desert และ Tundra หากผู้เล่นกดปุ่ม Back to Home จะกลับสู่หน้ากติกา และหากกดปุ่ม Next Map จะไปยัง map ถัดไป



ตัวอย่างหน้าต่างเกมหากแพ้ หากผู้เล่นกดปุ่ม Back to Home จะกลับสู่หน้ากติกา

การติดตั้งโปรแกรม

(Program Installation)

ขั้นตอนการติดตั้ง

1. ติดตั้งโปรแกรม JAVA รุ่นที่ติดตั้งเกมได้

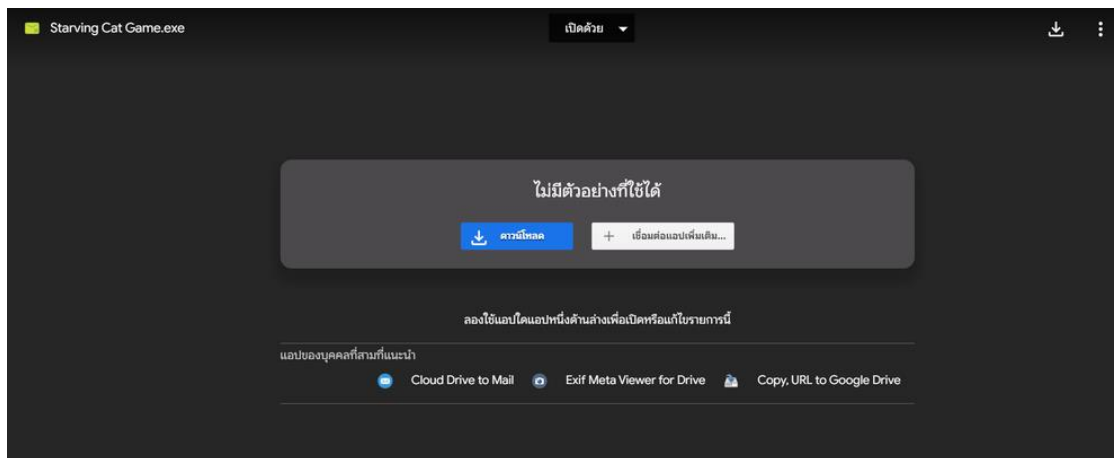
```
Command 'java' not found, but can be installed with:

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
```

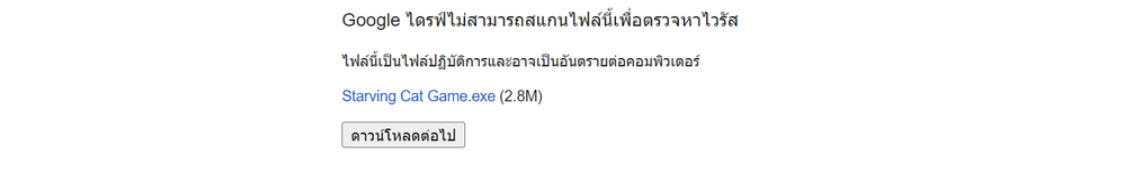
2. ดาวน์โหลดเกมโดยกดที่ลิงก์

https://drive.google.com/file/d/1UPPAXDPzI4wGjblGm8sZNNNBRvt16k1P/view?usp=share_link

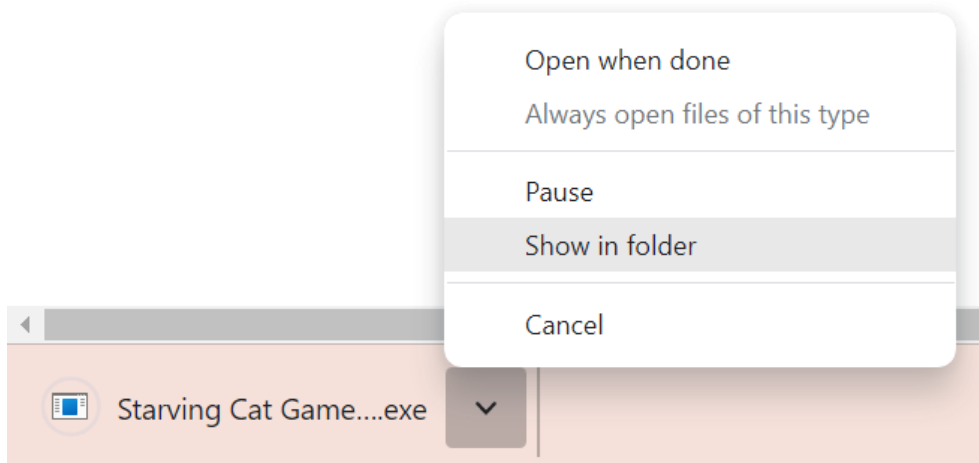
3. กดดาวน์โหลด



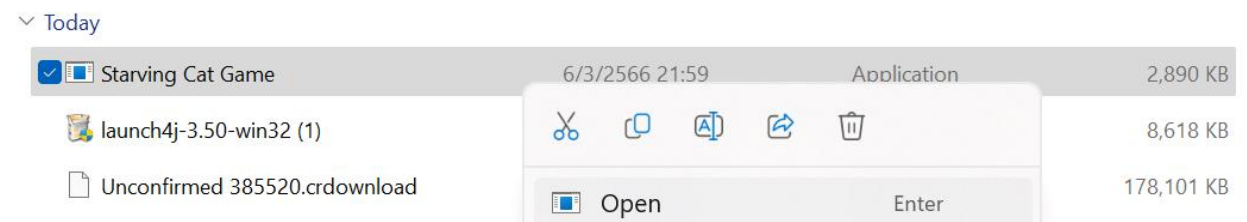
4. กดดาวน์โหลดต่อไป



5. กด Show in folder



6. คลิกขวาที่ Starving Cat Game.exe แล้วกด Open



แหล่งอ้างอิง

(Reference)

1. How to hide background of JButton

<https://stackoverflow.com/questions/8367500/how-to-hide-background-of-jbutton-which-containt-icon-image>

2. Pacman: A simple Java game

<https://github.com/arminkz/Pacman/blob/master/src/StartWindow.java>

3. Java Swing (GUI) | How to add a picture as the background to JFrame

<https://youtu.be/yGcYoz0s94E>

4. Pacman Game in Java

https://github.com/Gaspard/Pacman/tree/master?fbclid=IwAR0ybE9CbaWUWH05xNVREXl41z_n6DySu1LKyewluglj6Wf3x3zMmuDk_s0

5. Moving Character

<https://www.youtube.com/watch?v=4D3YIYPkit4>

6. Change size of image

<https://bigimage.11zon.com/th/image-resize/>

7. Piskel

<https://www.piskelapp.com/>

8. I love img

<https://www.iloveimg.com/th>

9. Closing JFrame with button click

https://stackoverflow.com/questions/2352727/closing-jframe-with-button-click?fbclid=IwAR0hyD7ztB_S24bU8pBv8QAnBP22887AU_2TXqTehwzu6ZPXbL2PO7WY5gE

10.Download Free Game Sound Effects

<https://mixkit.co/free-sound-effects/game/?page=3>

11.Audio Speed Changer

<https://mp3cut.net/th/change-speed>

12.Close window on pressing of certain key

<https://stackoverflow.com/questions/7000266/close-window-on-pressing-of-certain-key>

13.Just Fixed Launch4j this application requires a java runtime environment

<https://www.youtube.com/watch?v=Lexb3cpKPX4>

14.Export java project to exe file

<https://youtu.be/Zfkc-lepjuE>

15.How to Make a JAR File with Resources (Audio)

<https://www.youtube.com/watch?v=lgfcSr3NlG0>