Parental Controls: Game Time Limiter (*game_time_limiter*)

> *Application for setting a time limit for kids Windows games and applications.*

python 3.13.3   version 0.1.1   issues 0 open   ⬡ Forks 0   ⬡ Stars 0   license MIT

NOTE: This project was generated with Cookiecutter along with @clamytoe's toepack project template.

Initial setup

```
cd Projects
git clone https://github.com/clamytoe/game_time_limiter.git
cd game_time_limiter
```

Anaconda setup

If you are an Anaconda user, this command will get you up to speed with the base installation.

```
conda env create
conda activate game_time_limiter
```

Regular Python setup

If you are just using normal Python, this will get you ready, but I highly recommend that you do this in a virtual environment. There are many ways to do this, the simplest using *venv*.

```
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
```

Create Windows executable

```
pyinstaller --onefile --windowed --icon=OGS.ico game_time_limiter.py
```

Configuration

I've made it so that you can configure the app before it is run. This is done by creating a `config.json` file in the base directory of the application. Just modify the values in the file to suit your needs.

config.json:

```
{
    "limit_minutes": 120,
    "log_path": "C:/Users/clamy/AppData/Roaming/GameTimeLog.json",
    "apps_list": "C:/Users/clamy/Documents/apps_list.txt",
    "password": "mysecurepassword"
}
```

Additional tools

I've provided additional scripts to help with finding what Steam and Epic Games are installed on the system, along with another to display the currently running executables.

```
> python find_games.py

[Steam]
LockdownProtocol.exe
Myst.exe
HogwartsLegacy.exe
ClearThirdParty.exe
StillWakesTheDeep.exe
Warhammer 40,000 Boltgun.exe
Wuthering Waves.exe
Riven.exe

[EpicGames]
InfinityNikki.exe
```

and

```
> python active_processes.py

[Processes]
ASUS_FRQ_Control.exe
AacAmbientLighting.exe
AcPowerNotification.exe
AcrobatNotificationClient.exe
AdobeCollabSync.exe
AggregatorHost.exe
AppVShNotify.exe
AppleMobileDeviceService.exe
ApplicationFrameHost.exe
ArmouryCrate.Service.exe
ArmouryCrate.UserSessionHelper.exe
ArmouryCrate.exe
ArmouryHtmlDebugServer.exe
ArmourySocketServer.exe
ArmourySwAgent.exe
...
```

These will make it easy to populate the `apps_list.txt` file.

apps_list.txt:

```
javaw.exe
InfinityNikki.exe
Minecraft.Windows.exe
MinecraftDungeons.exe
RobloxPlayerBeta.exe
steam.exe
```

Auto Starting the Application

So that the application starts automatically when the user logs in we will add a new Task to the Task Scheduler.

Start Task Scheduler

1. Win+R: taskschd.msc
2. Click: OK

Adding New Task

1. Select Task Scheduler (Local)
2. From menu: Action > Create Task...
3. Type: Game Time Limiter
4. Under Security options: Select Run when user is logged on
5. Select Triggers
6. Click on New...
7. Begin the task: At log on
8. Settings: Any user
9. Advanced settings: Enabled
10. Click: OK
11. Select Actions
12. Click on New...
13. Actions: Start a program
14. Settings > Program/script: Browse to location of game_time_limiter.exe
15. Start in (optional): Enter the path to base directory of executable
16. Click: OK
17. Click: OK

Contributing

Contributions are welcomed. Tests can be run with with `pytest -v`, please ensure that all tests are passing and that you've checked your code with the following packages before submitting a pull request:

- black

- flake8
- isort
- mypy
- pytest-cov

I am not adhering to them strictly, but try to clean up what's reasonable.

License

Distributed under the terms of the MIT license, "game_time_limiter" is free and open source software.

Issues

If you encounter any problems, please file an issue along with a detailed description.

Changelog

- **v0.1.1** Modified badge url for license file from master to main branch.
- **v0.1.0** Initial commit.