Rogue Signal Protocol - Project Structure

Recommended Directory Structure





```
| ── tools/
| ── sprite_packer.py
| ── data_validator.py
| ── balance_analyzer.py
| └── level_viewer.py
| ── build/
| ── windows/
| ── linux/
| ── mac/
```

Initial Setup

1. Environment Setup

```
# Create virtual environment
python -m venv venv

# Activate (Windows)
venv\Scripts\activate

# Activate (Linux/Mac)
source venv/bin/activate

# Install dependencies
pip install -r requirements.txt
```

2. Requirements.txt

```
tcod>=13.8.1
numpy>=1.21.0
pygame>=2.1.0
Pillow>=9.0.0
jsonschema>=4.0.0
pytest>=7.0.0
```

3. Development Tools Setup

bash			

Install development tools
pip install black flake8 mypy

Pre-commit hooks (optional)
pip install pre-commit
pre-commit install

Git Configuration

.gitignore

```
# Python
__pycache__/
*.py[cod]
*$py.class
*.so
.Python
build/
develop-eggs/
dist/
downloads/
eggs/
.eggs/
lib/
lib64/
parts/
sdist/
var/
wheels/
*.egg-info/
.installed.cfg
*.egg
MANIFEST
# Virtual Environment
venv/
env/
ENV/
# IDE
.vscode/
.idea/
*.swp
*.swo
# Game specific
saves/
logs/
screenshots/
config.ini
# OS
```

.DS	_Store
Thu	mbs.db

Build Scripts

bulla.py			
python			

```
#!/usr/bin/env python3
"""Build script for Rogue Signal Protocol"""
import os
import shutil
import subprocess
import sys
from pathlib import Path
def build_executable():
  """Build standalone executable using PyInstaller"""
  cmd = [
    "pyinstaller",
    "--onefile",
    "--windowed",
    "--add-data", "data;data",
    "--add-data", "assets;assets",
    "--icon", "assets/icon.ico",
    "--name", "RogueSignalProtocol",
    "src/main.py"
  subprocess.run(cmd, check=True)
def package_release():
  """Package release with all necessary files"""
  release_dir = Path("release")
  if release_dir.exists():
    shutil.rmtree(release_dir)
  release_dir.mkdir()
  # Copy executable
  shutil.copy("dist/RogueSignalProtocol.exe", release_dir)
  # Copy data files
  shutil.copytree("data", release_dir / "data")
  shutil.copytree("assets", release_dir / "assets")
  # Copy documentation
  shutil.copy("README.md", release_dir)
  shutil.copy("LICENSE", release_dir)
```

```
print("Release packaged in release/")

if __name__ == "__main__":
    if len(sys.argv) > 1 and sys.argv[1] == "release":
        package_release()
    else:
        build_executable()
```

Development Workflow

1. Daily Development

```
# Start development
git pull origin main
source venv/bin/activate
python src/main.py

# Run tests
pytest tests/

# Code formatting
black src/
flake8 src/

# Commit changes
git add .
git commit -m "feat: implement stealth detection system"
git push origin feature-branch
```

2. Data Validation

```
bash

# Validate JSON data files

python tools/data_validator.py

# Check balance parameters

python tools/balance_analyzer.py
```

3. Asset Management

```
bash

# Pack sprite sheets
python tools/sprite_packer.py assets/sprites/

# Generate tileset
python tools/tileset_generator.py
```

Performance Considerations

Memory Management

- Load sprites lazily
- Cache frequently used assets
- Unload unused resources
- Profile memory usage regularly

Optimization Targets

- 60 FPS in ASCII mode
- 30 FPS minimum in graphics mode
- <100MB RAM usage
- <2 second startup time

Debugging Setup

Debug Configuration

```
python

# debug_config.py

DEBUG = True

SHOW_FPS = True

SHOW_MEMORY = True

LOG_LEVEL = "DEBUG"

ENABLE_PROFILER = True

SKIP_INTRO = True

UNLOCK_ALL_CLASSES = True
```

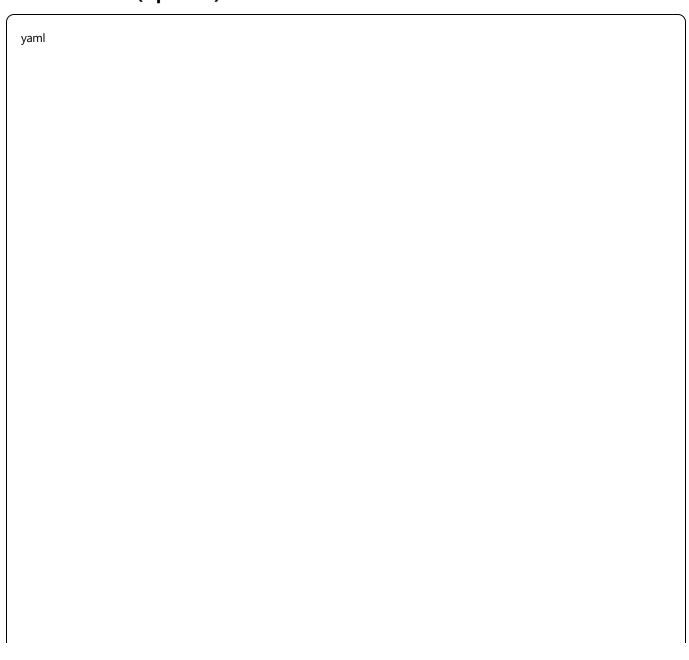
Logging Setup

```
import logging

def setup_logging():
    logging.basicConfig(
        level=logging.DEBUG,
        format='%(asctime)s - %(name)s - %(levelname)s - %(message)s',
        handlers=[
            logging.FileHandler('logs/game.log'),
            logging.StreamHandler()
        ]
        )
```

Continuous Integration

GitHub Actions (Optional)



```
# .github/workflows/test.yml
name: Test
on: [push, pull_request]
jobs:
 test:
  runs-on: ubuntu-latest
  steps:
  - uses: actions/checkout@v2
  - name: Set up Python
   uses: actions/setup-python@v2
   with:
    python-version: 3.9
  - name: Install dependencies
   run:
    pip install -r requirements.txt
  - name: Run tests
   run: pytest tests/
  - name: Validate data files
   run: python tools/data_validator.py
```