Team Space [Jonathan Quang, Brian Kwong, Datian Zhang]
APCS2 pd1
Asteroid Too

We are going to make a clone of asteroids, but with more features.

Our “level” will be a 1000 long by 800 pixel screen.
When playing, there will be a user interface that gives the player useful information, such as how many points they have, what “level” they are on, their current health, etc.

Class: Moveable

This class contains objects that have a position on the screen, a velocity vector, a max velocity. Objects of this class move via velocity vectors. In addition, there will also be a variable called radius because collisions between moveable objects will be treated as a collisions between circles.

Class Moveable
+ PVector position, - PVector velocity - int radius - float maxVel;
+ void updatePos() + void display()

Subclass: Asteroids

It will be populated with randomly generated asteroids whose collision boundaries will be treated as circles for the sake of simplicity. The player can shoot the asteroids. There will be three sizes of asteroids. When the smallest asteroid is destroyed, it disappears and points are awarded. When large or medium asteroids are destroyed, they points are awarded and the asteroids split into the next smallest size. When it is destroyed, three more asteroids of smaller sizes of random velocity are spawned. Size will be represented by int size. Asteroids will not be destroyed by collision with other asteroids.

Class Asteroid
-int size

Subclass: Ship

Ships will have an acceleration vector in addition to a velocity vector.. The max speed is there to prevent ridiculous speed calculations.

Ships can turn left, right, and move forward and back. They will be controlled via acceleration vectors. Ships also have a variable that controls how much health they have.

Class Ship
- float yaw - PVector accel - float maxVel - float maxAccel - int hp
+ void changeYaw() + void accelViaYaw()

Sub Subclass: Player Ship

This is the ship that the player controls with WASD to turn and accelerate. Spacebar shoots a bullet. Bullet velocity will depend on the orientation of the ship.

The player has an inventory system that contains all the powerups they have picked up in the order they were picked up. Each powerup has a money value based on what it can do, for example, dealing double damage will be worth $250 while a powerup that heals 15 HP would only be worth $100. Powerups are stored in an Arraylist called Inventory.

Upon pressing E, an inventory menu will appear and the game is paused. The inventory is sorted by quicksort upon opening the menu.

Weapons

Player would use DLL to switch between weapons that people have and have a mechanic like a power-up to pick up weapons.

Class PlayerShip
- ArrayList PowerUps - DLList Weapons - DLNode currentWeap + int money

+ void keyPressToAction + void openInventory + void usePowerUp + void useWeapon + void switchWeapon

Sub Subclass: Enemy ship

These are the ships that are controlled by a rudimentary AI that tries to go after the player. As a prototype/placeholder for early versions of the project, it will be just moving randomly. All enemy ships will be red and will be spawned in random places around the border of the screen. Enemy ships will not be damaged by colliding with asteroids. The order of things in which the AI of the enemy ship will attack will be determined by an ArrayPriorityQueue. Each enemy ship will only carry a basic gun..

Class EnemyShip
- ArrayPriorityQueue Targets
+ void moveToTarget + void attackTarget

Market

This is the place where the player can buy powerupss. If the market is destroyed, game over. The user can also purchase placeable walls to block asteroids and enemies, or storages to increase the amount of money the player can hold.
The market always spawns in the middle of the screen.

Class Market
+ int hp + ArrayList shopPowerUps
+ void buy() + void display(). + void place().

Placeable

These objects are either walls or money storages (depending on whether they can store money or not).

Class Placeables
+ int hp + boolean holdMoney? + PVector pos
+ void damage()

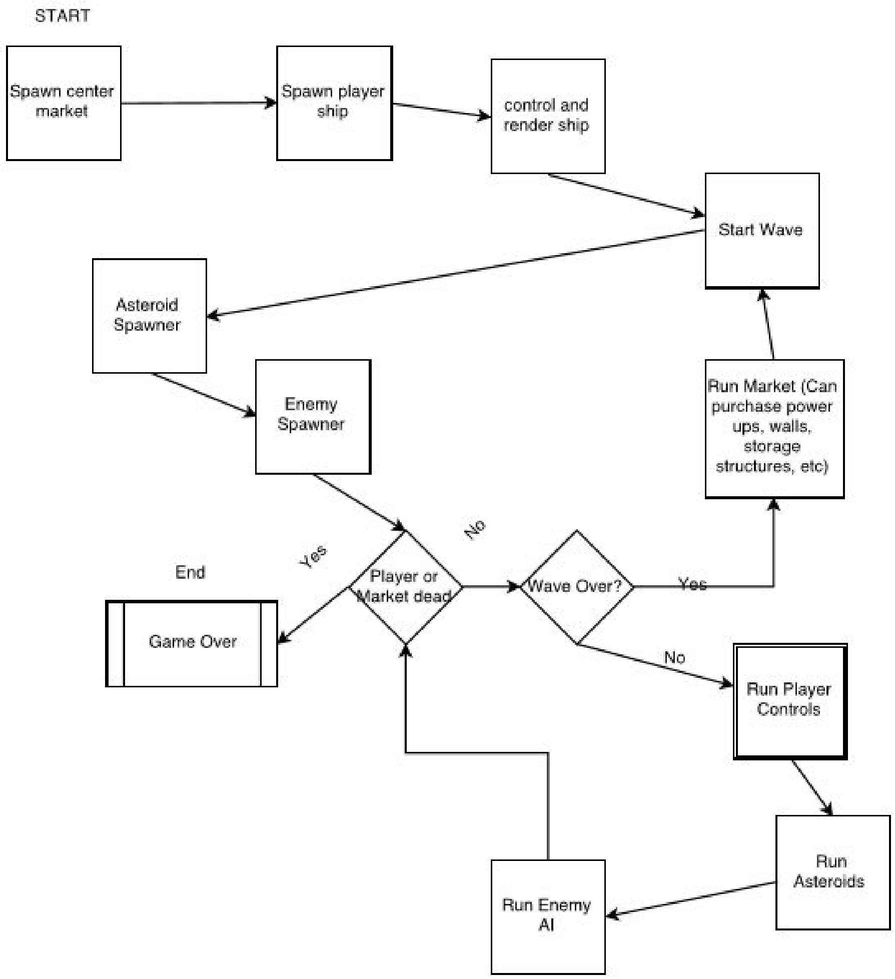
Powerups

These are objects in the level that randomly spawn and can be picked up by the player ship. This includes health packs or temporary damage upgrades.

Class Powerups
- int cost - String effect - float effectStrength
+ void applyEffect(Playership)

Level/Driver:

This file will be running our game. There will be an array containing the the asteroids, an arrayList containing enemies, and an arrayList containing walls/storages that the player has built. Refer to the diagram below:


START
Spawn center market
Spawn player ship
control and render ship
Start Wave
Asteroid Spawner
Enemy Spawner
Run Market (Can purchase power ups, walls, storage structures, etc)
End
Yes
Game Over
Player or Market dead
No
Wave Over?
Yes
No
Run Player Controls
Run Asteroids
Run Enemy AI