Player • - name : String • - game_started : boolean ANIKET GOEL • - colour : String • - current : int • - photo : ImageView 2020281 + setCurrent(curr) : void + setGame_started(curr) : Game ladder void GameController + isGame_started() : boolean • + p1_chance : boolean + getCurrent(): int • - start : int • - s : Stage • + p1_wins : boolean + setPhoto(photo) : void + setColour(colour) : void • - end : int • - p : Parent • + p2_wins : boolean • - sc : Scene • + snakes : + ladder(s,e) {} + getEnd() : int + getStart() : int + setEnd(end) : void • - back_exit : ImageView ArrayList<Snake> • + ladders : - dice_value : Label- player1_faded : ImageView ArrayList<ladder> <<interface>> • - player2 : ImageView Initializable • + players : <<extends>> - dice_val_2 : ImageView- exit_ok : ImageView ArrayList<Player> + setStart(start) : void • + path1 : Path • - exit_menu : ImageView • + path2 : Path • - dice_1 : ImageView 1..* • + paths : • - dice_2 : ImageView ArrayList<Path> • - dice_3 : ImageView • + first_chance_p1 : • - dice_4 : ImageView • - dice_5 : ImageView • - dice_6 : ImageView • + first_chance_p2 : int • - blue : ImageView • - green : ImageView Getters and Setters • - button : Button for all the components • - back_button : ImageView Board • - arrow : ImageView • - board : ImageView • - player1 : ImageView • - snakes : Arraylist<Snake> <<extends>> • - menu_1 : ImageView - ladders : Arraylist<ladder>- tiles : ArrayList<tile> <<implements > - p2_win : ImageView | p1_win : ImageView | p1_win : ImageView | p2_win : ImageView | p3_win : ImageView | p4_win : Imag replay_1 : ImageViewplayer2_faded : ImageView + getLadders() : ArrayList<ladder> <<extends>> getSnakes() : ArrayList<Snake> • + translateIT(img , x , y) : void • + open_green_token() : void + getTiles() : ArrayList<tile> + setLadders(ladders) : void + open_blue_token(): void + move_Token_upwards(img, path): void + move_Token_backwards(img, x, path): void + setSnakes(snakes) : void <<extends>> + setTiles(tiles) : void + move_Token_backwards(img, x, path) : void + move_Token_forward(img, x, path) : void + ladder_i(img , path) : void + snake_i(img , path) : void + backToMenu(mouseEvent) : void + fadeIT(img) : FadeTransition + fadeOut(img): FadeTransition • + fadeIn(img): FadeTransition Snake • + exitOK(mouseEvent) : void • + exitBack(mouseEvent) : void • + setBoard(snakes , ladders) : void • - head : int • + Chance(p1 , p2 , snakes, ladders , value) • - tail : int • - number : int • + initialize(url , resourceBundle) : void <<extends>> • + show_menu(mouseEvent) : void • + clickDice(mouseEvent) : void F Snake(num) {} + getNumber() : int • + replay(mouseEvent) : void Main_page_animations getHead(): int getTail(): int <<throws>> ava.lang.Exception - setHead(head) : void + fadeIT(img) : + setTail(tail) : void FadeTransition setNumber(number) : void Dice <<implements>> Controller • - s : Stage • - p : Parent tile + roll() : int • - sc : Scene <<extends>> • - heading : ImageView • - profilePic : • - isSnake : boolean ImageView • - isLadder : boolean • - watchvideo : **IOException** ImageView - mainButton: ImageView <<throws>> + setLadder(ladder) : void • - img1 : ImageView + setSnake(snake) : void • - img2 : ImageView + getisLadder() : boolean + getisSnake() : boolean ClassNotFoundException Main + initialize(ur; , resourceBundle) : void + attribute1:type = defaultValue switchToGame(mouseEvent) + attribute2:type Application : void attribute3:type <<extends>> + start(primaryStage) **4**-----Exception : void + main(String[] args) : void <<throws>>