

PROJECT TO THE MOTH



PLAY



CONTRIBUTORS



CHAIWAT UAINAN
COOKING
EVERYTHING KID.



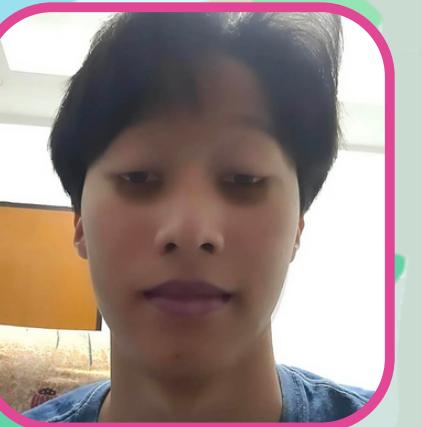
RACHATA THANANEHAK
EVERYTHING KID



AUTTAKORN CAMSOI
DOC KID



SIRICHET SENGKEAW
DRAWING KID



KONGSAN JANLUANG
UI KID

PROBLEM

6

PROBLEM ANALYSIS

โจทย์

ทางกลุ่มมีความสนใจในการพัฒนาเกม BOARD GAME ซึ่งมีความชัดเจนในการเขียนโปรแกรมเชิงวัตถุหรือ OBJECT ORIENTED PROGRAMMING ทำให้เกิดแรงบันดาลใจในการเริ่มต้นพัฒนาเกม MASTER CHESS โดยมีต้นแบบจาก BOARD GAME คือ CHESS GAME โดยใช้ความคิดและได้ฝึกฝนความสามารถในการวางแผนที่นี้ทางกลุ่มได้นำมาปรับเปลี่ยนให้ INTERFACE มีความทันสมัยและมีการเล่นที่ง่ายมากขึ้น พร้อมที่ทางกลุ่มยังได้ประสบการณ์การเขียนโปรแกรมเชิงวัตถุได้เป็นอย่างดีจากรูปแบบของเกมนี้

การวิเคราะห์โจทย์

- 1. การพัฒนาเกมโดยประยุกใช้ความรู้ทางด้านการเขียนโปรแกรมเชิงวัตถุ
- 2. การออกแบบเกมให้มีความน่าเล่นและทำให้ผู้เล่นสามารถเข้าใจได้ง่าย
- 3. การต่อยอด BOARD GAME ที่มีนานาให้เป็นเกมที่สามารถเล่นได้ทุกที่ทุกเวลา

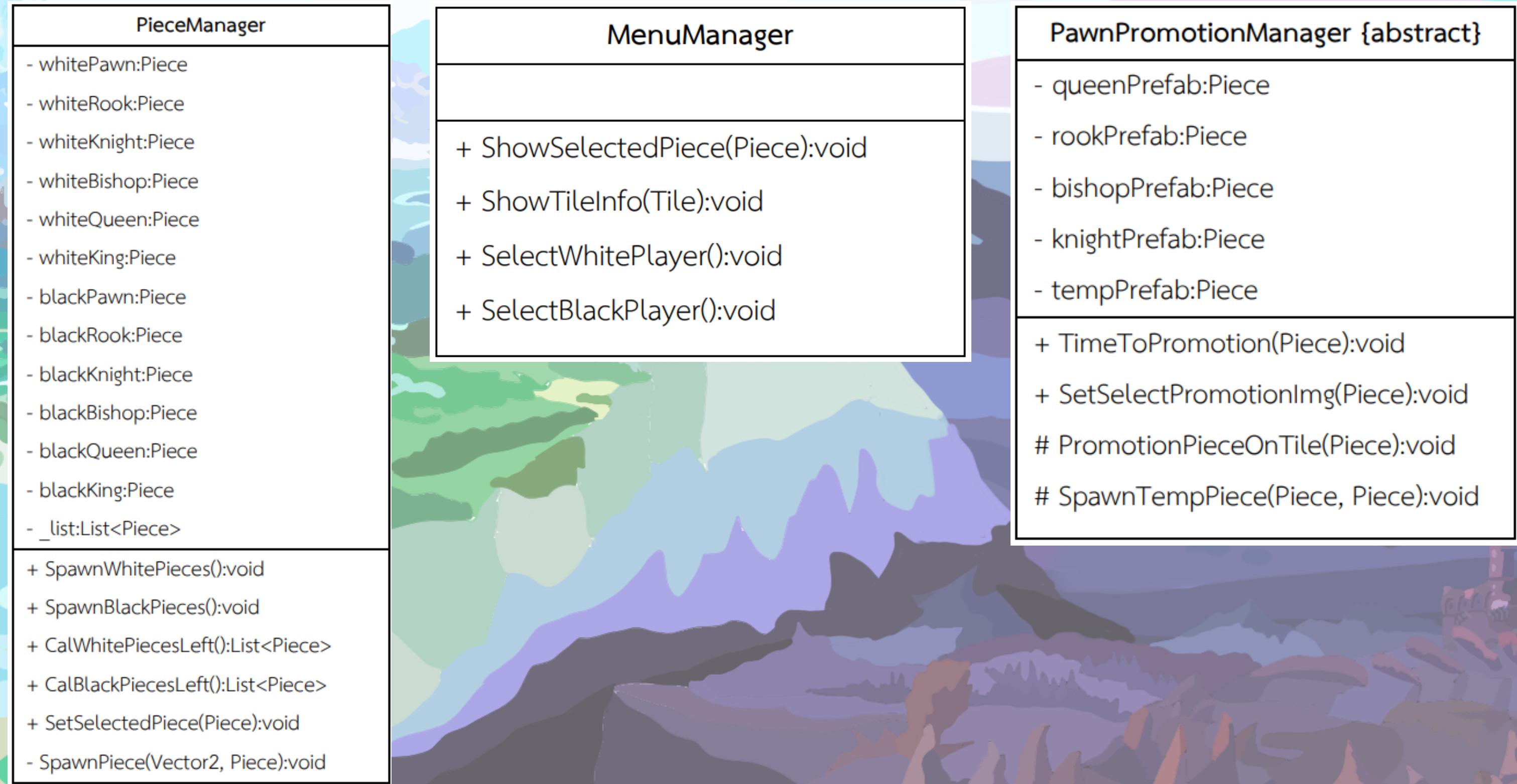
การแก้ปัญหา

- เริ่มต้นวางแผนการทำงานออกแบบคลาสต่างๆ เมืองต้น
- วิเคราะห์การทำงานในแต่ละส่วนของโปรแกรม ทำการสื่อสารกันที่ต้องทำ และทำการแยกจ่ายงาน
- แยกย้ายศึกษาข้อมูลต่างๆ เพิ่มเติม
- เริ่มการทำงานสร้างโปรเจคและจัดการเรื่องของ การสื่อสาร (GITHUB)
- ทำการทดสอบโปรแกรมและแก้ไขข้อผิดพลาด ของตัวโปรแกรมที่เกิดขึ้น
- ทำการสรุปผล

CLASS DIAGRAM

GameManager	Tile	TileManager
<p>+ State:GameState</p> <p>- _round:int</p> <p>- _blackPieces>List<Piece></p> <p>- _whitePieces>List<Piece></p> <p>- _isEnd:Boolean</p> <p>- UpdatePiecesLeft():void</p> <p>+ UpdateGameState(GameState):void</p> <p>- HandleWin():void</p> <p>- HandleBlackTurn():void</p> <p>- HandleWhiteTurn():void</p> <p>+ ChangeTurn():void</p>	<p>- WhiteMoveOccupiedTile:boolean</p> <p>- BlackMoveOccupiedTile:boolean</p> <p>- UnreachableTile:Boolean</p> <p>- SetPiece(Piece):void</p> <p>- MovePiece(Fiction):boolean</p>	<p>+ DictTiles:Dctionary<Vector2, Tile> {readonly}</p> <p>- Width:int {const}</p> <p>- Hight:int {const}</p> <p>+ GenerateTile():void</p> <p>+ GetTile(Vector2 pos):Tile</p> <p>+ Tiles():Dictionary</p>

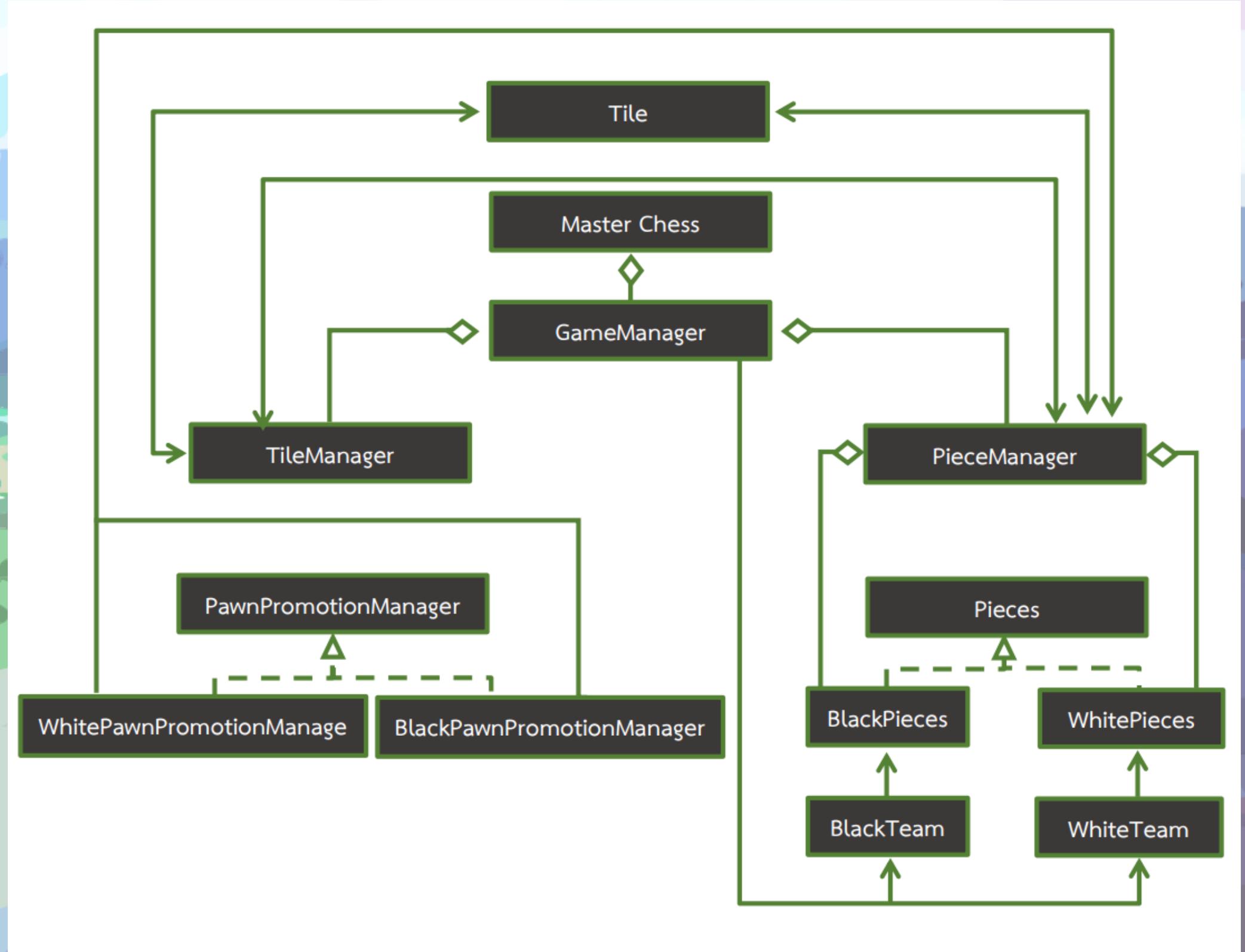
CLASS DIAGRAM



CLASS DIAGRAM

<p>Piece {abstract}</p> <ul style="list-style-type: none"> - 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} 	<table border="1"> <tr> <td>WhiteTeam</td><td>BlackTeam</td></tr> <tr> <td>- KingPos: Vector2</td><td>- KingPos: Vector2</td></tr> <tr> <td>+ FindKing(): void</td><td>+ FindKing(): void</td></tr> </table>	WhiteTeam	BlackTeam	- KingPos: Vector2	- KingPos: Vector2	+ FindKing(): void	+ FindKing(): void	<table border="1"> <tr> <td>WhitePieces</td><td>BlackPieces</td></tr> <tr> <td>- PromotionYPos: float {const}</td><td>- PromotionYPos: float {const}</td></tr> <tr> <td>+ CheckPawnPromotion(): void {override}</td><td>+ CheckPawnPromotion(): void {override}</td></tr> <tr> <td>+ PromotionPawn(Piece): void {override}</td><td>+ PromotionPawn(Piece): void {override}</td></tr> </table>	WhitePieces	BlackPieces	- PromotionYPos: float {const}	- PromotionYPos: float {const}	+ CheckPawnPromotion(): void {override}	+ CheckPawnPromotion(): void {override}	+ PromotionPawn(Piece): void {override}	+ PromotionPawn(Piece): void {override}	<table border="1"> <tr> <td>BlackPawnPromotionManager</td><td>WhitePawnPromotionManager</td></tr> <tr> <td></td><td></td></tr> <tr> <td>+ PromotionPieceOnTile(Piece): void {override}</td><td>+ PromotionPieceOnTile(Piece): void {override}</td></tr> <tr> <td>+ SpawnTempPiece(Piece, Piece): void {override}</td><td>+ SpawnTempPiece(Piece, Piece): void {override}</td></tr> </table>	BlackPawnPromotionManager	WhitePawnPromotionManager			+ PromotionPieceOnTile(Piece): void {override}	+ PromotionPieceOnTile(Piece): void {override}	+ SpawnTempPiece(Piece, Piece): void {override}	+ SpawnTempPiece(Piece, Piece): void {override}
WhiteTeam	BlackTeam																								
- KingPos: Vector2	- KingPos: Vector2																								
+ FindKing(): void	+ FindKing(): void																								
WhitePieces	BlackPieces																								
- PromotionYPos: float {const}	- PromotionYPos: float {const}																								
+ CheckPawnPromotion(): void {override}	+ CheckPawnPromotion(): void {override}																								
+ PromotionPawn(Piece): void {override}	+ PromotionPawn(Piece): void {override}																								
BlackPawnPromotionManager	WhitePawnPromotionManager																								
+ PromotionPieceOnTile(Piece): void {override}	+ PromotionPieceOnTile(Piece): void {override}																								
+ SpawnTempPiece(Piece, Piece): void {override}	+ SpawnTempPiece(Piece, Piece): void {override}																								

CLASS DIAGRAM



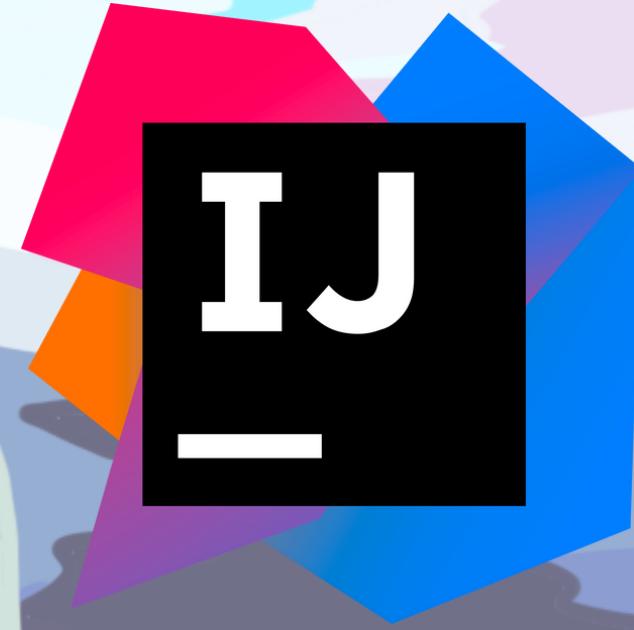
ក្រោមតីសប្បែសបុបនខែបាតា



UNITY



RIDER



INTELIJ



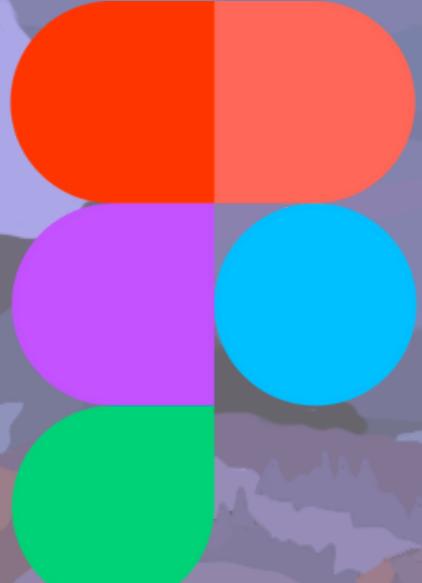
PROCREATE



VISUAL STUDIO



VS CODE

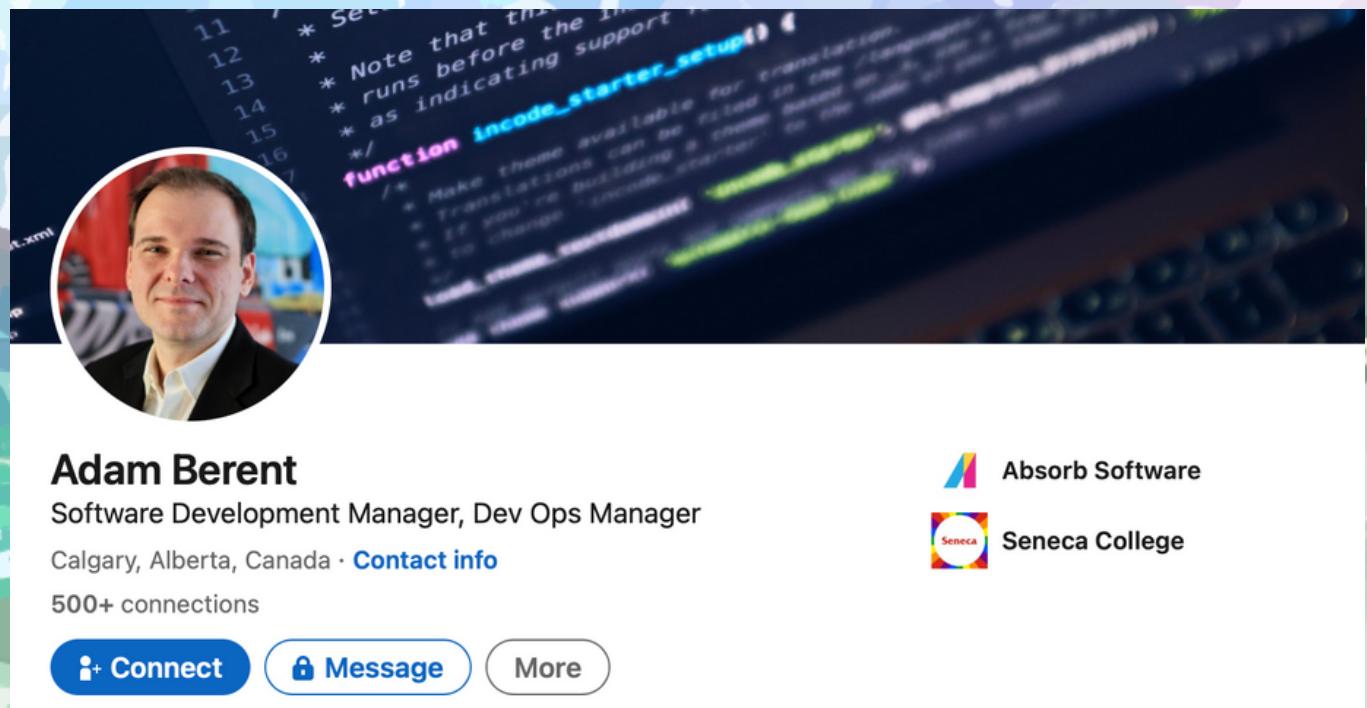


FIGMA



GITHUB

ແພວົມຄົດກາຣອອກແບບເກມຈາກ ຜູ້ປັບປະໂຫຍດຂຶ້ນ

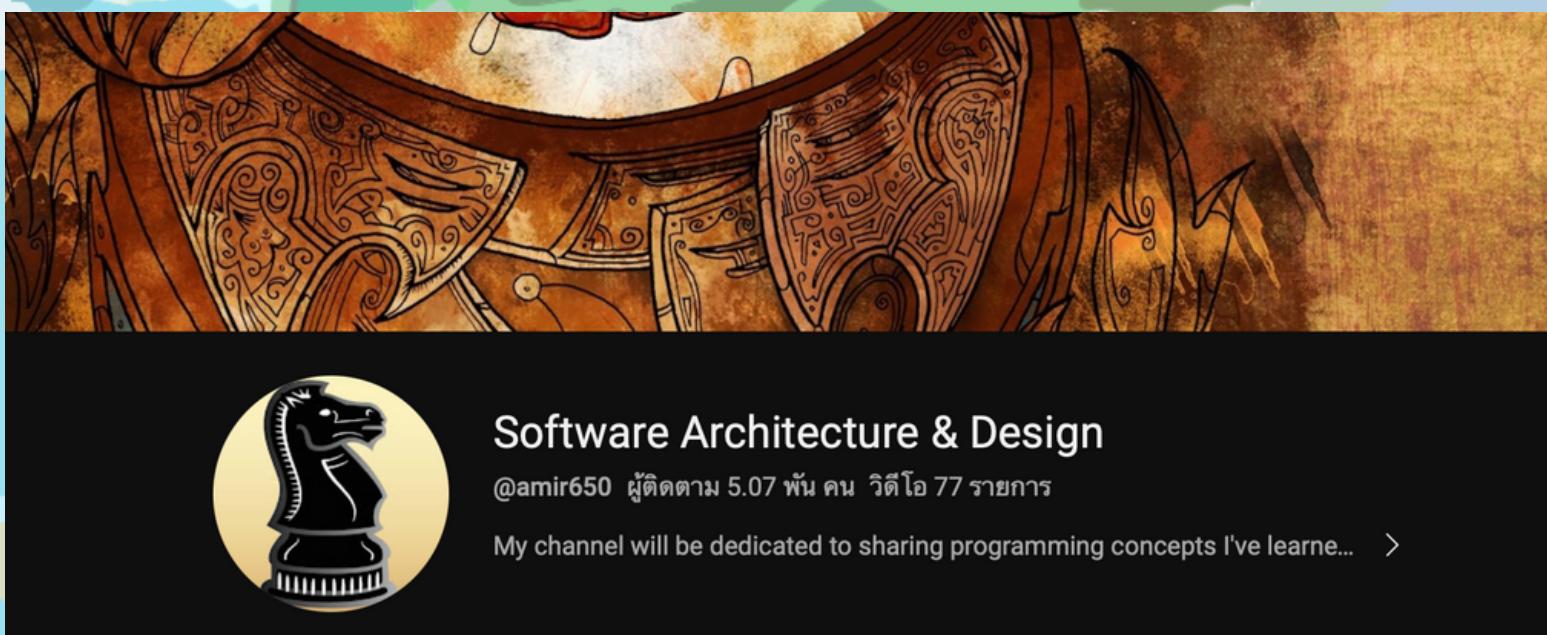


Adam Berent
Software Development Manager, Dev Ops Manager
Calgary, Alberta, Canada · [Contact info](#)
500+ connections

[Connect](#) [Message](#) [More](#)

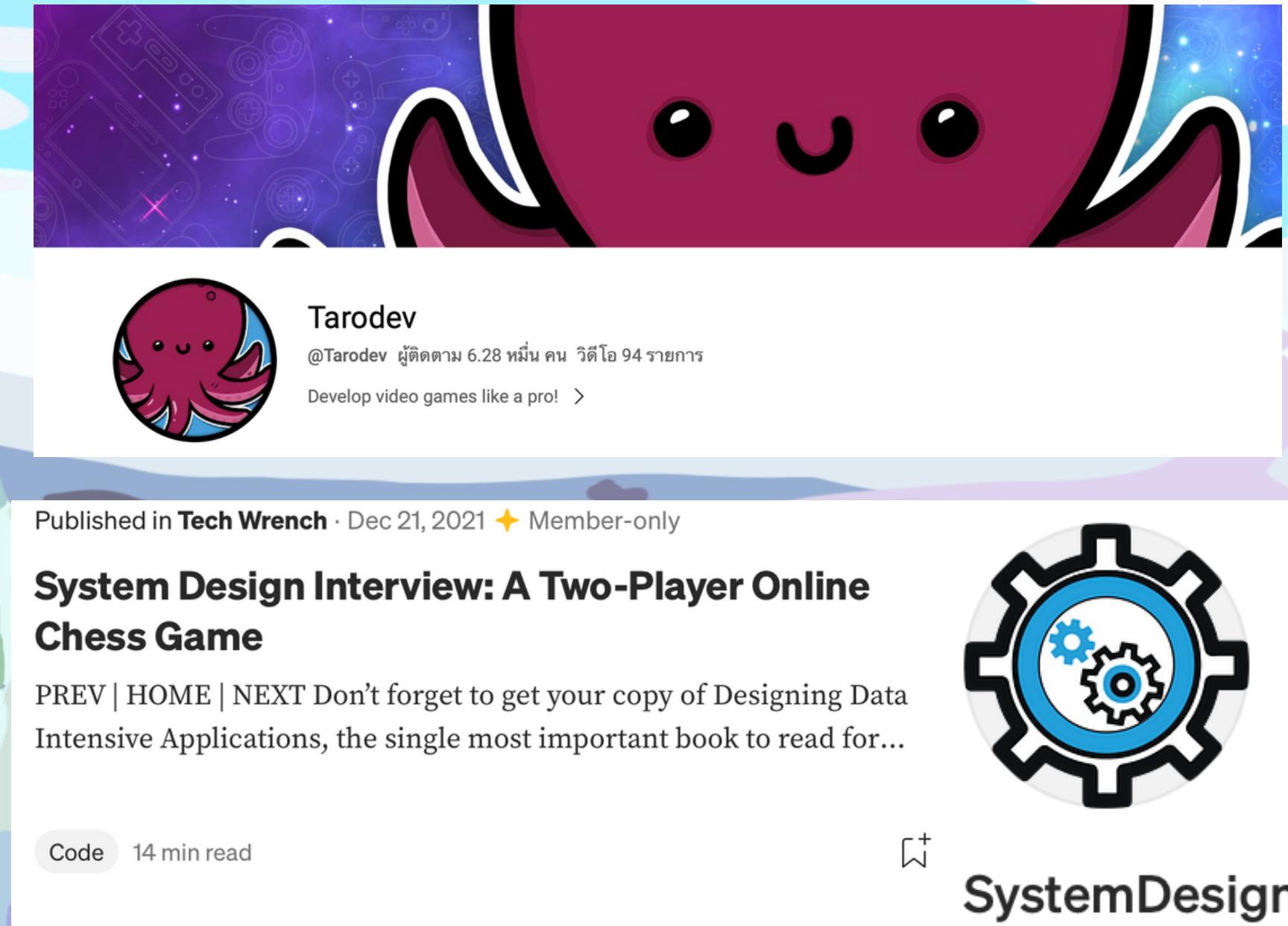
Absorb Software
Seneca College

A LinkedIn profile card for Adam Berent, showing his photo, title, location, connections, and social links. It also lists his employer, Absorb Software, and his college, Seneca College.



Software Architecture & Design
@amir650 ຜັດຕາມ 5.07 ພັນ ຜັນ ວິດໂອ 77 ຮາຍການ
My channel will be dedicated to sharing programming concepts I've learned... >

A YouTube channel thumbnail for "Software Architecture & Design" by @amir650. It features a knight chess piece icon and a background image of a classical building's facade.



Tarodev
@Tarodev ຜັດຕາມ 6.28 ພັນ ຜັນ ວິດໂອ 94 ຮາຍການ
Develop video games like a pro! >

Published in **Tech Wrench** · Dec 21, 2021 ✨ Member-only

System Design Interview: A Two-Player Online Chess Game

PREV | HOME | NEXT Don't forget to get your copy of Designing Data Intensive Applications, the single most important book to read for...

Code 14 min read

A Medium post by Tarodev titled "System Design Interview: A Two-Player Online Chess Game". The post is member-only and published in Tech Wrench on December 21, 2021. It discusses system design for a two-player online chess game and recommends the book "Designing Data Intensive Applications".



GeeksforGeeks

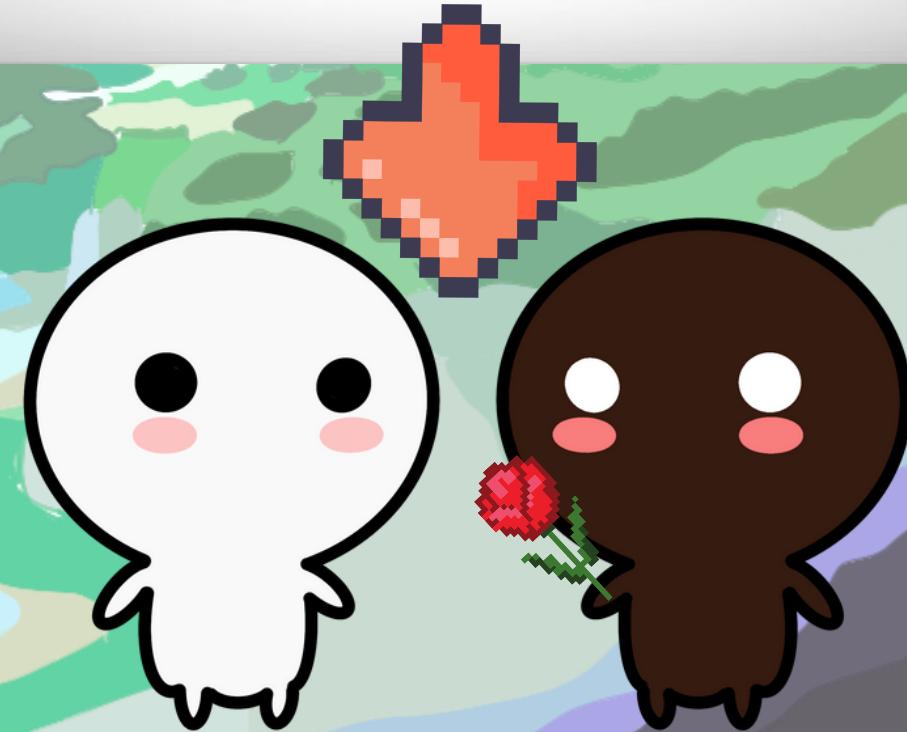
A-143, 9th Floor, Sovereign Corporate Tower,
Sector-136, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org

Contact information for GeeksforGeeks, including their address at A-143, 9th Floor, Sovereign Corporate Tower, Sector-136, Noida, Uttar Pradesh - 201305, and their email address for feedback, feedback@geeksforgeeks.org.



แรงบันดาลใจตัวละครจาก



สนับสนุน UI และ SPRITES
โดย UNITY ASSETS STORE



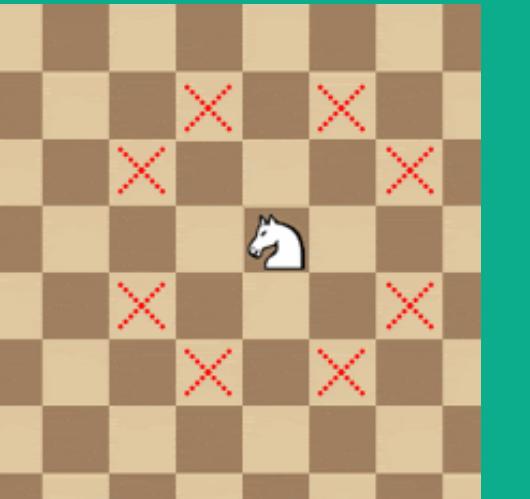
Unity Asset Store

กฎและกติกาหมากกระดานเบื้องต้นมีดังนี้

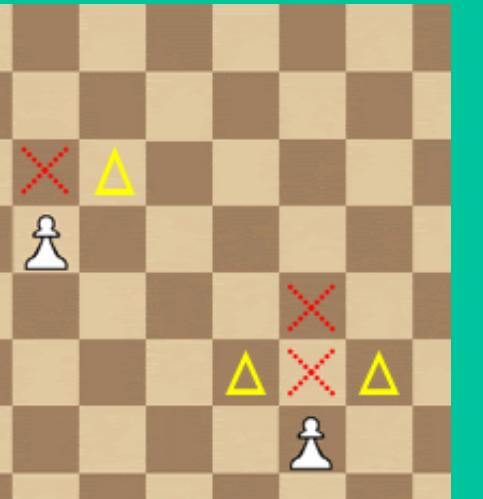
1. การวางแผน



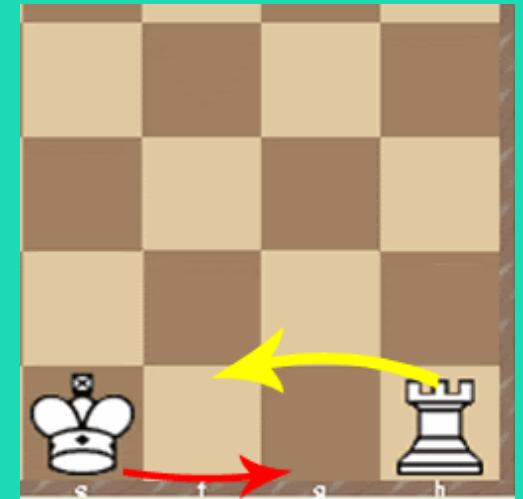
2. การเคลื่อนไหว



3. การโขมติ



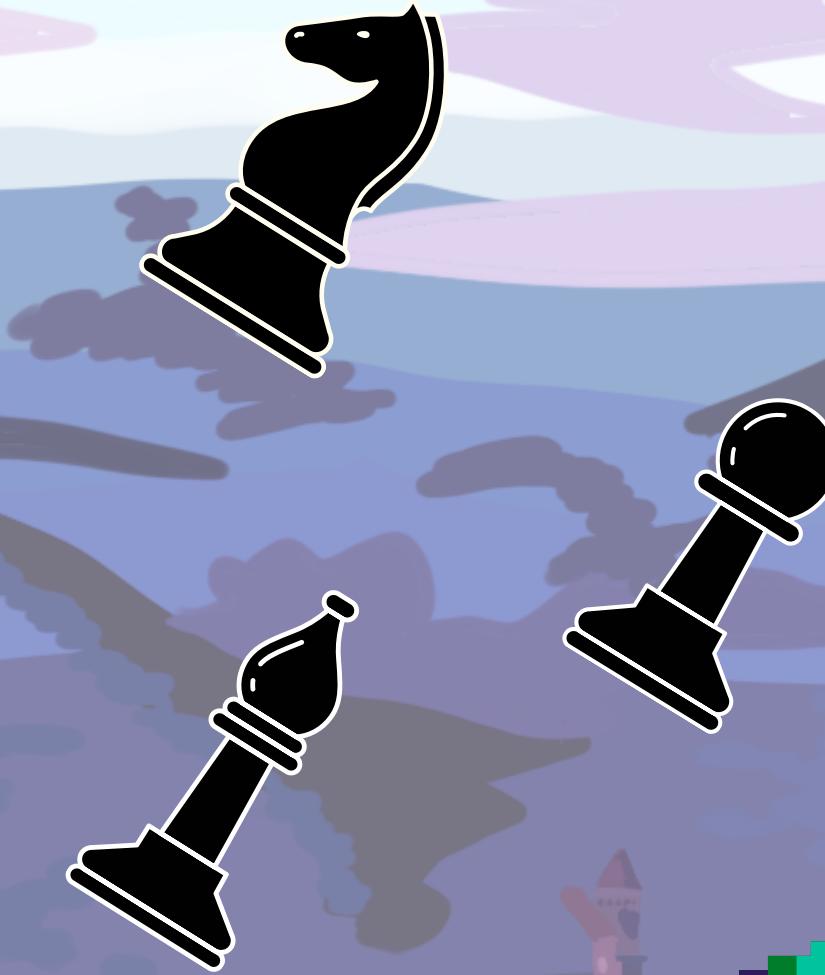
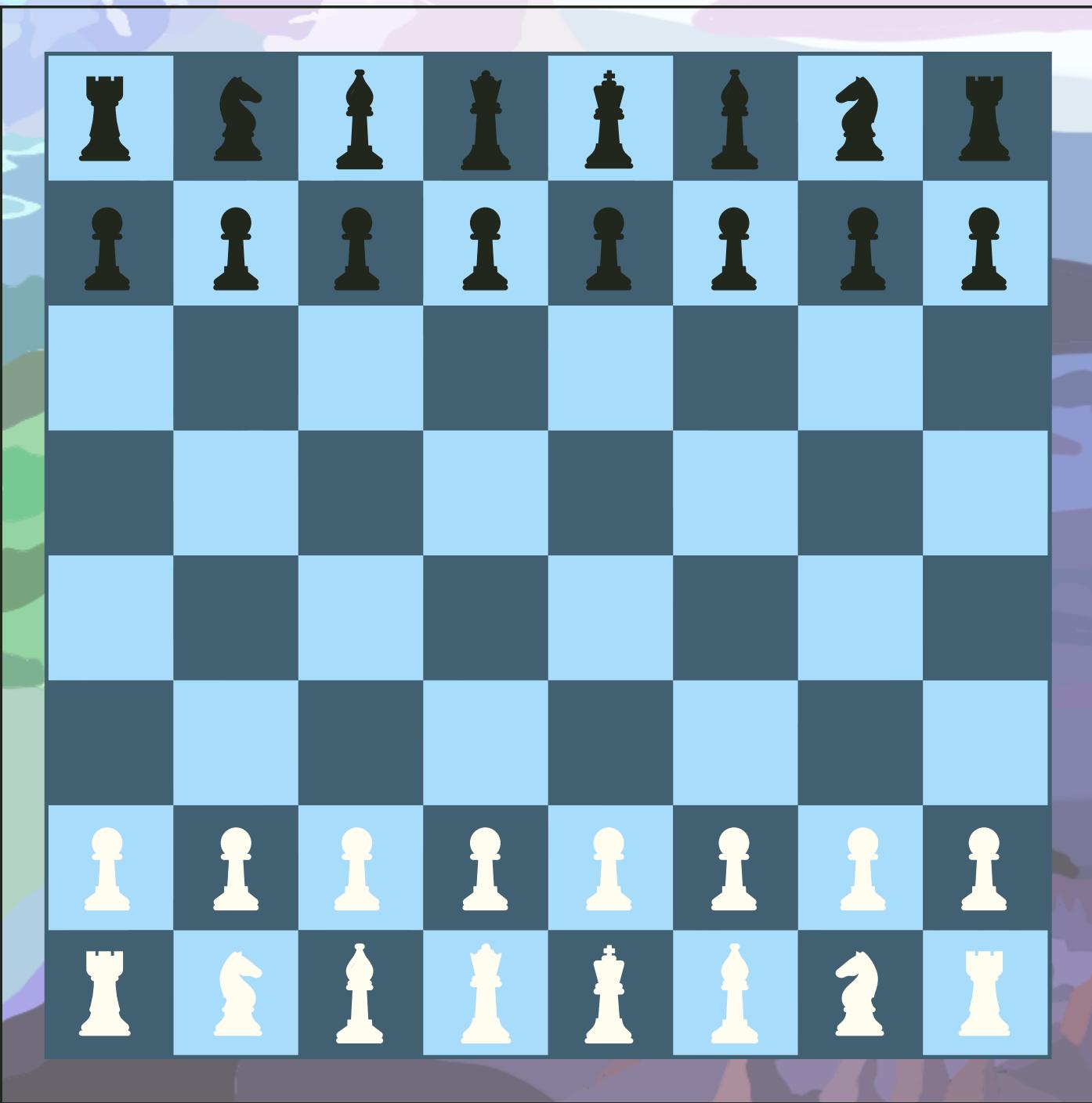
4. การชนจะ



CHESS GAME



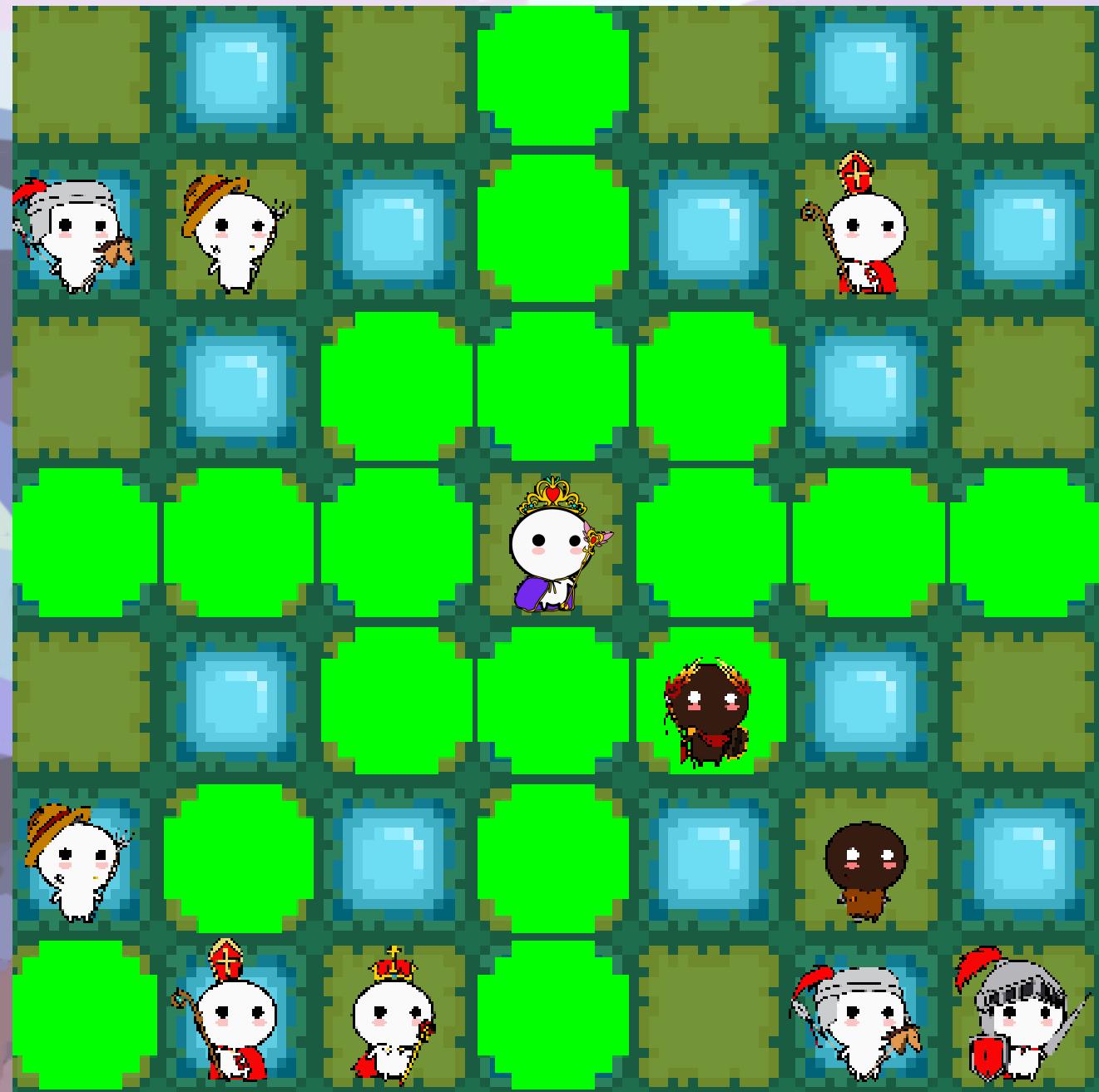
SETUP GAME



KING



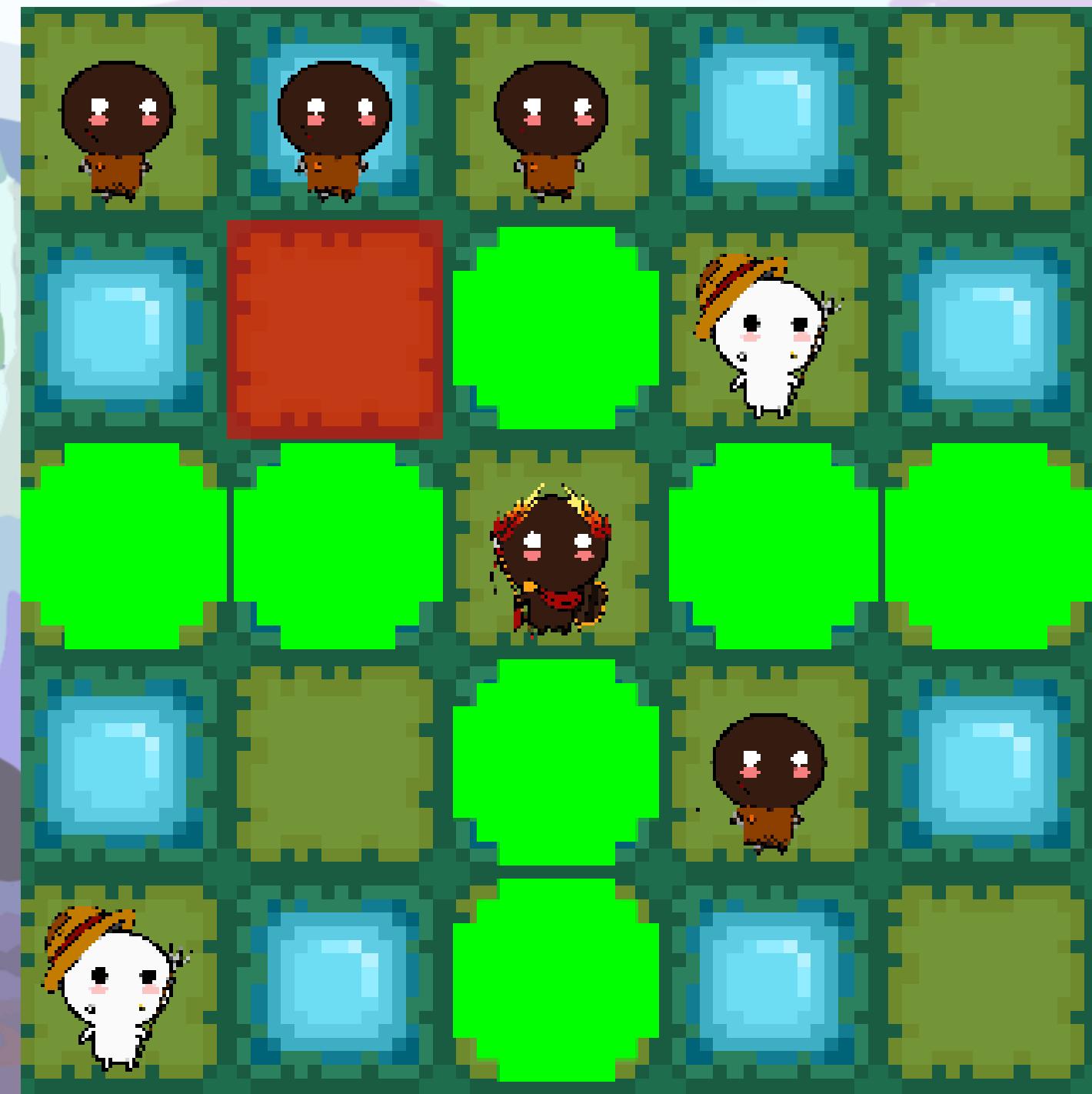
QUEEN



BISHOP



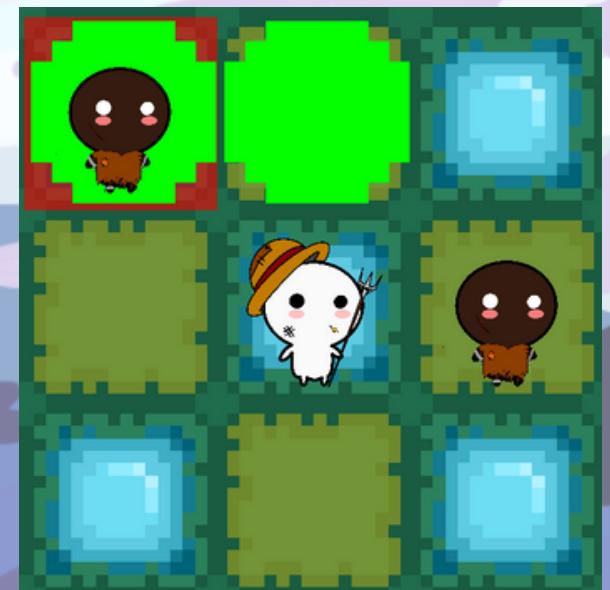
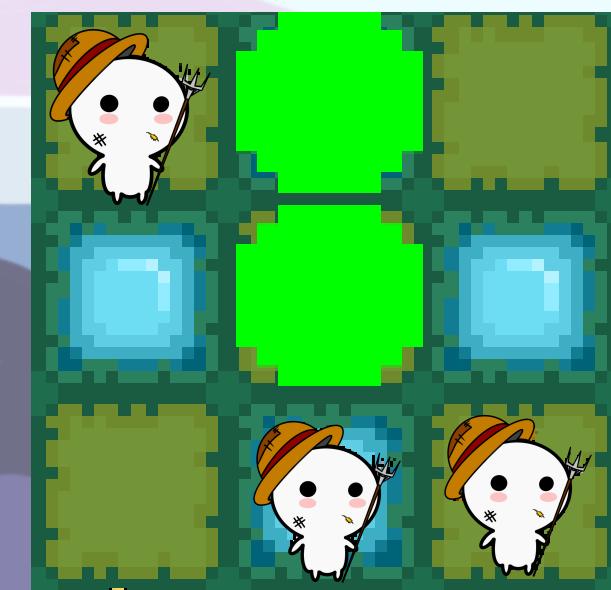
ROOK



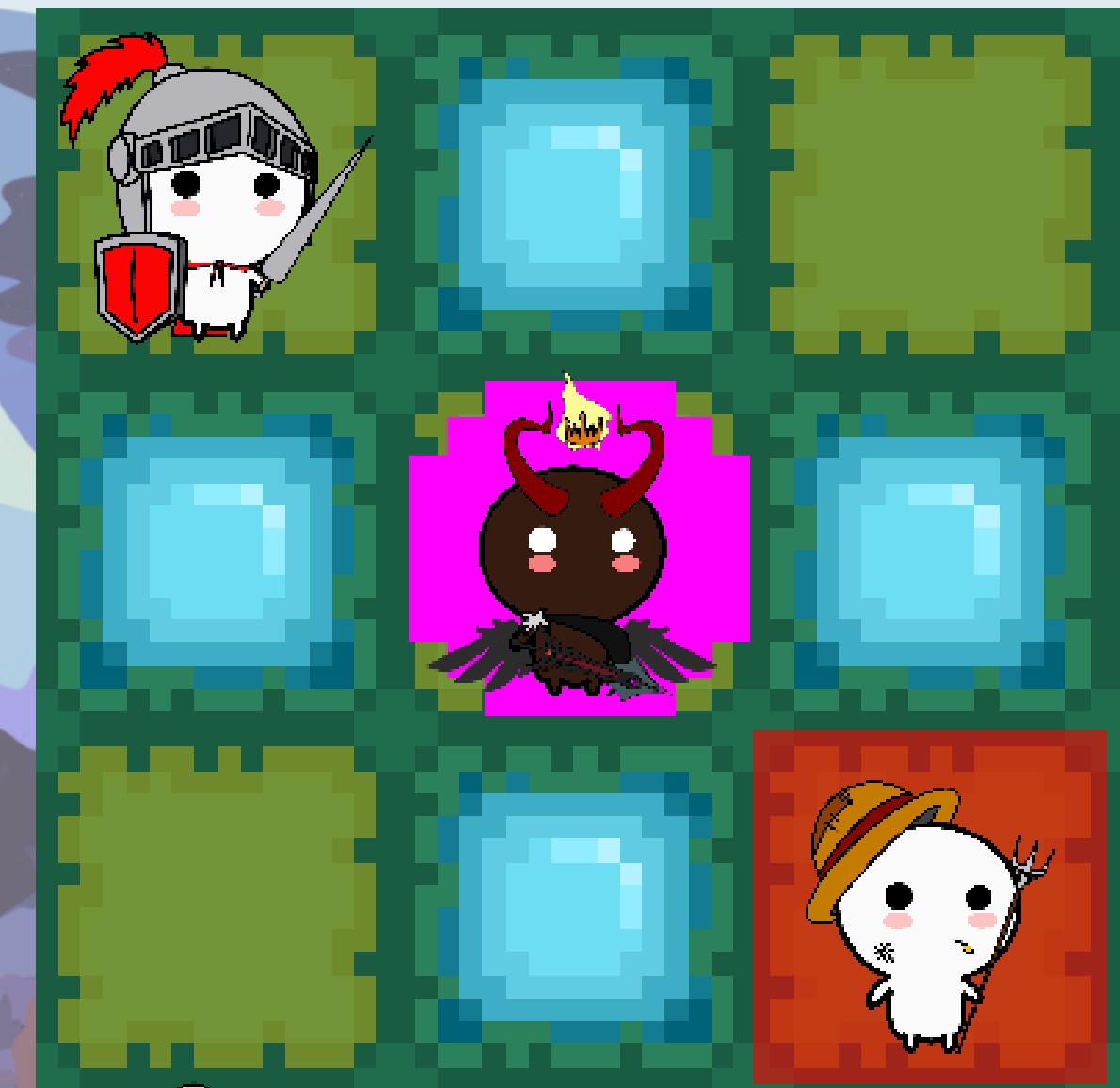
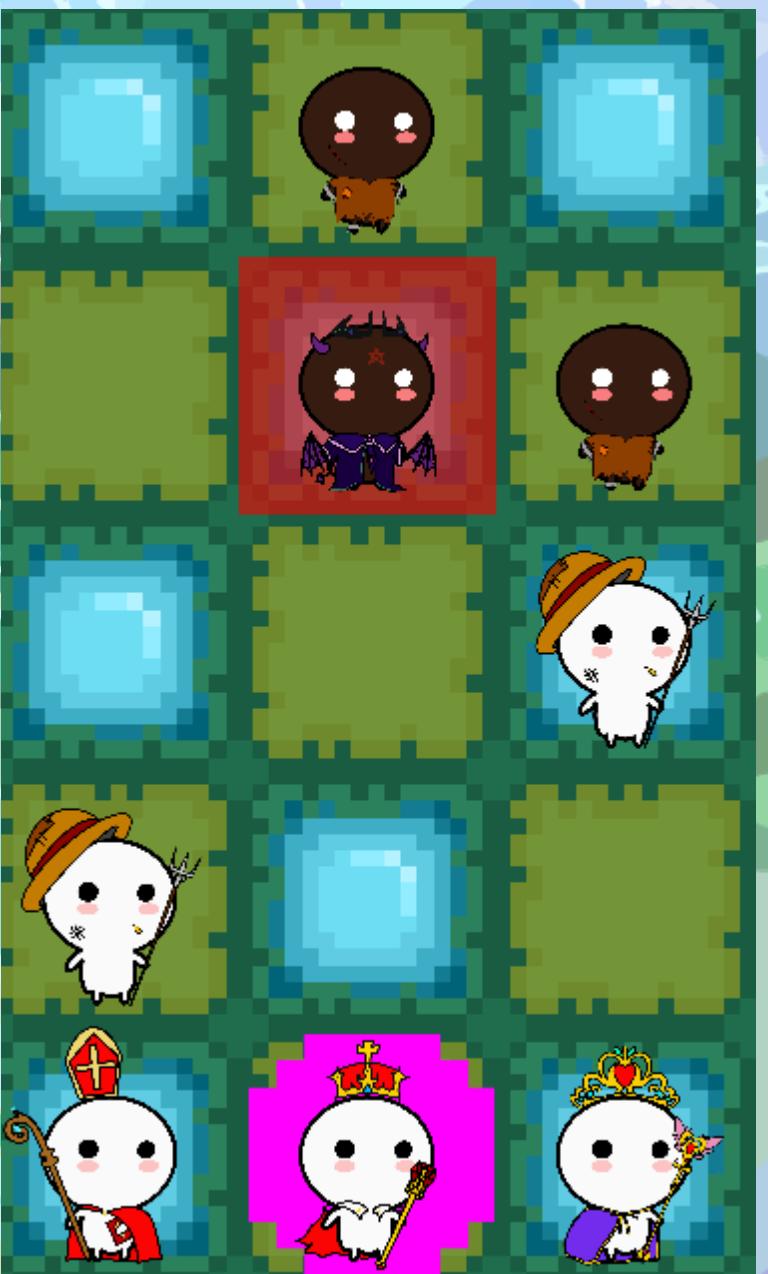
KNIGHT



PAWN



CHECK



CHECKMATE

NEW START



LET'S GO TO THE GAME

SCAN TO PLAY



PROJECT REPOSITORY

