**DOMINEERING**

class Game:

U fajlu **UserInterface.py** posedujemo klasu Game koja služi za iscrtavanje table i domina u GUI okruženju pomoću pygame ekstenzije

def show\_bg(self, surface, field):

Ova funkcija iscrtava pozadinu, tj šahovsku tablu na kojoj se postavljaju domine, kao i same domine

def HoverPlayerOne(self, rows:int, cols:int, field, surface):

def HoverPlayerTwo(self, rows:int, cols:int, field, surface):

Ove funkcije omogućavaju igraču da vidi kako bi izgledao njegov potez kada se odigra

U fajlu **GameEngine.py** imamo funkcije iza koje stoji logika igre i poteza

def CreateMatrix(rows: int, cols: int):

Kreiranje matrice poteza

def IsMoveValidOne(rows : int, cols : int, Mat):

def IsMoveValidTwo(rows : int, cols : int, Mat):

Provera da li su potezi oba igrača validni

def CalcAvalaibleMovesPlayerOne(Mat):

def CalcAvalaibleMovesPlayerTwo(Mat):

Računanje mogućih preostalih poteza za oba igrača

def PlayerOneMove(rows : int, cols : int, Mat):

def PlayerTwoMove(rows : int, cols : int, Mat):

Omogučava igru oba igrača i proverava koji je pobednik i da li je potez valjan i izbacuje odgovarajuće poruke

def PrintField(Field):

Štampanje celog polja u konzoli radi lakšeg pregleda (nije neophodno za GUI)

U fajlu **AI.py** imamo funkciju koja će služiti za pokretanje igre kada igra kompjuter u 2. fazi projekta

def getNextMove(matrix, isMoveValid):

U fajlu **Main.py** imamo klasu Main koja služi za pokretanje cele igre

class Main:

def \_\_init\_\_ (self):

Ova funkcija omogućava postavljanje neophodnih vrednosti za učitavanje table i igrača i iscrtavanje pygame prozora

def resetGame(Field):

Funkcija koja služi za ponovno pokretanje igre klikom na taster R

def mainLoop(self):

Glavna petlja koja crta tablu i igraču, i pamti prethodna stanja kao i koji igrač igra. Na klik miša omogućava postavljanje pločice ili resetovanje igre na pritisak tastera R

**Tim PathFinders**

Nenad Đorđević, 16080

Ivan Bogosavljević, 17561

Nikola Rašić, 17907