Capture The Flag User Manual

How to get started

In order to start the game, you must have all the files downloaded, while includes the CTF-env folder, the data folder where all of our media is stored, ai.py, ctf.py, gameobjects.py, images.py, and maps.py, which are the game files. Now you're all set up for starting up the game, and in order to launch the game, you'll have to navigate into the folder where your ctf.py file is located and open a terminal in this specific location. Then you want to, first of all, activate your virtual environment for Python by writing "*source ctf-env/bin/activate*" (*Note: you have to write this every time you open and close your terminal*). After this, you also want to install the required libraries in order to run this game. To do so, start off with writing "*pip install pymunk==5.7.0*" and enter, and then "*pip install pygame==2.0.1"* and enter.

Start commands

Now you have everything you need in order to start up Capture the Flag. Once again, have a terminal open in the location where your ctf.py file is located, and write "*python3 ctf.py --singleplayer*" in order to play in singleplayer mode, or if you'd like to play with or against a friend, type "*python3 ctf.py --hot-multiplayer*".

Gamemodes

- *"python3 ctf.py --singleplayer"*
- "*python3 ctf.py --hot-multiplayer*"

Maps and changing map

If you'd like to change the map, you can do so by opening the ctf.py file in any text editor program and navigating to line 43, or CTRL+F and searching for current_map. After that, you can change the maps.map0 to either maps.map1 or map.maps2, depending on which map you'd like to play. Maps.map0 is the default map with a moderate-sized map and 4 tanks, whereas Maps.map1 is the biggest map and also the most difficult one. Lastly, maps.map2 is the duel map, where you play against only one other tank on a smaller map.

Features

The features that have been implemented into this game are the following:

- Sounds - SFX when you shoot, destroy tanks and boxes, and when you capture a flag
- Hitpoints - The tank can take more than one shot to go down

- Respawn protection - Prevent griefing, you're invincible for a short period of time when you respawn
- Unfair AI - AIs have increased stats, such as movement speed and HP to make it more challenging
- Hot Seat Multiplayer - Choose between playing singleplayer mode or against someone else on the same device
    - Controls for Player 1 and 2 respectively:
        - Arrow Up, Arrow Down, Arrow Right, Arrow Left, and Return to accelerate, decelerate, turn right, turn left and shoot respectively
        - W, S, A, D, and Space to accelerate, decelerate, turn right, turn left and shoot respectively
- Fog of war - Your vision is a limited circle around your tank
- Explosions - Visual explosions whenever you destroy a tank or box

Functionality

Files

There are six game files in total, and they all have their own specific functionality, but the main file is ctf.py, which brings all of the functionalities of the other files together to run the game. Starting off with the "main file", ctf.py, it initializes first of all the display, a pygame clock, loads in audial media and handles physical collision between game objects, as well as it generates backgrounds, as well as the main loop for running the game itself, where it detects keyboard input and every game event such as AIs decision making, updating physics, display each tick.

It is also where all the game objects by using gameobjects.py are created, where classes for both game objects, physics for them as well as properties and functions for each of these game objects. It is also here where the handling of the shooting, creating bullets as well as functions such as movement of the tank and grabbing the flag.

Next up is AI.py, where our AI class and where everything from properties unique to the AI is defined, as well as the functions necessary for the AI to make its decision, for example, maybe_shoot which check whether it should shoot or not depending on if there's a tank or a wooden box in its line of shooting or how it's supposed to move, by iteratively going through all of the required movements to reach for example the flag and to return to the base in order to win. Basically, everything that has to do with the AIs' decision-making, from moving to shooting.

Second to last out is media.py, where the loading of images, as well as sound files from the data folder, happens.

Lastly, we have the maps.py file, which is where the map class is defined in the form of its size as well as which type of blocks they should have at which position, as well as the spawn points and the starting position of the flag.

How the main program calls each file

In ctf.py, the maps.py is used to select which map should be in use, define the size of the screen to the size of the map, create barriers, and create the map from the given map information as well as its components such as boxes, grass, bases, tanks, and flag. The fog of war functionality also uses properties from maps.py. The media file is called to load the SFX as well as to set the volume of the sounds, as well as the sounds, are being played, such as explosions when tanks and crates are being destroyed or when a flag has been grabbed. AI.py is being called when tanks are being created (the remaining tanks that aren't players) as well as inside the main loop, where every AI makes their decisions each tick. Lastly, the gameobjects.py file is whenever physics are being handled, such as collisions, movement, or shooting, since the property of the game objects called collision_type is used in such collision handling and the functions for movement and shooting is in gameobjects.py. As well as whenever a game object is being used, such as whenever you want to alter the HP of a tank, or its spawn protection, creating the game objects itself like tanks, map objects e.t.c. All the updating of the game objects in the main loop, be it visual or physical, also calls for the update/post_update function which is in gameobjects.py. The fog of war functionality also uses the properties of tanks which is linked to the tank class inside of gameobjects.py.