

Capture the flag!

Controll a tank and achieve victory on multiple different maps. Battle the AI or maybe you have someone you would like to destroy in hotseat multiplayer. Or perhaps you want to defeat the AI together with your friend then the CO-OP mode is for you.

Starting the game

To start the game you will need to have installed python3 along with our program. Navigate to our folder in your terminal and run Setup.sh with the following command to get the correct environment. Then you can run the game with the ctf.py file by the following command

- source setup.sh
- python3 ctf.py

Controls

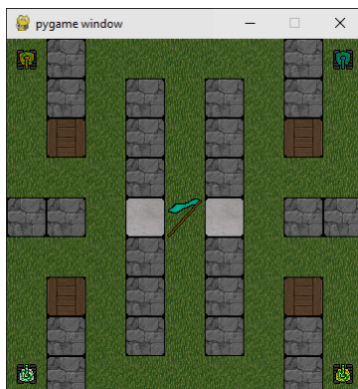
Player 1		Player 2 (when Multiplayer)	
Key	Function	Key	Function
ESC	Returns to main menu	ESC	Returns to main menu
UP	Drive Forward	W	Drive Forward
LEFT	Turn Left	A	Turn Left
DOWN	Drive Backwards	S	Drive Backwards
RIGHT	Turn Right	D	Turn Right
CTRL	Shoot	SPACE	Shoot

Maps:

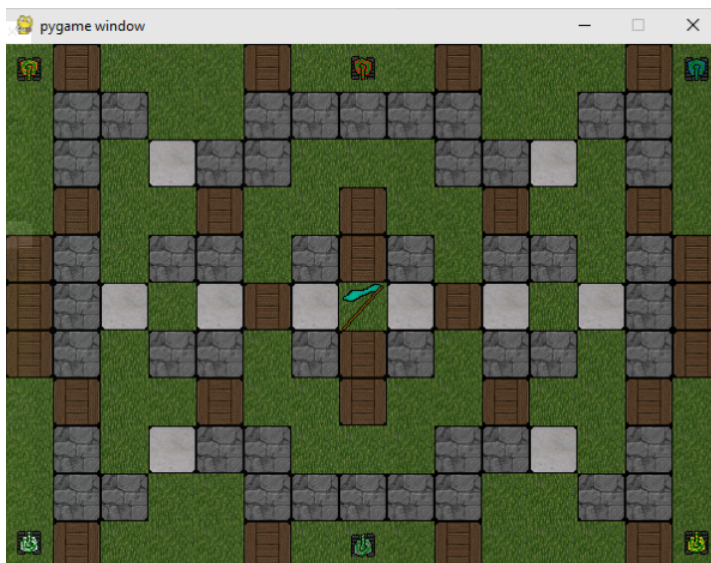
Small: Contains 2 Players and metal boxes. A good map to duel against your friend or AI.



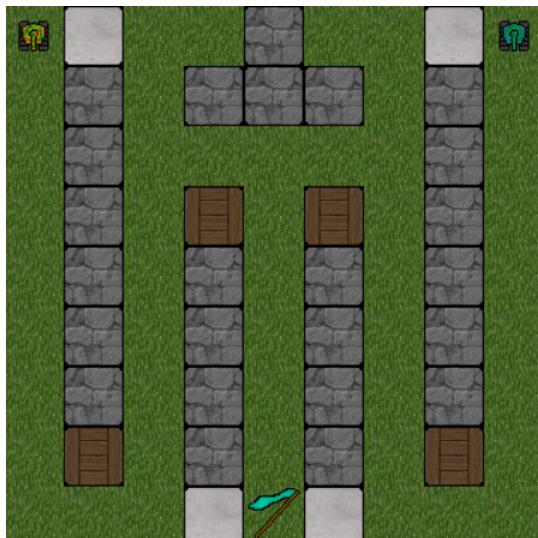
Medium: Contains 4 players and has a clear path to the flag. It is a good beginner map



Large: Contains 6 players and is a more complicated map to navigate. A good challenge



Custom: A race to the flag and back



Implemented features:

Easy Features

1. **Counting Score (1pt)**
 - Keep track of each player's score during the game.
 - Instead of ending the game immediately after a win, scores are printed to the console whenever someone captures the flag.
2. **Sounds (1pt)**

- Adds audio effects for events like flag captures, shooting, or destroying objects.
 - Plays background noise during the game.
 - 3. **Hot Seat Multiplayer (1pt)**
 - Adds two-player mode where both players use the same keyboard to control their tanks.
 - 4. **Hit Points (1.5pt)**
 - Tanks no longer get destroyed with a single hit.
 - Multiple hits are needed to destroy an object, making gameplay more challenging.
 - 5. **Respawn Protection (1.5pt)**
 - Prevents tanks from being destroyed immediately after respawning.
 - Adds a temporary shield after a tank reappears.
 - 6. **Unfair AI (1pt)**
 - Makes AI tanks harder to defeat by boosting their speed and bullet velocity.
-

Medium Features

- 7. **Read Maps from JSON File (2pts)**
 - Changes how maps are stored from Python files to JSON format.
- 8. **Welcome Screen (2.5pts)**
 - Displays a starting screen before the game begins.
 - Lets players select the game mode and map, showing a preview of the selected map.
- 9. **Score Screen (2.5pts)**
 - Extends the Counting Score feature by displaying scores directly on the screen after a flag capture.
 - Replaces the need for console output.
- 10. **Additional Win Condition (3pts)**
 - Adds three new ways to end the game:
 - Score Limit: Ends when a player/team reaches a certain score.
 - Time Limit: Ends after a set amount of time.
 - Round Limit: Ends after a specific number of rounds.
 - Also added a Freeplay to not end the game
- 11. **Co-op Mode (3pts)**
 - Teams compete instead of individuals.
 - Teammates share scores and can't hurt each other.
 - Teams can either have shared bases or individual ones.
- 12. **Recoil (2pts)**
 - When a tank fires, it experiences a backward push.
 - The recoil force makes the tank move in the opposite direction of the shot.
- 13. **Explosion (2pts)**
 - When bullet hit something an explosion happened instead of just disappearing

Brief explanation of each file and how the program calls them:

ai.py

AI controls the tanks through different functions so that the AI can control the tanks. They are programmed to find the flag first then return to their homebase, shooting wood boxes or tanks that come in their way. If a flag is taken the AI can also intercept and take a short cut to cut the flag carrier off. The ai file is imported at the top of the main file.

gameobjects.py

Game Objects handle all the physical objects in the game. Tanks, Boxes, Walls, Bullets and flags along with all their attributes of the different objects. It handles movement, destruction, shooting etc. The object calls its responding function when needed and is imported at the top of the main file.

images.py

Images are loaded here and the file is imported when needed. Imported at the top of the main file

maps.py

Maps are stored in the data folder in json format and the Maps.py calls on the json and converts it to readable format for our python game.

ctf.py

This is where the game is started from. It contains the menu and options.

sounds.py

Sound is loaded in sounds.py so each time a sound is needed it does not have to be imported during every event. It is imported at the beginning of each game

rungame.py

The main file of the game which contains all the main functions that create the game. It contains a main loop that continuously handles all functions and events such as updates or key presses. It runs every frame until the loop is ended which is 50 times per second.

setup.sh

A shell script that creates an environment so you can run the game. Is only run before the game is started