1 Implementation

This report explains our implementation of the project. Overall the projects uses the classes seen in Figure 1

GameLogic9: IGameLogic playerId: int opponentId: int gameBoard: GameBoard9 - initializeGame(columns: int, rows: int, playerId: int): void - gameFinished(): Winner - insertCoin(column: int, playerId: int): void - decideNextMove(): int - maxValue(state: GameBoard9, depth: int, alpha: int, beta: int): Utility9 - minValue(state: GameBoard9, depth: int, alpha: int, beta: int): Utility9

```
utility : int isTerminal : boolean

max(a : Utility9, b : Utility9) : Utility9
min(a : Utility9, b : Utility9) : Utility9
```

```
GameBoard9
state : int[][]
playerId: int
opponentId: int
columns: int
rows:int
- heuristic(): int
- result(action : int, playerId : int) : GameBoard9
- gameFinished() : Winner
- playerWonOn(column : int, row : int) : int
- utility(): Utillity9
- heuristicForPlayer(player: int, opponent: int, i: int, j: int): int
- awailableRowInColumn(column : int)
- availableActions() : ArrayList<Integer>
- getCopyState(): int[[[
- equals() : boolean
- hashCode(): int
```

Figure 1: UML class-diagram showing off classes implemented by group

1.1 Evaluation and cut-off function

1.2 α - β pruning