Projekat iz IS 2017

Backmmon igrica

Petar Trifunovic, 2014/72

# 

# Kratak prikaz implementiranog algoritma

+ (NSMutableDictionary \*)getMoveForPlayer:(NSInteger)player withBoard:(NSMutableArray \*)board dices:(NSMutableArray \*)dices

first:(NSInteger)first depth:(NSInteger)depth alpha:(CGFloat)alpha beta:(CGFloat)beta playAgain:(BOOL)playAgain;

{

if(depth == 0) {

return [NSMutableDictionary dictionaryWithDictionary: @{@"value" : @([self getValueForPlayer:first withBoard:board]),

@"type" : @"NODE"}];

}

NSMutableArray \*moves = [self getPosibleMovesForPlayer:player withBoard:board dices:dices first:first];

if([moves count] == 0) {

CGFloat value = 0;

for(NSUInteger i = 1; i <= 6; i++) {

@autoreleasepool {

NSMutableArray \*dice = [NSMutableArray arrayWithArray:@[@(i), @(j)]];

NSMutableDictionary \*dic= [self getMoveForPlayer:playAgain ? player : !player withBoard:board dices:dice first:first depth:depth-1 alpha:alpha beta:beta playAgain:NO];

value += [dic[@"value"] floatValue];

}

}

}

return [NSMutableDictionary dictionaryWithDictionary: @{@"value" : @(value/36),

@"type" : @"EXP"}];

}

if([ dices count] == 2) {

playAgain = [dices[0] isEqual:dices[1]];

}

if(depth >= 5){

//NSLog(@"%@", moves);

}

CGFloat minmax = player == first ? -1000000 : 1000000;

NSMutableDictionary \*minmaxMove;

for(NSMutableDictionary \*move in moves) {

CGFloat value;

@autoreleasepool {

NSMutableArray \*dice = [dices mutableCopy];

NSInteger index = [dice indexOfObject:move[@"dice"]];

[dice removeObjectAtIndex:index];

NSMutableDictionary \*dic= [self getMoveForPlayer:player withBoard:[self applayMove:move toBoard:board] dices:dice

first:first depth:depth-1 alpha:alpha beta:beta playAgain:playAgain];

value = [dic[@"value"] doubleValue];

}

if((player == first && minmax < value) ||

(player != first && minmax > value)) {

minmaxMove = move;

}

if(player == first) {

minmax = MAX(value, minmax);

alpha = MAX(value, alpha);

} else {

minmax = MIN(value, minmax);

beta = MIN(value, alpha);

}

if(beta <= alpha) {

break;

}

}

minmaxMove[@"value"] = @(minmax);

minmaxMove[@"type"] = first == player ? @"MAX" : @"MIN";

return minmaxMove;

}

Kratak opis realizovanih klasa sa opirima metoda

1) ViewController - Prikazuje tablu za backgamon

@interface ViewController : UIViewController

@property NSInteger gameLength;

@property NSInteger redBotDif;

@property NSInteger whiteBotDif;

@end

2) OptionViewController - Pocetna strana sa opcijama

@interface OptionsViewController : UIViewController

@end

3) BackgamoNavigationController - Koristi se za navigaciju izmedju ViewController-a

@interface BackgamonNavigationController : MaxelerNavigationController

@end

4) TriangleView - Trougaoni View

@interface TriangleView : UIView

@property TraingleSelected \*backgroundView;

@property TraingleSelected \*selectedView;

@property BOOL showBackgroundView;

@property UIColor \*bottomColor;

@property NSMutableArray<CircleView \*>\* circles;

// Prikazuje pozadinu

- (void)showBackgrooundView:(BOOL)show animated:(BOOL)animated;

@end

5) TriangleSelected - Pozadina za Trougaoni View

@interface TraingleSelected : UIView

@property CGFloat lineYOffset;

@property CGFloat lineSpacing;

@property CGFloat lineWidth;

@property UIColor \*lineColor;

@property CGFloat borderWidth;

@property CGFloat borderRadius;

@property UIColor \*borderColor;

@end

6) CircleSelected - Pozadina za kruzni View

@interface CircleSelecteed : UIView

@property UIColor \*outerColor;

@property UIColor \*innerColor;

@end

7) CirlceView - Kruzni View

@class CircleView;

@protocol CircleMoveDelegate <NSObject>

// Poziva se kad se pritisne na krug

- (void)circleDidStartMoving:(CircleView \*)circle;

//Poziva se kad se krug pomera

- (void)circleMoved:(CircleView \*)circle;

//Poziva se kad se zavrsi pomeranje kruga

- (void)circleDidEndMoving:(CircleView \*)circle;

@end

@interface CircleView : UIView

@property CircleSelecteed \*backgroundView;

@property BOOL showBackgroundView;

@property TriangleView \*parentView;

@property id<CircleMoveDelegate> delegate;

// Prikazuje pozadinu

- (void)showBackgrooundView:(BOOL)show animated:(BOOL)animated;

// Napravi krug unutar trougla

+ (CircleView \*)createCircleWithPerent:(TriangleView \*)parentView;

// Napravi crveni krug

+ (CircleView \*)createRedCircleWithPerent:(TriangleView \*)parentView;

// Napravi beli krug

+ (CircleView \*)createWhiteCircleWithPerent:(TriangleView \*)parentView;

- (BOOL)isRed;

- (BOOL)isWhite;

@end

8) Board - Implementacija same logike ige

@protocol BoardWon <NSObject>

// Player je pobedio i dobio won bodova

-(void)player:(NSUInteger)player won:(NSUInteger)won;

@end

@interface Board : UIView <CircleMoveDelegate>

@property NSMutableArray<TriangleView\*> \*board;

@property BOOL useRedBot;

@property BOOL useWhiteBot;

@property NSInteger redBotDeapth;

@property NSInteger whiteBotDeapth;

@property NSMutableArray \*diceNumbers;

@property NSMutableArray \*eaten;

@property NSMutableArray<NSMutableArray<UIView \*> \*> \*finished;

@property NSInteger player;

@property id<BoardWon> delegate;

// Kockci se pola na zadate brojeve

- (void)dicesDidLandOnNumbers:(NSMutableArray \*)numbers;

// Postavi zetone na tablu

- (void)layoutCirclesAnimated:(BOOL)animated;

@end

9) DieView - View koji se koristi za prikaz kockica

@interface DieView : UIView

#pragma mark Properties

@property (nonatomic, strong) UIImageView \*dieImage;

@property UIView \*dimView;

#pragma mark Methods

// Prikazi odgovarajucu sliku

- (void)showDieNumber:(int)num;

@end

10) Bot - Funkcije koje bot koristi

@interface Bot : NSObject

// Nadji najbolji potez za zadaqtu tablu, kockice, i ko je na potezu

+ (NSMutableDictionary \*)getMoveForPlayer:(NSInteger)player withBoard:(NSMutableArray \*)board dices:(NSMutableArray \*)dices

first:(NSInteger)first depth:(NSInteger)depth alpha:(CGFloat)alpha beta:(CGFloat)beta playAgain:(BOOL)playAgain;

// Heuristicka funkcija za procenu stanja table iz pogleda igraca

+ (CGFloat)getValueForPlayer:(NSInteger)player withBoard:(NSMutableArray \*)board;

//Funkcija koja nalazi sve moguce korake za igraca na tabli sa kockicama

+ (NSMutableArray \*)getPosibleMovesForPlayer:(NSInteger)player withBoard:(NSMutableArray \*)board dices:(NSMutableArray \*)dices first:(NSInteger)first;

// Funkcija koja uz pomocu poteza prelazi u sledece stanje

+ (NSMutableArray \*)applayMove:(NSMutableDictionary \*)move toBoard:(NSMutableArray \*) board;

@end

11) GraphViewController - Koristi se za prikaz stabla odlucivanja

@interface GraphViewController : UIViewController

@property NSMutableDictionary \*root;

@property NSMutableArray \*children;

@end