# Project 1:Shooter.

### What are we aiming for?

This is our first project. The aim is to make an advanced game. Much like the “Space Invaders” games your parents might know!

We will be using all of our skills to make this game: classes, for loops, collision detection, scoring..

## Steps to make this app:

1. **Make the tower!**
2. **Make the Alien (using a class!)**
3. **Make the bullets! (using a class!)**
4. **Tieing the basics together.**
5. **(Extra) Multiple Aliens!**

## 1: Make the tower!

This is the most simple step. You need to make the defense turret or whatever the player is going to be.

This should:

- Look good (have at least 2 different colours and 3 different shapes.
- Move left and right at the bottom of the screen.
- You know how to do this - go mad! Make it a cool as you can!

## 2: Make the Alien.

This will the be first class we make on our own. So pay attention!

1. Make a new tab:click the down arrow and select “New Tab”. Call it “Alien”
2. in that tab make the class container - everything in this class goes in this ***{codeBlock}***

```
class Alien {
  int alienHozPoz;
  int alienVertPoz;
  public Alien(int initialHosPos, int initialVertPos) {
    alienHozPoz = initialHosPos;
    alienVertPoz = initialVertPos;
  }
}
```

3. Add some basic Alien variables: `alienHozPoz`, `alienVertPoz`, etc.
4. Add a constructor (remember like the `setup` for a class). This should take the starting position as parameters. See above for an example.

5. Add a `drawAlien` method to the alien class.
    a. Go nuts with this! Make the Alien as cool as you like, two restrictions:
        i. The `vertPos, hozPos` **MUST** be the **center** of the Alien.
        ii. Everything is based off the `vertPos, hozPos.` If they change the whole alien must move!
            1. The most simple Alien would be a circle.
6. Add an `alienSize` variable. This should be set in the constructor. (we don’t need to pass it in). It should be as big as the CORE of the Alien. If a shot hits this - the alien is hit!
7. Add a `isHit` method. This should be a **`boolean`** (returns **`true`** - if the alien is hit, or **`false`** - if the alien is not hit.) It should take in 2 parameters - the `hoz` and `vert` position of what might hit the Alien. We need to use dist between the parameters and the `alienHozPoz, alienVertPoz.` And compare that to the `alienSize` variable.
    a. Ask if you get stuck here!

## 3: Make the Bullets

This is our second class! This is mostly up to you!

1. Using the same steps as we did for the new Alien class - make a new Bulletn class! It should:
    a. Also have position variables.
    b. Take in the starting postions in the constructor.
    c. Have a `drawBullet` function where you draw the bullet! (using the `bulletHozPos` and `bulletVertPos` position variables!) Go mad! Is it a laser? a rocket bullet? Whatever you want!
2. Make the bullet move!
    a. Add code in the `drawBullet` function so that every time that is called - the bullet moves closer to the top of the screen.
3. Try our your bullet! Does it move as it should?
4. Now add user input!
    a. Make one array of Bullet objects at the top of the program.
    b. When you click the mouse
    c. Create that Bullet were you tower is! See below code for a starter:

```
ArrayList<Bullet> bullets;

void setup() {
    bullets = new ArrayList<Bullet>();
…….

void draw() {
    for (int i=0; i<bullets.size(); i++ ) {
    bullets.get(i).drawBullet();
   }
```

```
…. .

void keyReleased() {
    Bullet newBullet = new Bullet(mouseX, height - 60);
    bullets.add(newBullet);
}
```

5. Try it out! You should be able to shoot as many bullets as you want!

## 3: Tieing the basics together.

Looking pretty good yeah!?
we need to add one more thing! For the basic game to work:

1. when you are drawing all of the Bullets in the array of bullets - check if any of them hit the Alien.
    a. How!?!? Use this `isHit` fucntion on the Alien class.
        i. For each of the Bullets - call this function on the Alien.
        ii. Use the `bullets.get(i).bulletHozPos` and `bullets.get(i).bulletVertPos` variables.
        iii. If this function returns true - do something!
2. More! You can now start to add stuff to your game!
    a. Make the Alien move! Does it move randomly? Or in a straight line? Does it have a speed? Is the speed random?
    b. Add health to the Alien! Needs a new varibale! Does it go down each time the Alien is hit? But the `isHit` only returns true if the alienHealth is < 0?
    c. Restrict the amount of bullets you have!
        i. Remove the bullets from the array if they go off the top of the screen.
        ii. Remove the bullets from the array if they hit the Alien.
        iii. Only add a new bullet if there are less than a certain amount of bullets in the array!

## (Extra) Multiple Aliens

Remember how we have multiple bullets? Do the same for Aliens!
But remember: you now need to replace each alien with a for loop that will loop through all of the Aliens.
You need to make them all at the start of the program - in the `setup` ***{codeBlock}***