

Chess Evolved 2 - technická dokumentácia

Generované programom Doxygen 1.10.0

1 Register hierarchie tried	1
1.1 Hierarchia tried	1
2 Register tried	3
2.1 Zoznam tried	3
3 Register súborov	5
3.1 Zoznam súborov	5
4 Dokumentácia tried	7
4.1 Dokumentácia triedy BoardController	7
4.1.1 Detailný popis	9
4.1.2 Dokumentácia k metódam	9
4.1.2.1 AddMovement()	9
4.1.2.2 AddPiece()	9
4.1.2.3 ClearGhost()	9
4.1.2.4 ClearHighlightedSquare()	9
4.1.2.5 ClearMovements()	10
4.1.2.6 ClearPieces()	10
4.1.2.7 CreateInitialPosition()	10
4.1.2.8 GetPiece()	10
4.1.2.9 GetPieceObject()	10
4.1.2.10 InitBoard()	11
4.1.2.11 MoveGameObject()	11
4.1.2.12 MovePiece()	11
4.1.2.13 OnMovementClicked()	12
4.1.2.14 OnPieceClicked()	12
4.1.2.15 PerformMove()	12
4.1.2.16 PerformPromotion()	12
4.1.2.17 SetBoardOrientation()	13
4.1.2.18 SetCheck()	13
4.1.2.19 SetGhost()	13
4.1.2.20 SetHighlightedSquare()	13
4.1.2.21 SetLegalMoves()	13
4.1.2.22 SetPly()	14
4.2 Dokumentácia triedy Chess	14
4.2.1 Detailný popis	15
4.2.2 Dokumentácia k členským enumeráciám	15
4.2.2.1 AI	15
4.2.2.2 Color	15
4.2.2.3 Essence	15
4.2.2.4 MovementType	15
4.2.2.5 PieceType	16

4.2.3 Dokumentácia k metódam	16
4.2.3.1 GetMovementImage()	16
4.2.3.2 GetPieceImage()	16
4.3 Dokumentácia triedy ClickableDropdown	17
4.3.1 Detailný popis	17
4.3.2 Dokumentácia k metódam	17
4.3.2.1 OnPointerClick()	17
4.4 Dokumentácia triedy CreateGameOverlay	18
4.4.1 Detailný popis	19
4.4.2 Dokumentácia k metódam	19
4.4.2.1 InitializeSettings()	19
4.4.2.2 OnMalakhColorSet()	20
4.4.2.3 OnMalakhRandomPiecesToggled()	20
4.4.2.4 OnPlayClick()	20
4.4.2.5 OnPlayerColorSet()	20
4.4.2.6 OnPlayerRandomPiecesToggled()	20
4.4.2.7 OnReturnClick()	22
4.4.2.8 SetEngineController()	22
4.4.2.9 SetPieceInfo()	22
4.5 Dokumentácia triedy EngineController	22
4.5.1 Detailný popis	23
4.5.2 Dokumentácia k metódam	23
4.5.2.1 MovePiece()	23
4.5.2.2 OnApplicationQuit()	24
4.5.2.3 ParseMove()	24
4.5.2.4 ReceiveEngineResponses()	24
4.5.2.5 RequestAvailableMoves()	24
4.5.2.6 RequestBestMove()	24
4.5.2.7 SendCommand()	25
4.5.2.8 Start()	25
4.5.2.9 StartEngine()	25
4.5.2.10 StartGame()	25
4.5.2.11 StopEngine()	25
4.5.2.12 Update()	25
4.5.3 Dokumentácia k dátovým členom	26
4.5.3.1 engineProcess	26
4.6 Dokumentácia triedy GameCanvas	26
4.6.1 Detailný popis	27
4.6.2 Dokumentácia k metódam	27
4.6.2.1 AddTurn()	27
4.6.2.2 ClearTurns()	27
4.6.2.3 OnBestMoveReceived()	28

4.6.2.4 OnCheckReceived()	28
4.6.2.5 OnExportClick()	28
4.6.2.6 OnGameCreated()	28
4.6.2.7 OnHelpButtonClicked()	28
4.6.2.8 OnLegalMovesReceived()	29
4.6.2.9 OnPlayClick()	29
4.6.2.10 OnPlayerMove()	29
4.6.2.11 OnPromotionChosen()	29
4.6.2.12 OnPromotionRequested()	30
4.6.2.13 OnResultReceived()	30
4.6.2.14 OnReturnClick()	30
4.6.2.15 SetBlackPly()	30
4.6.2.16 SetGame()	30
4.6.2.17 SetTurn()	31
4.7 Dokumentácia triedy Chess.GameSettings	31
4.7.1 Detailný popis	31
4.7.2 Dokumentácia konštruktoru a deštruktoru	32
4.7.2.1 GameSettings()	32
4.7.3 Dokumentácia k dátovým členom	32
4.7.3.1 ai	32
4.7.3.2 malakhBishop	32
4.7.3.3 malakhColor	33
4.7.3.4 malakhKnight	33
4.7.3.5 malakhPawn	33
4.7.3.6 malakhRook	33
4.7.3.7 playerBishop	33
4.7.3.8 playerColor	33
4.7.3.9 playerKnight	33
4.7.3.10 playerPawn	33
4.7.3.11 playerRook	34
4.8 Dokumentácia triedy Ghost	34
4.8.1 Detailný popis	34
4.8.2 Dokumentácia konštruktoru a deštruktoru	34
4.8.2.1 Ghost()	34
4.8.3 Dokumentácia k dátovým členom	35
4.8.3.1 parent	35
4.8.3.2 x	35
4.8.3.3 y	35
4.9 Dokumentácia triedy GhostObject	35
4.9.1 Detailný popis	35
4.9.2 Dokumentácia k vlastnosti	36
4.9.2.1 Ghost	36

4.10 Dokumentácia triedy Mobility	36
4.10.1 Detailný popis	36
4.10.2 Dokumentácia konštruktoru a deštruktoru	36
4.10.2.1 Mobility()	36
4.10.3 Dokumentácia k dátovým členom	37
4.10.3.1 direction_x	37
4.10.3.2 direction_y	37
4.10.3.3 limit	37
4.10.3.4 start_x	37
4.10.3.5 start_y	37
4.10.3.6 type	38
4.11 Dokumentácia triedy Move	38
4.11.1 Detailný popis	38
4.11.2 Dokumentácia konštruktoru a deštruktoru	38
4.11.2.1 Move() [1/2]	38
4.11.2.2 Move() [2/2]	39
4.11.3 Dokumentácia k metódam	39
4.11.3.1 ClearHasty()	39
4.11.3.2 ClearVigilant()	39
4.11.3.3 SetHasty()	39
4.11.3.4 SetVigilant()	40
4.11.4 Dokumentácia k dátovým členom	40
4.11.4.1 castling	40
4.11.4.2 hasty	40
4.11.4.3 hastyX	40
4.11.4.4 hastyY	40
4.11.4.5 vigilant	40
4.11.4.6 x1	40
4.11.4.7 x2	41
4.11.4.8 y1	41
4.11.4.9 y2	41
4.12 Dokumentácia triedy Movement	41
4.12.1 Detailný popis	41
4.12.2 Dokumentácia konštruktoru a deštruktoru	41
4.12.2.1 Movement()	41
4.12.3 Dokumentácia k dátovým členom	42
4.12.3.1 move	42
4.12.3.2 owner	42
4.12.3.3 type	42
4.13 Dokumentácia triedy MovementObject	42
4.13.1 Detailný popis	43
4.13.2 Dokumentácia k metódam	43

4.13.2.1 OnPointerClick()	43
4.13.3 Dokumentácia k vlastnosti	43
4.13.3.1 Movement	43
4.14 Dokumentácia triedy Piece	43
4.14.1 Detailný popis	44
4.14.2 Dokumentácia konštruktoru a deštruktoru	44
4.14.2.1 Piece()	44
4.14.3 Dokumentácia k dátovým členom	44
4.14.3.1 availableMoves	44
4.14.3.2 color	44
4.14.3.3 essence	45
4.14.3.4 type	45
4.14.3.5 x	45
4.14.3.6 y	45
4.15 Dokumentácia triedy PieceInfoController	45
4.15.1 Detailný popis	46
4.15.2 Dokumentácia k metódam	46
4.15.2.1 CreatePiece()	46
4.15.2.2 OnCloseButtonClicked()	46
4.15.2.3 UpdateDisplayedPiece()	46
4.15.3 Dokumentácia k vlastnosti	47
4.15.3.1 MyPiece	47
4.16 Dokumentácia triedy PieceObject	47
4.16.1 Detailný popis	48
4.16.2 Dokumentácia k metódam	48
4.16.2.1 OnPointerClick()	48
4.16.3 Dokumentácia k dátovým členom	49
4.16.3.1 highlighted	49
4.16.4 Dokumentácia k vlastnosti	49
4.16.4.1 HighlightColor	49
4.16.4.2 Piece	49
4.16.4.3 Selected	49
4.17 Dokumentácia triedy PlayerInfoController	49
4.17.1 Detailný popis	50
4.17.2 Dokumentácia k metódam	50
4.17.2.1 OnHelpButtonClicked()	50
4.17.2.2 SetPieces()	51
4.18 Dokumentácia triedy PromotionController	51
4.18.1 Detailný popis	52
4.18.2 Dokumentácia k metódam	52
4.18.2.1 OnPromotionBtnClicked()	52
4.18.2.2 SetColor()	52

4.18.2.3 SetPieces()	53
4.19 Dokumentácia triedy TitleCanvas	53
4.19.1 Detailný popis	53
4.19.2 Dokumentácia k metódam	54
4.19.2.1 OnGameCreated()	54
4.19.2.2 OnPlayClick()	54
4.19.2.3 OnQuitClick()	54
4.20 Dokumentácia triedy Turn	54
4.20.1 Detailný popis	55
4.20.2 Dokumentácia konštruktoru a deštruktoru	55
4.20.2.1 Turn()	55
4.20.3 Dokumentácia k metódam	55
4.20.3.1 GetBlackPly()	55
4.20.3.2 GetPly()	56
4.20.3.3 GetWhitePly()	56
4.20.3.4 SetBlackPly()	56
4.20.3.5 SetWhitePly()	56
4.20.4 Dokumentácia k dátovým členom	57
4.20.4.1 blackMove	57
4.20.4.2 blackMovementType	57
4.20.4.3 blackPieceType	57
4.20.4.4 blackPromotion	57
4.20.4.5 turnCounter	57
4.20.4.6 whiteMove	58
4.20.4.7 whiteMovementType	58
4.20.4.8 whitePieceType	58
4.20.4.9 whitePromotion	58
4.21 Dokumentácia triedy TurnObject	58
4.21.1 Detailný popis	59
4.21.2 Dokumentácia k metódam	59
4.21.2.1 UpdateBlackPly()	59
4.21.2.2 UpdateDisplayedText()	59
4.21.2.3 UpdateWhitePly()	59
4.21.3 Dokumentácia k vlastnosti	60
4.21.3.1 MyTurn	60
5 Dokumentácia súborov	61
5.1 Dokumentácia súboru Chess.cs	61
5.1.1 Detailný popis	61
5.2 Dokumentácia súboru Ghost.cs	61
5.2.1 Detailný popis	62
5.3 Dokumentácia súboru Mobility.cs	62

5.3.1 Detailný popis	62
5.4 Dokumentácia súboru Move.cs	62
5.4.1 Detailný popis	63
5.4.2 Dokumentácia enumeračných typov	63
5.4.2.1 Castling	63
5.5 Dokumentácia súboru Movement.cs	63
5.5.1 Detailný popis	63
5.6 Dokumentácia súboru Piece.cs	64
5.6.1 Detailný popis	64
5.7 Dokumentácia súboru Turn.cs	64
5.7.1 Detailný popis	64
5.8 Dokumentácia súboru EngineController.cs	64
5.8.1 Detailný popis	65
5.9 Dokumentácia súboru BoardController.cs	65
5.9.1 Detailný popis	65
5.10 Dokumentácia súboru GameCanvas.cs	65
5.10.1 Detailný popis	66
5.11 Dokumentácia súboru TitleCanvas.cs	66
5.11.1 Detailný popis	66
5.12 Dokumentácia súboru ClickableDropdown.cs	66
5.12.1 Detailný popis	67
5.13 Dokumentácia súboru CreateGameOverlay.cs	67
5.13.1 Detailný popis	67
5.14 Dokumentácia súboru PlayerInfoController.cs	67
5.14.1 Detailný popis	68
5.15 Dokumentácia súboru GhostObject.cs	68
5.15.1 Detailný popis	68
5.16 Dokumentácia súboru MovementObject.cs	68
5.16.1 Detailný popis	69
5.17 Dokumentácia súboru PieceObject.cs	69
5.17.1 Detailný popis	69
5.18 Dokumentácia súboru TurnObject.cs	69
5.18.1 Detailný popis	70
5.19 Dokumentácia súboru PromotionController.cs	70
5.19.1 Detailný popis	70
Register	71

Kapitola 1

Register hierarchie tried

1.1 Hierarchia tried

Tu nájdete zoznam, vyjadrujúci vzťah dedičnosti tried. Je zoradený približne (ale nie úplne) podľa abecedy:

Chess.GameSettings	31
Ghost	34
IPointerClickHandler	
ClickableDropdown	17
MovementObject	42
PieceObject	47
Mobility	36
MonoBehaviour	
BoardController	7
Chess	14
CreateGameOverlay	18
EngineController	22
GameCanvas	26
GhostObject	35
MovementObject	42
PieceInfoController	45
PieceObject	47
PlayerInfoController	49
PromotionController	51
TitleCanvas	53
TurnObject	58
Move	38
Movement	41
Piece	43
TMP_Dropdown	
ClickableDropdown	17
Turn	54

Kapitola 2

Register tried

2.1 Zoznam tried

Nasledujúci zoznam obsahuje predovšetkým identifikáciu tried, ale nachádzajú sa tu i ďalšie netriviálne prvky, ako sú štruktúry (struct), uniony (union) a rozhrania (interface). V zozname sú uvedené ich stručné popisy:

BoardController	7
Chess	14
ClickableDropdown	17
CreateGameOverlay	18
EngineController	22
GameCanvas	26
Chess.GameSettings	31
Ghost	34
GhostObject	35
Mobility	36
Move	38
Movement	41
MovementObject	42
Piece	43
PieceInfoController	45
PieceObject	47
PlayerInfoController	49
PromotionController	51
TitleCanvas	53
Turn	54
TurnObject	58

Kapitola 3

Register súborov

3.1 Zoznam súborov

Tu nájdete zoznam všetkých dokumentovaných súborov so stručnými popismi:

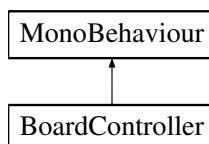
Chess.cs	Implementácia generických funkcionalít a šachových pojmov	61
Ghost.cs	Špeciálna figúrka vytvorená po vykonaní pohybu s hasty flagom	61
Mobility.cs	Konfigurácia mobility figúrky	62
Move.cs	Pohyb na šachovnici	62
Movement.cs	Pohyb figúrky	63
Piece.cs	Šachová figúrka	64
Turn.cs	Herný ťah	64
EngineController.cs	Ovládač šachového enginu Malakh	64
BoardController.cs	Ovládač rozhrania šachovnice	65
GameCanvas.cs	Ovládač rozhrania hernej obrazovky	65
TitleCanvas.cs	Ovládač rozhrania titulnej obrazovky	66
ClickableDropdown.cs	Rozšírenie dropdown komponentu	66
CreateGameOverlay.cs	Ovládač rozhrania vytvorenia novej hry	67
PlayerInfoController.cs	Ovládač nápovedy figúrky	67
GhostObject.cs	Ovládač objektu ducha	68
MovementObject.cs	Ovládač objektu pohybu	68
PieceObject.cs	Ovládač objektu figúrky	69
TurnObject.cs	Ovládač objektu herného ťahu	69
PromotionController.cs	Ovládač rozhrania výberu promócie pešiaka	70

Kapitola 4

Dokumentácia tried

4.1 Dokumentácia triedy BoardController

Diagram dedičnosti pre triedu BoardController



Verejné metódy

- void `InitBoard` (bool reversed, `Chess.GameSettings` gameSettings)
- void `SetPly` (`Chess.Color` currentPly)
- void `SetCheck` ()
- void `SetLegalMoves` (List< `Move` > legalMoves)
- `Piece` `GetPiece` (int x, int y)
- void `PerformMove` (`Move` legalMove)
- void `PerformPromotion` (`Chess.PieceType` promotedPieceType)

Verejné atribúty

- UnityEvent< `Movement`, `Chess.PieceType` > `onPlayerMove` = new()
- UnityEvent `onPromotionRequested` = new()

Privátne metódy

- void `SetBoardOrientation` (bool reversed)
- void `CreateInitialPosition` ()
- void `ClearPieces` ()
- void `ClearGhost` ()
- void `SetGhost` (`Ghost` ghost)
- void `ClearHighlightedSquare` ()
- void `SetHighlightedSquare` (int x, int y)

- void [AddPiece](#) ([Piece](#) piece, int x, int y)
- [Movement](#) [MovePiece](#) ([Move](#) move)
- [PieceObject](#) [GetPieceObject](#) (int x, int y)
- void [OnPieceClicked](#) ([Piece](#) piece)
- void [ClearMovements](#) ()
- void [AddMovement](#) ([Movement](#) movement)
- void [OnMovementClicked](#) ([Movement](#) movement)
- void [MoveGameObject](#) (GameObject gameObject, int x, int y)

Privátne atribúty

- TMP_Text **topColumn1**
- TMP_Text **topColumn2**
- TMP_Text **topColumn3**
- TMP_Text **topColumn4**
- TMP_Text **topColumn5**
- TMP_Text **topColumn6**
- TMP_Text **topColumn7**
- TMP_Text **topColumn8**
- TMP_Text **bottomColumn1**
- TMP_Text **bottomColumn2**
- TMP_Text **bottomColumn3**
- TMP_Text **bottomColumn4**
- TMP_Text **bottomColumn5**
- TMP_Text **bottomColumn6**
- TMP_Text **bottomColumn7**
- TMP_Text **bottomColumn8**
- TMP_Text **leftRow1**
- TMP_Text **leftRow2**
- TMP_Text **leftRow3**
- TMP_Text **leftRow4**
- TMP_Text **leftRow5**
- TMP_Text **leftRow6**
- TMP_Text **leftRow7**
- TMP_Text **leftRow8**
- TMP_Text **rightRow1**
- TMP_Text **rightRow2**
- TMP_Text **rightRow3**
- TMP_Text **rightRow4**
- TMP_Text **rightRow5**
- TMP_Text **rightRow6**
- TMP_Text **rightRow7**
- TMP_Text **rightRow8**
- AudioSource **audioSource**
- GameObject **board**
- bool **reversed** = false
- [Chess.Color](#) **currentPly**
- [Chess.GameSettings](#) **gameSettings**
- List< [PieceObject](#) > **pieceObjects** = new()
- [GhostObject](#) **ghostObject** = null
- GameObject **highlightedSquare** = null
- List< [MovementObject](#) > **movementObjects** = new()
- [Movement](#) **performedMovement** = null

4.1.1 Detailný popis

Táto trieda je zodpovedná za správu šachovnice.

4.1.2 Dokumentácia k metódam

4.1.2.1 AddMovement()

```
void BoardController.AddMovement (
    Movement movement ) [inline], [private]
```

Do šachovnice sa pridá nový pohyb.

Parametre

<i>movement</i>	Vykonaný pohyb.
-----------------	-----------------

4.1.2.2 AddPiece()

```
void BoardController.AddPiece (
    Piece piece,
    int x,
    int y ) [inline], [private]
```

Do šachovnice sa na danú pozíciu pridá nová figúrka.

Parametre

<i>piece</i>	Nová figúrka.
<i>x</i>	Koordinát X danej pozície.
<i>y</i>	Koordinát Y danej pozície.

4.1.2.3 ClearGhost()

```
void BoardController.ClearGhost ( ) [inline], [private]
```

Zo šachovnice sa odstráni duch (pre pravidlo En Passant).

4.1.2.4 ClearHighlightedSquare()

```
void BoardController.ClearHighlightedSquare ( ) [inline], [private]
```

Zo šachovnice sa odstráni znázornené políčko.

4.1.2.5 ClearMovements()

```
void BoardController.ClearMovements ( ) [inline], [private]
```

Zo šachovnice sa odstránia všetky pohyby.

4.1.2.6 ClearPieces()

```
void BoardController.ClearPieces ( ) [inline], [private]
```

Zo šachovnice sa odstránia všetky figúrky.

4.1.2.7 CreateInitialPosition()

```
void BoardController.CreateInitialPosition ( ) [inline], [private]
```

Zo šachovnice sa odstránia všetky figúrky a pridajú sa tu nové figúrky na počiatočnú pozíciu. Esencie sú priradené podľa aktuálnej hernej konfigurácie.

4.1.2.8 GetPiece()

```
Piece BoardController.GetPiece (
    int x,
    int y ) [inline]
```

Na danej pozícii sa vyhľadá figúrka.

Parametre

x	Koordinát X danej pozície.
y	Koordinát Y danej pozície.

Návratová hodnota

Nájdená figúrka na danej pozícii. Ak žiadan nebol nájdená tak sa vráti null.

4.1.2.9 GetPieceObject()

```
PieceObject BoardController.GetPieceObject (
    int x,
    int y ) [inline], [private]
```

Na šachovnici sa nájde objekt figúrky na danej pozícii.

Parametre

x	Koordinát X danej pozície.
x	Koordinát Y danej pozície.

Návratová hodnota

Nájdenny objekt figúrky.

4.1.2.10 InitBoard()

```
void BoardController.InitBoard (
    bool reversed,
    Chess::GameSettings gameSettings ) [inline]
```

Šachovnica sa inicializuje na počiatočnú pozíciu sa danou hernou konfiguráciou a orientáciou.

Parametre

<i>reversed</i>	Ak hráč hráč hrá za čierneho hráča, tak šachovnica je obrátená o 180 stupňov.
<i>gameSettings</i>	Herná konfigurácia.

4.1.2.11 MoveGameObject()

```
void BoardController.MoveGameObject (
    GameObject gameObject,
    int x,
    int y ) [inline], [private]
```

Generická metóda na presunutie herného objektu na danú pozíciu na šachovnici.

Parametre

<i>gameObject</i>	Presunutý herný objekt.
<i>x</i>	Koordinát X danej pozície.
<i>y</i>	Koordinát Y danej pozície.

4.1.2.12 MovePiece()

```
Movement BoardController.MovePiece (
    Move move ) [inline], [private]
```

Na šachovnici sa vykoná daný pohyb.

Parametre

<i>move</i>	Vykonaný pohyb.
-------------	-----------------

Návratová hodnota

Vykonaný pohyb spolu s jej figúrkou.

4.1.2.13 OnMovementClicked()

```
void BoardController.OnMovementClicked (
    Movement movement ) [inline], [private]
```

Po kliknutí na pohyb sa daný pohyb vykoná. Ak došlo k promócii tak sa vyžiada vykonanie promócie pešiaka.

Parametre

<i>movement</i>	Vybraný pohyb.
-----------------	----------------

4.1.2.14 OnPieceClicked()

```
void BoardController.OnPieceClicked (
    Piece piece ) [inline], [private]
```

Pri kliknutí na figúrku sa vymažú všetky aktuálne pohyby a zruší sa výber všetkých figúrok. Ak táto figúrka ešte nebola vybraná tak sa vyberie a znázornia sa jej dostupné pohyby.

Parametre

<i>piece</i>	Kliknutá figúrka.
--------------	-------------------

4.1.2.15 PerformMove()

```
void BoardController.PerformMove (
    Move legalMove ) [inline]
```

Vykoná sa daný pohyb. Vykonaný pohyb sa zapíše do lokálnej premennej

Parametre

<i>legalMove</i>	Vykonaný pohyb.
------------------	-----------------

4.1.2.16 PerformPromotion()

```
void BoardController.PerformPromotion (
    Chess::PieceType promotedPieceType ) [inline]
```

Vykoná sa promócia poslednej pohnutej figúrky.

Parametre

<i>promotedPieceType</i>	Nový typ figúrky priradený pešikovi.
--------------------------	--------------------------------------

4.1.2.17 SetBoardOrientation()

```
void BoardController.SetBoardOrientation (
    bool reversed ) [inline], [private]
```

Táto metóda nastaví orientáciu šachovnice.

Parametre

<i>reversed</i>	Ak hráč hráč hrá za čierneho hráča, tak šachovnica je obrátená o 180 stupňov.
-----------------	---

4.1.2.18 SetCheck()

```
void BoardController.SetCheck ( ) [inline]
```

Na šachovnici sa zobrazí napadnutie kráľa s červeným políčkوم pre hráča na rade.

4.1.2.19 SetGhost()

```
void BoardController.SetGhost (
    Ghost ghost ) [inline], [private]
```

Na šachovnici sa vytvorí nový duch (pre pravidlo En Passant).

Parametre

<i>ghost</i>	Nový duch.
--------------	------------

4.1.2.20 SetHighlightedSquare()

```
void BoardController.SetHighlightedSquare (
    int x,
    int y ) [inline], [private]
```

Do šachovnice sa na danú pozíciu pridá znázornené políčko.

Parametre

<i>x</i>	Koordinát X danej pozície.
<i>y</i>	Koordinát Y danej pozície.

4.1.2.21 SetLegalMoves()

```
void BoardController.SetLegalMoves (
    List< Move > legalMoves ) [inline]
```

Všetkým figúrkam sa odstránia legálne pohyby a priradia sa im nové.

Parametre

<i>legalMoves</i>	Zoznam legálnych pohybov.
-------------------	---------------------------

4.1.2.22 SetPly()

```
void BoardController.SetPly (
    Chess::Color currentPly ) [inline]
```

Nastaví sa lokálna premenná aktuálneho hráča na rade.

Parametre

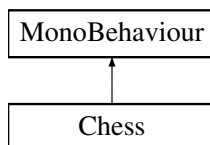
<i>currentPly</i>	Aktuálny hráč na rade.
-------------------	------------------------

Dokumentácia pre túto triedu bola generovaná z nasledujúceho súboru:

- [BoardController.cs](#)

4.2 Dokumentácia triedy Chess

Diagram dedičnosti pre triedu Chess



Triedy

- class [GameSettings](#)

Verejné typy

- enum [PieceType](#) {
[Pawn](#) = 0 , [Knight](#) = 1 , [Bishop](#) = 2 , [Rook](#) = 3 ,
[Queen](#) = 4 , [King](#) = 5 }
- enum [Essence](#) { [Classic](#) = 0 , [Red](#) = 1 , [Blue](#) = 2 }
- enum [Color](#) { [White](#) = 0 , [Black](#) = 1 , [Random](#) = 2 }
- enum [MovementType](#) { [Move](#) = 0 , [Attack](#) = 1 , [AttackMove](#) = 2 }
- enum [AI](#) { [Basic](#) = 0 , [Ensemble](#) = 1 }

Statické verejné metódy

- static Sprite [GetPiecImage](#) ([PieceType](#) type, [Color](#) color, [Essence](#) essence)
- static Sprite [GetMovementImage](#) ([MovementType](#) type)

4.2.1 Detailný popis

Táto trieda slúži podobne ako Common trieda.

4.2.2 Dokumentácia k členským enumeráciám

4.2.2.1 AI

enum `Chess.AI`

Enumerátor reprezentujúci typ šachovej inteligencie.

Hodnoty enumerácií

Basic	Základná inteligencia využíva na evaluáciu šachovej pozície ručne nastavenú evaluačnú funkciu.
Ensemble	Základná inteligencia využíva na evaluáciu šachovej pozície súbor CNN modelov.

4.2.2.2 Color

enum `Chess.Color`

Farba šachovej figúrky.

Hodnoty enumerácií

White	Biela figúrka.
Black	Čierna figúrka.
Random	Náhodná figúrka.

4.2.2.3 Essence

enum `Chess.Essence`

Enumerátor reprezentujúci esenciu šachovej figúrky

Hodnoty enumerácií

Classic	Klasická esencia.
Red	Červená esencia.
Blue	Modrá esencia.

4.2.2.4 MovementType

enum `Chess.MovementType`

Enumerátor reprezentujúci typ mobility.

Hodnoty enumerácií

Move	Tento pohyb umožňuje len presunúť sa na prázdnu pozíciu.
Attack	Tento pohyb umožňuje len napadnúť protivníkovu figúrku.
AttackMove	Tento pohyb umožňuje presunúť sa na prázdnu pozíciu a napadnúť protivníkovu figúrku.

4.2.2.5 PieceType

enum `Chess.PieceType`

Enumerátor reprezentujúci typ šachovej figúrky.

Hodnoty enumerácií

Pawn	Pešiak.
Knight	Rytier.
Bishop	Strelec.
Rook	Veža.
Queen	Kráľovná.
King	Kráľ.

4.2.3 Dokumentácia k metódam

4.2.3.1 GetMovementImage()

```
static Sprite Chess.GetMovementImage (
    MovementType type ) [inline], [static]
```

Táto metóda načíta obrázok požadovaného pohybu

Parametre

<i>type</i>	Typ pohybu.
-------------	-------------

Návratová hodnota

Obrázok daného pohybu.

4.2.3.2 GetPieceImage()

```
static Sprite Chess.GetPieceImage (
    PieceType type,
    Color color,
    Essence essence ) [inline], [static]
```

Táto metóda načíta obrázok požadovanej šachovej figúrky.

Parametre

<i>type</i>	Typ figúrky.
<i>color</i>	Farba figúrky.
<i>essence</i>	Esencia figúrky.

Návratová hodnota

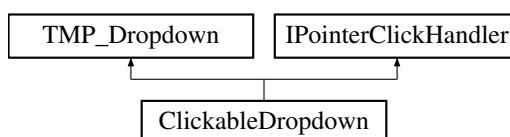
Obrázok danej figúrky.

Dokumentácia pre túto triedu bola generovaná z nasledujúceho súboru:

- [Chess.cs](#)

4.3 Dokumentácia triedy ClickableDropdown

Diagram dedičnosti pre triedu ClickableDropdown



Verejné metódy

- override void [OnPointerClick](#) (PointerEventData eventData)

Verejné atribúty

- UnityEvent **onClicked**

4.3.1 Detailný popis

Táto trieda je identická s dropdownom, jediné čo sme mu pridali je onClick event. Toto správanie potrebujeme v rozhraní vytvorenia hry.

4.3.2 Dokumentácia k metódam

4.3.2.1 OnPointerClick()

```
override void ClickableDropdown.OnPointerClick (
    PointerEventData eventData ) [inline]
```

Pri kliknutí na tento objekt sa to oznámi poslucháčom jeho onClick eventu.

Parametre

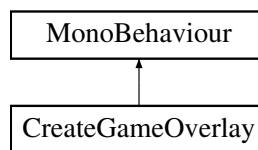
<code>eventData</code>	Informácie o kliknutí.
------------------------	------------------------

Dokumentácia pre túto triedu bola generovaná z nasledujúceho súboru:

- [ClickableDropdown.cs](#)

4.4 Dokumentácia triedy CreateGameOverlay

Diagram dedičnosti pre triedu CreateGameOverlay



Verejné metódy

- void [InitializeSettings](#) ([Chess.GameSettings](#) gameSettings)
- void [SetEngineController](#) ([EngineController](#) engineController)

Verejné atribúty

- UnityEvent< [Chess.GameSettings](#) > **onGameCreated**

Privátne metódy

- void **Awake** ()
- void [OnPlayerColorSet](#) ([Chess.Color](#) color)
- void [OnMalakhColorSet](#) ([Chess.Color](#) color)
- void [OnPlayerRandomPiecesToggled](#) (bool value)
- void [OnMalakhRandomPiecesToggled](#) (bool value)
- void [OnReturnClick](#) ()
- void [OnPlayClick](#) ()
- void [SetPieceInfo](#) ([Chess.PieceType](#) pieceType, [Chess.Essence](#) essence)

Privátne atribúty

- Toggle **aiBasicToggle**
- Toggle **aiEnsembleToggle**
- Toggle **playerWhiteColorToggle**
- Toggle **playerBlackColorToggle**
- Toggle **playerRandomColorToggle**
- Toggle **malakhWhiteColorToggle**
- Toggle **malakhBlackColorToggle**
- Toggle **malakhRandomColorToggle**
- ClickableDropdown **playerPawnDropdown**
- ClickableDropdown **playerKnightDropdown**
- ClickableDropdown **playerBishopDropdown**
- ClickableDropdown **playerRookDropdown**
- ClickableDropdown **malakhPawnDropdown**
- ClickableDropdown **malakhKnightDropdown**
- ClickableDropdown **malakhBishopDropdown**
- ClickableDropdown **malakhRookDropdown**
- Toggle **playerRandomPiecesToggle**
- Toggle **malakhRandomPiecesToggle**
- TMP_Dropdown **pieceDropdown**
- TMP_Dropdown **essenceDropdown**
- TMP_Text **pieceLabel**
- TMP_Text **essenceLabel**
- Button **returnBtn**
- Button **playBtn**
- GameObject **board**
- GameObject **piecePrefab**
- GameObject **movementPrefab**
- EngineController **engineController**
- PieceObject **pieceObject**
- List< MovementObject > **movements** = new()

4.4.1 Detailný popis

Táto trieda je zodpovedná za správu rozhrania vytvorenia novej hry.

4.4.2 Dokumentácia k metódam**4.4.2.1 InitializeSettings()**

```
void CreateGameOverlay.InitializeSettings (
    Chess::GameSettings gameSettings ) [inline]
```

V tomto rozhraní inicializujeme hernú konfiguráciu predošlej hry.

Parametre

<i>gameSettings</i>	Herná konfigurácia.
---------------------	---------------------

4.4.2.2 OnMalakhColorSet()

```
void CreateGameOverlay.OnMalakhColorSet (
    Chess::Color color ) [inline], [private]
```

Pri nastavení farby Malakha aktualizujeme všetky relevantné hráčove grafické komponenty. Následne nastavíme farbu hráča.

Parametre

<i>color</i>	Nová farba Malakha.
--------------	---------------------

4.4.2.3 OnMalakhRandomPiecesToggled()

```
void CreateGameOverlay.OnMalakhRandomPiecesToggled (
    bool value ) [inline], [private]
```

Pri nastavení náhodných figúrok pre Malakha nastavíme interaktívnosť výberu esencie pre Malakhove figúrky.

Parametre

<i>value</i>	Ak je hodnota true, tak sú figúrky náhodné. V opačnom prípade je false.
--------------	---

4.4.2.4 OnPlayClick()

```
void CreateGameOverlay.OnPlayClick ( ) [inline], [private]
```

Pri stlačení tlačidla spustenia hry sa všetkým poslucháčom odošle informácia o novej hernej konfigurácii (tie spustia novú hru). Ak boli nastavené náhodné esencie, tak sa tu esencie náhodne vyberú.

4.4.2.5 OnPlayerColorSet()

```
void CreateGameOverlay.OnPlayerColorSet (
    Chess::Color color ) [inline], [private]
```

Pri nastavení farby hráča aktualizujeme všetky relevantné hráčove grafické komponenty. Následne nastavíme farbu Malakha.

Parametre

<i>color</i>	Nová farba hráča.
--------------	-------------------

4.4.2.6 OnPlayerRandomPiecesToggled()

```
void CreateGameOverlay.OnPlayerRandomPiecesToggled (
    bool value ) [inline], [private]
```

Pri nastavení náhodných figúrok pre hráča nastavíme interaktívnosť výberu esencie pre hráčove figúrky.

Parametre

<i>value</i>	Ak je hodnota true, tak sú figúrky náhodné. V opačnom prípade je false.
--------------	---

4.4.2.7 OnReturnClick()

```
void CreateGameOverlay.OnReturnClick ( ) [inline], [private]
```

Pri stlačení tlačidla vrátenia vymažeme toto rozhranie.

4.4.2.8 SetEngineController()

```
void CreateGameOverlay.SetEngineController (
    EngineController engineController ) [inline]
```

Táto funkcia nastaví lokálna premenná ovládača enginu. Táto trieda je potrebná na čítanie mobilít figúrok.

4.4.2.9 SetPieceInfo()

```
void CreateGameOverlay.SetPieceInfo (
    Chess::PieceType pieceType,
    Chess::Essence essence ) [inline], [private]
```

V nápovede sa zobrazí mobilita danej figúrky.

Parametre

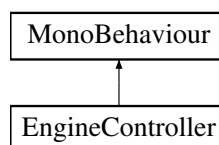
<i>pieceType</i>	Typ figúrky.
<i>essence</i>	Esencia figúrky.

Dokumentácia pre túto triedu bola generovaná z nasledujúceho súboru:

- [CreateGameOverlay.cs](#)

4.5 Dokumentácia triedy EngineController

Diagram dedičnosti pre triedu EngineController



Verejné metódy

- void `StartEngine` ()
- void `StopEngine` ()
- void `StartGame` (`Chess.GameSettings` gameSettings)
- void `RequestAvailableMoves` ()
- void `RequestBestMove` ()
- void `MovePiece` (int x1, int y1, int x2, int y2, `Chess.PieceType` promotedPieceType)

Verejné atribúty

- string `engineName` = string.Empty
- string `author` = string.Empty
- Dictionary< `Chess.PieceType`, Dictionary< `Chess.Essence`, List< `Mobility` > > `mobilities` = new()
- UnityEvent< List< `Move` > > `onLegalMovesReceived` = new()
- UnityEvent< `Move`, `Chess.PieceType` > `onBestMoveReceived` = new()
- UnityEvent< `Chess.Color` > `onResultReceived` = new()
- UnityEvent `onCheckReceived` = new()

Privátne metódy

- void `Start` ()
- void `Update` ()
- void `OnApplicationQuit` ()
- void `SendCommand` (string command)
- void `ReceiveEngineResponses` ()
- `Move ParseMove` (string move)

Privátne atribúty

- Process `engineProcess`
- `Move receivedBestMove` = null
- `Chess.PieceType promotedPieceType` = `Chess.PieceType.Pawn`
- bool `unhandledResult` = false
- `Chess.Color receivedResult` = `Chess.Color.Random`
- bool `receivedCheck` = false

4.5.1 Detailný popis

Táto trieda je zodpovedná za komunikáciu s enginom Malakh pomocou UCI protokolu.

4.5.2 Dokumentácia k metódam

4.5.2.1 MovePiece()

```
void EngineController.MovePiece (
    int x1,
    int y1,
    int x2,
    int y2,
    Chess::PieceType promotedPieceType ) [inline]
```

Enginu prikážeme aby vykonal daný pohyb.

Parametre

<i>x1</i>	Koordinát X zdrojovej pozície.
<i>y1</i>	Koordinát Y zdrojovej pozície.
<i>x2</i>	Koordinát X cieľovej pozície.
<i>y2</i>	Koordinát Y cieľovej pozície.
<i>promotedPieceType</i>	Typ figúrky na ktorú sa zmení pešiak pri promócii.

4.5.2.2 OnApplicationQuit()

```
void EngineController.OnApplicationQuit ( ) [inline], [private]
```

Pri ukončení aplikácie terminuje šachový engine.

4.5.2.3 ParseMove()

```
Move EngineController.ParseMove (
    string move ) [inline], [private]
```

String konvertujeme na pohyb figúrky.

Parametre

<i>move</i>	Slovo reprezentujúce pohyb figúrky v algebraickej notácii.
-------------	--

Návratová hodnota

Pohyb figúrky.

4.5.2.4 ReceiveEngineResponses()

```
void EngineController.ReceiveEngineResponses ( ) [inline], [private]
```

V tejto metóde periodicky prijímame a spracúvame prijaté správy od enginu.

4.5.2.5 RequestAvailableMoves()

```
void EngineController.RequestAvailableMoves ( ) [inline]
```

Od enginu si vyžiadame všetky dostupné legálne pohyby.

4.5.2.6 RequestBestMove()

```
void EngineController.RequestBestMove ( ) [inline]
```

Od enginu si vyžiadame najlepší pohyb.

4.5.2.7 SendCommand()

```
void EngineController.SendCommand (
    string command ) [inline], [private]
```

Generická metóda na odoslanie ľubovoľnej správy enginu.

4.5.2.8 Start()

```
void EngineController.Start ( ) [inline], [private]
```

Pri spustení tohto skriptu zapneme engine.

4.5.2.9 StartEngine()

```
void EngineController.StartEngine ( ) [inline]
```

Táto funkcia spustí engine Malakh v UCI režime.

4.5.2.10 StartGame()

```
void EngineController.StartGame (
    Chess::GameSettings gameSettings ) [inline]
```

Hra je spustená s danou hernou konfiguráciou.

gameSettings Nastavenia novej hry.

4.5.2.11 StopEngine()

```
void EngineController.StopEngine ( ) [inline]
```

Táto funkcia terminuje engine pomocou UCI príkazu.

4.5.2.12 Update()

```
void EngineController.Update ( ) [inline], [private]
```

Pri každej aktualizácii kontrolujeme, či sme nedostali od enginu novú nespracovanú správu. Niektoré operácie musia byť vykonané cez update procedúru a nemôžu byť vykonané priamo.

4.5.3 Dokumentácia k dátovým členom

4.5.3.1 engineProcess

Process EngineController.engineProcess [private]

Inicializátor:

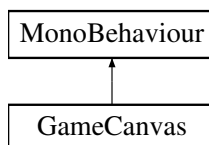
```
= new()
{
    StartInfo = new()
    {
        FileName = "Malakh-chess-engine.exe",
        UseShellExecute = false,
        RedirectStandardInput = true,
        RedirectStandardOutput = true,
        CreateNoWindow = true
    }
}
```

Dokumentácia pre túto triedu bola generovaná z nasledujúceho súboru:

- [EngineController.cs](#)

4.6 Dokumentácia triedy GameCanvas

Diagram dedičnosti pre triedu GameCanvas



Verejné metódy

- void [SetGame](#) (Chess.GameSettings gameSettings)
- void [AddTurn](#) (Chess.PieceType whitePieceType, Chess.PieceType whitePromotion, Chess.MovementType whiteMovementType, [Move](#) whiteMove)
- void [SetBlackPly](#) (Chess.PieceType blackPieceType, Chess.PieceType blackPromotion, Chess.MovementType blackMovementType, [Move](#) blackMove)
- void [ClearTurns](#) ()
- void [OnExportClick](#) ()

Privátne metódy

- void **Awake** ()
- void **Start** ()
- void [OnLegalMovesReceived](#) (List< [Move](#) > legalMoves)
- void [OnBestMoveReceived](#) ([Move](#) bestMove, Chess.PieceType promotedPieceType)
- void [OnPlayerMove](#) ([Movement](#) movement, Chess.PieceType promotedPieceType)
- void [OnResultReceived](#) (Chess.Color victor)
- void [OnCheckReceived](#) ()
- void [OnPromotionRequested](#) ()
- void [OnPromotionChosen](#) (Chess.PieceType pieceType)
- void [SetTurn](#) (Chess.Color currentPly)
- void [OnHelpButtonClicked](#) (Chess.PieceType pieceType, Chess.Color color, Chess.Essence essence)
- void [OnReturnClick](#) ()
- void [OnPlayClick](#) ()
- void [OnGameCreated](#) (Chess.GameSettings gameSettings)

Privátne atribúty

- [BoardController](#) **boardController**
- [PlayerInfoController](#) **playerInfoController**
- [PlayerInfoController](#) **malakhInfoController**
- [PieceInfoController](#) **pieceInfoController**
- [PromotionController](#) **promotionController**
- [EngineController](#) **engineController**
- TMP_Text **turnDisplay**
- Button **returnBtn**
- Button **playBtn**
- Button **exportBtn**
- [TitleCanvas](#) **titleCanvas**
- [Chess.GameSettings](#) **gameSettings**
- GameObject **turnListContent**
- GameObject **turnObjectPrefab**
- Scrollbar **turnScrollbar**
- List< [TurnObject](#) > **turnObjectList** = new()
- int **turnCounter** = 0
- GameObject **createGameOverlay** = null

4.6.1 Detailný popis

Táto trieda je zodpovedná za správu hernej obrazovky.

4.6.2 Dokumentácia k metódam**4.6.2.1 AddTurn()**

```
void GameCanvas.AddTurn (
    Chess::PieceType whitePieceType,
    Chess::PieceType whitePromotion,
    Chess::MovementType whiteMovementType,
    Move whiteMove ) [inline]
```

Do histórie hry pridáme nový pohyb bieleho hráča.

Parametre

<i>whitePieceType</i>	Typ bielej figúrky.
<i>whitePromotion</i>	Promócia bieleho pešiaka.
<i>whiteMovementType</i>	Typ pohybu bielej figúrky.
<i>whiteMove</i>	Vykonaný pohyb bielej figúrky.

4.6.2.2 ClearTurns()

```
void GameCanvas.ClearTurns ( ) [inline]
```

Vymažeme históriu hry.

4.6.2.3 OnBestMoveReceived()

```
void GameCanvas.OnBestMoveReceived (
    Move bestMove,
    Chess::PieceType promotedPieceType ) [inline], [private]
```

Pri prijatí najlepšieho pohybu od enginu tento pohyb vykonáme na šachovnici a pohyb vložíme do histórie hry. Následne prikážeme enginu aby daný pohyb vykonal a vyžiadame si legálne pohyby.

Parametre

<i>bestMove</i>	Najlepší pohyb podľa enginu.
<i>promotedPieceType</i>	ak bola vykonaná promócia tak sa pešiak premení na tento typ figúrky.

4.6.2.4 OnCheckReceived()

```
void GameCanvas.OnCheckReceived ( ) [inline], [private]
```

Pri prijatí informácie o napadnutí kráľa od enginu to nastavíme na šachovnici.

4.6.2.5 OnExportClick()

```
void GameCanvas.OnExportClick ( ) [inline]
```

Pri kliknutí na tlačidlo exportu výsledok hry exportujeme do výstupného súboru. Do súboru ukladáme informácie ako herná konfigurácia, výsledok hry a história hry.

4.6.2.6 OnGameCreated()

```
void GameCanvas.OnGameCreated (
    Chess::GameSettings gameSettings ) [inline], [private]
```

Pri prijatí hernej konfigurácie od rozhrania vytvorenia novej hry nastavíme túto konfiguráciu v tomto rozhraní.

Parametre

<i>gameSettings</i>	Herná konfigurácia.
---------------------	---------------------

4.6.2.7 OnHelpButtonClicked()

```
void GameCanvas.OnHelpButtonClicked (
    Chess::PieceType pieceType,
    Chess::Color color,
    Chess::Essence essence ) [inline], [private]
```

Pri kliknutí na tlačidlo nápovedy pri figúrke v bočnom paneli zobrazíme nápovedu pre danú figúrku.

Parametre

<i>pieceType</i>	Typ figúrky.
<i>color</i>	Vlastník figúrky.
<i>essence</i>	Esencia figúrky.

4.6.2.8 OnLegalMovesReceived()

```
void GameCanvas.OnLegalMovesReceived (
    List< Move > legalMoves ) [inline], [private]
```

Pri prijatí legálnych pohybov od enginu prepošleme túto informáciu ovládaču šachovnice.

4.6.2.9 OnPlayClick()

```
void GameCanvas.OnPlayClick ( ) [inline], [private]
```

Po stlačení tlačidla novej hry zobrazíme rozhranie vytvorenia novej hry.

4.6.2.10 OnPlayerMove()

```
void GameCanvas.OnPlayerMove (
    Movement movement,
    Chess::PieceType promotedPieceType ) [inline], [private]
```

Po prijatí pohybu hráča od ovládača šachovnice sa vykonaný pohyb zapíše do histórie hry. Následne prikážeme enginu aby daný pohyb vykonal a vyžiadame si najlepší pohyb.

Parametre

<i>movement</i>	Vykonaný pohyb hráča.
<i>promotedPieceType</i>	Ak bola vykonaná promócia pešiaka, tak toto bol typ figúrky na ktorú sa premenil.

4.6.2.11 OnPromotionChosen()

```
void GameCanvas.OnPromotionChosen (
    Chess::PieceType pieceType ) [inline], [private]
```

Pri prijatí promócia od rozhrania promócie vykonáme promóciu na šachovnici.

Parametre

<i>pieceType</i>	Vybraná promócia pešiaka.
------------------	---------------------------

4.6.2.12 OnPromotionRequested()

```
void GameCanvas.OnPromotionRequested ( ) [inline], [private]
```

Pri prijatí žiadosti o výber promócie od ovládača šachovnice zobrazíme rozhranie na výber promócie.

4.6.2.13 OnResultReceived()

```
void GameCanvas.OnResultReceived (
    Chess::Color victor ) [inline], [private]
```

Pri prijatí výsledku hry od enginu hra je terminovaná a zobrazí sa výsledok hry.

Parametre

<i>victor</i>	Výherný hráč.
---------------	---------------

4.6.2.14 OnReturnClick()

```
void GameCanvas.OnReturnClick ( ) [inline], [private]
```

Po stlačení tlačidla návratu sa vrátíme do rozhrania hlavnej obrazovky.

4.6.2.15 SetBlackPly()

```
void GameCanvas.SetBlackPly (
    Chess::PieceType blackPieceType,
    Chess::PieceType blackPromotion,
    Chess::MovementType blackMovementType,
    Move blackMove ) [inline]
```

Do histórie hry pridáme nová pohyb čierneho hráča.

Parametre

<i>blackPieceType</i>	Typ čiernej figúrky.
<i>blackPromotion</i>	Promócia čierneho pešiaka.
<i>blackMovementType</i>	Typ pohybu čiernej figúrky.
<i>blackMove</i>	Vykonaný pohyb čiernej figúrky.

4.6.2.16 SetGame()

```
void GameCanvas.SetGame (
    Chess::GameSettings gameSettings ) [inline]
```

Inicializujeme toto rozhranie aby sa spustila hra s danou hernou konfiguráciou.

Parametre

<i>gameSettings</i>	Herná konfigurácia novej hry
---------------------	------------------------------

4.6.2.17 SetTurn()

```
void GameCanvas.SetTurn (
    Chess::Color currentPly ) [inline], [private]
```

Nastavenie hráča na rade. Informáciu zobrazíme v nadpise. Ak je na rade hráč, tak si vyžiadame dostupné pohyby. Ak je na rade Malakh, tak si vyžiadame najlepší pohyb.

Parametre

<i>currentPly</i>	Hráč na rade.
-------------------	---------------

Dokumentácia pre túto triedu bola generovaná z nasledujúceho súboru:

- [GameCanvas.cs](#)

4.7 Dokumentácia triedy Chess.GameSettings

Verejné metódy

- [GameSettings](#) ([AI ai](#), [Color playerColor](#), [Essence playerPawn](#), [Essence playerKnight](#), [Essence playerBishop](#), [Essence playerRook](#), [Color malakhColor](#), [Essence malakhPawn](#), [Essence malakhKnight](#), [Essence malakhBishop](#), [Essence malakhRook](#))

Verejné atribúty

- [AI ai](#)
- [Color playerColor](#)
- [Color malakhColor](#)
- [Essence playerPawn](#)
- [Essence playerKnight](#)
- [Essence playerBishop](#)
- [Essence playerRook](#)
- [Essence malakhPawn](#)
- [Essence malakhKnight](#)
- [Essence malakhBishop](#)
- [Essence malakhRook](#)

4.7.1 Detailný popis

Trieda reprezentujúca hernú konfiguráciu.

4.7.2 Dokumentácia konštruktoru a deštruktoru

4.7.2.1 GameSettings()

```
Chess.GameSettings.GameSettings (
    AI ai,
    Color playerColor,
    Essence playerPawn,
    Essence playerKnight,
    Essence playerBishop,
    Essence playerRook,
    Color malakhColor,
    Essence malakhPawn,
    Essence malakhKnight,
    Essence malakhBishop,
    Essence malakhRook ) [inline]
```

Konštruktor hernej konfigurácie.

Parametre

<i>ai</i>	Typ šachovej inteligencie.
<i>playerColor</i>	Farba hráča.
<i>playerPawn</i>	Esencia hráčovho pešiaka.
<i>playerKnight</i>	Esencia hráčovho rytiera.
<i>playerBishop</i>	Esencia hráčovho strelca.
<i>playerRook</i>	Esencia hráčovej veže.
<i>malakhColor</i>	Farba Malakha
<i>malakhPawn</i>	Esencia Malakhovho pešiaka.
<i>malakhKnight</i>	Esencia Malakhovho rytiera.
<i>malakhBishop</i>	Esencia Malakhovho strelca.
<i>malakhRooks</i>	Esencia Malakhovej veže.

4.7.3 Dokumentácia k dátovým členom

4.7.3.1 ai

```
AI Chess.GameSettings.ai
```

Typ šachovej inteligencie.

4.7.3.2 malakhBishop

```
Essence Chess.GameSettings.malakhBishop
```

Esencia Malakhovho strelca.

4.7.3.3 malakhColor

`Color Chess.GameSettings.malakhColor`

Farba Malakha.

4.7.3.4 malakhKnight

`Essence Chess.GameSettings.malakhKnight`

Esencia Malakhovho rytiera.

4.7.3.5 malakhPawn

`Essence Chess.GameSettings.malakhPawn`

Esencia Malakhovho pešiaka.

4.7.3.6 malakhRook

`Essence Chess.GameSettings.malakhRook`

Esencia Malakhovej veže.

4.7.3.7 playerBishop

`Essence Chess.GameSettings.playerBishop`

Esencia hráčovho strelca.

4.7.3.8 playerColor

`Color Chess.GameSettings.playerColor`

Farba hráča.

4.7.3.9 playerKnight

`Essence Chess.GameSettings.playerKnight`

Esencia hráčovho rytiera.

4.7.3.10 playerPawn

`Essence Chess.GameSettings.playerPawn`

Esencia hráčovho pešiaka.

4.7.3.11 playerRook

Essence `Chess.GameSettings.playerRook`

Esencia hráčovej veže.

Dokumentácia pre túto triedu bola generovaná z nasledujúceho súboru:

- [Chess.cs](#)

4.8 Dokumentácia triedy Ghost

Verejné metódy

- [Ghost](#) (`int x`, `int y`, [Piece parent](#))

Verejné atribúty

- `int x`
- `int y`
- [Piece parent](#)

4.8.1 Detailný popis

Táto trieda reprezentuje špeciálnu figúrku vytvorenú po vykonaní pohybu s hasty flagom.

4.8.2 Dokumentácia konštruktoru a deštruktoru

4.8.2.1 Ghost()

```
Ghost.Ghost (
    int x,
    int y,
    Piece parent ) [inline]
```

Konštruktor špeciálnej figúrky.

Parametre

<i>x</i>	Koordinát X na šachovnici.
<i>y</i>	Koordinát Y na šachovnici.
<i>parent</i>	Figúrka, ktorá vlastní túto špeciálnu figúrku.

4.8.3 Dokumentácia k dátovým členom

4.8.3.1 parent

`Piece Ghost.parent`

Figúrka, ktorá vlastní túto špeciálnu figúrku.

4.8.3.2 x

`int Ghost.x`

Koordinát X na šachovnici.

4.8.3.3 y

`int Ghost.y`

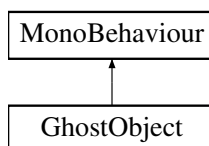
Koordinát Y na šachovnici.

Dokumentácia pre túto triedu bola generovaná z nasledujúceho súboru:

- [Ghost.cs](#)

4.9 Dokumentácia triedy GhostObject

Diagram dedičnosti pre triedu GhostObject



Vlastnosti

- [Ghost Ghost](#) [get, set]

Privátne atribúty

- [Ghost ghost](#) = null

4.9.1 Detailný popis

Táto trieda spravuje objekt ducha.

4.9.2 Dokumentácia k vlastnosti

4.9.2.1 Ghost

`Ghost` `GhostObject.Ghost` [get], [set]

Duch priradený tomuto objektu.

Dokumentácia pre túto triedu bola generovaná z nasledujúceho súboru:

- [GhostObject.cs](#)

4.10 Dokumentácia triedy Mobility

Verejné metódy

- `Mobility` (`Chess.MovementType` `type`, `int` `start_x`, `int` `start_y`, `int` `direction_x`, `int` `direction_y`, `int` `limit`)

Verejné atribúty

- `Chess.MovementType` `type`
- `int` `start_x`
- `int` `start_y`
- `int` `direction_x`
- `int` `direction_y`
- `int` `limit`

4.10.1 Detailný popis

Táto trieda reprezentuje konfiguráciu mobility figúrky.

4.10.2 Dokumentácia konštruktoru a deštruktoru

4.10.2.1 Mobility()

```
Mobility.Mobility (  
    Chess::MovementType type,  
    int start_x,  
    int start_y,  
    int direction_x,  
    int direction_y,  
    int limit ) [inline]
```

Konštruktor mobility.

Parametre

<i>type</i>	Typ pohybu.
<i>start_x</i>	Koordinát X počiatočnej pozície.
<i>start_y</i>	Koordinát Y počiatočnej pozície.
<i>direction↔ _x</i>	Koordinát X cieľovej pozície.
<i>direction↔ _y</i>	Koordinát Y cieľovej pozície.
<i>limit</i>	Limit pohybu.

4.10.3 Dokumentácia k dátovým členom

4.10.3.1 direction_x

```
int Mobility.direction_x
```

Koordinát X cieľovej pozície.

4.10.3.2 direction_y

```
int Mobility.direction_y
```

Koordinát Y cieľovej pozície.

4.10.3.3 limit

```
int Mobility.limit
```

Limit pohybu. Ak je hodnota 0, tak pohyb nemá limit.

4.10.3.4 start_x

```
int Mobility.start_x
```

Koordinát X počiatočnej pozície.

4.10.3.5 start_y

```
int Mobility.start_y
```

Koordinát Y počiatočnej pozície.

4.10.3.6 type

`Chess.MovementType` `Mobility.type`

Typ pohybu.

Dokumentácia pre túto triedu bola generovaná z nasledujúceho súboru:

- [Mobility.cs](#)

4.11 Dokumentácia triedy Move

Verejné metódy

- `Move` (int `x1`, int `y1`, int `x2`, int `y2`)
- `Move` (`Castling` `castling`)
- void `SetHasty` (int `hastyX`, int `hastyY`)
- void `ClearHasty` ()
- void `SetVigilant` ()
- void `ClearVigilant` ()

Verejné atribúty

- readonly int `x1`
- readonly int `y1`
- readonly int `x2`
- readonly int `y2`
- bool `hasty` = false
- bool `vigilant` = false
- int `hastyX`
- int `hastyY`
- readonly `Castling` `castling` = `Castling.None`

4.11.1 Detailný popis

Trieda reprezentujúca pohyb na šachovnici.

4.11.2 Dokumentácia konštruktoru a deštruktoru

4.11.2.1 `Move()` [1/2]

```
Move.Move (
    int x1,
    int y1,
    int x2,
    int y2 ) [inline]
```

Konštruktor normálneho pohybu na šachovnici.

Parametre

<i>x1</i>	Koordinát X počiatočnej pozície.
<i>y1</i>	Koordinát Y počiatočnej pozície.
<i>x2</i>	Koordinát X cieľovej pozície.
<i>y2</i>	Koordinát Y cieľovej pozície.

4.11.2.2 Move() [2/2]

```
Move.Move (
    Castling castling ) [inline]
```

Konštruktor rošády.

Parametre

<i>castling</i>	Typ rošády.
-----------------	-------------

4.11.3 Dokumentácia k metódam

4.11.3.1 ClearHasty()

```
void Move.ClearHasty ( ) [inline]
```

Zrušenie hasty flagu.

4.11.3.2 ClearVigilant()

```
void Move.ClearVigilant ( ) [inline]
```

Zrušenie vigilant flagu.

4.11.3.3 SetHasty()

```
void Move.SetHasty (
    int hastyX,
    int hastyY ) [inline]
```

Nastavenie hasty flagu.

Parametre

<i>hastyX</i>	Koordinát X vytvoreného ducha.
<i>hastyY</i>	Koordinát Y vytvoreného ducha.

4.11.3.4 SetVigilant()

```
void Move.SetVigilant ( ) [inline]
```

Nastavenie vigilant flagu.

4.11.4 Dokumentácia k dátovým členom

4.11.4.1 castling

```
readonly Castling Move.castling = Castling.None
```

Typ rošády.

4.11.4.2 hasty

```
bool Move.hasty = false
```

Hasty flag pohybu (pohyb vytvorí ducha).

4.11.4.3 hastyX

```
int Move.hastyX
```

Koordinát X vytvoreného ducha.

4.11.4.4 hastyY

```
int Move.hastyY
```

Koordinát Y vytvoreného ducha.

4.11.4.5 vigilant

```
bool Move.vigilant = false
```

Vigilant flag pohybu (pohyb môže napadnúť ducha).

4.11.4.6 x1

```
readonly int Move.x1
```

Koordinát X zdrojovej pozície.

4.11.4.7 x2

```
readonly int Move.x2
```

Koordinát X cieľovej pozície.

4.11.4.8 y1

```
readonly int Move.y1
```

Koordinát Y zdrojovej pozície.

4.11.4.9 y2

```
readonly int Move.y2
```

Koordinát Y cieľovej pozície.

Dokumentácia pre túto triedu bola generovaná z nasledujúceho súboru:

- [Move.cs](#)

4.12 Dokumentácia triedy Movement

Verejné metódy

- [Movement](#) ([Piece owner](#), [Chess.MovementType type](#), [Move move](#))

Verejné atribúty

- [Piece owner](#)
- [Chess.MovementType type](#)
- [Move move](#)

4.12.1 Detailný popis

Táto trieda reprezentuje pohyb figúrky.

4.12.2 Dokumentácia konštruktoru a deštruktoru

4.12.2.1 Movement()

```
Movement.Movement (
    Piece owner,
    Chess::MovementType type,
    Move move ) [inline]
```

Konštruktor pohybu figúrky.

Parametre

<i>owner</i>	Vlastník pohybu.
<i>type</i>	Typ pohybu.
<i>move</i>	Pohyb.

4.12.3 Dokumentácia k dátovým členom

4.12.3.1 move

[Move](#) `Movement.move`

Pohyb.

4.12.3.2 owner

[Piece](#) `Movement.owner`

Vlastník pohybu - šachová figúrka.

4.12.3.3 type

[Chess.MovementType](#) `Movement.type`

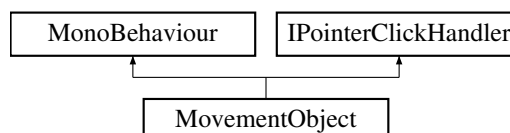
Typ pohybu.

Dokumentácia pre túto triedu bola generovaná z nasledujúceho súboru:

- [Movement.cs](#)

4.13 Dokumentácia triedy MovementObject

Diagram dedičnosti pre triedu MovementObject



Verejné metódy

- void [OnPointerClick](#) (PointerEventData eventData)

Verejné atribúty

- `UnityEvent< Movement > onClick`

Vlastnosti

- `Movement Movement` [get, set]

Privátne atribúty

- `Image image`
- `Movement movement = null`

4.13.1 Detailný popis

Táto trieda spravuje objekt pohybu.

4.13.2 Dokumentácia k metódam

4.13.2.1 OnPointerClick()

```
void MovementObject.OnPointerClick (
    PointerEventData eventData ) [inline]
```

Pri kliknutí na tento objekt sa to oznámi poslucháčom jeho onClick eventu.

Parametre

<code>eventData</code>	Informácie o kliknutí.
------------------------	------------------------

4.13.3 Dokumentácia k vlastnosti

4.13.3.1 Movement

```
Movement MovementObject.Movement [get], [set]
```

Pohyb priradený tomuto objektu. Ak ho zmeníme, tak sa aktualizuje jeho zobrazenie.

Dokumentácia pre túto triedu bola generovaná z nasledujúceho súboru:

- [MovementObject.cs](#)

4.14 Dokumentácia triedy Piece

Verejné metódy

- `Piece (Chess.PieceType type, Chess.Color color, Chess.Essence essence)`

Verejné atribúty

- [Chess.PieceType](#) type
- [Chess.Color](#) color
- [Chess.Essence](#) essence
- int x = -1
- int y = -1
- List< [Movement](#) > availableMoves = new()

4.14.1 Detailný popis

Táto trieda reprezentuje šachovú figúrku.

4.14.2 Dokumentácia konštruktoru a deštruktoru

4.14.2.1 Piece()

```
Piece.Piece (
    Chess::PieceType type,
    Chess::Color color,
    Chess::Essence essence ) [inline]
```

Konštruktor figúrky.

Parametre

<i>type</i>	Typ figúrky.
<i>color</i>	Farba figúrky.
<i>essence</i>	Esencia figúrky.

4.14.3 Dokumentácia k dátovým členom

4.14.3.1 availableMoves

```
List<Movement> Piece.availableMoves = new()
```

Zoznam dostupných pohybov figúrky.

4.14.3.2 color

```
Chess.Color Piece.color
```

Farba šachovej figúrky.

4.14.3.3 essence

`Chess.Essence` `Piece.essence`

Esencia šachovej figúrky.

4.14.3.4 type

`Chess.PieceType` `Piece.type`

Typ šachovej figúrky.

4.14.3.5 x

`int` `Piece.x` = -1

Koordinát X umiestnenia figúrky na šachovnici.

4.14.3.6 y

`int` `Piece.y` = -1

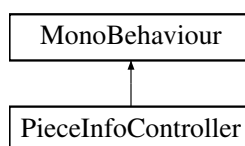
Koordinát Y umiestnenia figúrky na šachovnici.

Dokumentácia pre túto triedu bola generovaná z nasledujúceho súboru:

- [Piece.cs](#)

4.15 Dokumentácia triedy PieceInfoController

Diagram dedičnosti pre triedu PieceInfoController



Vlastnosti

- [Piece MyPiece](#) [get, set]

Privátne metódy

- void **Awake** ()
- void **Start** ()
- void **CreatePiece** ()
- void **UpdateDisplayedPiece** ()
- void **OnCloseButtonClicked** ()

Privátne atribúty

- Button **closeBtn**
- GameObject **board**
- **EngineController** **engineController**
- GameObject **piecePrefab**
- GameObject **movementPrefab**
- **PieceObject** **pieceObject**
- List< **MovementObject** > **movements** = new()
- **Piece** **myPiece**

4.15.1 Detailný popis

Táto trieda je zodpovedná za správu nápovedy figúrky.

4.15.2 Dokumentácia k metódam

4.15.2.1 CreatePiece()

```
void PieceInfoController.CreatePiece ( ) [inline], [private]
```

Objekt figúrky sa vytvorí a umiestni v strede šachovnice.

4.15.2.2 OnCloseButtonClicked()

```
void PieceInfoController.OnCloseButtonClicked ( ) [inline], [private]
```

Po kliknutí tlačidla zatvorenia sa toto rozhranie deaktivuje.

4.15.2.3 UpdateDisplayedPiece()

```
void PieceInfoController.UpdateDisplayedPiece ( ) [inline], [private]
```

Zobrazená figúrka sa aktualizuje a zobrazia sa jej dostupné pohyby.

4.15.3 Dokumentácia k vlastnosti

4.15.3.1 MyPiece

`Piece` PieceInfoController.MyPiece [get], [set]

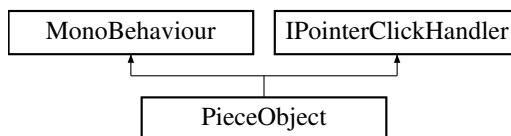
Aktuálne zobrazená figúrka. Ak ju zmeníme, tak sa aktualizuje jej zobrazenie.

Dokumentácia pre túto triedu bola generovaná z nasledujúceho súboru:

- PieceInfoController.cs

4.16 Dokumentácia triedy PieceObject

Diagram dedičnosti pre triedu PieceObject



Verejné metódy

- void `OnPointerClick` (PointerEventData eventData)

Verejné atribúty

- UnityEvent< `Piece` > `onPieceClicked`

Vlastnosti

- `Piece` `Piece` [get, set]
- bool `Selected` [get, set]
- bool `Highlighted` [get, set]
- Color `HighlightColor` [get, set]

Privátne atribúty

- Image `pieceImage`
- Image `outlineImage`
- Image `backgroundImage`
- `Piece` `piece` = null
- bool `selected` = false
- bool `highlighted` = false

4.16.1 Detailný popis

Táto trieda spravuje objekt figúrky.

4.16.2 Dokumentácia k metódam

4.16.2.1 OnPointerClick()

```
void PieceObject.OnPointerClick (  
    PointerEventData eventData ) [inline]
```

Pri kliknutí na tento objekt sa to oznámi poslucháčom jeho onClick eventu.

Parametre

<i>eventData</i>	Informácie o kliknutí.
------------------	------------------------

4.16.3 Dokumentácia k dátovým členom

4.16.3.1 highlighted

```
bool PieceObject.highlighted = false [private]
```

Informácia o znázornení figúrky. Ak sa zmení znázornenie figúrky, tak sa aktivuje alebo deaktivuje pozadie figúrky.

4.16.4 Dokumentácia k vlastnosti

4.16.4.1 HighlightColor

```
Color PieceObject.HighlightColor [get], [set]
```

Aktuálna farba znázornenia figúrky. Ak sa táto vlastnosť zmení, tak sa aktualizuje farba pozadia.

4.16.4.2 Piece

```
Piece PieceObject.Piece [get], [set]
```

Figúrka priradená tomuto objektu. Ak ju zmeníme, tak sa aktualizuje jej zobrazenie.

4.16.4.3 Selected

```
bool PieceObject.Selected [get], [set]
```

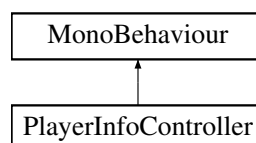
Informácia o výbere figúrky. Ak sa zmení výber figúrky, tak sa aktivuje alebo deaktivuje outline figúrky.

Dokumentácia pre túto triedu bola generovaná z nasledujúceho súboru:

- [PieceObject.cs](#)

4.17 Dokumentácia triedy PlayerInfoController

Diagram dedičnosti pre triedu PlayerInfoController



Verejné metódy

- void **SetPieces** ([Chess.Color](#) color, [Chess.Essence](#) pawnEssence, [Chess.Essence](#) knightEssence, [Chess.Essence](#) bishopEssence, [Chess.Essence](#) rookEssence)

Verejné atribúty

- UnityEvent< [Chess.PieceType](#), [Chess.Color](#), [Chess.Essence](#) > **onHelpButtonClicked** = new()

Privátne metódy

- void **Start** ()
- void **OnHelpButtonClicked** ([Chess.PieceType](#) pieceType, [Chess.Color](#) color, [Chess.Essence](#) pieceEssence)

Privátne atribúty

- Image **pawnImage**
- Image **knightImage**
- Image **bishopImage**
- Image **rookImage**
- Image **queenImage**
- Image **kingImage**
- Button **pawnButton**
- Button **knightButton**
- Button **bishopButton**
- Button **rookButton**
- Button **queenButton**
- Button **kingButton**
- [Chess.Color](#) **color**
- [Chess.Essence](#) **pawnEssence**
- [Chess.Essence](#) **knightEssence**
- [Chess.Essence](#) **bishopEssence**
- [Chess.Essence](#) **rookEssence**

4.17.1 Detailný popis

Táto trieda spravuje rozhranie hráčovych informácií.

4.17.2 Dokumentácia k metódam

4.17.2.1 OnHelpButtonClicked()

```
void PlayerInfoController.OnHelpButtonClicked (  
    Chess::PieceType pieceType,  
    Chess::Color color,  
    Chess::Essence pieceEssence ) [inline], [private]
```

Po kliknutí na tlačidlo nápovedy sa informácia o vyžiadanej nápovede prepošle poslucháčom (tí otvoria okno nápovedy).

Parametre

<i>pieceType</i>	Typ figúrky.
<i>color</i>	Farba figúrky.
<i>pieceEssence</i>	Esencia figúrky.

4.17.2.2 SetPieces()

```
void PlayerInfoController.SetPieces (
    Chess::Color color,
    Chess::Essence pawnEssence,
    Chess::Essence knightEssence,
    Chess::Essence bishopEssence,
    Chess::Essence rookEssence ) [inline]
```

Táto metóda nastaví farbu a esencie figúrok hráča.

Parametre

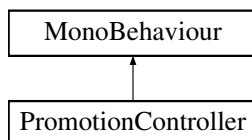
<i>color</i>	Farba hráča.
<i>pawnEssence</i>	Esencia pešiaka.
<i>knightEssence</i>	Esencia rytiera.
<i>bishopEssence</i>	Esencia strelca.
<i>rookEssence</i>	Esencia veže.

Dokumentácia pre túto triedu bola generovaná z nasledujúceho súboru:

- [PlayerInfoController.cs](#)

4.18 Dokumentácia triedy PromotionController

Diagram dedičnosti pre triedu PromotionController



Verejné metódy

- void [SetPieces](#) ([Chess.Essence](#) rookEssence, [Chess.Essence](#) bishopEssence, [Chess.Essence](#) knight↔ Essence)
- void [SetColor](#) ([Chess.Color](#) color)

Verejné atribúty

- UnityEvent< [Chess.PieceType](#) > [onPromotionChosen](#)

Privátne metódy

- void **Start** ()
- void **OnPromotionBtnClicked** ([Chess.PieceType](#) pieceType)

Privátne atribúty

- Button **queenBtn**
- Button **rookBtn**
- Button **bishopBtn**
- Button **knightBtn**
- Image **queenImage**
- Image **rookImage**
- Image **bishopImage**
- Image **knightImage**
- [Chess.Essence](#) **rookEssence**
- [Chess.Essence](#) **bishopEssence**
- [Chess.Essence](#) **knightEssence**

4.18.1 Detailný popis

Táto trieda je zodpovedná za správu rozhrania výberu promócie pešiaka.

4.18.2 Dokumentácia k metódam

4.18.2.1 OnPromotionBtnClicked()

```
void PromotionController.OnPromotionBtnClicked (
    Chess::PieceType pieceType ) [inline], [private]
```

Po vybratí promócie sa táto informácia odošle poslucháčom (tí vykonajú promóciu pešiaka).

Parametre

<i>pieceType</i>	Typ vybranej figúrky.
------------------	-----------------------

4.18.2.2 SetColor()

```
void PromotionController.SetColor (
    Chess::Color color ) [inline]
```

Táto metóda nastaví farbu hráča pre dostupné promócie pešiaka.

Parametre

<i>color</i>	Farba hráča, ktorý si vyberá promóciu.
--------------	--

4.18.2.3 SetPieces()

```
void PromotionController.SetPieces (
    Chess::Essence rookEssence,
    Chess::Essence bishopEssence,
    Chess::Essence knightEssence ) [inline]
```

Pri výbere promócie sa pešiak zmení na esenciu vybraného typu figúrky. Táto metóda nastaví prislúchajúce esencie dostupným typom figúrok.

Parametre

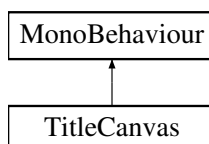
<i>rookEssence</i>	Esencia veže.
<i>bishopEssence</i>	Esencia strelca.
<i>knightEssence</i>	Esencia rytiera.

Dokumentácia pre túto triedu bola generovaná z nasledujúceho súboru:

- [PromotionController.cs](#)

4.19 Dokumentácia triedy TitleCanvas

Diagram dedičnosti pre triedu TitleCanvas



Privátne metódy

- void **Start** ()
- void [OnPlayClick](#) ()
- void [OnGameCreated](#) ([Chess.GameSettings](#) gameSettings)
- void [OnQuitClick](#) ()

Privátne atribúty

- Button **playBtn**
- Button **quitBtn**
- [GameCanvas](#) **gameCanvas**
- [EngineController](#) **engineController**
- GameObject **createGameOverlay** = null

4.19.1 Detailný popis

Táto trieda je zodpovedná za správu titulnej obrazovky.

4.19.2 Dokumentácia k metódam

4.19.2.1 OnGameCreated()

```
void TitleCanvas.OnGameCreated (
    Chess::GameSettings gameSettings ) [inline], [private]
```

Pri prijatí hernej konfigurácie od rozhrania vytvorenia novej hry sa presunieme do rozhrania hernej obrazovky s danou konfiguráciou.

Parametre

<code>gameSettings</code>	Herná konfigurácia.
---------------------------	---------------------

4.19.2.2 OnPlayClick()

```
void TitleCanvas.OnPlayClick ( ) [inline], [private]
```

Po stlačení tlačidla spustenia hry sa zobrazí rozhranie vytvorenia novej hry.

4.19.2.3 OnQuitClick()

```
void TitleCanvas.OnQuitClick ( ) [inline], [private]
```

Po stlačení tlačidla vypnutia hry sa terminuje aplikácia.

Dokumentácia pre túto triedu bola generovaná z nasledujúceho súboru:

- [TitleCanvas.cs](#)

4.20 Dokumentácia triedy Turn

Verejné metódy

- [Turn](#) (int [turnCounter](#))
- string [GetWhitePly](#) ()
- string [GetBlackPly](#) ()
- void [SetWhitePly](#) (Chess.PieceType [whitePieceType](#), Chess.PieceType [whitePromotion](#), Chess.MovementType [whiteMovementType](#), Move [whiteMove](#))
- void [SetBlackPly](#) (Chess.PieceType [blackPieceType](#), Chess.PieceType [blackPromotion](#), Chess.MovementType [blackMovementType](#), Move [blackMove](#))

Verejné atribúty

- int `turnCounter`
- `Chess.PieceType` `whitePieceType` = `Chess.PieceType.Pawn`
- `Chess.PieceType` `whitePromotion` = `Chess.PieceType.Pawn`
- `Chess.MovementType` `whiteMovementType` = `Chess.MovementType.Move`
- `Move` `whiteMove` = null
- `Chess.PieceType` `blackPieceType` = `Chess.PieceType.Pawn`
- `Chess.PieceType` `blackPromotion` = `Chess.PieceType.Pawn`
- `Chess.MovementType` `blackMovementType` = `Chess.MovementType.Move`
- `Move` `blackMove` = null

Privátne metódy

- string `GetPly` (`Chess.PieceType` `pieceType`, `Chess.MovementType` `movementType`, `Move` `move`, `Chess.PieceType` `promotionType`)

4.20.1 Detailný popis

Trieda reprezentujúca herný ťah.

4.20.2 Dokumentácia konštruktoru a deštruktoru

4.20.2.1 Turn()

```
Turn.Turn (
    int turnCounter ) [inline]
```

Konštruktor herného ťahu.

Parametre

<code>turnCounter</code>	Poradie herného ťahu.
--------------------------	-----------------------

4.20.3 Dokumentácia k metódam

4.20.3.1 GetBlackPly()

```
string Turn.GetBlackPly ( ) [inline]
```

Táto metóda vráti string reprezentujúci čierny polťah.

Návratová hodnota

Čierny polťah v algebraickej notácii.

4.20.3.2 GetPly()

```
string Turn.GetPly (
    Chess::PieceType pieceType,
    Chess::MovementType movementType,
    Move move,
    Chess::PieceType promotionType ) [inline], [private]
```

Táto metóda vytvorí algebraickú notáciu daného pohybu.

Parametre

<i>pieceType</i>	Typ presunutej figúrky.
<i>movementType</i>	Typ vykonaného pohybu.
<i>promotionType</i>	Typ vykonanej promócie pešiaka.

4.20.3.3 GetWhitePly()

```
string Turn.GetWhitePly ( ) [inline]
```

Táto metóda vráti string reprezentujúci biely polťah.

Návratová hodnota

Biely polťah v algebraickej notácii.

4.20.3.4 SetBlackPly()

```
void Turn.SetBlackPly (
    Chess::PieceType blackPieceType,
    Chess::PieceType blackPromotion,
    Chess::MovementType blackMovementType,
    Move blackMove ) [inline]
```

Táto metóda nastaví čierny polťah.

Parametre

<i>blackPieceType</i>	Typ presunutej figúrky.
<i>blackPromotion</i>	Promócia pešiaka.
<i>blackMovementType</i>	Typ pohybu.
<i>blackMove</i>	Vykonaný pohyb

4.20.3.5 SetWhitePly()

```
void Turn.SetWhitePly (
    Chess::PieceType whitePieceType,
```

```
Chess::PieceType whitePromotion,  
Chess::MovementType whiteMovementType,  
Move whiteMove ) [inline]
```

Táto metóda nastaví biely polťah.

Parametre

<i>whitePieceType</i>	Typ presunutej figúrky.
<i>whitePromotion</i>	Promócia pešiaka.
<i>whiteMovementType</i>	Typ pohybu.
<i>whiteMove</i>	Vykonaný pohyb

4.20.4 Dokumentácia k dátovým členom

4.20.4.1 blackMove

```
Move Turn.blackMove = null
```

Vykonaný pohyb čierneho hráča (čierny polťah).

4.20.4.2 blackMovementType

```
Chess.MovementType Turn.blackMovementType = Chess.MovementType.Move
```

Typ čierneho pohybu (čierny polťah).

4.20.4.3 blackPieceType

```
Chess.PieceType Turn.blackPieceType = Chess.PieceType.Pawn
```

Typ čiernej figúrky (čierny polťah).

4.20.4.4 blackPromotion

```
Chess.PieceType Turn.blackPromotion = Chess.PieceType.Pawn
```

Typ čiernej promócie (čierny polťah).

4.20.4.5 turnCounter

```
int Turn.turnCounter
```

Poradie ťahu.

4.20.4.6 whiteMove

```
Move Turn.whiteMove = null
```

Vykonaný pohyb bieleho hráča (biely polťah).

4.20.4.7 whiteMovementType

```
Chess.MovementType Turn.whiteMovementType = Chess.MovementType.Move
```

Typ bieleho pohybu (biely polťah).

4.20.4.8 whitePieceType

```
Chess.PieceType Turn.whitePieceType = Chess.PieceType.Pawn
```

Typ bielej figúrky (biely polťah).

4.20.4.9 whitePromotion

```
Chess.PieceType Turn.whitePromotion = Chess.PieceType.Pawn
```

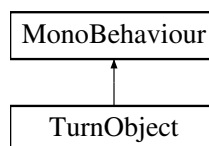
Typ bielej promócie (biely polťah).

Dokumentácia pre túto triedu bola generovaná z nasledujúceho súboru:

- [Turn.cs](#)

4.21 Dokumentácia triedy TurnObject

Diagram dedičnosti pre triedu TurnObject



Verejné metódy

- void [UpdateWhitePly](#) ([Chess.PieceType](#) pieceType, [Chess.PieceType](#) piecePromotion, [Chess.MovementType](#) movementType, [Move](#) move)
- void [UpdateBlackPly](#) ([Chess.PieceType](#) pieceType, [Chess.PieceType](#) piecePromotion, [Chess.MovementType](#) movementType, [Move](#) move)

Vlastnosti

- [Turn MyTurn](#) [get, set]

Privátne metódy

- void [UpdateDisplayedText](#) ()

Privátne atribúty

- TMP_Text **display**
- [Turn myTurn](#) = null

4.21.1 Detailný popis

Táto trieda spravuje objekt herného ťahu.

4.21.2 Dokumentácia k metódam

4.21.2.1 UpdateBlackPly()

```
void TurnObject.UpdateBlackPly (
    Chess::PieceType pieceType,
    Chess::PieceType piecePromotion,
    Chess::MovementType movementType,
    Move move ) [inline]
```

Upravíme čierny polťah herného ťahu.

Parametre

<i>pieceType</i>	Typ pohnutej figúrky.
<i>piecePromotion</i>	Promócia pešiaka.
<i>movementType</i>	Typ pohybu.
<i>move</i>	Vykonaný pohyb.

4.21.2.2 UpdateDisplayedText()

```
void TurnObject.UpdateDisplayedText ( ) [inline], [private]
```

Aktualizujeme zobrazený text herného ťahu. Herný ťah ma formát: @turnCounter. @whitePly @blackPly

4.21.2.3 UpdateWhitePly()

```
void TurnObject.UpdateWhitePly (
    Chess::PieceType pieceType,
```

```
Chess::PieceType piecePromotion,  
Chess::MovementType movementType,  
Move move ) [inline]
```

Upravíme biely polťah herného ťahu.

Parametre

<i>pieceType</i>	Typ pohnutej figúrky.
<i>piecePromotion</i>	Promócia pešiaka.
<i>movementType</i>	Typ pohybu.
<i>move</i>	Vykonaný pohyb.

4.21.3 Dokumentácia k vlastnosti

4.21.3.1 MyTurn

```
Turn TurnObject.MyTurn [get], [set]
```

Herný ťah priradený tomuto objektu. Ak ho zmeníme, tak sa aktualizuje zobrazený text.

Dokumentácia pre túto triedu bola generovaná z nasledujúceho súboru:

- [TurnObject.cs](#)

Kapitola 5

Dokumentácia súborov

5.1 Dokumentácia súboru Chess.cs

Implementácia generických funkcionalít a šachových pojmov.

Triedy

- class [Chess](#)
- class [Chess.GameSettings](#)

5.1.1 Detailný popis

Implementácia generických funkcionalít a šachových pojmov.

Autor

Martin Šváb

Dátum

Máj 2024

5.2 Dokumentácia súboru Ghost.cs

Špeciálna figúrka vytvorená po vykonaní pohybu s hasty flagom.

Triedy

- class [Ghost](#)

5.2.1 Detailný popis

Špeciálna figúrka vytvorená po vykonaní pohybu s hasty flagom.

Autor

Martin Šváb

Dátum

Máj 2024

5.3 Dokumentácia súboru Mobility.cs

Konfigurácia mobility figúrky.

Triedy

- class [Mobility](#)

5.3.1 Detailný popis

Konfigurácia mobility figúrky.

Autor

Martin Šváb

Dátum

Máj 2024

5.4 Dokumentácia súboru Move.cs

Pohyb na šachovnici.

Triedy

- class [Move](#)

Enumerácie

- enum [Castling](#) { [None](#) , [QueenSide](#) , [KingSide](#) }

5.4.1 Detailný popis

Pohyb na šachovnici.

Autor

Martin Šváb

Dátum

Máj 2024

5.4.2 Dokumentácia enumeračných typov

5.4.2.1 Castling

enum `Castling`

Enumerátor reprezentujúci typ rošády.

Hodnoty enumerácií

None	Null hodnota
QueenSide	Rošáda z kráľovninej strany.
KingSide	Rošáda z kráľovej strany.

5.5 Dokumentácia súboru Movement.cs

Pohyb figúrky.

Triedy

- class `Movement`

5.5.1 Detailný popis

Pohyb figúrky.

Autor

Martin Šváb

Dátum

Máj 2024

5.6 Dokumentácia súboru Piece.cs

Šachová figúrka.

Triedy

- class [Piece](#)

5.6.1 Detailný popis

Šachová figúrka.

Autor

Martin Šváb

Dátum

Máj 2024

5.7 Dokumentácia súboru Turn.cs

Herný ťah.

Triedy

- class [Turn](#)

5.7.1 Detailný popis

Herný ťah.

Autor

Martin Šváb

Dátum

Máj 2024

5.8 Dokumentácia súboru EngineController.cs

Ovládač šachového enginu Malakh.

Triedy

- class [EngineController](#)

5.8.1 Detailný popis

Ovládač šachového enginu Malakh.

Autor

Martin Šváb

Dátum

Máj 2024

5.9 Dokumentácia súboru BoardController.cs

Ovládač rozhrania šachovnice.

Triedy

- class [BoardController](#)

5.9.1 Detailný popis

Ovládač rozhrania šachovnice.

Autor

Martin Šváb

Dátum

Máj 2024

5.10 Dokumentácia súboru GameCanvas.cs

Ovládač rozhrania hernej obrazovky.

Triedy

- class [GameCanvas](#)

5.10.1 Detailný popis

Ovládač rozhrania hernej obrazovky.

Autor

Martin Šváb

Dátum

Máj 2024

5.11 Dokumentácia súboru TitleCanvas.cs

Ovládač rozhrania titulnej obrazovky.

Triedy

- class [TitleCanvas](#)

5.11.1 Detailný popis

Ovládač rozhrania titulnej obrazovky.

Autor

Martin Šváb

Dátum

Máj 2024

5.12 Dokumentácia súboru ClickableDropdown.cs

Rozšírenie dropdown komponentu.

Triedy

- class [ClickableDropdown](#)

5.12.1 Detailný popis

Rozšírenie dropdown komponentu.

Autor

Martin Šváb

Dátum

Máj 2024

5.13 Dokumentácia súboru CreateGameOverlay.cs

Ovládač rozhrania vytvorenia novej hry.

Triedy

- class [CreateGameOverlay](#)

5.13.1 Detailný popis

Ovládač rozhrania vytvorenia novej hry.

Autor

Martin Šváb

Dátum

Máj 2024

5.14 Dokumentácia súboru PlayerInfoController.cs

Ovládač nápoedy figúrky.

Triedy

- class [PlayerInfoController](#)

5.14.1 Detailný popis

Ovládač náповedy figúrky.

Ovládač rozhrania hráčovych informácií.

Autor

Martin Šváb

Dátum

Máj 2024

5.15 Dokumentácia súboru GhostObject.cs

Ovládač objektu ducha.

Triedy

- class [GhostObject](#)

5.15.1 Detailný popis

Ovládač objektu ducha.

Autor

Martin Šváb

Dátum

Máj 2024

5.16 Dokumentácia súboru MovementObject.cs

Ovládač objektu pohybu.

Triedy

- class [MovementObject](#)

5.16.1 Detailný popis

Ovládač objektu pohybu.

Autor

Martin Šváb

Dátum

Máj 2024

5.17 Dokumentácia súboru PieceObject.cs

Ovládač objektu figúrky.

Triedy

- class [PieceObject](#)

5.17.1 Detailný popis

Ovládač objektu figúrky.

Autor

Martin Šváb

Dátum

Máj 2024

5.18 Dokumentácia súboru TurnObject.cs

Ovládač objektu herného ťahu.

Triedy

- class [TurnObject](#)

5.18.1 Detailný popis

Ovládač objektu herného ťahu.

Autor

Martin Šváb

Dátum

Máj 2024

5.19 Dokumentácia súboru PromotionController.cs

Ovládač rozhrania výberu promócie pešiaka.

Triedy

- class [PromotionController](#)

5.19.1 Detailný popis

Ovládač rozhrania výberu promócie pešiaka.

Autor

Martin Šváb

Dátum

Máj 2024

Register

- AddMovement
 - BoardController, [9](#)
- AddPiece
 - BoardController, [9](#)
- AddTurn
 - GameCanvas, [27](#)
- AI
 - Chess, [15](#)
- ai
 - Chess.GameSettings, [32](#)
- Attack
 - Chess, [16](#)
- AttackMove
 - Chess, [16](#)
- availableMoves
 - Piece, [44](#)
- Basic
 - Chess, [15](#)
- Bishop
 - Chess, [16](#)
- Black
 - Chess, [15](#)
- blackMove
 - Turn, [57](#)
- blackMovementType
 - Turn, [57](#)
- blackPieceType
 - Turn, [57](#)
- blackPromotion
 - Turn, [57](#)
- Blue
 - Chess, [15](#)
- BoardController, [7](#)
 - AddMovement, [9](#)
 - AddPiece, [9](#)
 - ClearGhost, [9](#)
 - ClearHighlightedSquare, [9](#)
 - ClearMovements, [9](#)
 - ClearPieces, [10](#)
 - CreateInitialPosition, [10](#)
 - GetPiece, [10](#)
 - GetPieceObject, [10](#)
 - InitBoard, [11](#)
 - MoveGameObject, [11](#)
 - MovePiece, [11](#)
 - OnMovementClicked, [11](#)
 - OnPieceClicked, [12](#)
 - PerformMove, [12](#)
 - PerformPromotion, [12](#)
 - SetBoardOrientation, [12](#)
 - SetCheck, [13](#)
 - SetGhost, [13](#)
 - SetHighlightedSquare, [13](#)
 - SetLegalMoves, [13](#)
 - SetPly, [14](#)
- BoardController.cs, [65](#)
- Castling
 - Move.cs, [63](#)
- castling
 - Move, [40](#)
- Chess, [14](#)
 - AI, [15](#)
 - Attack, [16](#)
 - AttackMove, [16](#)
 - Basic, [15](#)
 - Bishop, [16](#)
 - Black, [15](#)
 - Blue, [15](#)
 - Classic, [15](#)
 - Color, [15](#)
 - Ensemble, [15](#)
 - Essence, [15](#)
 - GetMovementImage, [16](#)
 - GetPieceImage, [16](#)
 - King, [16](#)
 - Knight, [16](#)
 - Move, [16](#)
 - MovementType, [15](#)
 - Pawn, [16](#)
 - PieceType, [16](#)
 - Queen, [16](#)
 - Random, [15](#)
 - Red, [15](#)
 - Rook, [16](#)
 - White, [15](#)
- Chess.cs, [61](#)
- Chess.GameSettings, [31](#)
 - ai, [32](#)
 - GameSettings, [32](#)
 - malakhBishop, [32](#)
 - malakhColor, [32](#)
 - malakhKnight, [33](#)
 - malakhPawn, [33](#)
 - malakhRook, [33](#)
 - playerBishop, [33](#)
 - playerColor, [33](#)

- playerKnight, [33](#)
 - playerPawn, [33](#)
 - playerRook, [33](#)
- Classic
 - Chess, [15](#)
- ClearGhost
 - BoardController, [9](#)
- ClearHasty
 - Move, [39](#)
- ClearHighlightedSquare
 - BoardController, [9](#)
- ClearMovements
 - BoardController, [9](#)
- ClearPieces
 - BoardController, [10](#)
- ClearTurns
 - GameCanvas, [27](#)
- ClearVigilant
 - Move, [39](#)
- ClickableDropdown, [17](#)
 - OnPointerClick, [17](#)
- ClickableDropdown.cs, [66](#)
- Color
 - Chess, [15](#)
- color
 - Piece, [44](#)
- CreateGameOverlay, [18](#)
 - InitializeSettings, [19](#)
 - OnMalakhColorSet, [19](#)
 - OnMalakhRandomPiecesToggled, [20](#)
 - OnPlayClick, [20](#)
 - OnPlayerColorSet, [20](#)
 - OnPlayerRandomPiecesToggled, [20](#)
 - OnReturnClick, [22](#)
 - SetEngineController, [22](#)
 - SetPieceInfo, [22](#)
- CreateGameOverlay.cs, [67](#)
- CreateInitialPosition
 - BoardController, [10](#)
- CreatePiece
 - PieceInfoController, [46](#)
- direction_x
 - Mobility, [37](#)
- direction_y
 - Mobility, [37](#)
- EngineController, [22](#)
 - engineProcess, [26](#)
 - MovePiece, [23](#)
 - OnApplicationQuit, [24](#)
 - ParseMove, [24](#)
 - ReceiveEngineResponses, [24](#)
 - RequestAvailableMoves, [24](#)
 - RequestBestMove, [24](#)
 - SendCommand, [24](#)
 - Start, [25](#)
 - StartEngine, [25](#)
 - StartGame, [25](#)
 - StopEngine, [25](#)
 - Update, [25](#)
- EngineController.cs, [64](#)
- engineProcess
 - EngineController, [26](#)
- Ensemble
 - Chess, [15](#)
- Essence
 - Chess, [15](#)
- essence
 - Piece, [44](#)
- GameCanvas, [26](#)
 - AddTurn, [27](#)
 - ClearTurns, [27](#)
 - OnBestMoveReceived, [27](#)
 - OnCheckReceived, [28](#)
 - OnExportClick, [28](#)
 - OnGameCreated, [28](#)
 - OnHelpButtonClicked, [28](#)
 - OnLegalMovesReceived, [29](#)
 - OnPlayClick, [29](#)
 - OnPlayerMove, [29](#)
 - OnPromotionChosen, [29](#)
 - OnPromotionRequested, [29](#)
 - OnResultReceived, [30](#)
 - OnReturnClick, [30](#)
 - SetBlackPly, [30](#)
 - SetGame, [30](#)
 - SetTurn, [31](#)
- GameCanvas.cs, [65](#)
- GameSettings
 - Chess.GameSettings, [32](#)
- GetBlackPly
 - Turn, [55](#)
- GetMovementImage
 - Chess, [16](#)
- GetPiece
 - BoardController, [10](#)
- GetPieceImage
 - Chess, [16](#)
- GetPieceObject
 - BoardController, [10](#)
- GetPly
 - Turn, [55](#)
- GetWhitePly
 - Turn, [56](#)
- Ghost, [34](#)
 - Ghost, [34](#)
 - GhostObject, [36](#)
 - parent, [35](#)
 - x, [35](#)
 - y, [35](#)
- Ghost.cs, [61](#)
- GhostObject, [35](#)
 - Ghost, [36](#)
- GhostObject.cs, [68](#)
- hasty

- Move, [40](#)
- hastyX
 - Move, [40](#)
- hastyY
 - Move, [40](#)
- HighlightColor
 - PieceObject, [49](#)
- highlighted
 - PieceObject, [49](#)
- InitBoard
 - BoardController, [11](#)
- InitializeSettings
 - CreateGameOverlay, [19](#)
- King
 - Chess, [16](#)
- KingSide
 - Move.cs, [63](#)
- Knight
 - Chess, [16](#)
- limit
 - Mobility, [37](#)
- malakhBishop
 - Chess.GameSettings, [32](#)
- malakhColor
 - Chess.GameSettings, [32](#)
- malakhKnight
 - Chess.GameSettings, [33](#)
- malakhPawn
 - Chess.GameSettings, [33](#)
- malakhRook
 - Chess.GameSettings, [33](#)
- Mobility, [36](#)
 - direction_x, [37](#)
 - direction_y, [37](#)
 - limit, [37](#)
 - Mobility, [36](#)
 - start_x, [37](#)
 - start_y, [37](#)
 - type, [37](#)
- Mobility.cs, [62](#)
- Move, [38](#)
 - castling, [40](#)
 - Chess, [16](#)
 - ClearHasty, [39](#)
 - ClearVigilant, [39](#)
 - hasty, [40](#)
 - hastyX, [40](#)
 - hastyY, [40](#)
 - Move, [38, 39](#)
 - SetHasty, [39](#)
 - SetVigilant, [39](#)
 - vigilant, [40](#)
 - x1, [40](#)
 - x2, [40](#)
 - y1, [41](#)
 - y2, [41](#)
- move
 - Movement, [42](#)
- Move.cs, [62](#)
 - Castling, [63](#)
 - KingSide, [63](#)
 - None, [63](#)
 - QueenSide, [63](#)
- MoveGameObject
 - BoardController, [11](#)
- Movement, [41](#)
 - move, [42](#)
 - Movement, [41](#)
 - MovementObject, [43](#)
 - owner, [42](#)
 - type, [42](#)
- Movement.cs, [63](#)
- MovementObject, [42](#)
 - Movement, [43](#)
 - OnPointerClick, [43](#)
- MovementObject.cs, [68](#)
- MovementType
 - Chess, [15](#)
- MovePiece
 - BoardController, [11](#)
 - EngineController, [23](#)
- MyPiece
 - PieceInfoController, [47](#)
- MyTurn
 - TurnObject, [60](#)
- None
 - Move.cs, [63](#)
- OnApplicationQuit
 - EngineController, [24](#)
- OnBestMoveReceived
 - GameCanvas, [27](#)
- OnCheckReceived
 - GameCanvas, [28](#)
- OnCloseButtonClicked
 - PieceInfoController, [46](#)
- OnExportClick
 - GameCanvas, [28](#)
- OnGameCreated
 - GameCanvas, [28](#)
 - TitleCanvas, [54](#)
- OnHelpButtonClicked
 - GameCanvas, [28](#)
 - PlayerInfoController, [50](#)
- OnLegalMovesReceived
 - GameCanvas, [29](#)
- OnMalakhColorSet
 - CreateGameOverlay, [19](#)
- OnMalakhRandomPiecesToggled
 - CreateGameOverlay, [20](#)
- OnMovementClicked
 - BoardController, [11](#)
- OnPieceClicked

- BoardController, [12](#)
- OnPlayClick
 - CreateGameOverlay, [20](#)
 - GameCanvas, [29](#)
 - TitleCanvas, [54](#)
- OnPlayerColorSet
 - CreateGameOverlay, [20](#)
- OnPlayerMove
 - GameCanvas, [29](#)
- OnPlayerRandomPiecesToggled
 - CreateGameOverlay, [20](#)
- OnPointerClick
 - ClickableDropdown, [17](#)
 - MovementObject, [43](#)
 - PieceObject, [48](#)
- OnPromotionBtnClicked
 - PromotionController, [52](#)
- OnPromotionChosen
 - GameCanvas, [29](#)
- OnPromotionRequested
 - GameCanvas, [29](#)
- OnQuitClick
 - TitleCanvas, [54](#)
- OnResultReceived
 - GameCanvas, [30](#)
- OnReturnClick
 - CreateGameOverlay, [22](#)
 - GameCanvas, [30](#)
- owner
 - Movement, [42](#)
- parent
 - Ghost, [35](#)
- ParseMove
 - EngineController, [24](#)
- Pawn
 - Chess, [16](#)
- PerformMove
 - BoardController, [12](#)
- PerformPromotion
 - BoardController, [12](#)
- Piece, [43](#)
 - availableMoves, [44](#)
 - color, [44](#)
 - essence, [44](#)
 - Piece, [44](#)
 - PieceObject, [49](#)
 - type, [45](#)
 - x, [45](#)
 - y, [45](#)
- Piece.cs, [64](#)
- PieceInfoController, [45](#)
 - CreatePiece, [46](#)
 - MyPiece, [47](#)
 - OnCloseButtonClicked, [46](#)
 - UpdateDisplayedPiece, [46](#)
- PieceObject, [47](#)
 - HighlightColor, [49](#)
 - highlighted, [49](#)
 - OnPointerClick, [48](#)
 - Piece, [49](#)
 - Selected, [49](#)
- PieceObject.cs, [69](#)
- PieceType
 - Chess, [16](#)
- playerBishop
 - Chess.GameSettings, [33](#)
- playerColor
 - Chess.GameSettings, [33](#)
- PlayerInfoController, [49](#)
 - OnHelpButtonClicked, [50](#)
 - SetPieces, [51](#)
- PlayerInfoController.cs, [67](#)
- playerKnight
 - Chess.GameSettings, [33](#)
- playerPawn
 - Chess.GameSettings, [33](#)
- playerRook
 - Chess.GameSettings, [33](#)
- PromotionController, [51](#)
 - OnPromotionBtnClicked, [52](#)
 - SetColor, [52](#)
 - SetPieces, [52](#)
- PromotionController.cs, [70](#)
- Queen
 - Chess, [16](#)
- QueenSide
 - Move.cs, [63](#)
- Random
 - Chess, [15](#)
- ReceiveEngineResponses
 - EngineController, [24](#)
- Red
 - Chess, [15](#)
- RequestAvailableMoves
 - EngineController, [24](#)
- RequestBestMove
 - EngineController, [24](#)
- Rook
 - Chess, [16](#)
- Selected
 - PieceObject, [49](#)
- SendCommand
 - EngineController, [24](#)
- SetBlackPly
 - GameCanvas, [30](#)
 - Turn, [56](#)
- SetBoardOrientation
 - BoardController, [12](#)
- SetCheck
 - BoardController, [13](#)
- SetColor
 - PromotionController, [52](#)
- SetEngineController
 - CreateGameOverlay, [22](#)

SetGame
 GameCanvas, 30
 SetGhost
 BoardController, 13
 SetHasty
 Move, 39
 SetHighlightedSquare
 BoardController, 13
 SetLegalMoves
 BoardController, 13
 SetPieceInfo
 CreateGameOverlay, 22
 SetPieces
 PlayerInfoController, 51
 PromotionController, 52
 SetPly
 BoardController, 14
 SetTurn
 GameCanvas, 31
 SetVigilant
 Move, 39
 SetWhitePly
 Turn, 56
 Start
 EngineController, 25
 start_x
 Mobility, 37
 start_y
 Mobility, 37
 StartEngine
 EngineController, 25
 StartGame
 EngineController, 25
 StopEngine
 EngineController, 25
 TitleCanvas, 53
 OnGameCreated, 54
 OnPlayClick, 54
 OnQuitClick, 54
 TitleCanvas.cs, 66
 Turn, 54
 blackMove, 57
 blackMovementType, 57
 blackPieceType, 57
 blackPromotion, 57
 GetBlackPly, 55
 GetPly, 55
 GetWhitePly, 56
 SetBlackPly, 56
 SetWhitePly, 56
 Turn, 55
 turnCounter, 57
 whiteMove, 57
 whiteMovementType, 58
 whitePieceType, 58
 whitePromotion, 58
 Turn.cs, 64
 turnCounter
 Turn, 57
 TurnObject, 58
 MyTurn, 60
 UpdateBlackPly, 59
 UpdateDisplayedText, 59
 UpdateWhitePly, 59
 TurnObject.cs, 69
 type
 Mobility, 37
 Movement, 42
 Piece, 45
 Update
 EngineController, 25
 UpdateBlackPly
 TurnObject, 59
 UpdateDisplayedPiece
 PieceInfoController, 46
 UpdateDisplayedText
 TurnObject, 59
 UpdateWhitePly
 TurnObject, 59
 vigilant
 Move, 40
 White
 Chess, 15
 whiteMove
 Turn, 57
 whiteMovementType
 Turn, 58
 whitePieceType
 Turn, 58
 whitePromotion
 Turn, 58
 x
 Ghost, 35
 Piece, 45
 x1
 Move, 40
 x2
 Move, 40
 y
 Ghost, 35
 Piece, 45
 y1
 Move, 41
 y2
 Move, 41