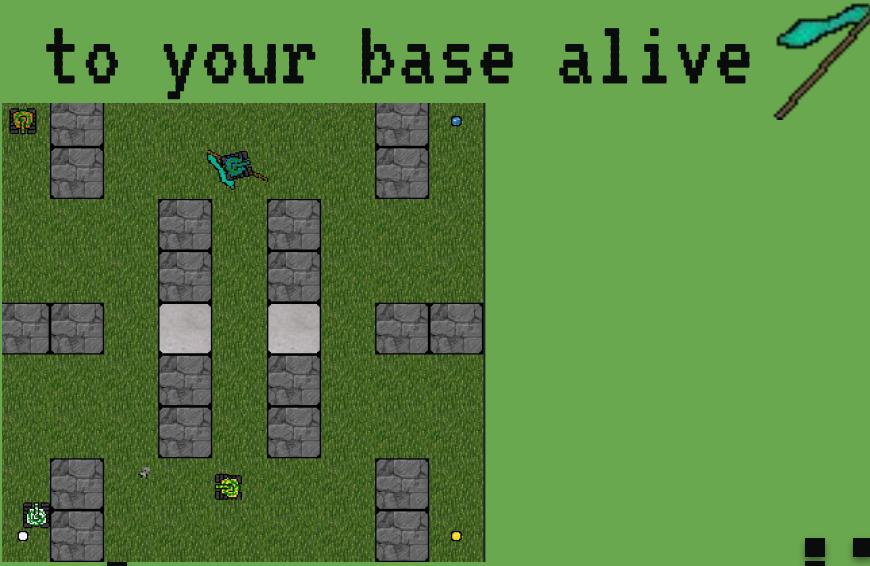




# CAPTURE THE FLAG

Vanessa Slopi, Rikard Boström, Oliver Lindberg

In this game you play as a tank,  
the goal is to snatch the flag  
before your enemies and make it  
back to your base alive!



# HOW TO PLAY CAPTURE THE FLAG

Before you enjoy this game there is a few steps you need to follow.

1. Install python3.
2. Run following commands in the terminal.
  1. source setup.sh
  2. python3 menu.py

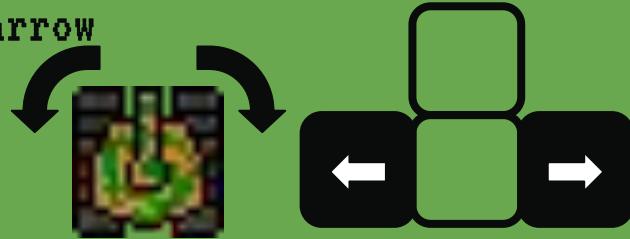
You are now all set and ready to begin!

You also have a multiplayer option and the option to choose a desired map

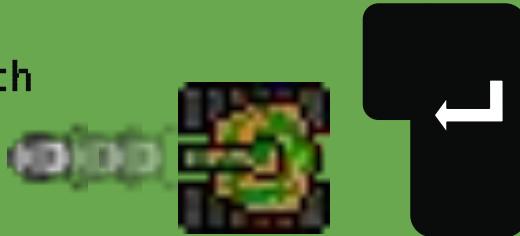


# Advanced operation manual for tank

Rotate  
with the  
side-arrow  
key!



Shoot with  
enter!



Accelerate  
with the  
forward  
key!



# System

**01** `ctf.py`

Main file for the game

**02** `ai.py`

Contains function and classes for the Artificial Intelligence used in the game

**03** `gameobjects.py`

Defines classes for visual and physical objects in the game

**04** `maps.py`

Includes different map options for the game

**05** `menu.py`

Defines the main menu for the game

## Ai Class

Revolves around moving along a path decided by an algorithm and shooting when convenient.

`decide():`

    Calls `maybe_shoot` and `move_cycle`

`maybe_shoot():`

    Shoots if appropriate

`move_cycle():`

    Gets best path and travels through it

# Pathfinding

`filter_tile_neighbors():`

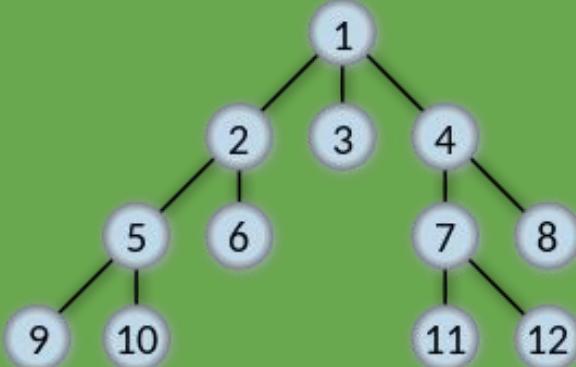
Used to filter out tiles out of bounds along with tiles with metal or stone boxes.

`get_tile_neighbors():`

Returns the filtered neighbors of current tile.

`find_shortest_path():`

Uses breadth first search to get the shortest possible path, using `get_tile_neighbors()` to get the nodes.



# Classes in gameobjects.py

01

**GameObject**

Mostly handles visual aspect of an object.

02

**GamePhysicsObject**

Handles physical objects and their interactions

03

**GameVisibleObject**

For object that are visible on screen but have no physical representation (ex. flag)

04

**Bullet**

Handles bullets

05

**Tank**

Handles aspects which are specific for the tanks

06

**Box**

Handle box objects

# Shoot and Bullet

**01** `shoot()`

Returns Bullet object

**02** `Bullet class`

Subclass of GamePhysicsObject

**03** `Creation`

Bullet created based of tank location and orientation

**04** `update()`

Sets the velocity of the bullet with a rotated vector

**05** `Cooldown`

Counts amount of ticks after shot

# Collision

**01** Collision type

Number objects

**02** Collision handlers

Calls functions when two specified collision types collides

**03** Arbiters

Colliding objects

**04** update()

Sets the velocity of the bullet with a rotated vector

**05** Cooldown

Counts amount of ticks after shot

# Features

## Recoil

Recoil when the tank shoots

## Explosion

Explosion vfx on bullet collision

## Fog of war

## Welcome screen

A main menu for the game

## JSON map files

Import maps as files

## Networked multiplayer

(unfinished)

# Menu

## 01 Display text

Creates a text with customizable font, size and location

## Buttons

02

Buttons that light up with color when hovered over

## Actions

03

Each button has an action that either starts the game in a specified mode or changes the menu page

## Pages

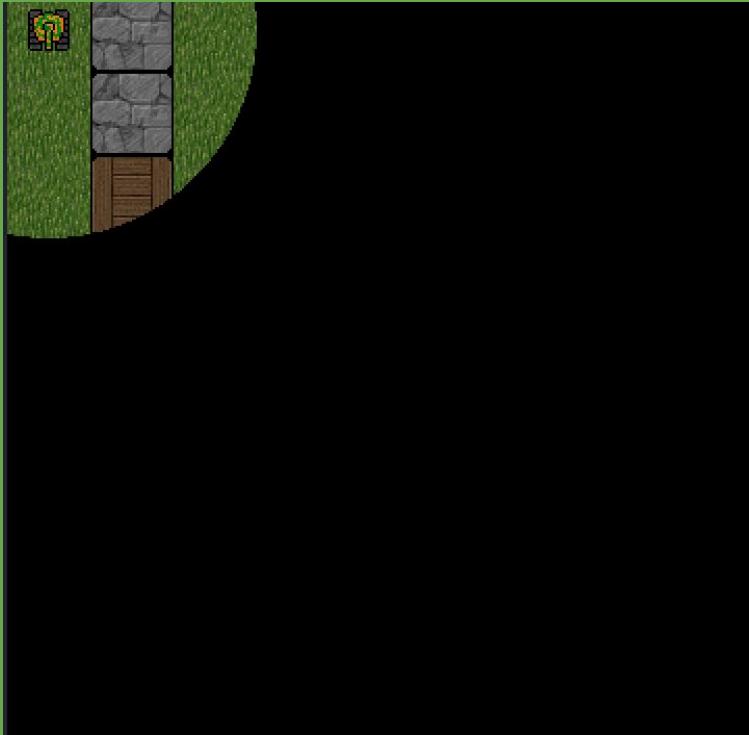
04

Sets the velocity of the bullet with a rotated vector

## Display maps

05

Miniature images of maps displayed on the map picker page



## Fog of war

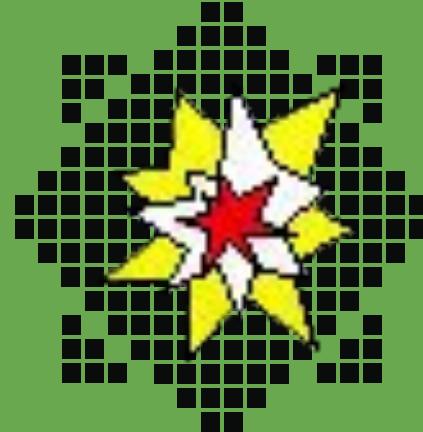
Limiting the player's visibility to only their tanks position. Unexplored areas remain covered by the black fog preventing the player from seeing enemies movements or structures

# Explosion

## A new gameobject

When an explosion gets created, frames are counted down.

The explosion stays big for a few frames, and then begins shrinking with the descending frames until finally removing itself from the loop.



# JSON map files



JSON file gets created

JSON file is loaded

Loaded JSON object is used to create map object

# Questions?

- Ask away!

