UML Diagram of Will Hero- Grp_126 Shelja, Vishnu GameState **Application** -score: int **SavedGames** -obstacle: Obstacle -GameID: int -SavedGames: -Arraylist<Arraylist<Object>> +GameState(): void Will Hero +displaySavedGames(): void #displayState(): void +onMouseClick(event:MouseEvent): void +exitGame(): void +addAGame(game:GameState):void Settings +displaySettings():Void -world +start(primaryStage: Stage): void -Music +stop(): void +setWorld() :Void +resume(): void +setMusic(): Void +main(args: Strings[]): void +CustomCostume(): Void +getlocation(): Location kill Game <<interface>> Helmet -noOfObstacles: int +pigHelmet() +throwingAxe(): void +addWeapon(): +Game() +getWeapons() +hammer(): void +Game(gamestate: GameState) +upgrade() +coins(): +loadObstacle(): void Controller +loadUser(): void Elimination +Start() +HitObstacle(): boolean GameOver +Pause(): void +onMouseClick(event: MouseEvent): void +continue(): void +StartNewGame(): void -expression: label +Orc(): void axe +location(): void Hammer +ResumeGame(savedgames: -resurrect: boolean +Obstacle():void -noOfAxe :int SavedGames): void -noOfHammer: int +InBetweenLand(): void +displayHighScores(highscore: +addWeapon(): +addWeapon(): #GameOver() +getWeapons():Axe HighScore): void +getWeapons():Hammer -displayMessage(): void +exitApp(): void -Resurrect(): void +displayOptions(): void +savegame(): void <<interface>> obstacle Boss chest #noOfCoins: Int +move(): void #nOfweapons: ArrayList<<Helmet>> +move(): void +stopMoving(): void +updateCoins: Void +stopMoving(): void +getOrientation(): Object +updateWeapons(Helmet hel): Void +getOrientation(): Object +boss() +move(): void HighScore

Orcs

+move(): void

+stopMoving(): void

+getOrientation(): Object

+greenOrc()

+move(): void

Windmill

+move(): void

+stopMoving(): void

+getOrientation(): Object

+windMill()

+round(): void

-Score: int

-List: Arraylist<Arraylist<Object>>

#getScores(): Arraylist<Arraylist<Object>>

#addScore(Score:int, Position: position, Obstacle: obstacle)

#displayScores(): void

+deleteScore(): void

