Readme.pdf

Fritz Keyzer – 19856202

Stellar Crunch

1. Main class to be compiled: “StellarCrunch.java”
2. I have no interface inheritance
3. PlayerObject class extends GameObject class. This is class inheritance. This is an example of dynamic polymorphism.
4.
    - Added a shooting mechanic that shoots a part of your player mass, while conserving momentum.
    - Any two asteroids that collide with enough speed will shatter. Shattering conserves momentum, and the collision speed is calculated from the normal of the collision plane.
    - Added a level system, each increasing level spawns more asteroids. You complete each level by being the last asteroid left. You can merge with smaller asteroids, but have to shoot larger asteroids in order to divide and conquer. You can restart at any time to retry the level.
    - Added text above each planet showing its mass and distance from you. The size of the text and position are calculated as if it were a physical sign above the planet, however there is a minimum text size.
    - All asteroids have a border, a red border if it is larger than you and you cannot absorb it, a green ring if it is smaller and you can, and yellow if it is smaller, but it (or you) are moving too fast.
    - Added a death screen, that shows a top down view, that is zoomed out far enough to capture all the asteroids. It also shows where you died.
    - Added highscores, players can enter their names at the start, and highscores are saved in a text file. Highscores are in the form of completion time, there is a highscore per level, storing the fastest completion time, for that particular level.
    - Added a rotating minimap to aid in orientation. The minimap has arrows indicating towards asteroids that are too far away to appear on the minimap.
    - Modified the first person view algorithm to “pretend” all asteroids are sliding along an invisible floor. (as opposed to having all the centres at the same height) This greatly helps make the first person perspective more immersive.
    - Added a Heads up display showing velocity, mass, current level, highscore, current time etc.
    - Added a compass to the top of the screen showing you the direction you are heading in and also a bar above the compass showing the location of asteroids in the distance.
    - Added a very small amount of drag, to make the simulation feel more natural. And compensate partly for inaccuracies.
    - Asteroids spawn randomly within a belt. The belt has a small rotational velocity.
    - Asteroids are randomly given colours. When asteroids merge, the colours mix.
    - The colour of an asteroid fades to black as it moves away, assisting with immersion.
5. I didn’t use the provided vector class. I preferred to have more control over individual components. I could extend this project to all 3 dimensions, as opposed to having all the objects on a plane. But I mostly just enjoyed implementing all the math myself.

6. No additional libraries required.
7. I didn't use any of the textbook libraries.
8.