Documentation

Functionality and Usage:

This program is a game where you defend the earth from incoming asteroids. It maintains a database of high scores that is displayed at the end of the program and updated if the player achieves a new high score.

Upon starting the program you will be greeted by a title screen that lists instructions on how to play. Once you press the Start button the game will begin.

For the duration of the game asteroids will spawn around the edges of the screen and move towards the planet, your goal is to destroy them before they make contact.
You can move the player character around the central planet using the left and right arrow keys. Pressing spacebar will fire a missile from the player character directly away from the planet. This missile will destroy an asteroid if it hits it.

Every asteroid you destroy will increase your score and score multiplier, also after destroying enough asteroids will move you to the next stage which increases the spawn rate and speed of asteroids and increase your number of lives. If an asteroid hits the planet or player your lives will decrease and your score multiplier will be reset.

Upon hitting 0 lives the game will end and your score will be checked against the saved high scores. If your score is higher or there are not enough saved scores then you will be prompted to enter a name (between 2 and 15 characters) and once you have entered your name (or if your score was not high enough) you will be taken to a screen displaying all high scores. You can press the exit button on this screen to exit the program.

Method Descriptions:

Main Methods

main(String[] args)
- Launches the application

start(Stage primaryStage)
- Sets up properties of primary stage, then calls method to load start screen

GUI Methods

loadStart(Scene theScene, StackPane root)
- Loads the start screen with title, instructions, and button to start game

loadStatusDisplay(StackPane root)
- Loads display bar for score, combo, stage, and lives at the top of the screen that is updated automatically

loadGame(Scene theScene, StackPaneroot)
- Loads and runs the game; contains main gameplay loop; calls functions to load status display and listener for game end

updateMissles(GraphicsContext gc, Iterator<Missile> missileIter)
- Updates the positions of all missiles and draws them

updateAsteroids(GraphicsContext gc, Iterator<Asteroid>asteroidIter)
- Updates the positions of all asteroids and draws them

checkCollision(Iterator<Missile> missleIterr,Iterator<AsteroidAsteroidIter
- Checks for collisions between every missile and asteroid and destroys both and updates game state when a collision is detected

loadNameEntry(StackPane root)
- Loads screen for name entry when user gets a new high score

loadHighScores(StackPane root)
- Loads screen to display all high scores to the user, also has button to exit program

getScoreLabels(String scorePos,String name,String score, Boolean)
- Creates labels for each high score to be displayed, new high scores use red font

addGameListener(StackPane root)
- Creates a change listener that detects when the game has ended and removes game elements from the screen, then loads either name entry or high scores screen depending on whether the user achieved a new high score

addNameListener(StackPane root)
- Creates a change listener that detects when the user has entered a valid name and removes name entry screen, then loads high scores screen

HighScores Methods

checkDataBase()
- Checks if database and high scores table exist and creates them if they do not

addHighScore(String name, int score, long timestamp)
- Adds a score to the database

removeHighScore()
- Removes the lowest/most recent high score from the database

enoughHighScores()

- Checks whether there are 10 high scores in the database

higherScore(int score)
- Checks a given score is higher than any in the database

getHighScoreList()
- Returns a 2D arraylist containing all the scores in the database

Missile Methods
update()
- used to change the missile's position based on its current velocity.

render(GraphicsContext gc)
- used to draw the missile on screen

getBoundary()
- used to define a rectangular boundary at the position of the missile, used for collision detection

Player Methods
update()
- used to change the player's position based on its position counter

render(GraphicsContext gc)
- used to draw the player on screen

getBoundary()
- used to define a rectangular boundary at the position of the player, used for collision detection

incPosCounter()
- Increments the position counter to move the player clockwise

decPosCounter()
- Decrements the position counter to move the player counter-clockwise

Asteroid Methods

update()
- used to change the asteroid's position based on its current velocity.

render(GraphicsContext gc)

- used to draw the asteroid on screen

getBoundary()
- used to define a rectangular boundary at the position of the asteroid, used for collision detection

getRealVel(double vel)
- used to calculate either X or Y velocity component, depending on spawn point of the asteroid

getCompVel(double vel)
- uses the velocity component calculated by getRealVel to calculate the complementary velocity component

intersects(Player player)
intersects(Missile missile)
intersects(Planet planet)
- used to detect collisions between an Asteroid and a Player/Missile/Planet respectively

Planet Methods

render(GraphicsContext gc)
- used to draw the planet on screen

getBoundary()
- used to define a rectangular boundary at the position of the planet, used for collision detection

GameState Methods

asteroidDestroyed()
- used when an asteroid is destroyed to increase score, combo, and a counter for number of asteroids destroyed in the stage. Calls upstage method once enough asteroids destroyed in stage

asteroidHit()
- Used when an asteroid hits the player or planet to decrease lives and reset combo as well as to set game end flag if lives hits 0

upStage()
- Used when enough asteroids are destroyed to increase stage, lives, and score