Period 1
Melanie Lin
Group Name: Error 404
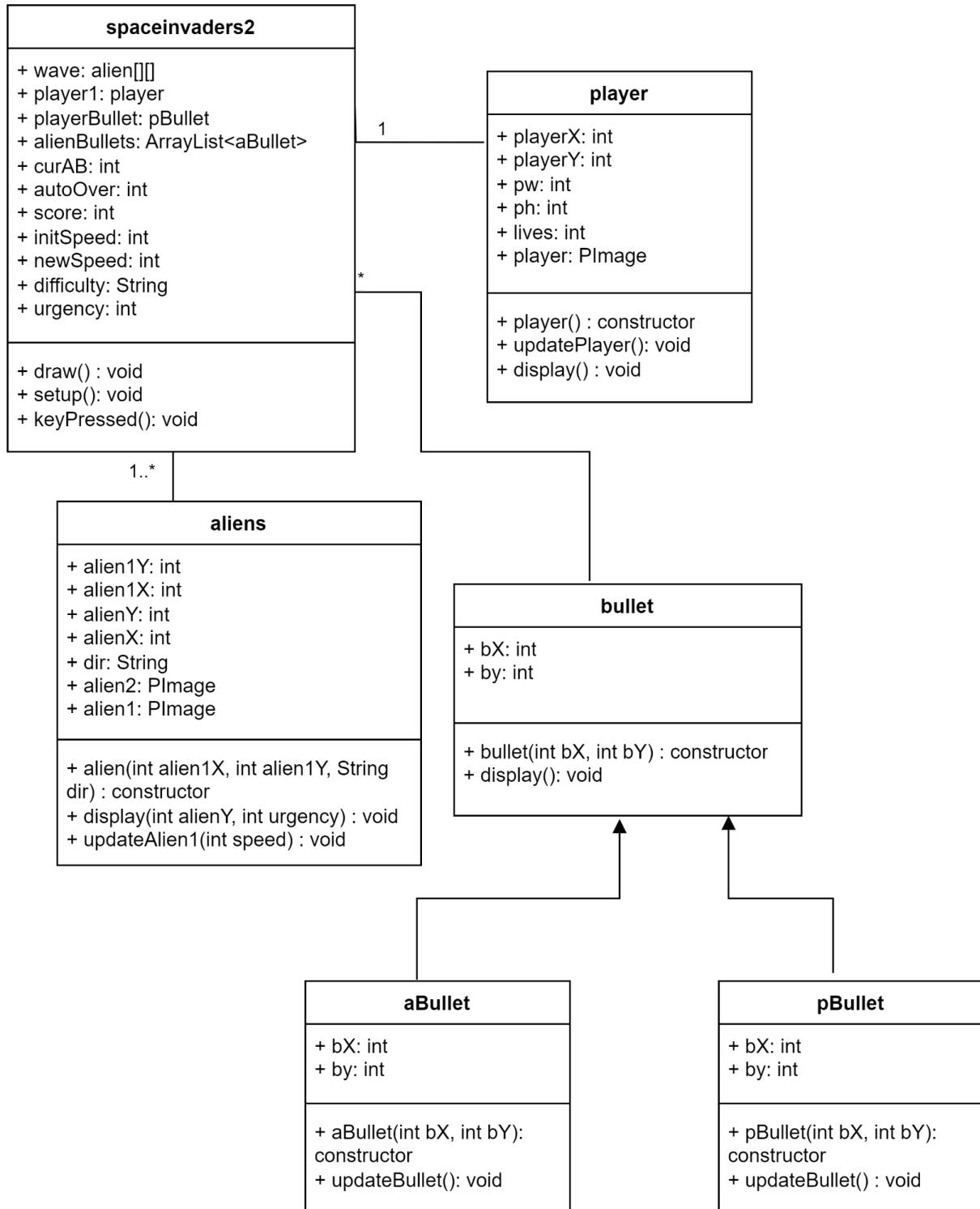
Project Title: Processing Invaders
Description: This project is a Processing version of the popular classic arcade game Space Invaders, in which the player moves across the bottom of the screen, shooting up at aliens moving down in a wave from the top of the screen.

Functionalities:
- The player spawns and can be moved horizontally across the bottom of the screen
- A wave of 3 lines of aliens spawns
- The wave of aliens slowly makes its way down the screen
- The aliens periodically shoot down at the player
- The player can shoot up
- The score, displayed on the top of the screen, increases for every alien that is killed
- The aliens progressively move faster as more are killed
- The player will die and the game ends, displaying the score, once it gets hit by alien bullets 3 times or the aliens reach the bottom of the screen.
    - The player's number of lives left is displayed on the top of the screen
- The player will win and the game ends once every alien is killed
- There are 3 different difficulties that change the speed of the aliens, the current of which is displayed on the top of the screen.

UML Diagram:

```
┌─────────────────────────────────┐
│         spaceinvaders2          │
├─────────────────────────────────┤
│ + wave: alien[][]               │
│ + player1: player               │
│ + playerBullet: pBullet         │
│ + alienBullets: ArrayList<aBullet> │
│ + curAB: int                    │
│ + autoOver: int                 │
│ + score: int                    │
│ + initSpeed: int                │
│ + newSpeed: int                 │
│ + difficulty: String            │
│ + urgency: int                  │
├─────────────────────────────────┤
│ + draw() : void                 │
│ + setup(): void                 │
│ + keyPressed(): void            │
└─────────────────────────────────┘
```

```
┌─────────────────────────────────┐
│             player              │
├─────────────────────────────────┤
│ + playerX: int                  │
│ + playerY: int                  │
│ + pw: int                       │
│ + ph: int                       │
│ + lives: int                    │
│ + player: PImage                │
├─────────────────────────────────┤
│ + player() : constructor        │
│ + updatePlayer(): void          │
│ + display() : void              │
└─────────────────────────────────┘
```

1

*

```
┌─────────────────────────────────┐
│             aliens              │
├─────────────────────────────────┤
│ + alien1Y: int                  │
│ + alien1X: int                  │
│ + alienY: int                   │
│ + alienX: int                   │
│ + dir: String                   │
│ + alien2: PImage                │
│ + alien1: PImage                │
├─────────────────────────────────┤
│ + alien(int alien1X, int alien1Y, String │
│ dir) : constructor              │
│ + display(int alienY, int urgency) : void │
│ + updateAlien1(int speed) : void │
└─────────────────────────────────┘
```

1..*

```
┌─────────────────────────────────┐
│             bullet              │
├─────────────────────────────────┤
│ + bX: int                       │
│ + by: int                       │
├─────────────────────────────────┤
│ + bullet(int bX, int bY) : constructor │
│ + display(): void               │
└─────────────────────────────────┘
```

```
┌─────────────────────────────────┐
│            aBullet              │
├─────────────────────────────────┤
│ + bX: int                       │
│ + by: int                       │
├─────────────────────────────────┤
│ + aBullet(int bX, int bY):      │
│ constructor                     │
│ + updateBullet(): void          │
└─────────────────────────────────┘
```

```
┌─────────────────────────────────┐
│            pBullet              │
├─────────────────────────────────┤
│ + bX: int                       │
│ + by: int                       │
├─────────────────────────────────┤
│ + pBullet(int bX, int bY):      │
│ constructor                     │
│ + updateBullet() : void         │
└─────────────────────────────────┘
```

How does it work?

The game automatically starts when the project is run.

The objective of the game is to shoot and kill all of the aliens before they shoot the player three times or move down to the bottom of the screen.

The user must move the player to avoid being hit by alien bullets.

The player can move left or right across the bottom of the screen and can't go off-screen. To move left, hold the left arrow key. To move right, hold the right arrow key.

To shoot, press the '1' key.

To switch difficulties:
- Easy: press '2' (also default)
- Medium: press '3'
- Hard: press '4'

Once the player dies or wins, the game ends and to replay it, the project must be run again.