

technocamps

Inspiring | Creative | Fun

Ysbrydoledig | Creadigol | Hwyl



Greenfoot: Tanks!

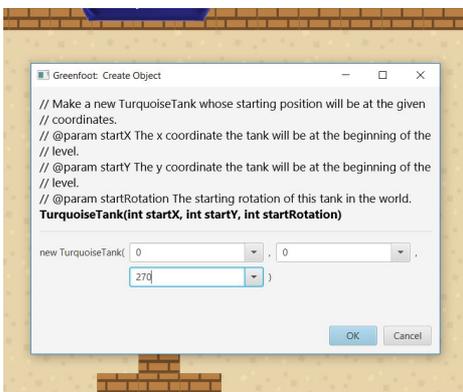
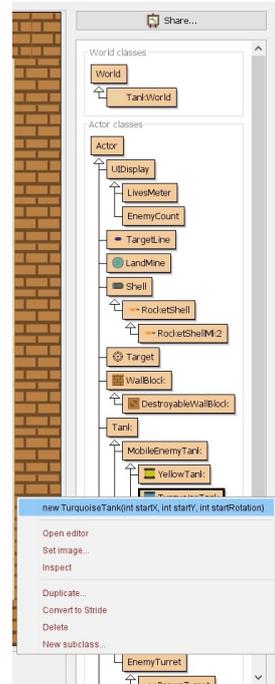
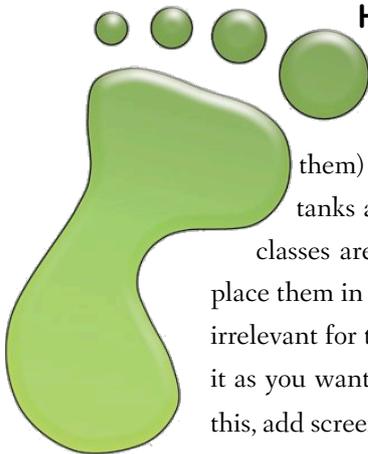
How to make a new level

Open any existing level of the game in Greenfoot, pause the game and start rearranging the wall blocks and tanks (by left clicking on them and dragging them) however you want your level to look. You can add new wall blocks and new tanks also. This is done by going to the right pane of Greenfoot where all the Actor classes are, click on the wall block or tank you want, select new WallBlock/Tank, and place them in the world. For Tanks, you will be prompted to put in 3 integers. The first 2 are irrelevant for this stage, the third is the rotation this tank will have at the start of the level, set it as you want to see how the tanks will look at the start of your level. (Explain how you do this, add screenshot too)

Now, put Greenfoot on one half of the screen and on the other open the Greenfoot Code editor for the TankWorld class. Make a new method called “ private void prepareLevelX()

“ (where X is the number of your level, should be the number after the last current level) after the last prepareLevel method. In this, you will add the walls you made by using the private method addWall(int startX, int startY, boolean destroyable, boolean isHorizontal, int nrOfBlocks) in the TankWorld class:

- startX is the x position of the first block of the wall.
- startY is the y position of the first block of the wall.
- destroyable is true if the wall will be made from destroyable wall blocks, false if not. Note that if in your level you had a wall made from destroyable and non destroyable blocks, it will have to be written in the method as different walls.
- isHorizontal is true if the wall will be horizontal, false if vertical.
- nrOfBlocks is the number of blocks this wall will have. If the number is negative, the blocks will be put to the left (or above if the wall is vertical) the first block. If it is positive, they will be put to the right (or under if the wall is vertical).



Look at the starting blocks of the walls you made in the level. Use inspect to find out their x and y position. Use this as startX and startY, then put in if the wall will be destroyable or not, vertical or not, and the number of blocks the wall will have. Do this until every wall you placed with the mouse is written in the method. Then use inspect to find the x and y position of the tanks you put in the level. These values will be put as the first 2 argument of the Tank constructors, which are the starting coordinates of the tank in the level. The third argument is the starting rotation the tank will have. Make the new tanks with these arguments, and put them in the world at the starting coordinates you put in their constructors (the method at the start of the class, declared with public followed by the name of the class and then the arguments. It is the method that creates a new object of that class).

Then go to the prepare() method in the TankWorld class, and before the keyword "default" add:

```
case X:
    prepareLevelX()
    break;
```

, where X is the number of your level, like in the screenshot.

That is it! Your new level is done.

```
case 5:
    prepareLevel5();
    break;
case 6:
    prepareLevel6();
    break;
case 7:
    prepareLevel7();
    break;
case 8:
    prepareLevel8();
    break;
case 9:
    prepareLevel9();
    break;
case 10:
    prepareLevel10();
    break;
/*add your new levels here*/
case 11:
    prepareLevel11();
    break;
/*If the last level has been cleared, then the level counter has been
* incremented beyond the number of levels in the game, so it will
* resort to calling the method that display the game win message and
* stop the game.*/
default:
    gameWin();
    return;
```

Creating a new tank

If you want to make a new type of tank that is stationary, make a new subclass of StaticEnemyTank. To make a new mobile enemy tank, make your new class as a subclass of MobileEnemyTank instead, then do the same. For both cases, copy-paste the code from the BrownTank class, change the constructor to match the name of your new tank, and if you make a new type of turret for this tank, put the name of that in the addedToWorld method after "new". For mobile enemy tanks, then override (make a new version of these methods in your class, which will replace the ones in the MobileEnemyTank) the getSpeed, getMaxTurnSpeed, getMineAvoidanceDistance and

```
public class GreenTank extends Tank
{
    /**
     * Makes a new Green Tank, with it's start x and y coordinates the ones given
     * as arguments.
     * @param startX The starting x coordinate of this tank.
     * @param startY The starting y coordinate of this tank.
     * @param startRotation The starting rotation of this tank in the world.
     */
    //this is a constructor
    public GreenTank(int startX, int startY, int startRotation)
    {
        //simply calls the constructor of the superclass.
        super(startX, startY, startRotation);
    }
}
```

getShellAvoidanceDistance and make them return custom values that you choose for your tank. If you want your new type of tank to lay mines, also override getNumberOfMines and getMineLayingPeriod and put custom values that you choose. Look in the MobileEnemyTank class at these methods and read the comments before each one to understand what they are supposed to do.

If you want to make a new type of enemy turret, then make a new subclass of EnemyTurret, make a constructor that just passes the tank argument to the superclass constructor, then override the

getFireCooldown, getLiveShellLimit and the getAimAngle methods and make them return custom values of your own. If you want your turret to fire rocket shells, override getShellType so that it returns RocketShell.class or RocketShellMk2.class. Look in the EnemyTurret class at these methods and read the comments before each one to understand what they are supposed to do.

If you want a new type of tank and turret with completely new behaviours, you will have to be a bit more skilled and creative! The code is thoroughly commented so if you want to you can understand what I did and expand upon it!

```
/**
 * Overrides the superclass addedToWorld method so that a Green Turret will be
 * placed on this tank not a simple Turret object.
 */
@Override
protected void addedToWorld(World world)
{
    setRotation(startRotation);
    tankTurret=new /*put name of your new Turret here*/GreenTurret(this);
}
```

You can find all of the code at: github.com/Alexandru-Dascalu/Greenfoot_Tanks



www.technocamps.com

