

TeamPuwadol Documentation

จัดทำโดย

นายอชิวรรณ ดีโป 6231372021

นายภูวดล วิจิตพันธุ์ 6231350121

2110215 Programming Methodology

Semester 2 Year 2019

Chulalongkorn University

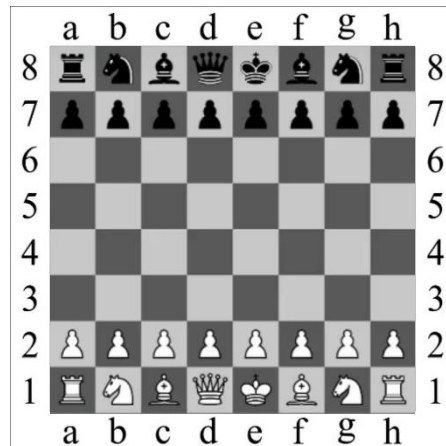
Multichess game

เกมนี้ประกอบด้วย เกมหมากรุกสากลหลายประเภท ได้แก่ Normal, Atomic, King of hill, Three check, Chess960 และ Horde โดยแต่ละประเภทจะมีกฎกติกาพิเศษที่แตกต่างกัน

กฎกติกา

โดยเกมหมากรุกจะเป็นเกมวางแผนผู้เล่น 2 คน บนกระดาน 8×8 ช่อง โดยจะชนะก็ต่อเมื่อ king ฝ่ายตรงข้ามถูกรุกและไม่สามารถเดินไปไหนได้ และ เสมอ เมื่อ king และตัวอื่นๆที่มีไม่สามารถเดินได้แล้ว

โดยแต่ละผู้เล่นจะเริ่มโดยมี 16 ตัว: one king, one queen, two rooks, two knights, two bishops, and eight pawns.



King	เดินได้ 1 ช่องรอบๆ โดยการเดินนั้นต้องไม่ทำให้ตัวเองถูกกิน
Queen	เดินได้หลายช่อง 8 ทิศทางรอบๆ
Rook	เดินได้หลายช่อง 4 ทิศทางในแนวตรง
Knight	เดินไปฝั่งตรงข้ามของทุกๆตาราง 2×3 ช่อง
Bishop	เดินได้หลายช่อง 4 ทิศทางในแนวทแยง
Pawn	เดินได้ 2 ช่อง (เมื่ออยู่ที่ 2 ช่องล่างสุดสำหรับขาว และ 2 ช่องบนสุดสำหรับดำ) โดยปกติจะเดิน 1 ช่องตรงๆและกิน 1 ช่องทแยงไปด้านหน้า

กฎกติกาพิเศษ

1. Castling

การเข้าป้อม (castling) จะทำเมื่อช่องระหว่าง king และ rook รวมทั้งจุดสิ้นสุดไม่มีตัวหมากมาขึ้น และทางเดินของขุนไม่ถูกกินโดยฝั่งตรงข้ามโดยมี 2 ทางโดย Chess960 ก็ทำได้เช่นกัน

- king มาที่ตำแหน่ง c และ rook มาที่ตำแหน่ง d
- king มาที่ตำแหน่ง g และ rook มาที่ตำแหน่ง f

2. En passant

การกินตัด (En passant) คือการที่เบี้ยเดิน 2 ขั้น (ในครั้งแรก) มาชนาบข้างกับเบี้ยฝั่งตรงข้าม เบี้ยฝั่งตรงข้ามสามารถกินโดยการเดินเหมือนกินธรรมดาและเบี้ยจะถูกกิน

3. Promotion

การเลื่อนขั้น (promotion) คือการเปลี่ยนเบี้ยเป็น queen, rook, knight หรือ bishop เมื่อเบี้ยเดินไปจนสุดกระดาน

กระดานต่างๆ

ในโปรแกรมนี้จะมีเกมหมากรุกสากลหลายประเภท ได้แก่ Normal, Atomic, King of hill, Three check, Chess960 และ Horde

1. Normal

เป็นกระดานธรรมดา ไม่มีกฎกติกาพิเศษ

2. Atomic

เมื่อกินตัวเองและตัวที่ถูกกินรวมทั้งช่องรอบๆกเว้นเบี้ยจะถูกระเบิด

3. King of hill

สามารถชนะได้โดยการรุกจนหรือ king เดินไป 4 ช่องตรงกลาง

4. Three check

สามารถชนะได้โดยการรุกจนหรือ รุกฝั่งตรงข้าม 3 รอบ

5. Chess960

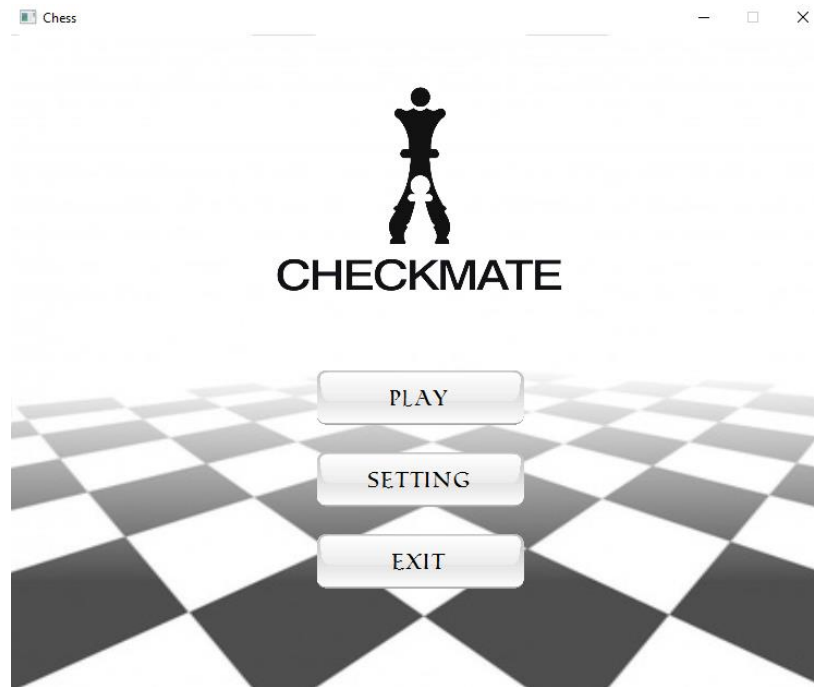
เริ่มต้นด้วยกระดานแบบสุ่ม โดย bishop ต้องอยู่บนช่องคนละสี และ king ต้องอยู่ระหว่าง rook ทั้ง 2 ตัว

6. Horde

เริ่มต้นด้วยฝั่งขาวมาเบี้ย 36 ตัว โดยฝั่งดำจะชนะเมื่อกินเบี้ยทั้งหมด

Game instruction

Menu scene



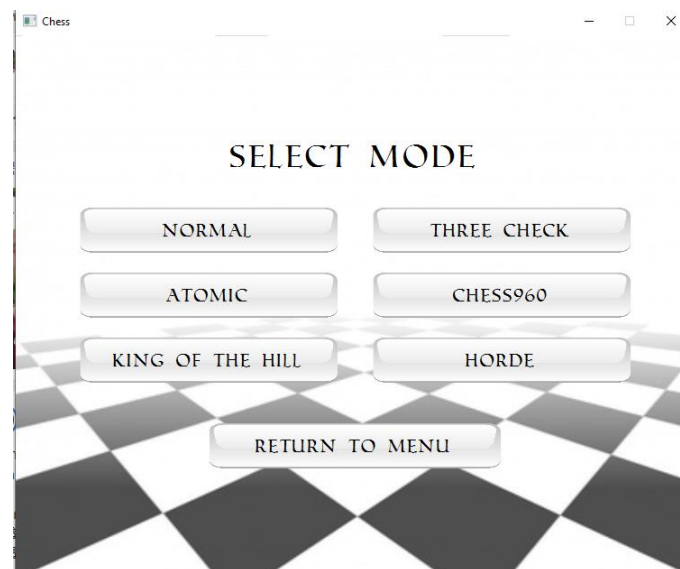
- Play กดเพื่อเข้าสู่หน้าเลือกโหมด
- Setting กดเพื่อเข้าสู่หน้า setting สามารถปรับเสียงได้
- Exit กดเพื่อออกจากเกม

Setting scene



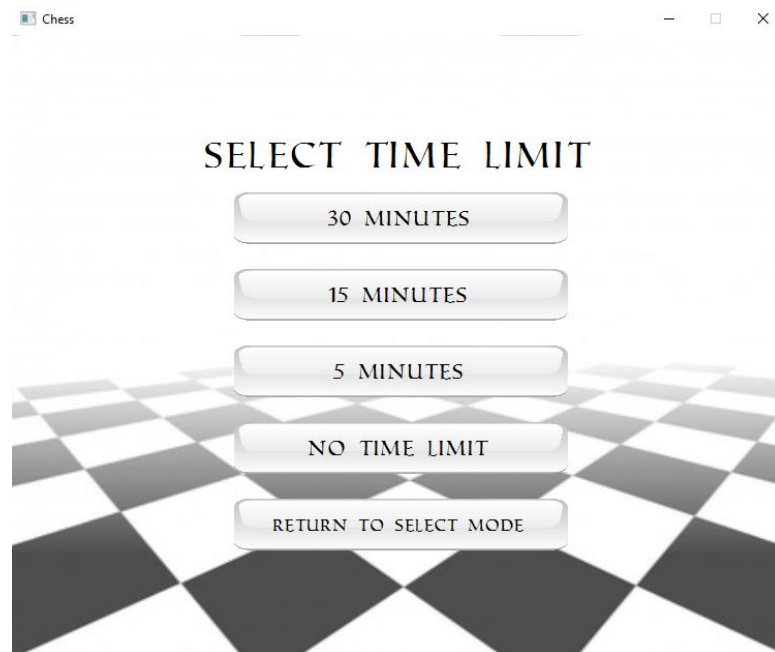
- สามารถปรับเสียง BGM และ SFX ได้ในหน้านี้

Select mode scene



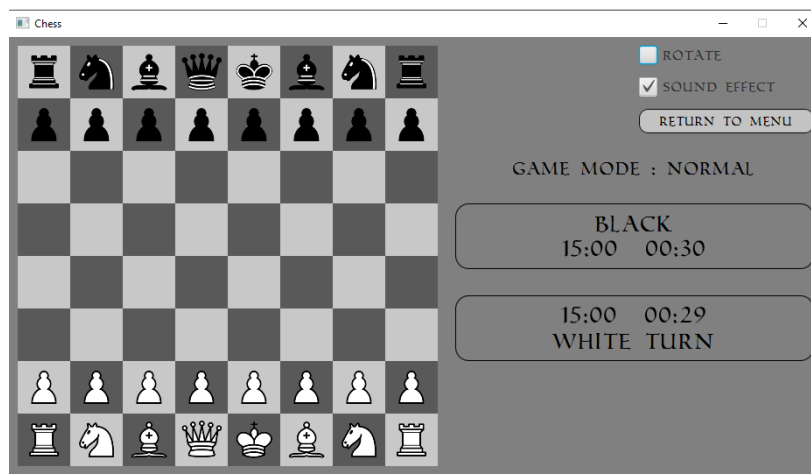
- สามารถเลือกโหมดที่ต้องการเล่นได้在這นี้ โดยกติกาของแต่ละโหมดสามารถดูได้จากกติกาของกระดานต่าง ๆ ด้านบน

Select time scene



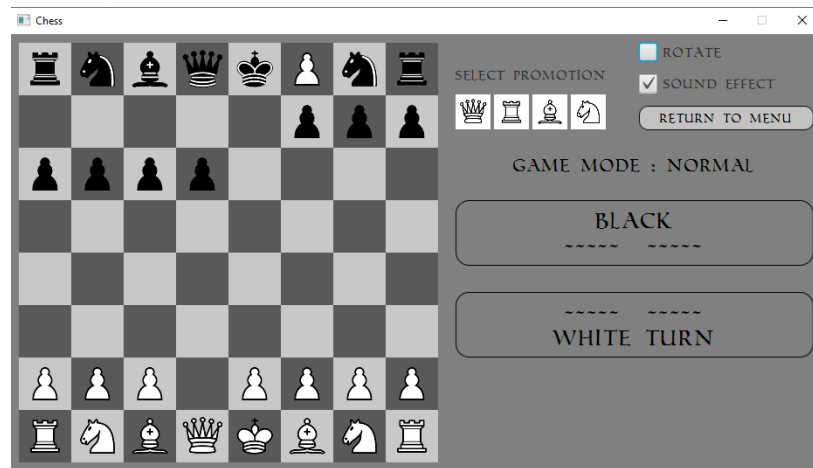
- หน้าเลือกเวลาจำกัด เป็นเวลาที่แต่ละคนใช้ได้ ใน 1 เกม โดยถ้าเวลาหมดจะเป็นฝ่ายแพ้

Game scene



- หน้าเล่นเกม สามารถกดเดินตัวหมากได้โดยการคลิกที่ตัวหมากนั้น ๆ และคลิกช่องที่ต้องการเดินไป
- สามารถกด ROTATE เพื่อหมุนกระดานได้ และสามารถกด SOUND EFFECT เพื่อเปิด/ปิดเสียงกดได้

Promotion scene



- หากเบี้ยเดินไปจนสุดกระดาน จะปรากฏหน้าต่าง select promotion ให้เลือกตัวที่ต้องการ

Winning scene



- หน้าแสดงผล จะแสดงฝ่ายที่ชนะ กดปุ่ม RETURN TO MENU เพื่อกลับไปหน้าจอ Menu

1. Package entity.base

1.1. Class Entity

ตัวหมากแต่ละตัว

Fields

# Point point	เก็บตำแหน่งของแต่ละ Entity ในขณะเล่นเกม
# Side side	เก็บฝั่งของตัวหมาก ขาว/ดำ ไม่สามารถเปลี่ยนได้

Methods

+ <i>String</i> getSymbol()	get normal piece
+ <i>String</i> getHighlightSymbol()	get piece with highlight
+ <i>ArrayList</i> <Point> moveList(Board board)	get move list this entity
# <i>ArrayList</i> <Point> moveList(Board board, Point point, Point vector)	get move list when go in vector direction use for rook, bishop and queen
+ Point getPoint() + Side getSide() + void setPoint(Point point)	getter & setter
+ <i>String</i> toString()	to string

1.2. interface HaveCastling

การเข้าป้อมของ King และ Rook

Methods

+ void setNeverMove();	set neverMove to be false
+ boolean isNeverMove();	getter

2. Package entity

2.1. Class King extend Entity implement HaveClastling

Methods

+ String getSymbol	ให้ชื่อไฟล์รูปภาพตัวหมากปกติ
+ String getHighlightSymbol	ให้ชื่อไฟล์รูปภาพตัวหมากที่สนใจ
+ ArrayList<Point> moveList	ให้ list ของตำแหน่งตัวหมากที่เดินไปได้
+ String toString	toString
+ void setNeverMove	เก็บคำว่าตัวหมากนี้เคยเดินแล้ว (neverMove)
+ boolean isNeverMove	getter & setter

2.2. Class Rook extend Entity implement HaveClastling

Methods

+ String getSymbol	ให้ชื่อไฟล์รูปภาพตัวหมากปกติ
+ String getHighlightSymbol	ให้ชื่อไฟล์รูปภาพตัวหมากที่สนใจ
+ ArrayList<Point> moveList	ให้ list ของตำแหน่งตัวหมากที่เดินไปได้
+ String toString	toString
+ void setNeverMove	เก็บคำว่าตัวหมากนี้เคยเดินแล้ว (neverMove)
+ boolean isNeverMove	getter & setter

2.3. Class Queen extend Entity

Methods

+ String getSymbol	ให้ชื่อไฟล์รูปภาพตัวหมากปกติ
+ String getHighlightSymbol	ให้ชื่อไฟล์รูปภาพตัวหมากที่สนใจ
+ ArrayList<Point> moveList	ให้ list ของตำแหน่งตัวหมากที่เดินไปได้
+ String toString	toString

2.4. Class Bishop extend Entity

Methods

+ String getSymbol	ให้ชื่อไฟล์รูปภาพตัวหมากปกติ
+ String getHighlightSymbol	ให้ชื่อไฟล์รูปภาพตัวหมากที่สนใจ
+ ArrayList<Point> moveList	ให้ list ของตำแหน่งตัวหมากที่เดินไปได้
+ String toString	toString

2.5. Class Knight extend Entity

Methods

+ String getSymbol	ให้ชื่อไฟล์รูปภาพตัวหมากปกติ
+ String getHighlightSymbol	ให้ชื่อไฟล์รูปภาพตัวหมากที่สนใจ
+ ArrayList<Point> moveList	ให้ list ของตำแหน่งตัวหมากที่เดินไปได้
+ String toString	toString

2.6. Class Pawn extend Entity

Methods

+ String getSymbol	ให้ชื่อไฟล์รูปภาพตัวหมากปกติ
+ String getHighlightSymbol	ให้ชื่อไฟล์รูปภาพตัวหมากที่สนใจ
+ ArrayList<Point> moveList	ให้ list ของตำแหน่งตัวหมากที่เดินไปได้
+ String toString	toString

3. Package game.base

3.1. Class Board

กระดานเก็บตำแหน่งของตัวหมาก เช็การชนะ การเดิน

Fields

# Point twoWalkPawn	เก็บตำแหน่งเบี้ยที่เดิน 2 ชั้น สำหรับ en passant
- Cell[][] cellMap	เก็บ cell map
- Entity whiteKing, blackKing	เก็บ King ของแต่ละฝั่ง
# ArrayList<Point> removePoint	เก็บ จุดที่จะต้องลบหลัง move animation
# Entity movePiece, castlingRook	เก็บ Entity ที่ move
# Point movePoint, newRookPoint	เก็บ Point ของ Entity ที่ move
# <u>final Point[] knightWalk</u>	เก็บ list ของทางเดินของ Knight
# <u>final Point[] KingWalk</u>	เก็บ list ของทางเดินของ King
- private int width, height	เก็บขนาดของกระดาน

Methods

+ boolean isWin(Side side)	เช็การชนะ
+ boolean isDraw(Side side)	เช็การเสมอ
+ boolean isCheck(Side side)	เช็การรุก

+ void startAnimation(Point oldPoint, Point newPoint)	เริ่มการเดิน โดยการเก็บตัวที่เดินและลบตัวที่เดินออกจากกระดาน แล้วสั่ง AppManager.startAnimation
+ void continueMove()	เดินต่อโดยนำตัวที่เดินมาที่จุดสิ้นสุด และลบตัวอื่นๆที่ถูกระเบิดสำหรับ atomicBoard
+ ArrayList<Point> moveList(Point point)	ให้ list ของจุดที่เดินได้จริงๆ
# ArrayList<Point> editMovePoint(Point oldPoint, ArrayList<Point> movePoint)	ลบจุดที่เดินไม่ได้จากการเปิดรุกหรืออื่นๆ
# boolean checkCannotMovePoint (ArrayList<Point> oldPoint, Point newPoint, Point kingPoint, Side side)	เช็คว่าคุณนั้นๆเดินได้ไหมโดยใช้ checkOther, checkRook และ checkBishop
# boolean checkOther(Point newPoint, Point kingPoint, Side side)	เช็คสำหรับ king, pawn และ knight
# boolean checkRook(Point point, Point vector, ArrayList<Point> oldPoint, Point newPoint, Side side)	เช็คการเดินตรงสำหรับ rook และ queen
# boolean checkBishop(Point point, Point vector, ArrayList<Point> oldPoint, Point newPoint, Side side)	เช็คการเดินตรงสำหรับ bishop และ queen
+ void havePromotion(Point point, Side side)	Pawn เดินสุดจน และสั่ง GameController.setPromotion
+ void promotion(Point point, Side side, String piece)	เมื่อผู้เล่นเลือกตัว promotion และเปลี่ยน pawn เป็นตัวนั้นๆ
# ArrayList<Point> castingPoint(Side side)	ให้จุดที่ king สามารถ castling ได้
+ boolean isCastlingPoint(Side side, Point point)	เช็คว่าคุณกำลัง castling ีเปล่า
+ void castling(Side side, Point oldPoint, Point newPoint)	ทำการ castling และ สั่ง AppManager.startCastlingAnimation
# <u>int min(int i1, int i2, int i3, int i4)</u>	สำหรับ castling
# <u>int max(int i1, int i2, int i3, int i4)</u>	สำหรับ castling
# boolean isFree(Point start, Point stop, Side side)	เช็คว่าคุณไม่โดนกิน สำหรับ castling
# boolean isNull(Point oldKing, Point oldRook, Point newKing, Point newRook)	เช็คว่าคุณว่าง สำหรับ castling

# boolean isEatenPoint(Point point, Side side)	เช็คว่าคุณกินโดยฝั่งตรงข้ามได้รึเปล่า
# void remove(Point p)	เอา Entity ออกจาก cellMap
# void addEntity(Entity e, Point p)	เพิ่ม Entity ลงใน cellMap
+ Entity getEntity(Point p)	ให้ Entity จาก cellMap
+ boolean isInBoard(Point p)	เช็คว่ายู่ในกระดานรึเปล่า
# ArrayList<Entity> getAllPieces(Side side)	ให้ list ของ Entity ฝั่งนั้นๆ
# Entity getKing(Side side)	ให้ King ของฝั่งนั้นๆ
# Side getAnotherSide(Side side)	ให้ another side
+ Point[] getKnightWalk()	ให้ list ของทางเดินของ Knight
+ Point[] getKingWalk()	ให้ list ของทางเดินของ King
# Point addPoint(Point p1, Point p2)	ให้ผลรวมของคู่อันดับ
+ Cell[][] getCellMap()	ให้ cellMap

3.2. Interface CheckMateAble

สำหรับกระดานที่มีการรุก และชนะด้วยการรุกจนได้

Fields

+ boolean winByCheckmate(Side side)	เช็คการชนะจากการรุกจน
+ boolean drawCannotMove(Side side)	เช็คการเสมอจากการอับ
+ boolean isCheck(Side side)	เช็คการรุก

3.3. Class Games

Fields

+ static final String NORMAL
+ static final String ATOMIC

+ static final String KINGOFTHEHILL
+ static final String CHESS960
+ static final String THREECHECK
+ static final String HORDE

4. Package game

4.1. Abstract Class AtomicBoard extends Board implements CheckMateAble

เกมที่เมื่อกินตัวเองและตัวที่ถูกกินรวมทั้งช่องรอบๆกวนเบี้ยจะถูกระเบิด

Methods

+ boolean isWin(Side side)	เช็คการชนะ
+ boolean winByLosingKing(Side side)	เช็คการชนะจากเสีย king
+ boolean winByCheckmate(Side side)	เช็คการชนะจากการรุกจน
+ boolean isDraw(Side side)	เช็คการเสมอ
+ boolean drawCannotMove(Side side)	เช็คการเสมอจากการอัป
+ boolean drawByLosing2King()	เช็คการเสมอจากการเสีย king
+ boolean isCheck(Side side)	เช็คการรุก
+ void startAnimation(Point oldPoint, Point newPoint)	Override การระเบิด
+ protected ArrayList<Point> editMovePoint(Point oldPoint, ArrayList<Point> movePoint)	Override การระเบิด king ตัวเอง
# boolean checkOther(Point newPoint, Point kingPoint, Side side)	Override king สามารถติดกันได้
- void explosion(Point point)	การระเบิด

4.2. Abstract Class Chess960Board extends Board implements CheckMateAble

เกมที่เริ่มต้นด้วยกระดานแบบสุ่ม โดย bishop ต้องอยู่บนช่องคนละสี และ king ต้องอยู่ระหว่าง rook ทั้ง 2 ตัว

Methods

+ boolean isWin(Side side)	เช็คการชนะ
+ boolean winByCheckmate(Side side)	เช็คการชนะจากการรุกจน
+ boolean isDraw(Side side)	เช็คการเสมอ
+ boolean drawCannotMove(Side side)	เช็คการเสมอจากการอัป
+ boolean isCheck(Side side)	เช็คการรุก
+ boolean isCastlingPoint(Side side, Point point)	Override special castling
# ArrayList<Point> castingPoint(Side side)	Override special castling

4.3. Abstract Class HordeBoard extends Board implements CheckMateAble

เกมที่เริ่มต้นด้วยฝูงขาวมาเบี่ย 36 ตัว โดยฝั่งดำจะชนะเมื่อกินเบี่ยทั้งหมด

Methods

+ boolean isWin(Side side)	เช็คการชนะ
+ boolean winByEatenAll(Side side)	เช็คการชนะจากกินเบี่ยขาวทั้งหมด
+ boolean winByCheckmate(Side side)	เช็คการชนะจากการรุกจน
+ boolean isDraw(Side side)	เช็คการเสมอ
+ boolean drawCannotMove(Side side)	เช็คการเสมอจากการอัป
+ boolean isCheck(Side side)	เช็คการรุก
# protected ArrayList<Point> editMovePoint(Point oldPoint, ArrayList<Point> movePoint)	Override avoid check white king

4.4. Abstract Class KingOfHillBoard extends Board implements CheckMateAble

เกมที่สามารถชนะได้โดยการรุกจนหรือ king เดินไป 4 ช่องตรงกลาง

Field

- ArrayList<Point> centerPoint;	4 ช่องตรงกลาง
---------------------------------	---------------

Methods

+ boolean isWin(Side side)	เช็คการชนะ
+ boolean winByCenterKing(Side side)	ชนะโดยการเดินขุนไปตรงกลาง
+ boolean winByCheckmate(Side side)	เช็คการชนะจากการรุกจน
+ boolean isDraw(Side side)	เช็คการเสมอ
+ boolean drawCannotMove(Side side)	เช็คการเสมอจากการอับ
+ boolean isCheck(Side side)	เช็คการรุก

4.5. Abstract Class NormalBoard extends Board implements CheckMateAble

เกมธรรมดา

Methods

+ boolean isWin(Side side)	เช็คการชนะ
+ boolean winByCheckmate(Side side)	เช็คการชนะจากการรุกจน
+ boolean isDraw(Side side)	เช็คการเสมอ
+ boolean drawCannotMove(Side side)	เช็คการเสมอจากการอับ
+ boolean isCheck(Side side)	เช็คการรุก

4.6. Abstract Class ThreeCheckBoard extends Board implements CheckMateAble

เกมที่สามารถชนะได้โดยการรุกจนหรือ รุกฝั่งตรงข้าม 3 รอบ

Fields

- int whiteCheck, blackCheck;	จำนวนการรุกของแต่ละฝั่ง
-------------------------------	-------------------------

Methods

+ boolean isWin(Side side)	เช็คการชนะ
+ boolean winByThreeCheck(Side side)	ชนะโดยการรุก 3 ครั้ง
+ boolean winByCheckmate(Side side)	เช็คการชนะจากการรุกจน
+ boolean isDraw(Side side)	เช็คการเสมอ
+ boolean drawCannotMove(Side side)	เช็คการเสมอจากการอับ
+ boolean isCheck(Side side)	เช็คการรุก
+ void continueMove()	Override นับจำนวนรุก
+ int[] getCheckNumber()	ให้จำนวนการรุกของแต่ละฝั่ง

5. Package logic

5.1. Class Cell

ช่องแต่ละช่องในตาราง

Field

+ Entity myEntity	ตัวหมากในช่องนั้น
-------------------	-------------------

Methods

+ void setEntity(Entity e)	
+ Entity getEntity()	getter & setter
+ void removeEntity()	remove entity

5.2. Class GameController

สร้างกระดานและเชื่อมต่อระหว่าง Board และ Frontend

Fields

+ <u>Board board</u>	board ของเกม
+ <u>Side turn</u>	ตาเดิน
+ <u>Point size</u>	ขนาดกระดาน
+ <u>Point promotionPoint</u>	จุดที่มีการ promotion
+ <u>Side promotionSide</u>	ฝั่งที่มีการ promotion
+ <u>final String[][] normalBoard</u>	กระดานปกติ
+ <u>final String[][] hordeBoard</u>	กระดานสำหรับ hordeBoard

Methods

+ <u>void InitializeMap(String gameType)</u>	สร้างกระดาน
+ <u>int[] randomKey()</u>	สุ่มตัวหมากสำหรับ Chess960
+ <u>int[] put(int[] l, int value, int index)</u>	สำหรับ randomKey
+ <u>Cell[][] getDisplayCellMap()</u>	ให้ cellMap
+ <u>boolean isWin()</u>	เช็คการชนะ
+ <u>boolean isDraw()</u>	เช็คการเสมอ
+ <u>boolean isCheck()</u>	เช็คการเสมอ
+ <u>void startAnimation(Point oldPoint, Point newPoint)</u>	เริ่มการเดิน โดยการเก็บตัวที่เดินและลบตัวที่เดินออกจากกระดาน
+ <u>void continueMove()</u>	เดินต่อโดยนำตัวที่เดินมาที่จุดสิ้นสุด และลบตัวอื่นๆที่ถูกระเบิดสำหรับ atomicBoard
+ <u>ArrayList<Point> moveList(Point p)</u>	ให้ list ของจุดที่เดินได้
+ <u>void setPromotion(Point point, Side side)</u>	กำหนดฝั่งและตำแหน่งการ promotion
+ <u>boolean isPromotion()</u>	มีการ promotion ี่เปล่า

+ <u>void promotion(String piece)</u>	ผู้เล่นเลือกตัว promotion และเปลี่ยน pawn เป็นตัวนั้นๆ
+ <u>boolean isTurn(Point p, Side turn)</u>	ใช่ตาเดินรีเปล่า
+ <u>void nextTurn()</u>	เปลี่ยนตาเดิน
+ <u>Side getTurn()</u>	getter
+ <u>Side getAnotherSide(Side side)</u>	ให้ another side
+ <u>int[] getCheckNumber()</u>	ให้จำนวนการรุกของ ThreeCheckBoard

5.3. Enum Side

ฝั่งของผู้เล่น มีขาวและดำ

WHITE, BLACK

6. Package application

6.1. Class AppManager

ควบคุมการทำงานของโปรแกรม เป็นตัวกลางในการเรียกใช้งานคลาสต่าง ๆ

Fields

- <u>Stage stage</u>	เก็บ stage จาก Main เพื่อการเรียกใช้ภายใน AppManager
- <u>Scene scene</u>	เก็บ scene จาก Main เพื่อการเรียกใช้ภายใน AppManager
- <u>MenuPane menuPane</u>	เก็บ menuPane เพื่อการเรียกใช้ใน class อื่น ๆ
- <u>SettingMenu settingMenu</u>	เก็บ settingMenu เพื่อการเรียกใช้ใน class อื่น ๆ
- <u>SelectModePane selectModePane</u>	เก็บ selectModePane เพื่อการเรียกใช้ใน class อื่น ๆ
- <u>TimeSelectPane timeSelectPane</u>	เก็บ timeSelectPane เพื่อการเรียกใช้ใน class อื่น ๆ
- <u>GamePane gamePane</u>	เก็บ gamePane เพื่อการเรียกใช้ใน class อื่น ๆ
- <u>BoardPane boardPane</u>	เก็บ boardPane เพื่อการเรียกใช้ใน class อื่น ๆ
- <u>PromotionPane promotionPane</u>	เก็บ promotionPane เพื่อการเรียกใช้ใน class อื่น ๆ
- <u>SettingPane setting</u>	เก็บ setting เพื่อการเรียกใช้ใน class อื่น ๆ
- <u>String gameType</u>	เก็บ gameType เพื่อการเรียกใช้ใน class อื่น ๆ
- <u>int spareTime</u>	เก็บ spareTime เพื่อการเรียกใช้ใน class อื่น ๆ

- <u>PlayerStatusDisplay whiteDisplay</u>	เก็บ whiteStatusDisplay เพื่อการเรียกใช้ใน class อื่น ๆ
- <u>PlayerStatusDisplay blackDisplay</u>	เก็บ blackStatusDisplay เพื่อการเรียกใช้ใน class อื่น ๆ
- <u>boolean rotateStatus</u>	เก็บค่า rotate status ใช้ในการ rotate board
- <u>Canvas transitionCanvas</u>	เก็บ canvas ที่ใช้ใน startAnimation()

Methods

<u>+ void showMenu()</u>	set scene root ด้วย menuPane เพื่อแสดง หน้า menu
<u>+ void hidePromotion()</u>	เรียกใช้ hidePromotionPane() ของ promotionPane
<u>+ void showPromotion()</u>	เรียกใช้ showPromotionPane() ของ promotionPane
<u>+ void showSettingMenu()</u>	set scene root ด้วย settingMenu เพื่อ แสดงหน้า setting
<u>+ void showSelectMode()</u>	set scene root ด้วย selectModePane เพื่อ แสดงหน้า select mode
<u>+ void showTimeSelect()</u>	set scene root ด้วย timeSelectPane เพื่อ แสดงหน้า time select
<u>+ void showGamePane()</u>	set scene root ด้วย gamePane เพื่อแสดง หน้าเล่นเกม
<u>+ void setPromotionListener(String text)</u>	เรียก promotion() ใน boardPane เพื่อ set listener ของ promotion button
<u>+ void displayMessage(String message)</u>	เปลี่ยน display message ใน console ตอนเล่นเกม
<u>+ void rotateBoard()</u>	เรียก method rotateBoard() ของ boardPane
<u>+ void setSliderStyle()</u>	set slider style ตอนเข้าหน้า setting menu
<u>+ void stopTimer()</u>	หยุด timer ของทั้งสองฝั่ง
<u>+ void startAnimation(Point start, Point end, Entity entity)</u>	แสดง animation การขยับของตัวหมาก

<u>+ void startCastlingAnimation(Point startKing, Point endKing, Entity king, Point startRook, Point endRook, Entity rook)</u>	แสดง animation ของการ castling
<u>+ void removeTransitionCanvas()</u>	ลบ canvas ที่เกิดจากการแสดง animation
<u>+ void updateSettingMenu()</u>	update setting menu ตาม parameter ใน SoundManager
<u>+ setters and getters for each field as needed</u>	

6.2. Class Main extends Application

ใช้สำหรับการรันโปรแกรม

Methods

<u>+ void start(Stage primaryStage)</u>	set ค่าเริ่มต้นต่าง ๆ สำหรับการรันแอป
<u>+ void main(String[] args)</u>	An entry point of the application

6.3. Class SoundManager

ควบคุมเสียงทั้งหมดในโปรแกรม

Fields

<u>- MediaPlayer clickingSound</u>	เสียง click mouse
<u>- MediaPlayer bgm</u>	Background music สำหรับหน้า menu
<u>- MediaPlayer clockTick</u>	เสียงนาฬิกานับถอยหลังตอนที่เวลาใกล้หมด
<u>- MediaPlayer wrongSelected</u>	เสียงเตือนตอนที่เลือกตัวมากฝั่งตรงข้ามหรือไม่ยอมเลือกตัว promotion
<u>- MediaPlayer promotionSound</u>	sound effect หลังจากกดเลือกตัว promotion
<u>- MediaPlayer winning</u>	sound effect ตอนจบเกม ไม่ว่าจะชนะหรือเสมอ
<u>- boolean soundEffectStatus</u>	เก็บ status ของ sound effect ว่าจะให้เล่นเสียงเอฟเฟคใหม่

Methods

<u>+ void playClickingSound()</u>	เล่นเสียงคลิกเม้าส์โดยใช้ thread
-----------------------------------	----------------------------------

<u>+ void playWrongSelected()</u>	เล่นเสียง wrong selected โดยใช้ thread
<u>+ void playPromotionSound()</u>	เล่นเสียง promotion sound โดยใช้ thread
<u>+ void playMenuBgm()</u>	เล่นเสียง menu bgm แบบวนลูป โดยใช้ thread
<u>+ void stopMenuBgm()</u>	หยุดการเล่นเสียง bgm
<u>+ void playClockTick()</u>	เล่นเสียงนาฬิกาจับเวลา โดยใช้ thread
<u>+ void stopClockTick()</u>	หยุดเสียงนาฬิกาจับเวลา
<u>+ void playWinningSound()</u>	เล่นเสียงตอนจบเกมโดยใช้ thread
<u>+ double getMenuBgmVolume()</u> <u>+ void setMenuBgmVolume(double volume)</u>	getter/setter ของ bgm
<u>+ double getSoundEffectVolume()</u> <u>+ setSoundEffectVolume(double volume)</u>	getter/setter volume ของ clickingSound, clockTick, wrongSelected, promotionSound, winning
<u>+ boolean getSoundEffectStatus()</u> <u>+ setSoundEffectStatus(boolean soundEffectStatus)</u>	getter/setter ของ soundEffectStatus

7. Package application.board

7.1. Class BoardCell extends Pane

ช่องแต่ละช่องในตารางหมากรุก

Fields

- Cell myCell	เก็บ cell เพื่อนำค่าจาก cell มาใช้ในการแสดงผล
- Point p	เก็บตำแหน่งพิกัดของ BoardCell แต่ละอัน
- Color color	สีพื้นหลังของ BoardCell
- boolean isClicked	true ถ้า BoardCell ที่ถูกคลิกมีตัวหมากอยู่
- boolean moveable	true ถ้าตัวหมากที่ถูกคลิกสามารถเดินไปที่ช่องนั้นได้

Constructor

+ BoardCell(Cell cell, Point p, Color color)	<ul style="list-style-type: none"> - ตั้งค่าเริ่มต้นให้กับตัวแปรแต่ละตัว - ตั้งขนาดไว้ที่ 60*60 - ตั้ง background หากใน cell มีหมากให้ใช้รูปหมาก หากไม่มีให้ตั้งพื้นหลังเป็นสี color - หากเป็น โหมด King of the hill ช่องที่มีพิกัด (3,3),(3,4),(4,3),(4,4) ให้ตั้งค่า border เป็นสีแดง
--	---

Methods

+ boolean hasEntity()	return true ถ้าใน cell มีตัวหมากอยู่
+ void setBackgroundTileColor()	ตั้งค่าพื้นหลังด้วย color
+ void setBackgroundTileColor(Image image)	ตั้งค่าพื้นหลังด้วย color และรูป image
+ void setBackgroundTileColor(Image redDot, Image entity)	ตั้งค่าพื้นหลังด้วย color และ redDot กับ entity หากช่องนั้น moveable เป็น true และมีหมากฝั่งตรงข้ามอยู่
+ void update()	ตั้งค่าพื้นหลังตามสถานะของ cell และตั้งค่า isClicked และ moveable เป็น false
+ void setMyCell(Cell myCell) + Cell getMyCell()	setter/getter ของ myCell
+ void setP(Point p) + Point getP()	setter/getter ของ p
+ boolean isClicked() + void setClicked(boolean isClicked)	setter/getter ของ isClicked
+ boolean isMoveable() + void setMoveable(boolean moveable)	setter/getter ของ moveable

7.2. Class BoardPane extends GridPane

ตัวกระดานหมากรุก นำ BoardCell แต่ละอันมาเรียงใน GridPane

Fields

- ObservableList<BoardCell> boardCellList	เก็บลิสต์ของ BoardCell
- Cell[][] cellMap	เก็บ array ของ cell ที่เอามาจาก GameController
- int row	เก็บค่าแถวของกระดาน เป็นค่า final มีค่า 8
- int column	เก็บค่าคอลัมน์ของกระดาน เป็นค่า final มีค่า 8
- Color whiteTile	ค่าสีช่องสีขาว(เทาอ่อน)
- Color blackTile	ค่าสีช่องสีดำ(เทาเข้ม)
- BoardCell bc	ใช้ตอน initialize BoardCell
- Point currentSelectedPoint	เก็บพิกัดของตัวหมากที่ถูกเลือก
- ArrayList<Point> currentSelectedMoveList	เก็บลิสต์ของพิกัดช่องที่หมากที่ถูกเลือกสามารถเดินไปได้
- boolean move	true ถ้ามีการเดินหมาก
- String promotionPiece	เก็บชื่อของหมากที่เบี้ยจะทำการ promotion

Constructor

+ BoardPane(String gameType)	<ul style="list-style-type: none">- initialize map ผ่าน GameController ตาม gamType ที่ใส่เข้ามา- initialize cellMap โดยเรียกผ่าน GameController- เรียก method createBoardCell()- เรียก method setBoardCellListener()
------------------------------	---

Methods

- void createBoardCell()	<ul style="list-style-type: none">- สร้าง BoardCell โดยใช้ cellMap และพิกัด row column แล้วเพิ่มเข้าไปใน boardCellList- เพิ่ม BoardCell ที่สร้างเข้าไปพิกัดแต่ละจุดของ BoardPane ตามพิกัดของแต่ละ BoardCell
- void setBoardCellListener()	ตั้งค่า listener ของแต่ละ boardCell โดยเรียก addOnClickHandler(BoardCell myBoardCell)

- void addOnClickHandler(BoardCell myBoardCell) throws Exception	<p>ตั้งค่า listener ให้ BoardCell</p> <ul style="list-style-type: none"> - ถ้ามีตัวหมากอยู่ให้ highlight และ เรียก method showWalkPath() และ set isClicked เป็น true - ถ้ามีตัวหมากและ isClicked เป็น true ให้ เรียก method updateBoard() - ถ้าตัวหมากสามารถเดินไปช่องนั้นได้ให้ทำการเดินหมากไปที่ช่องนั้น
- void disableBoardCell()	- ทำให้ BoardCell ทั้งหมดไม่สามารถกดได้ระหว่างการแสดง moving animation
- void enableBoardCell()	- ทำให้ BoardCell ทั้งหมดกดได้อีกครั้งตอนเริ่มตาถัดไป
- boolean checkEndGame()	- ตรวจสอบว่ามีการชนะเกมหรือการเสมอกเกิดขึ้นหรือยัง ถ้ามีให้เรียก method showEndGameWindow(String text)
- void showWalkPath()	- แสดงช่องที่ตัวหมากสามารถเดินไปได้ และ set moveable ของช่องนั้น ๆ เป็น true
+ void showEndGameWindow(String text)	- แสดงหน้าจบเกม
+ void updateBoard(BoardCell myBoardCell)	- เรียก update ของ BoardCell ทุกตัว ยกเว้น myBoardCell
+ void updateBoard()	- เรียก update ของ BoardCell ทุกตัว ยกเว้น myBoardCell
+ void rotateBoard()	- rotate board 1 ครั้ง
+ void promotion(String text)	- listener ของ Promotion button

8. Package application.console

8.1. Class GamePane extends StackPane

หน้าจอตอนเล่นเกม ประกอบด้วยตัวกระดานและ console

Field

- Console console	console ของเกม
-------------------	----------------

Constructor

+ GamePane(BoardPane boardPane)	<ul style="list-style-type: none">- ตั้งค่าพื้นหลังด้วยสีเทา- ตั้งค่าความสูงที่ 500- set spacing ให้เป็น 20- add boardPane และ console เข้าไปใน children ของ gameBox
---------------------------------	---

Methods

+ void rotateStatusDisplay()	เรียก rotateStatusDisplay() ของ console
------------------------------	---

8.1.1. Class Console extends VBox

Constructor

+ Console()	<ul style="list-style-type: none">- สร้าง whiteDisplay blackDisplay gameMode Label- console ทั้งหมดจะประกอบด้วย promotionPane setting gameMode blackDisplay และ whiteDisplay
-------------	---

Methods

+ void rotateStatusDisplay()	สลับตำแหน่งระหว่าง blackDisplay และ whiteDisplay
------------------------------	--

8.2. Class PlayerStatusDisplay extends VBox

แสดงเวลาและเทิร์นของ player

Fields

- Label turn	แสดง turn
- Side side	เก็บค่า Side ว่าอยู่ฝั่งขาวหรือดำ
- int spareTime	เวลาทั้งหมดที่ใช้ได้ในแต่ละเกมของแต่ละฝั่ง
- int timePerTurn	เวลาในแต่ละเทิร์น
- boolean hasTimeLimit	true ถ้าเวลาที่เลือกไม่ใช่ 0 (No time Limit)
- Thread timerThread	ตัวจับเวลา
- Text spareTimeText	text ของ spareTime
- Text timePerTurnText	text ของ timePerTurn
- HBox timePane	ช่องแสดงเวลา

Constructor

+ PlayerStatusDisplay(Side side)	ตั้งค่าเริ่มต้นให้ตัวแปรต่าง ๆ
----------------------------------	--------------------------------

Methods

+ void rotateDisplay()	สลับตำแหน่งระหว่างช่องแสดงเวลาและ Label turn
+ void startTurn()	เริ่ม turn โดยถ้ามีเวลาจำกัดจะเริ่มนับเวลาลอยหลัง
+ void endTurn()	ถ้ามีเวลาจำกัด หยุดนาฬิกาจับเวลาและรีเซ็ต timePerTurn เป็น 30 หลังจากนั้นเรียก setTurnText(false)
+ void stop() throws Exception	หยุดนาฬิกาจับเวลา
+ void update()	อัปเดต spareTimeText และ timePerTurnText
- void setSpareTimeText()	ตั้งค่า spareTimeText ตาม spareTime
- void setTimePerTurnText()	ตั้งค่า timePerTurnText ตาม timePerTurn
+ void setTurnText(boolean isTurn)	- หาก isTurn เป็น true ให้ตั้งค่า turn ด้วย side + "TURN" - หากเป็น false ให้ตั้งค่าด้วย side.toString()

8.3. Class PromotionButton extends Button

ปุ่มสำหรับ promote pawn

Fields

- String text	สำหรับตั้งค่า background
- String imgUrl	รูปสำหรับ background

Constructor

+ PromotionButton(String text)	<ul style="list-style-type: none">- ตั้งค่าเริ่มต้นของ text- ตั้งค่าขนาด 40*40- เรียก method setBackgroundWithImage()- เรียก method setListener()
--------------------------------	--

Methods

+ void setListener()	<p>ตั้งค่า listener โดย</p> <ul style="list-style-type: none">- ถ้าคลิก ให้เล่นเสียง promotion และซ่อนแถบ promotion พร้อม ทั้งเปลี่ยนตัวเบี้ยให้เป็นตัวที่เลือก- ถ้าเอาเมาส์ชี้ให้ highlight- ถ้าเอาเมาส์ออกให้ unhighlight
- void highlightButton()	ตั้งค่า border เป็นสีแดง
- void unhighlightButton()	ตั้งค่า border เป็นสีเทา
- void setBackgroundWithImage()	ตั้งค่าพื้นหลังตาม text

8.4. Class PromotionPane extends VBox

แถบสำหรับ promote เบี้ย และแสดง warning message

Fields

- Text text	ใช้แสดง warning message
- VBox promotion	ช่องเลือก promotion
- HBox selection	แถบปุ่ม promotionButton

Constructor

+ PromotionPane()	ตั้งค่าเริ่มต้นต่าง ๆ และใส่เข้าไปใน PromotionPane
-------------------	--

Methods

+ void hidePromotionPane()	ลบช่องเลือก promotion ออกจาก PromotionPane
+ void showPromotionPane()	เพิ่มช่องเลือก promotion เข้าไปใน PromotionPane และตั้งค่า background ของปุ่ม
+ void setMessage(String message)	set warning message

8.5. Class SettingPane extends VBox

แถบ setting ใน console ประกอบด้วย rotate และ sound effect

Constructor

+ SettingPane()	สร้าง rotate check box และ sound effect check box และ return button รวมถึงตั้งค่าเริ่มต้นต่าง ๆ
-----------------	---

Methods

- void setRotationListener(MyCheckBox rotation)	ตั้งค่า listener ของ rotate check box โดยถ้าคลิกถูก จะทำการ rotate board ทุกครั้งที่เปลี่ยนเทิร์น
- void setSoundEffectListener(MyCheckBox clickSound)	ตั้งค่า listener ของ sound effect check box โดยถ้าคลิกถูก จะเล่นเสียง sfx หากทำ action ต่าง ๆ
- void setReturnBtnListener(ReturnButton returnBtn)	ตั้งค่า listener ของ return button ให้สามารถกดย้อนกลับ ไปหน้าเมนูได้ตลอดเวลา

8.5.1. Class MyCheckBox extends CheckBox

ตัว check box ใน setting console

Constructor

+ MyCheckBox(String text)	ตั้งค่าเริ่มต้นของ check box
---------------------------	------------------------------

8.5.2. Class ReturnButton extends MyButton

ปุ่มกดย้อนกลับไปหน้าจอเมนู

Field

- Color color	สีพื้นหลังของ return button
---------------	-----------------------------

Constructor

+ ReturnButton(String text, double fontSize)	ตั้งค่าเริ่มต้นต่าง ๆ ของ return button
--	---

Methods

# void setMouseEnteredTextFont()	override method ของ MyButton
# void setMouseExitedTextFont()	override method ของ MyButton

9. Package application.menu

9.1. Class MenuPane extends StackPane implements UnchangeableBackground

หน้าจอที่แสดงตอนเข้าเกม

Constructor

+ MenuPane()	ตั้งค่าต่าง ๆ ของหน้าจอ ซึ่งมีส่วนประกอบหลัก ๆ คือ - ปุ่ม Play กดเพื่อเล่นเกม - ปุ่ม Setting ใช้สำหรับตั้งค่าเสียง - ปุ่ม Exit ใช้สำหรับออกเกม
--------------	---

Methods

- void setTitleImage(GraphicsContext gc)	ตั้งค่ารูปไอคอนของเกม
+ void setBackgroundWithImage()	ตั้งค่าพื้นหลังของหน้าจอ
- void setExitButtonListener(MyButton exitButton)	set listener ให้ปุ่ม Exit
- void setPlayButtonListener(MyButton playButton)	set listener ให้ปุ่ม Play
- void setSettingButtonListener(MyButton settingButton)	set listener ให้ปุ่ม Setting

9.2. Class MyButton extends Button implements ChangeableBackground

base ของปุ่มต่าง ๆ ในเกม

Field

# double fontSize	ขนาดของ font ตัวอักษรบนปุ่ม
-------------------	-----------------------------

Constructor

+ MyButton(String text, double fontSize)	ตั้งค่าพื้นฐานต่าง ๆ ของปุ่ม set พื้นหลังด้วยรูปที่เตรียมไว้
--	--

Methods

# void setListener()	set listener การเลื่อนเมาส์เข้า/ออกปุ่ม
# void setMouseEnteredTextFont()	เมื่อเมาส์เลื่อนมาทับปุ่ม ให้เพิ่มขนาดตัวอักษรและ highlight ปุ่ม
# void setMouseExitedTextFont()	เมื่อเลื่อนเมาส์ออกให้ลดขนาดตัวอักษรกลับมาเท่าเดิมและ unhighlight ปุ่ม
+ void setBackgroundWithImage(Image img)	ตั้งค่าพื้นหลังด้วยรูป

9.3. Class SelectModeButton extends MyButton

ปุ่มเลือกโหมดที่ต้องการเล่น

Field

- String gameType	เก็บโหมดของเกม
-------------------	----------------

Constructor

+ SelectModeButton(String text, double fontSize)	ตั้งค่าพื้นฐานของปุ่ม และ set gameType ของปุ่มตาม text ที่ได้เข้ามา
--	---

Methods

- void setSelectListener()	set listener การคลิกให้ปุ่ม โดยการเรียก setGame()
- void setGame()	ตั้งค่าเกมเบื้องต้นและสร้าง BoardPane ให้ AppManager ตาม gameType ของปุ่ม

9.4. Class SelectModePane extends VBox implements UnchangeableBackground

หน้าเมนู select mode หลังจากกดปุ่ม play

Constructor

+ SelectModePane()	ตั้งค่าพื้นฐานต่าง ๆ ใส่ปุ่มเลือกโหมดทั้งหมด 6 โหมด ประกอบด้วย normal, atomic, king of the hill, three check, chess960 และ horde
--------------------	--

Methods

- void setBackgroundWithImage()	ตั้งค่าพื้นหลังด้วยรูป
- void setReturnBtnListener(MyButton returnBtn)	ตั้งค่า listener ของปุ่ม return to menu

9.5. Class SettingMenu extends VBox implements UnchangeableBackground

หน้า setting ใช้ปรับเสียง

Fields

- Slider bgmSlider	ตัวสไลด์เพิ่มลดเสียง bgm
- Label bgmValue	ค่าความดังของ bgm
- SoundButton bgmSoundButton	ปุ่มกด mute/unmute ของ bgm
- Slider sfxSlider	ตัวสไลด์เพิ่มลดเสียง sfx
- Label sfxValue	ค่าความดังของ sfx
- SoundButton sfxSoundButton	ปุ่มกด mute/unmute ของ sfx

Constructor

+ SettingMenu()	ตั้งค่าพื้นฐานต่าง ๆ initialize slider, value และ sound button
-----------------	--

Methods

- void setBgmSliderListener()	ตั้งค่า listener ให้กับ bgm slider ให้เพิ่ม/ลดเสียงหากกดเลื่อนซ้ายขวา
- void setSoundEffectSliderListener()	ตั้งค่า listener ให้กับ sfx slider ให้เพิ่ม/ลดเสียงหากกดเลื่อนซ้ายขวา

- void setReturnBtnListener(MyButton returnBtn)	ตั้งค่า listener ให้กับปุ่ม return to menu
+ void setBackgroundWithImage()	ตั้งค่าพื้นหลังด้วยรูปภาพ
+ void update()	update status ของ sfx ตามค่า soundEffectStatus ของ SoundManager
+ Slider getBgmSlider()	getter ของ bgmSlider
+ Slider getSfxSlider()	getter ของ sfxSlider

9.5.1. Class SoundButton extends Button implements ChangeableBackground

ปุ่มกด mute/unmute

Field

- boolean isMuted	true ถ้ากด mute
-------------------	-----------------

Constructor

+ SoundButton(String sound, Slider slider, Label value)	ตั้งค่าพื้นฐานของปุ่มและ set listener ให้ mute/unmute เมื่อกดปุ่ม
---	---

Methods

+ void setBackgroundImageByVolume(Slider slider)	ตั้งค่ารูปพื้นหลังของปุ่มตามระดับความดัง
+ void setBackgroundWithImage(Image img)	ตั้งค่าพื้นหลังด้วยรูปลำโพง muted
+ void setMuted(boolean isMuted)	setter ของ isMuted

9.6. Class TimeSelectButton extends MyButton

Field

- int spareTime	time limit ของแต่ละปุ่ม
-----------------	-------------------------

Constructor

+ TimeSelectButton(String text, double fontSize)	ตั้งค่าพื้นฐานต่าง ๆ และตั้งค่า spareTime ตาม text ที่ใส่เข้ามา
--	---

Methods

- void setSelectListener()	ตั้งค่า listener ของปุ่มด้วย method startGame()
- void startGame()	ตั้งค่าต่าง ๆ ของเกมต่อจาก select mode สร้าง GamePane และเริ่มเกม

9.7. Class TimeSelectPane extends VBox implements UnchangeableBackground

หน้าเมนูเลือก time limit

Constructor

+ TimeSelectPane()	ตั้งค่าพื้นฐานต่าง ๆ ใส่ปุ่มเลือกเวลาจำกัดโดยมีเวลาให้เลือกดังนี้ 30 นาที, 15 นาที, 5 นาที และ ไม่มีเวลาจำกัด
--------------------	--

Methods

+ void setBackgroundWithImage()	ตั้งค่าพื้นหลังด้วยรูป
- void setReturnBtnListener(MyButton returnBtn)	ตั้งค่า listener ให้ปุ่ม return to select mode

9.8. Interface ChangeableBackground

สำหรับคลาสที่มีการเปลี่ยนพื้นหลังโดยใช้รูปตามสถานการณ์

Methods

+ void setBackgroundWithImage(Image img)	ตั้งค่าพื้นหลังตาม img
--	------------------------

9.9. Interface UnchangeableBackground

สำหรับคลาสที่มีการตั้งพื้นหลังโดยใช้รูป และไม่มีการเปลี่ยนอีก

Methods

+ void setBackgroundWithImage()	ตั้งค่าพื้นหลังด้วยรูป
---------------------------------	------------------------

10. Package myException

10.1. Class MyException extends Exception

abstract class MyException ใช้ป้องกันความผิดพลาดจาก user

Field

- String message	ข้อความแสดงปัญหาที่เกิดขึ้น
------------------	-----------------------------

Constructor

+ MyException(String message)	ตั้งค่า message ตาม message ที่ได้มา
-------------------------------	--------------------------------------

Methods

+ String getMessage()	getter ของ message
-----------------------	--------------------

10.2. Class IsPromotingException extends MyException

exception หากไม่ยอมเลือกตัว promotion ของเบี้ย

Constructor

+ IsPromotingException(String message)	รับ message ส่งไปให้ super
--	----------------------------

10.3. Class WrongPieceException extends MyException

exception หากกดเลือกหมากของฝ่ายตรงข้าม

Constructor

+ WrongPieceException(String message)	รับ message ส่งไปให้ super
---------------------------------------	----------------------------

11. Package Resource

11.1. Class Resource

เตรียมโหลด path ของ resource ต่าง ๆ มาเก็บไว้

Fields

+ <u>String BACKGROUND</u>	เก็บ path ของ background.jpg
+ <u>String ICON</u>	เก็บ path ของ icon.png
+ <u>String BUTTON_FRAME</u>	เก็บ path ของ buttonFrame.png
+ <u>String HIGHLIGHT_BUTTON_FRAME</u>	เก็บ path ของ highlightButtonFrame.png
+ <u>String VERY_LOW_VOLUME_IMAGE</u>	เก็บ path ของ veryLowVolume.png
+ <u>String LOW_VOLUME_IMAGE</u>	เก็บ path ของ lowVolume.png
+ <u>String MID_VOLUME_IMAGE</u>	เก็บ path ของ midVolume.png
+ <u>String HIGH_VOLUME_IMAGE</u>	เก็บ path ของ highVolume.png
+ <u>String MUTE_VOLUME_IMAGE</u>	เก็บ path ของ muteVolume.png
+ <u>String BUTTON_CLICK</u>	เก็บ path ของ buttonClick.mp3
+ <u>String CLOCK_TICKING</u>	เก็บ path ของ clockTicking.wav
+ <u>String WRONG_SELECTED</u>	เก็บ path ของ wrongPieceSelectedSound.mp3
+ <u>String GAME_MENU</u>	เก็บ path ของ gameMenuLoop.mp3
+ <u>String PROMOTION_SOUND</u>	เก็บ path ของ promotionSound.mp3
+ <u>String WINNING_SOUND</u>	เก็บ path ของ winningSound.wav
+ <u>String ROMAN_FONT</u>	เก็บ path ของ Roman.ttf

11.2.Class Sprites

เตรียมโหลด path ของ sprites ตัวหมาก มาเก็บไว้

Fields

<u>+ String BLANK</u>	เก็บ String "blank"
<u>+ String WALKPATH</u>	เก็บ path ของ redDot.png
<u>+ String W_KING</u>	เก็บ path ของ W_King.png
<u>+ String W_QUEEN</u>	เก็บ path ของ W_Queen.png
<u>+ String W_BISHOP</u>	เก็บ path ของ W_Bishop.png
<u>+ String W_KNIGHT</u>	เก็บ path ของ W_Knight.png
<u>+ String W_ROOK</u>	เก็บ path ของ W_Rook.png
<u>+ String W_PAWN</u>	เก็บ path ของ W_Pawn.png
<u>+String B_KING</u>	เก็บ path ของ B_King.png
<u>+ String B_QUEEN</u>	เก็บ path ของ B_Queen.png
<u>+ String B_BISHOP</u>	เก็บ path ของ B_Bishop.png
<u>+ String B_KNIGHT</u>	เก็บ path ของ B_Knight.png
<u>+ String B_ROOK</u>	เก็บ path ของ B_Rook.png
<u>+ String B_PAWN</u>	เก็บ path ของ B_Pawn.png
<u>+ String HIGHLIGHT_W_KING</u>	เก็บ path ของ highlight_W_King.png
<u>+ String HIGHLIGHT_W_QUEEN</u>	เก็บ path ของ highlight_W_Queen.png
<u>+ String HIGHLIGHT_W_BISHOP</u>	เก็บ path ของ highlight_W_Bishop.png
<u>+ String HIGHLIGHT_W_KNIGHT</u>	เก็บ path ของ highlight_W_Knight.png
<u>+ String HIGHLIGHT_W_ROOK</u>	เก็บ path ของ highlight_W_Rook.png
<u>+ String HIGHLIGHT_W_PAWN</u>	เก็บ path ของ highlight_W_Pawn.png
<u>+ String HIGHLIGHT_B_KING</u>	เก็บ path ของ highlight_B_King.png
<u>+ String HIGHLIGHT_B_QUEEN</u>	เก็บ path ของ highlight_B_Queen.png

+ <u>String HIGHLIGHT_B_BISHOP</u>	เก็บ path ของ highlight_B_Bishop.png
+ <u>String</u> <u>HIGHLIGHT_B_KNIGHT</u>	เก็บ path ของ highlight_B_Knight.png
+ <u>String HIGHLIGHT_B_ROOK</u>	เก็บ path ของ highlight_B_Rook.png
+ <u>String HIGHLIGHT_B_PAWN</u>	เก็บ path ของ highlight_B_Pawn.png