Project to the moon

Master Chess

คณะผู้จัดทำ

นายคมสัน จันทร์หลวง 650510601  
นายรชต ธนัญชัย 650510638  
นายศิริเชษฐ์ แสงแก้ว 650510644  
นายอรรถกร คำสร้อย 650510691  
นายชัยภัทร ใจน่าน 650610606

อาจารย์ที่ปรึกษา

ผศ.ดร.เบญจมาศ ปัญญางาม  
ผศ.ดร.เมทินี เขียวกันยะ

โครงงานนี้เป็นส่วนหนึ่งของกระบวนวิชา 204211 การเขียนโปรแกรมเชิงวัตถุ  
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์  
มหาวิทยาลัยเชียงใหม่

สารบัญ

**หัวข้อ หน้า**

[บทที่ 1 บทนำ (Introduction) 1](#_Toc129014726)

[บทที่ 2 ปัญหาและการวิเคราะห์ปัญหา (Problem and Problem Analysis) 2](#_Toc129014727)

[โจทย์ปัญหา 2](#_Toc129014728)

[การวิเคราะห์โจทย์ 2](#_Toc129014729)

[ขั้นตอนการแก้ไขปัญหา 2](#_Toc129014730)

[บทที่ 3 การออกแบบคลาส (Class Design) 3](#_Toc129014731)

[รายละเอียดของคราส 3](#_Toc129014732)

[ความสัมพันธ์ของคลาส 14](#_Toc129014733)

[บทที่ 4 เครื่องมือและเทคนิค (Tool and Technique) 15](#_Toc129014734)

[โปรแกรมสำหรับการสร้างหรือปรับแต่งรูปภาพรูปภาพ 15](#_Toc129014735)

[โปรแกรมสำหรับการเขียนโปรแกรม 16](#_Toc129014736)

[บทที่ 5 ภาพการทำงานของโปรแกรม (Screen Capture) 18](#_Toc129014737)

[บทที่ 6 การติดตั้งโปรแกรม (Program Installation) 23](#_Toc129014738)

[การอ้างอิง (Reference) 24](#_Toc129014739)

# **บทที่ 1 บทนำ (Introduction)**

ชื่อกลุ่ม : Project to the moon  
ชื่อหัวข้อโครงงาน : Master Chess

จัดทำโดย

นายคมสัน จันทร์หลวง 650510601 SEC001  
นายรชต ธนัญชัย 650510638 SEC002  
นายศิริเชษฐ์ แสงแก้ว 650510644 SEC001  
นายอรรถกร คำสร้อย 650510691 SEC002  
นายชัยภัทร ใจน่าน 650610606 SEC001

# **บทที่ 2 ปัญหาและการวิเคราะห์ปัญหา (Problem and Problem Analysis)**

## โจทย์ปัญหา

ทางกลุ่มมีความสนใจในการพัฒนาเกม Board Game ซึ่งมีความชัดเจนในการเขียนโปรแกรมเชิงวัตถุหรือ Object Oriented Programming ทำให้เกิดแรงบันดาลใจในการเริ่มต้นการพัฒนาเกม Master Chess โดยมีต้นแบบจาก Board Game คือ Chess Game โดยใช้ความคิดและฝึกฝนความสามารถในด้านการวางแผน ทั้งนี้ทางกลุ่มได้นำมาปรับเปลี่ยนให้ Interface มีความทันสมัยและมีการเล่นมีง่ายมากยิ่งขึ้น พร้อมทั้งทางกลุ่มยังสามารถฝึกฝนการเขียนโปรแกรมเชิงวัตถุได้เป็นอย่างดีจากรูปแบบของเกมนี้

## การวิเคราะห์โจทย์

1. การพัฒนาเกมโดยประยุคใช้ความรู้ทางด้านการเขียนโปรแกรมเชิงวัตถุ  
2. การออกแบบเกมให้มีความน่าเล่นและทำให้ผู้เล่นสามารถเข้าใจได้ง่าย  
3. การต่อยอด Board Game ที่มีมานานให้เป็นเกมที่สามารถเล่นได้โดยไม่จำเป็นต้องมีอุปกรณ์อื่นๆ

## ขั้นตอนการแก้ไขปัญหา

1. เริ่มต้นวางแผนการทำงานออกแบบคลาสต่างๆ เบื้องต้น  
2. วิเคราะห์การทำงานในแต่งส่วนของโปรแกรม ทำรายการสิ่งที่ต้องทำ และทำการแจกจ่ายงาน  
3. แยกย้ายศึกษาข้อมูลต่างๆ เพิ่มเติม  
4. เริ่มการทำงานสร้างโปรเจคและจัดการเรื่องของการสื่อสาร (Git)  
5. ทำการทดสอบโปรแกรมและแก้ไขข้อผิดพลาดของตัวโปรแกรมที่เกิดขึ้น  
6. ทำการสรุปผล

# **บทที่ 3 การออกแบบคลาส (Class Design)**

## รายละเอียดของคราส

|  |
| --- |
| GameManager |
| + State:GameState  - \_round:int  - \_blackPieces:List<Piece>  - \_whitePieces:List<Piece>  - \_isEnd:Boolean |
| - UpdatePiecesLeft():void  + UpdateGameState(GameState):void  - HandleWin():void  - HandleBlackTurn():void  - HandleWhiteTurn():void  + ChangeTurn():void |

GameManager เป็นคลาสที่การออกคำสั่งต่างๆ ให้กับระบบส่วนกลาง เล่น การเปลี่ยน Turn การสั่งระบบประกาศการแพ้ชนะ การเลื่อนขั้นของ Pawn ฯลฯ

มี Attribute ดังนี้

1. State มีประเภทเป็น GameState เป็นค่าที่มีไว้สำหรับการสร้าง UI ต่างๆ ของระบบเกม
2. \_round มีประเภทเป็น Integer ใช้เพื่อนับจำนวนรอบที่ผ่านมา
3. \_blackPieces มีประเภทเป็น List ของ Piece ใช้เพื่อเก็บข้อมูลของตัวเบี้ยฝั่งดำ
4. \_whitePieces มีประเภทเป็น List ของ Piece ใช้เพื่อเก็บข้อมูลของตัวเบี้ยฝั่งขาว
5. \_isEnd มีประเภทเป็น Boolean เป็นค่าที่ไม่ไว้เพื่อหยุดเกม

มี Method ดังนี้

1. UpdatePiecesLeft เป็น Method ไม่มีการคืนค่า (Void) เป็น Method ที่มีไว้เลื่อนขึ้นขัน Pawn ที่ได้ไปถึงแนวหลังของอีกฝั่งได้ จะเรียกใช้ Interface เลื่อนตัวที่จะเลื่อนขึ้นให้ของฝั่งนั้นๆ
2. Awake, Start เป็น Method ที่ไม่การคืนค่า (Void) เป็น Method ที่มีใน Unity เพื่อเรียกใช้
3. UpdateGameState เป็น Method ที่ไม่มีการคืนค่า (Void) รับค่าต่างๆ ที่หลังจากจบรอบก่อนหน้า เช่น จำนวนและตำแหน่งของเบี้ยที่เปลี่ยนไปจากรอบที่แล้ว
4. HandleWin เป็น Method ที่ไม่มีการคืนค่า (Void) ใช้ในการประกาศผู้ชนะ
5. HandleBlackTurn เป็น Method ที่ไม่มีการคืนค่า (Void) ใช้จัดการเมื่อเข้าสู่รอบของฝั่งดำ
6. HandleWhiteTurn เป็น Method ที่ไม่มีการคืนค่า (Void) ใช้จัดการเมื่อเข้าสู่รอบของฝั่งขาว
7. ChangeTurn เป็น Method ที่ไม่มีการคืนค่า (Void) ใช้เปลี่ยนรอบการเล่นหลังจากที่ Action ต่างๆ ได้จบลง

|  |
| --- |
| Tile |
| - WhiteMoveOccupiedTile:boolean  - BlackMoveOccupiedTile:boolean  - UnreachableTile:Boolean |
| - SetPiece(Piece):void  - MovePiece(Fiction):boolean |

Tile เป็นคราส ที่สร้างช่องขึ้นมา และมี Method เกี่ยวกับการจัดการ Piece บนตัวของมันเอง

มี Attribute ดังนี้

1. WhiteMoveOccupiedTile, BlackMoveOccupiedTile มีประเภทเป็น Method เป็น Attribute ที่เก็บ Method ไว้เพื่อคืนค่า Boolean เพื่อตรวจสอบว่าตัวเองมี Object อยู่หรือไม่ หากมีเป็นมิตรหรือศัตรู สามารถเดินได้หรือไม่
2. UnreachableTile มีประเภทเป็น Boolean เพื่อนำไปตรวจว่าตัวเองสามารถถูกเข้าถึงได้หรือเปล่า

มี Method ดังนี้

1. SetPiece เป็น Method ที่ไม่มีการคืนค่า (void) รับค่า Piece ที่จะมันอยู่บนช่องนี้
2. MovePiece เป็น Method มีการคืนค่าเป็น Boolean และรับค่าเป็นตำแหน่งที่กำลังจะย้าย เพื่อย้าย Piece ที่อยู่บนช่องออกจากช่อง

|  |
| --- |
| TileManager |
| + DictTiles:Dcitionary<Vector2, Tile> {readonly}  - Width:int {const}  - Hight:int {const} |
| + GenerateTile():void  + GetTile(Vector2 pos):Tile  + Tiles():Dictionary |

TileManager เป็นคลาสที่มีหน้าที่ในการตรวจจับการเปลี่ยนแปลงต่างๆ ของแต่ละช่องและทำการส่งข้อมูลต่างๆ หรือสื่งที่ต้องทำให้กับ GameManager หรือ PiecesManager ไปสั่งระบบย่อยต่อไป

มี Attribute ดังนี้

1. DictTiles มีประเภทเป็น Dictionary ของ Vector2 และ Tile เป็น Attribute ที่ใช้สำหรับอ่านเพียงอย่างเดียว
2. Width มีประเภทเป็น Integer ซึ่งเป็นค่าคงที่ ไม่สามารถเปลี่ยนแปลงได้ เป็นตัวปรับขนาดของสนามการเล่น
3. Height มีประเภทเป็น Integer ซึ่งเป็นค่าคงที่ ไม่สามารถเปลี่ยนแปลงได้ เป็นตัวปรับขาดของสนามการเล่น

มี Method ดังนี้

1. GenerateTile เป็น Method ที่ไม่มีการคืนค่า (Void) สร้างช่องในการเดิน
2. GetTile เป็น Method ที่มีการคืนค่าเป็นคลาสของแต่ละช่อง (Tile) รับค่าตำแหน่งเขามาเพื่อหาช่องนั้นๆ
3. Tiles เป็น Method ที่มีการคืนค่าเป็น Dictionary ของ Tile ใช้เพื่อรับข้อมูลช่องทั้งหมด

|  |
| --- |
| PieceManager |
| - whitePawn:Piece  - whiteRook:Piece  - whiteKnight:Piece  - whiteBishop:Piece  - whiteQueen:Piece  - whiteKing:Piece  - blackPawn:Piece  - blackRook:Piece  - blackKnight:Piece  - blackBishop:Piece  - blackQueen:Piece  - blackKing:Piece  - \_list:List<Piece> |
| + SpawnWhitePieces():void  + SpawnBlackPieces():void  + CalWhitePiecesLeft():List<Piece>  + CalBlackPiecesLeft():List<Piece>  + SetSelectedPiece(Piece):void  - SpawnPiece(Vector2, Piece):void |

PieceManager เป็นคราสสำหรับจัดการ Piece ทุกประเภท เช่น การเรียกใช้เดิน การเลือก Piece การคำนวนระยะทางต่างๆ ของ Piece แต่ละตัวจะถูกเก็บไว้ใน PieceManager ทั้งหมด

มี Attribute ดังนี้

1. whitePawn มีประเภทเป็น Piece เก็บข้อมูลของ Piece ที่เป็น Pawn ของฝั่งสีขาว
2. whiteRook มีประเภทเป็น Piece เก็บข้อมูลของ Piece ที่เป็น Rook ของฝั่งสีขาว
3. whiteKnight มีประเภทเป็น Piece เก็บข้อมูลของ Piece ที่เป็น Knight ของฝั่งสีขาว
4. whiteBishop มีประเภทเป็น Piece เก็บข้อมูลของ Piece ที่เป็น Bishop ของฝั่งสีขาว
5. whiteQueen มีประเภทเป็น Piece เก็บข้อมูลของ Piece ที่เป็น Queen ของฝั่งสีขาว
6. whiteKing มีประเภทเป็น Piece เก็บข้อมูลของ Piece ที่เป็น King ของฝั่งสีขาว
7. blackPawn มีประเภทเป็น Piece เก็บข้อมูลของ Piece ที่เป็น Pawn ของฝั่งสีดำ
8. blackRook มีประเภทเป็น Piece เก็บข้อมูลของ Piece ที่เป็น Rook ของฝั่งสีดำ
9. blacknight มีประเภทเป็น Piece เก็บข้อมูลของ Piece ที่เป็น Knight ของฝั่งสีดำ
10. blackBishop มีประเภทเป็น Piece เก็บข้อมูลของ Piece ที่เป็น Bishop ของฝั่งสีดำ
11. blackQueen มีประเภทเป็น Piece เก็บข้อมูลของ Piece ที่เป็น Queen ของฝั่งสีดำ
12. blackKing มีประเภทเป็น Piece เก็บข้อมูลของ Piece ที่เป็น King ของฝั่งสีดำ
13. \_list มีประเภทเป็น List ของ Piece เก็บข้อมูล Piece ของทั้ง 2 ฝั่ง

มี Method ดังนี้

1. SpawnWhitePiece เป็น Method ที่ไม่การคืนค่า (Void) ใช้เพื่อสร้าง Piece ฝั่งสีขาว
2. SpawnBlackPiece เป็น Method ที่ไม่การคืนค่า (Void) ใช้เพื่อสร้าง Piece ฝั่งสีขาว
3. CalWhitePiecesLeft เป็น Method ที่มีการคืนค่า List ของ Piece สีขาว ใช้ในการคำนวณหาจำนวน Piece ฝั่งสีขาวที่เหลืออยู่ ณ ปัจจุบัน
4. CalBlackPiecesLeft เป็น Method ที่มีการคืนค่า List ของ Piece สีดำ ใช้ในการคำนวณหาจำนวน Piece ฝั่งสีดำที่เหลืออยู่ ณ ปัจจุบัน
5. SetSelectPiece เป็น Method ที่ไม่มีการคืนค่า แต่จะรับค่าของ Piece ที่เราต้องการเลือก
6. SpawnPiece เป็น Method ที่ไม่มีการคืนค่า (Void) รับค่าตำแหน่งที่จะสร้าง pos มีประเภทเป็น Vector2 และ Piece ที่จะสร้าง

|  |
| --- |
| PieceManager |
|  |
| + ShowSelectedPiece(Piece):void  + ShowTileInfo(Tile):void  + SelectWhitePlayer():void  + SelectBlackPlayer():void |

MenuManager เป็นคราสที่ทำการแสดงข้อมูลของแต่ละ Piece ไว้ทางด้านขวาของหน้าจอ เพื่อบอก คุณสมบัติ ตำแหน่ง ความสามารถ ให้กับผู้เล่นให้ได้รับรู้

มี Method ดังนี้

1. ShowSelectedPiece เป็น Method ที่ไม่มีการคืนค่า (Void) รับค่า Piece มาเพื่อแสดงข้อมูลของ Piece ที่เลือก
2. ShowTileInfo เป็น Method ที่ไม่มีการคืนค่า (Void) รับค่าของช่อง (Tile) มาเพื่อแสดงข้อมูลภายในช่องๆ นั้น เช่น ตำแหน่ง ฯลฯ
3. SelectWhitePlayer, SelectBlackPlayer เป็น Method ที่ไม่มีการคืนค่า (Void) ใช้เพื่อแสดงข้อมูลของผู้เล่น หรือประกาศว่าเป็นรอบของฝั่งไหน

|  |
| --- |
| PawnPromotionManager {abstract} |
| - queenPrefab:Piece  - rookPrefab:Piece  - bishopPrefab:Piece  - knightPrefab:Piece  - tempPrefab:Piece |
| + TimeToPromotion(Piece):void  + SetSelectPromotionImg(Piece):void  # PromotionPieceOnTile(Piece):void  # SpawnTempPiece(Piece, Piece):void |

PawnPromotionManager เป็นคราส Abstract โดยเป็นคราส Parent ของ BlackPawn  
PromotionManger และ WhitePawnPromotionManager เป็นคราสที่มีหน้าที่ในการให้ผู้เล่นเลือกการเลื่อนขั้นของ Pawn ที่สามารถบุกเข้าไปถึงแนวหลังของฝั่งตรงข้ามได้

มี Attribute ดังนี้

1. queenPrefab, rookPrefab, bishopFrefab, knightPrefab มีประเภทเป็น Piece สำหรับใช้ในการกำหนดประเภทต่างๆ ของ Pawn ที่กำลังเลื่อนขั้น
2. tempPiece มีประเภทเป็น Piece สำหรับในการแทนที่ชั่วคราวระหว่างรอการเลื่อนขั้น

มี Method ดังนี้

1. TimeToPromotion เป็น Method ที่ไม่มีการคืนค่า รับค่า Piece ที่เลือก เพื่อเปลี่ยน Pawn ที่ได้บุกเข้าไปในแนวหลังของฝั่งตรงข้าม
2. SetSelectPromotionImg เป็น Method ทีไม่มีการคืนค่า รับค่า Piece ที่เลือก เพื่อเปลี่ยนลักษณะ (รูปภาพ) ให้เป็นไปตามขั้นที่เราเลือก
3. PromotionPieceOnTile เป็น Method ที่ไม่มีการคืนค่า รับค่า Piece ที่เลือก โดย Method นี้เป็น Method ที่บังคับให้สร้างไว้ในคราส Child ของตัวเอง (Abstract Method) เพื่อทำการเลื่อนขั้นให้ถูกกับสี
4. SpawnTempPiece เป็น Method ที่ไม่มีการคืนค่า รับค่า Piece ที่เลือก โดย Method นี้เป็น Method ที่บังคับให้สร้างไว้มนคราส Child ของตนเอง (Abstract Method) เพื่อแสดง Piece ชั่วคราวตามสีของฝ่ายตัวเอง

|  |  |
| --- | --- |
| BlackPawnPromotionManager | WhitePawnPromotionManager |
|  |  |
| + PromotionPieceOnTile(Piece):void {override}  + SpawnTempPiece(Piece, Piece):void {override} | + PromotionPieceOnTile(Piece):void {override}  + SpawnTempPiece(Piece, Piece):void {override} |

BlackPawnPromotionManager, WhitePawnPromotionManager เป็นคราสที่มีความคล้ายคลึงกัน โดยคราสทั้งสองคราสเป็นคราสที่ใช้สำหรับการเลื่อนขั้น Pawn ที่บุกเข้าไปในแนวหลังของฝั่งตรงข้ามได้ จะแบ่งออกเป็น 2 สี

มี Method ดังนี้

1. PromotionPieceOnTile เป็น Method ที่ไม่การคืนค่า รับค่าเป็น Piece ที่เลือก โดย Method นี้เป็น Method ที่ถูกบังคับให้สร้างจากคราส Parent มีหน้าที่ในการสร้างตัวขึ้นมาทับ Pawn ที่ได้เข้าถึงแนวหลัง
2. SpawnTempPiece เป็น Method ที่ไม่มีการคืนค่า รับค่าเป็น Piece ชั่วคราวระหว่างรอการเลื่อนขั้น โดย Method นี้เป็น Method ที่ถูกบังคับให้สร้างจากคราส Parent มีหน้าที่สร้าง Piece ชั่วคราวระหว่างรอการเลื่อนขั้น

|  |
| --- |
| Piece {abstract} |
| - occupiedTile:Tile  + faction: Faction  + roll:Roll  + isFirstMove:boolean  +pos:Vector2  + CurrentPieceMove:List<Vector2> {static}  + AttackMove:List<Vector2> {static} |
| + CalculateLegalMove(Piece):void  - CurrentLegalMove(IEnumerable<Vector2>, Piece):Vector2[]  - CurrentLegalMove(IEnumerable<Vector2>, Faction):Vector2[]  - CurrentLegalMove(IEnumerable<Vector2[]>, Fiction):Vector2[]  - RookWalk(float, float. Fiction):Vector2[]  - BishopWalk(float, float. Fiction):Vector2[]  - KnightWalk(float, float. Fiction):Vector2[]  - KingWalk(float, float. Fiction):Vector2[]  - PawnWalk(float, float. Fiction):Vector2[]  - ShowLegalMove(IEnumerable<Vector2>):void  - ShowHighlight(Vector2):Vector2  + CheckPawnPromotion():void {abstract}  + PromotionPawn(Piece):void {abstract} |

Piece เป็นคราว Abstract โดยเป็นคราส Parent ของ WhitePieces และ BlackPieces มีหน้าที่สืบทอดความสามารถการเดินของ Piece แต่ละประเภทไว้ พร้อมทั้งการตรวจสอบเพื่อสร้างวิถีการเดินของ Piece แต่ละตัว

มี Attribute ดังนี้

1. occupiedTile มีประเภทเป็น Tile เก็บค่าของช่องที่ตัวเองอยู่ ณ ปัจจุบัน
2. faction มีประเภทเป็น Faction เพื่อระบุฝั่งของตัว Piece
3. roll มีประเภทเป็น Roll ระบุประเภทของ Piece
4. isFirstMove มีประเภทเป็น Boolean เพื่อใช้กับ Pawn ที่มีการเดินครั้งแรกไม่เหมือนกับครั้งต่อไป
5. pos มีประเภทเป็น Vector2 เก็บค่าตำแหน่งปัจจุบัน
6. CurrentPieceMove มีประเภทเป็น List ของ Vector2 เก็บค่าการเดินที่เป็นไปได้ของตัวเองทั้งหมดไว้
7. AttackMove มีประเภทเป็น List ของ Vector2 เก็บค่าการเดินที่สามารถโจมตีฝั่งตรงข้ามได้

มี Method ดังนี้

1. CalculateLegalMove เป็น Method ที่ไม่มีการคืนค่า (Void) รับค่าเป็นประเภทของ Piece ของตัวเอง เพื่อคำนวณหาเส้นทางในการเดิน
2. CurrentLegalMove เป็น Overload Method ที่มีการคืนค่า Array ของ Vector2 หรือ Vector2 ที่ตัวนั้นสามารถเดินได้ โดยมีการรับค่าตาม Fiction ของตัวเองและสร้างเส้นทางที่สามารถเดินไปได้ในตอนนี้ โดยจะถูกเรียกใช้ที่ CalculateLegalMove
3. RookWalk, BishopWalk, KnightWalk, KingWalk, PawnWalk เป็น Method ที่มีการคืนค่าตำแหน่งเส้นทางของ Rook ที่สามารถเดินได้ (Vector2) และรับค่าเป็นตำแหน่ง x, y และฝ่ายของตัวนั้นเพื่อสร้างเส้นทางในการเดินได้ถูกต้อง
4. ShowLegalMove เป็น Method ที่ไม่มีการคืนค่า (Void) โดยรับค่าเป็น IEnumerable ของ Vector2 ถูกเรียกใช้เพื่อให้เรียกใช้ ShowHighLight ต่อ
5. ShowHighLight เป็น Method ที่มีการคืนค่าของตำแหน่งที่สามารถเดินได้ โดยเปลี่ยนสีพื้นเป็นสีเขียว โดยรับค่าตำแหน่งมา (Vector2)
6. CheckPawnPromotion เป็น Method ที่บังคับให้คราส Child สร้าง Method นี้ด้วย (Abstract Method) และไม่มีการคืนค่า (Void) โดยจะตรวจสอบ Pawn ที่ได้เข้าไปในแนวหลังของฝั่งตรงข้าม
7. PromotionPawn เป็น Method ที่บังคับให้คราส Child สร้าง Method นี้ด้วย (Abstract Method) และไม่มีการคืนค่า (Void) โดยเป็น Method สำหรับการเลื่อนขั้น Pawn

|  |  |
| --- | --- |
| WhitePieces | BlackPieces |
| - PromotionYPos:float {const} | - PromotionYPos:float {const} |
| + CheckPawnPromotion():void {override}  + PromotionPawn(Piece):void {override} | + CheckPawnPromotion():void {override}  + PromotionPawn(Piece):void {override} |

WhitePieces และ BlackPieces เป็นคราส Child ที่สืบทอดมาจาก Pieces โดยจะเป็นคราสที่ใช้ในการเลื่อนขั้น Pawn

มี Attribute ดังนี้

1. PromotionYPos มีประเภทเป็น float และเป็นตัวแปรคงที่

มี Method ดังนี้

1. CheckPawnPomotion เป็น Method ที่ถูกบังคับให้สร้างจากคราส Parent ไม่มีการคืนค่า (Void) มีหน้าที่ตรวจสอบการเลื่อนขั้น
2. PromotionPawn เป็น Method ที่ถูกบังคับให้สร้างจากคราส Parent ไม่มีการคืนค่า (Void) รับค่า Piece ตัวปัจจุบัน เพื่อเลื่อนขั้นต่อไป

|  |  |
| --- | --- |
| WhiteTeam | BlackTeam |
| - KingPos:Vetor2 | - KingPos:Vetor2 |
| + FindKing():void | + FindKing():void |

WhiteTeam, BlackTeam เป็นคราสที่มีไว้เก็บข้อมูลของแต่ละฝั่งว่า King ตอนนี้อยู่ ณ ตำแหน่งใด เพื่อทำการตรวจสอบว่า King กำลังอยู๋ในอันตราย

มี Attribute ดังนี้

1. KingPos มีประเภทเป็น Vector2 เก็บตำแหน่ง King ของฝั่งตัวเอง

มี Method ดังนี้

1. FindKing เป็น Method ที่ไม่มีการคืนค่า ใช้สำหรับค้นหาตัวหน่อยของ King ในปัจจุบันของแต่ละฝั่ง

## ความสัมพันธ์ของคลาส

Tile

PawnPromotionManager

BlackPawnPromotionManager

WhitePawnPromotionManager

WhiteTeam

BlackTeam

WhitePieces

BlackPieces

Pieces

TileManager

PieceManager

Master Chess

GameManager

# **บทที่ 4 เครื่องมือและเทคนิค (Tool and Technique)**

## โปรแกรมสำหรับการสร้างหรือปรับแต่งรูปภาพรูปภาพ

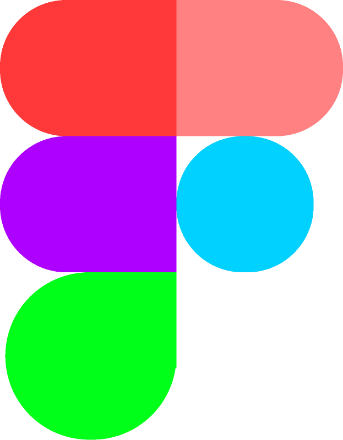
1. Procreate

แอปพลิเคชั่นเครื่องมือสำหรับสร้างหรือวาดรูปภาพเพื่อนำมาใช้ในบอร์ดเกม Master Chess เช่น ตัวเบี้ยต่างๆ ตัวกระดาน ตัวปุ่มต่างๆ ในโปรแกรมส่วนใหญ่ถูกวาดมาจากโปรแกรม Procreate



1. Figma

แอปพลิเคชั่นสำหรับการปรับขนาดรูปภาพและตัดพื้นหลังรูปภาพที่ต้องการ รวมถึงการปรับสีของรูปภาพ



## โปรแกรมสำหรับการเขียนโปรแกรม

1. Unity

ซอฟต์แวร์ที่ใช้ในการเขียนตัวเกม Master Chess ขึ้นมา โดยเป็นซอร์ฟแวร์สำแบบข้ามแพลตฟอร์ม ใช้เพื่อสำหรับการพัฒนาซอฟต์แวร์และการจำลองต่างๆ โดยเป็น Framework ของภาษา C# โดยมีการประยุคใช้ การเขียนโปรแกรมเชิงวัตถุ หรือ Object Oriented Programming โดยจะความคล้ายกับตัวภาษา Java ในหลายๆ ส่วน พร้อมมี Library อำนวยความสะดวกสำหรับการพัฒนาเกม



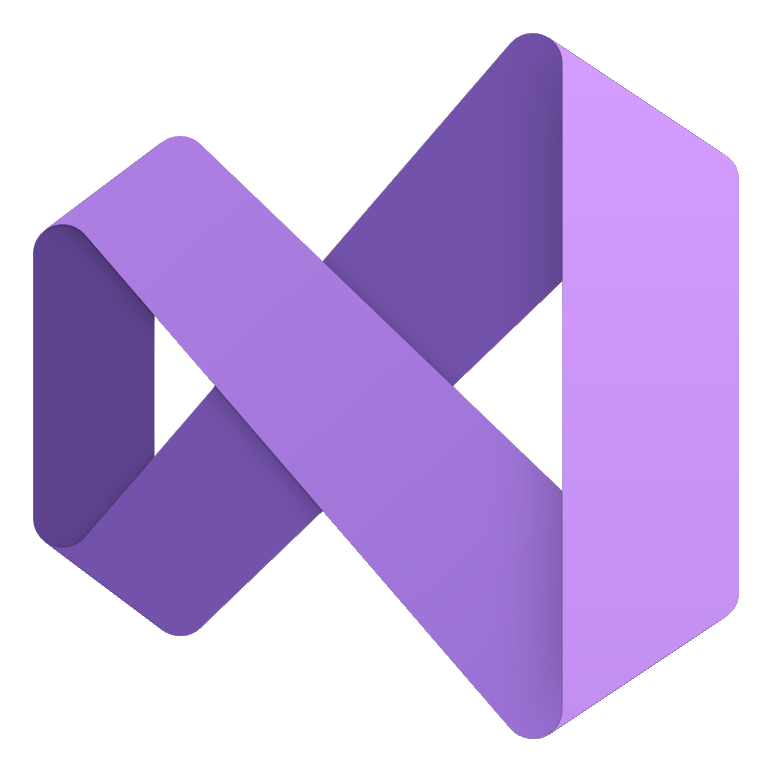
2. Jetbrain Rider

JetBrains Rider เป็น .NET IDE ข้ามแพลตฟอร์มที่ใช้แพลตฟอร์ม IntelliJ และ ReSharper มีความสามารถในการิเคราะห์ตัวโปรแกรมแจ้งเตือนเกี่ยวกับโปรแกรมที่ใช้ไม่ถูกวิธีหรือเกิดข้อผิดพลาด รวมถึงการทดสอบอย่างง่าย

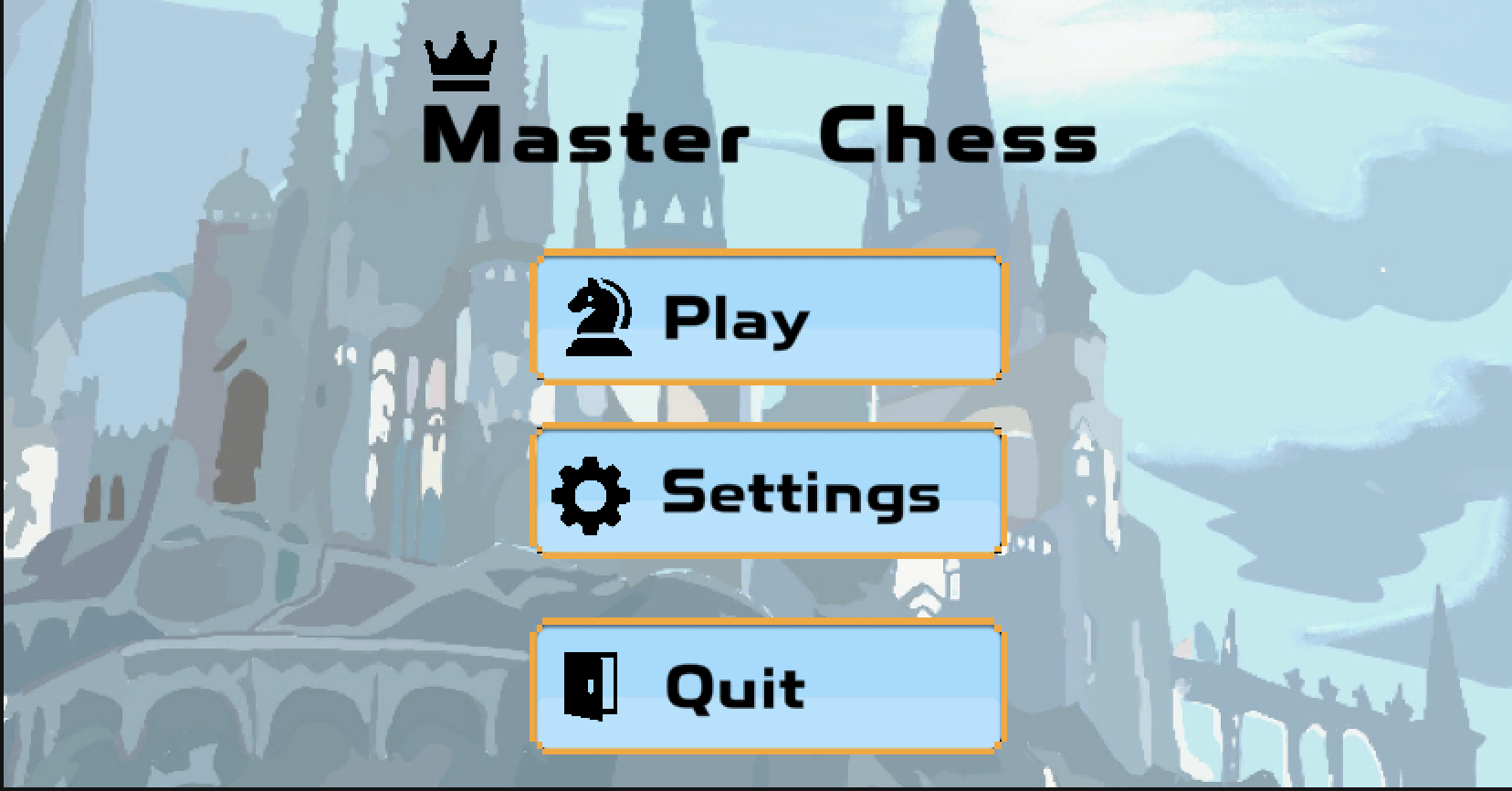


3. Visual Code Studio

Visual Studio IDE เป็น Launching Pad ที่สร้างสรรค์ซึ่งคุณสามารถใช้แก้ไข ดีบัก และสร้างโค้ด จากนั้นจึงเผยแพร่แอป ตัวแก้ไขมาตรฐานและตัวดีบักเกอร์ที่ IDE ส่วนใหญ่มีให้ Visual Studio มีคอมไพเลอร์ เครื่องมือเติมโค้ด ตัวออกแบบกราฟิก และคุณสมบัติอื่นๆ อีกมากมายเพื่อปรับปรุงกระบวนการพัฒนาซอฟต์แวร์



# **บทที่ 5 ภาพการทำงานของโปรแกรม (Screen Capture)**



ภาพที่ หน้าต่าง Menu



ภาพที่ หน้าต่างตั้งค่า



ภาพที่ หน้าต่างยืนยันการเล่นเกม

Graphical user interface

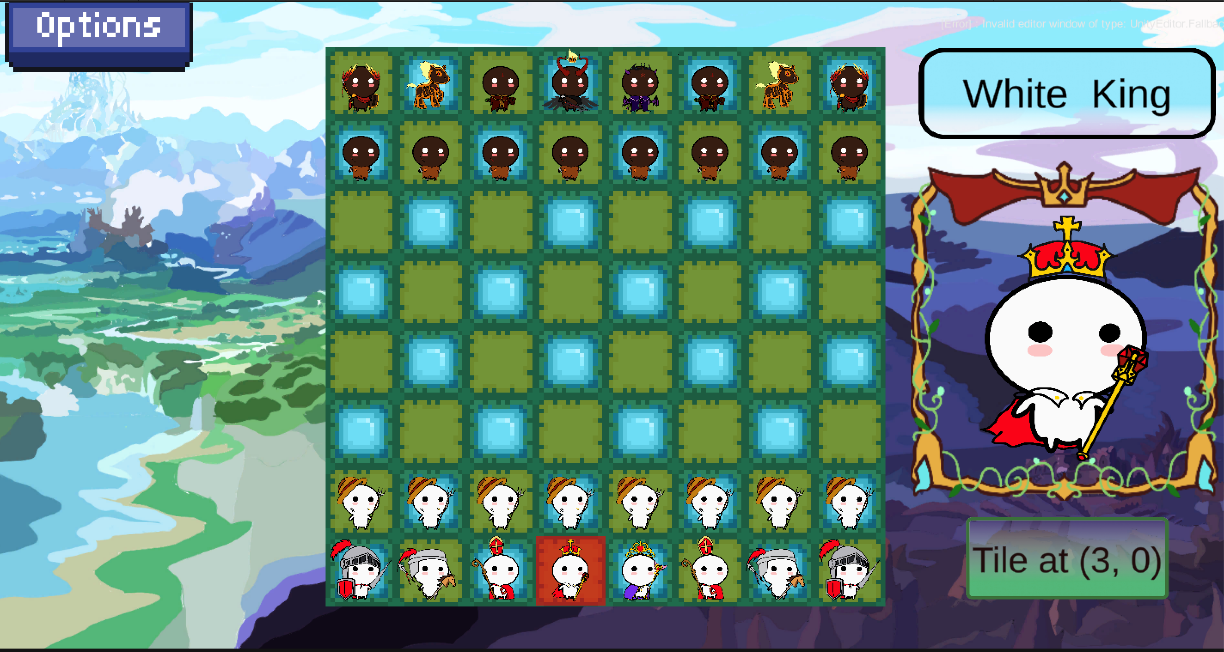
Description automatically generated

ภาพที่ การเลือกฝ่ายเริ่มก่อน

A screenshot of a computer

Description automatically generated with medium confidence

ภาพที่ ขณะเล่นเกม

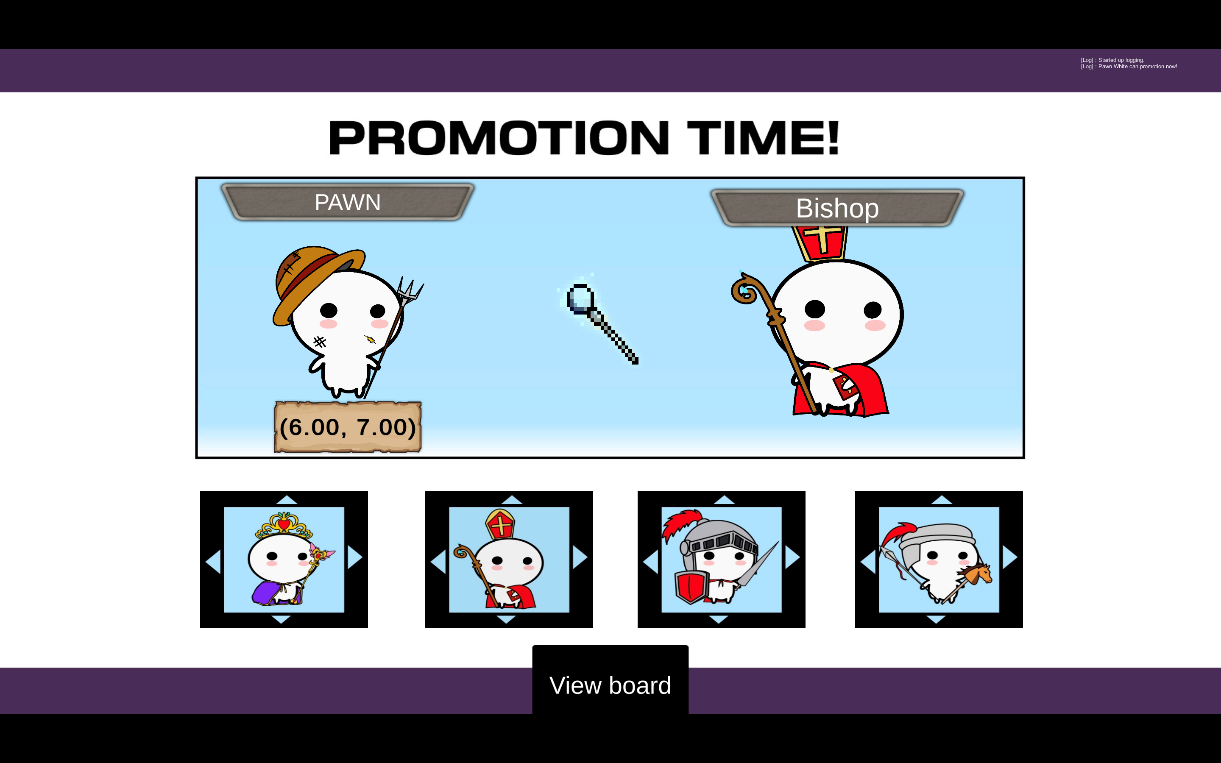


ภาพที่ ขณะเล่นเกม

Graphical user interface, application

Description automatically generated

ภาพที่ ขณะเล่นเกม



ภาพที่ Pawn เลื่อนขั้น



ภาพที่ ชนะเกม

# **บทที่ 6 การติดตั้งโปรแกรม (Program Installation)**

ใช้ QR Code นี้เพื่อเข้าสู่เว็บไชต์ของเกม

Qr code

Description automatically generated

# **การอ้างอิง (Reference)**

Brackeys. (2560). BUILD LEVELS QUICKLY in Unity with SNAPS!. ค้นเมื่อ 8 กุมภาพันธ์ 2566. จาก https://www.youtube.com/watch?v=b4oqOdBCy3c.

Microsoft. (2566). await operator - asynchronously await for a task to complete. สืบค้นเมื่อ 12 กุมภาพันธ์ 2566, จาก https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/operators/await.

Microsoft. (2566). Lambda expression (=>) operator defines a lambda expression. สืบค้นเมื่อ 12 กุมภาพันธ์ 2566, จาก https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/operators/lambda-operator.

Tarodev. (2563). Create a grid in Unity - Perfect for tactics or turn-based games! Part 2. สืบค้นเมื่อ 10 กุมภาพันธ์ 2566. จาก https://www.youtube.com/watch?v=f5pm29yhaTs.

Tarodev. (2565). Game Manager - Controlling the flow of your game [Unity Tutorial]. สืบค้นเมื่อ 11 กุมภาพันธ์2566. จาก https://www.youtube.com/watch?v=4I0vonyqMi8.

Unity. Unity User Manual 2021.3 (LTS). สืบค้นเมื่อ 6 มีนาคม 2566, จาก https://www.youtube.com  
/watch?v=b4oqOdBCy3c