

App
+ WIDTH: int = 864 + HEIGHT : int = 640 + SMOOTHING AVG : int = 32 + INITIAL PARACHUTES : int = 3 + FPS : int = 60 - configPath : String + playerColoursConfig : JSONObject - fuelCanImg : PImage + parachuteImg : PImage - bigboilimg : PImage - windRightImg : PImage - windLeftImg : PImage + players : Map <Character, Player> - arrowTimer : int - finalScoreBoardDelayTimer : int - levelDelaytimer : int - playerIndex : int - isGameOver : boolean - isLevelOver : boolean - rand : Random + currentLevel : int + currentPlayingLevel : Level + levels : ArrayList<Level> + playerIterator : Iterator<Character> + drawPlayerIterator : Iterator<Character> + playerListIterator : ListIterator<Character> + playerList : List<Character> + currentPlayer : Character + currentTank : Tank
+ settings () : void + setup () : void + keyPressed (KeyEvent event) : void + draw () : void - componentGUI () : void - windGUI () : void - healthBarGUI () : void - playerArrow () : void - inGameScoreboard () : void - finalScoreboard () : void - changePlayerTurn () : void - changeLevel () : void - restartGame () : void + setRGBValues (String input) : int [] + main (String [] args) : void

Level
- rand : Random - wind : int - background : PImage - treeSprite : PImage - layoutInput : String - height : int [] - foregroundRGBValues : int [] - screenLayout : Character [][] - layout : Character [][] - levelObjects : List<LevelObject> - trees : ArrayList<Tree> - projectiles : ArrayList<Projectile> - playerTanks : Map<Character, Tank>
+ setLayout (String layoutInput) : void + setProjectiles (Projectile projectile) : void + getProjectiles () : ArrayList<Projectile> + setWind (Integer wind) : void + getWind () : int + setBackground (PImage background) : void + setForegroundColour (String foregroundColour) : void + set TreeSprite (PImage trees) : void + get TreeSprite () : PImage + getHeight () : int [] + setHeight(int [] inputHeight) : void + getPlayerTurn () : TreeSet<Character> + getPlayerTanks () : Map<Character, Tank> - createLevel () : void + restartLevel () : void - calculateMovingAverage (int [] data) : int [] + draw (App app) : void

Player
+ playerChar : Character + rgbColors : int [] - parachute : int - score : int - bigProjectile : boolean
+ getParachute () : int + setParachute (int parachute) : void + getScore () : int + setScore () : void + setBigProjectileActive () : void + setBigProjectileInactive () : void + isBigProjectile () : boolean + resetPlayer () : void

