public void GriviTripsGame() {  
 boolean turn = false;  
 boolean winner = false;  
 int playerToPlay;  
 int selectedGameMode;  
 char player;  
  
 selectedGameMode = selectGameMode();  
 if (selectedGameMode == 1) {  
 field.drawEmptyField();  
 while (winner == false) {  
 if (turn == false) {  
 playerToPlay = humanPlayer.playersTurn(*FIRSTPLALYER*);  
 turn = turnInGame(playerToPlay, turn);  
 player = PlayersChars(playerToPlay);  
 winner = winnerMessege(player, playerToPlay);  
 } else {  
 playerToPlay = humanPlayer.playersTurn(*SECONDPLAYER*);  
 turn = turnInGame(playerToPlay, turn);  
 player = PlayersChars(playerToPlay);  
 winner = winnerMessege(player, playerToPlay);  
 }  
 }  
 } else {  
 field.drawEmptyField();  
 while (winner == false) {  
 if (turn == false) {  
 playerToPlay = humanPlayer.playersTurn(*FIRSTPLALYER*);  
 turn = turnInGame(playerToPlay, turn);  
 player = PlayersChars(playerToPlay);  
 winner = winnerMessege(player, playerToPlay);  
 } else {  
 playerToPlay = aiPlayer.turn();  
 turn = turnInGame(playerToPlay, turn);  
 player = PlayersChars(playerToPlay);  
 winner = winnerMessege(player, playerToPlay);  
 }  
 }  
 }  
}

}

Главный метод моей игры

избавился от ненужного корипаста, оставил только 4 переменные от которых зависит гэймплэй

Заэкстактить ещё 4 строчки неполучаеться птому что виннер мне нужен в этом методе, в следствии чего так же нужны playerToPlay, player !

Метод выбора игры.

selectGameMode();

private int selectGameMode() {  
 int selectedGameMode;  
 selectedGameMode = humanPlayer.gameSelection();  
 return selectedGameMode;  
}

public int gameSelection() {  
 String input;  
 gameSelectionMesseges();  
 input = inputCheck('1', '2');  
 chosen = (int) input.charAt(0) - 48;  
 return chosen;  
}

Игрок вводит 1 или 2 для игры с игроком или с компом.

Метод ввода числа колонки

public int playersTurn(int firstOrSecondPlayer) {  
 String input;  
 if (firstOrSecondPlayer == 1) {  
 outputMesseges1stPlayer();  
 input = inputCheck('1', '7');  
 chosen = (int) input.charAt(0) - 48;  
 }  
 if (firstOrSecondPlayer == 2){  
 outputMesseges2ndPlayer();  
 input = inputCheck('1', '7');  
 chosen = (int) input.charAt(0) - 48;  
 }  
  
 return firstOrSecondPlayer;  
}

Ходит 1 или 2 игрок в зависимости от игрока, меняется **outputMesseges1stPlayer();**

**Возвращает** int 1 или 2, в зависимости от игрока

Метод хода

public boolean turnInGame(int playerToPlay, boolean turn) {  
 boolean fullToGame;  
 int columnToPutt;  
 char player;  
 columnToPutt = PlayerChose(playerToPlay);  
 fullToGame = field.columnFullChecking(columnToPutt - 1);  
 player = PlayersChars(playerToPlay);  
 turn = puttingChipToField(fullToGame, columnToPutt, playerToPlay, player, turn);  
  
 return turn;  
}

**В метод передается номер игрока и очередь хода**

**Метод проверки колонны**

public boolean columnFullChecking(int columnToPutt) {  
 int toBeFull = 0;  
   
 if (columnToPutt > *COLUMNS*) {  
 System.*out*.println("This column doesn't exist");  
 }  
 for (int rows = 0; rows < *ROWS*; rows++) {  
 if (columnToPutt < *ROWS*) {  
 if (field[rows][columnToPutt] != *EMPTY*) {  
 toBeFull++;  
 setWhereToPutChip(toBeFull);  
 fullToGame = fullToChecking(toBeFull);  
 }  
 }  
 setWhereToPutChip(toBeFull);  
 }  
 return fullToGame;  
}

**Метод переключения между humanPlayer и aiPlayer**

private int playerChooseTheNumber(int playerToPlay) {  
 int playerChosenNumber;  
  
 if ((playerToPlay == *FIRSTPLALYER*) || (playerToPlay == *SECONDPLAYER*)) {  
 playerChosenNumber = humanPlayer.getChosen();  
 } else {  
 playerChosenNumber = aiPlayer.getChosen();  
  
 }  
 return playerChosenNumber;  
}

**возвращает выбранное игроками число**

**Метод положить чип в колонну в выбранную**

public void puttingTheChip(char player, int columnToPutt) {  
 fullToGame = false;  
  
 if (fullToGame == false) {  
 if (whereToPutChip != *ROWS*) {  
 field[(*ROWS* - 1) - (whereToPutChip)][columnToPutt] = player;  
 //field[4][0] = player;  
 fieldPrint();  
 }  
 } else {  
 System.*out*.println("The column " + (columnToPutt + 1) + " is full ");  
 System.*out*.println("chose another column!!!");  
 }

**Метод выбора символа игрока**

private char PlayersChars(int player1or2) {  
 char playersChar = 0;  
 if (player1or2 == *FIRSTPLALYER*) {  
 playersChar = field.*PLAYER1*;  
 }  
 if ((player1or2 == *SECONDPLAYER*) || (player1or2 == *AI*)) {  
 playersChar = field.*PLAYER2*;  
 }  
 return playersChar;  
}

**возвращает выбранный символ**

**Метод проверки и определения победителя**

public boolean winnerMessege(char player, int playerToPlay) {  
 boolean winner;  
  
 winner = field.winnerCheck(player);  
 if (winner == true) {  
 if (playerToPlay == *FIRSTPLALYER*) {  
 System.*out*.println(" 1st player is game winner !!!!");  
 }  
 if (playerToPlay == *SECONDPLAYER*) {  
 System.*out*.println(" 2nd player is game winner !!!!");  
 }  
 }  
 return winner;  
}

**Ещё есть немного локальных методов**

**если нужно, могу их описать**