

Big Space Game



player 1



331 M/S

Trevor Hedstrom

Program Operation

- Follow the menu to play the game
- Up to 32 player multiplayer
- Right click to thrust
- Left click to shoot
- Middle click to special shoot (big ships only)

Demonstration

Multiplayer:

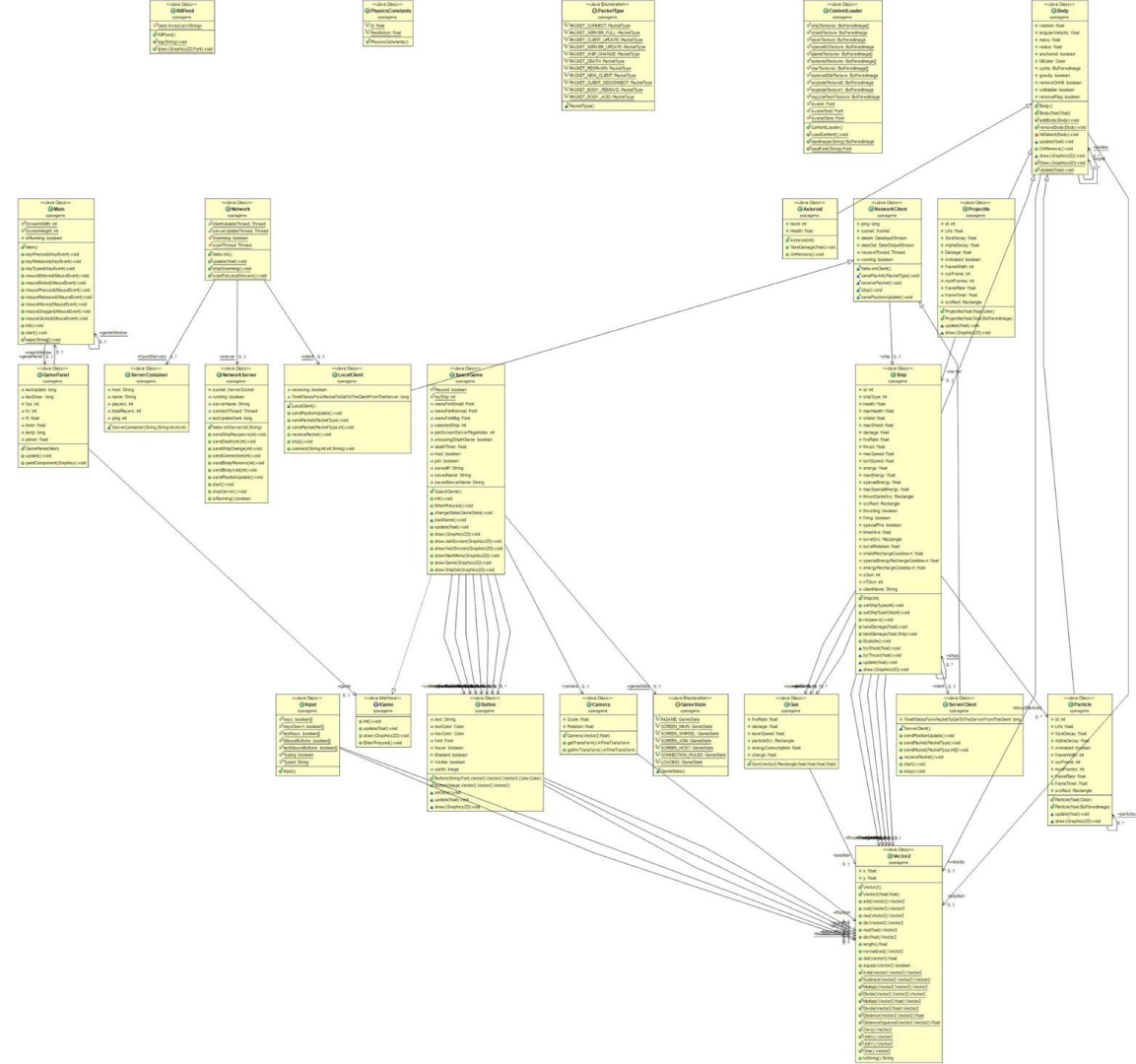


Demonstration

Planets and orbits (single-player only):

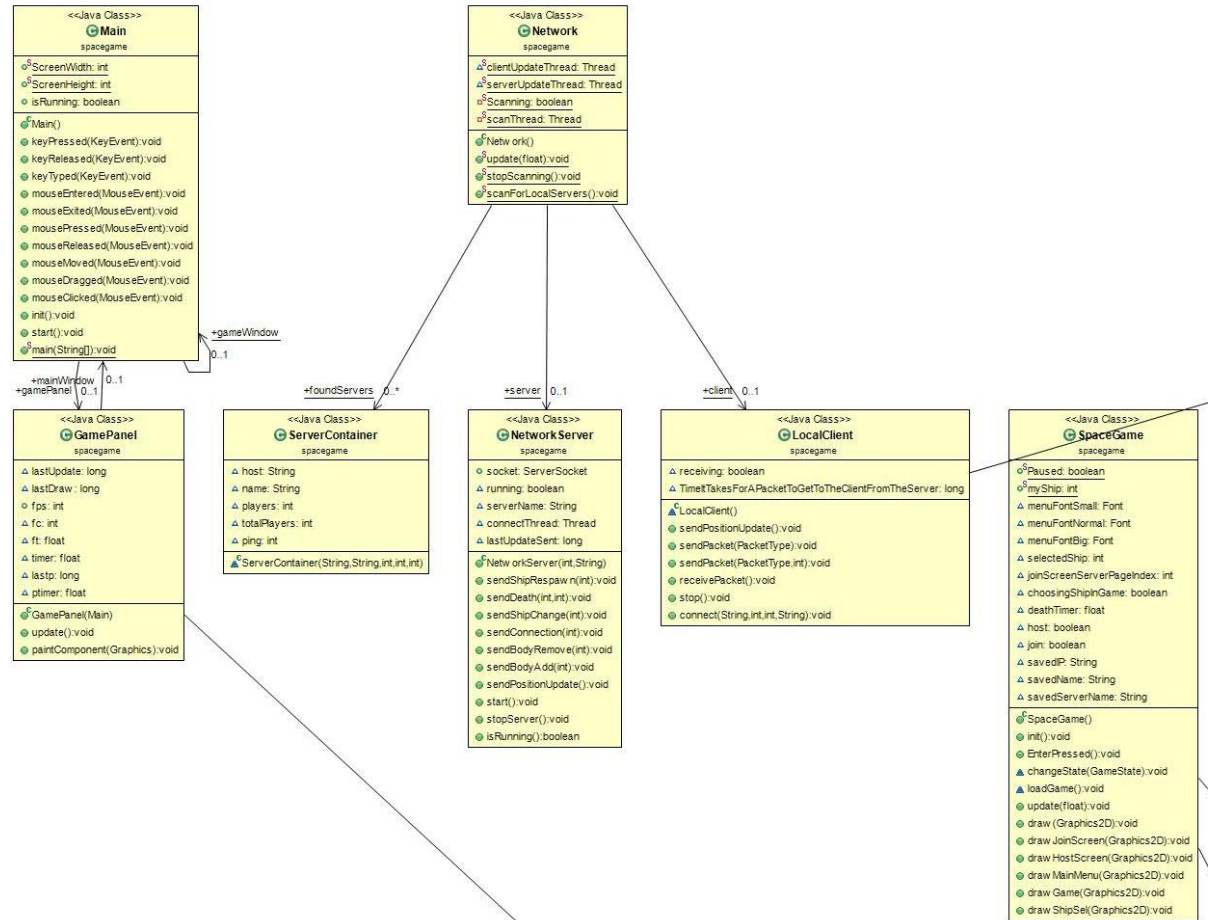


Too big for one slide



UML Diagram

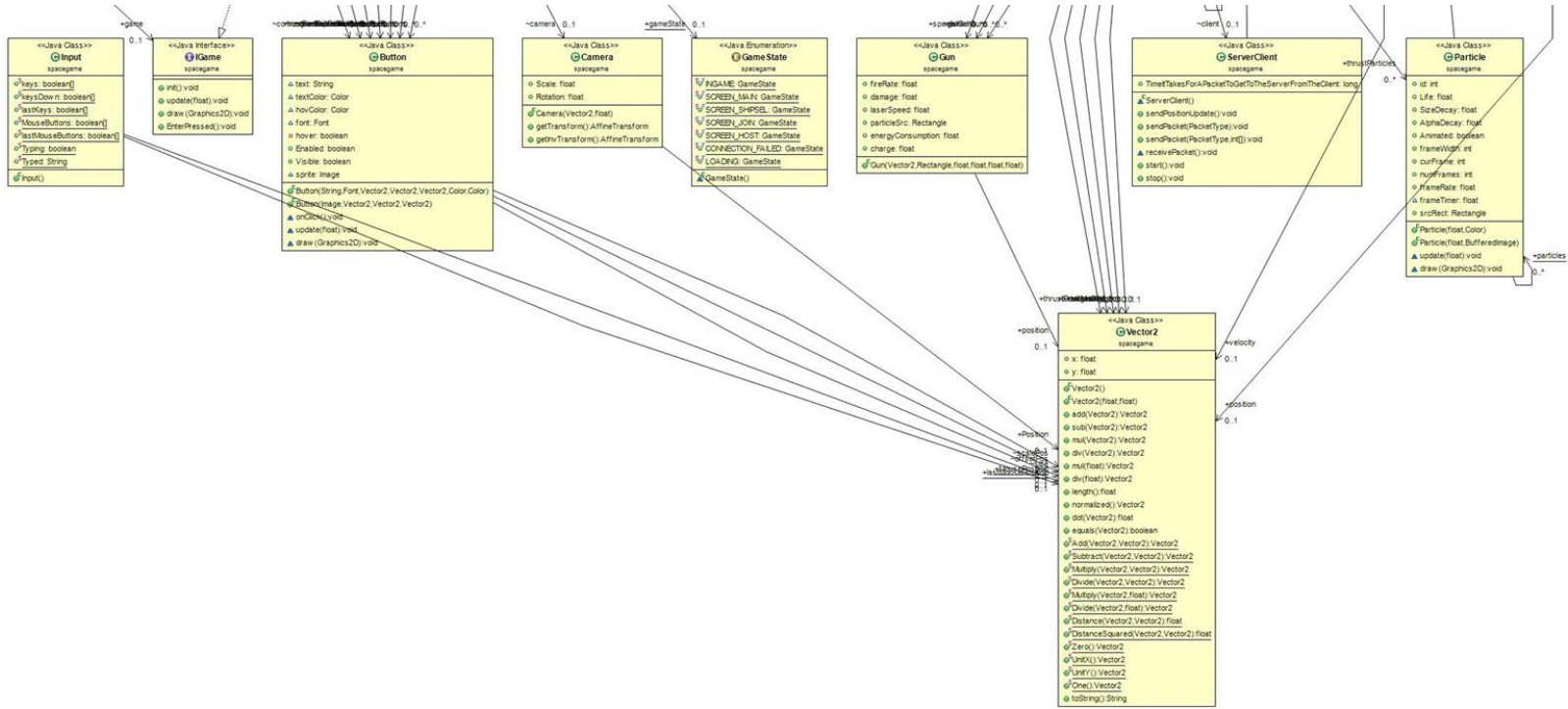
The basics



UML Diagram

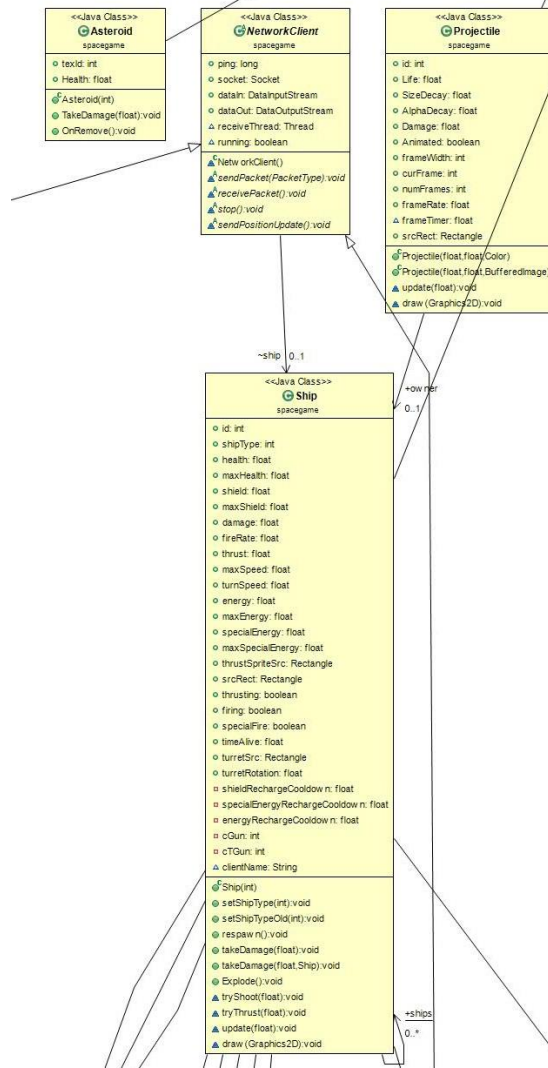
UI and more basics

Also some network stuff



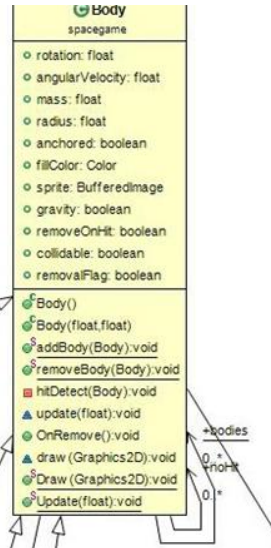
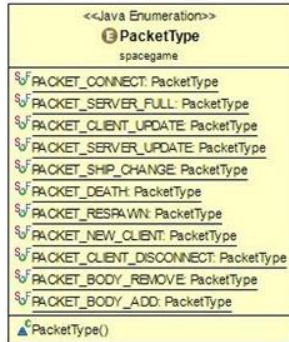
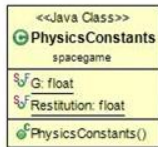
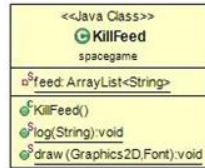
UML Diagram

Network client and projectiles



UML Diagram

Static things (constants, resources) and Bodies



Classes

- Vector2
- Body
- Asteroid extends Body
- Projectile extends Body
- Ship extends Body
- Gun
- Button
- Camera
- ContentLoader
- KillFeed
- Network

Class Interaction

Program starts in Main -> creates a Main class (the main window)

Main takes input and stores it in Input

Main class creates GamePanel (for graphics within the main window)

Main class calls the GamePanel's Draw() and Update() method

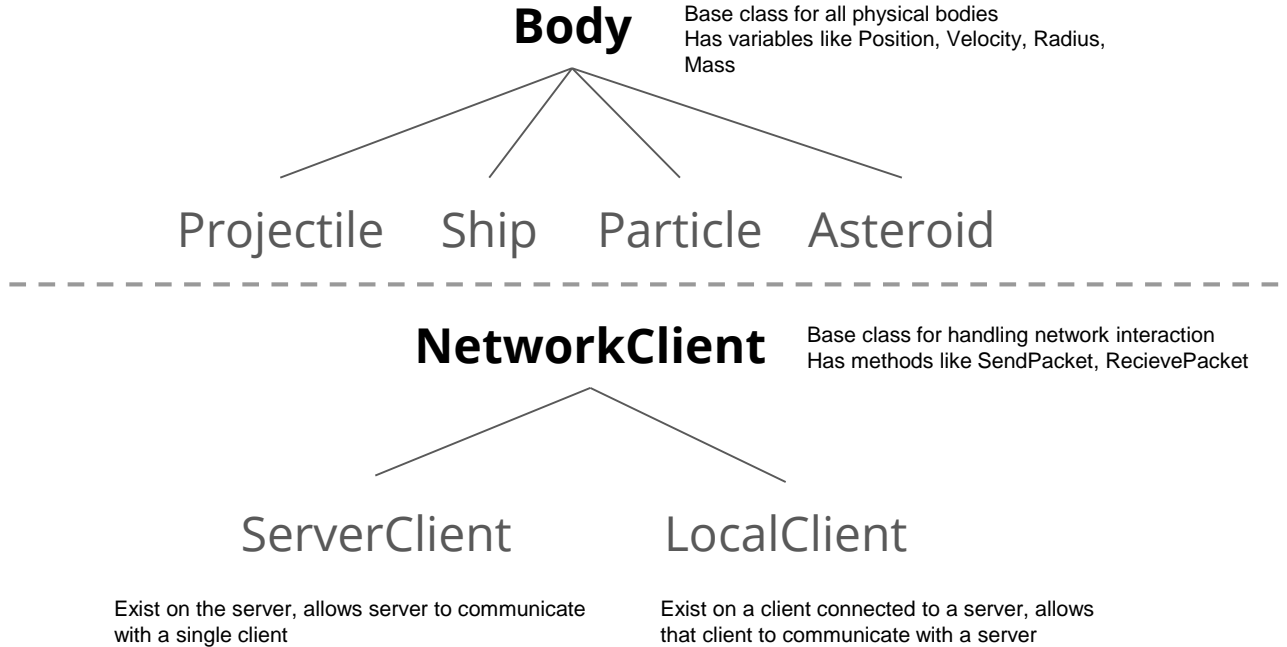
Tries to call Draw() 60 times per second, and Update() in between.

GamePanel creates IGame interface, relays Draw() and Update()

SpaceGame implements IGame, receiving Update and Draw calls

SpaceGame calls Update() on Body, Projectile, Ship, Particle, NetworkServer, LocalClient, and any active Buttons

Inheritance Hierarchy



Polymorphism

`takeDamage()` makes a ship take damage, gets called when a ship hits another body.

The first `takeDamage()` method simply calls the other `takeDamage()` giving it `null` for the other parameter.

```
/**
 * This method is created just to implement polymorphism
 * @param dmg Damage to take
 */
public void takeDamage(float dmg){
    takeDamage(dmg, null);
}

/**
 * Applies damage to Shield, then to Health
 * @param dmg Damage to take
 */
public void takeDamage(float dmg, Ship other){
    if (health > 0){
        if (shield > 0){
            shield -= dmg;
            if (shield < 0){
                health += shield;
                shield = 0;
            }
        }else{
            health -= dmg;
        }

        shieldRechargeCooldown = 2;

        if (health <= 0){
```

Interfaces

IGame is used to create an interface for GamePanel so that GamePanel can update and draw and class that implements IGame, creating an outline for any game to be made with this platform.

Special Features

Can scan the local network (to a degree) for any servers running, and lists them.

Uses TCP for maximum reliability with data sent across the network.

A variable in LocalClient is:

`TimeItTakesForAPacketToGetToTheClientFromTheServer`
Allowing for the client (and server!) to calculate ping

When a third player joins, sometimes the game crashes. If it doesn't crash, the second player who connected to the server will have planets flying across their screen for no reason. I did not implement this. I do not know why this happens.

Known bugs

Local server browser won't find servers with IP addresses that have different first 3 numbers.

When a new client joins, there's a chance the entire game will crash for everyone but the host of the server.

It's written in Java, so it will run out of memory or have a `NullPointerException` because of JVM garbage collection after a while.

“Second-party” code

- Didn't use any
- I barely know what “second-party” means
- If you count JVM as third-party code, then sure... still not second party tho

Conclusion

Pretty fun game

Java is slow

Questions

Why would anyone write a real game in Java

Minecraft was written in Java

It's really slow