Rogue Signal Protocol - Project Structure

Recommended Directory Structure

```
rogue_signal_protocol/
├── README.md
├── requirements.txt
├── setup.py
├── .gitignore
├── docs/
│   ├── design_document.md
│   ├── technical_architecture.md
│   ├── api_reference.md
│   └── development_log.md
├── src/
│   ├── __init__.py
│   ├── main.py
│   ├── game/
│   │   ├── __init__.py
│   │   ├── engine.py
│   │   ├── state_manager.py
│   │   └── config.py
│   ├── entities/
│   │   ├── __init__.py
│   │   ├── player.py
│   │   ├── enemy.py
│   │   ├── exploit.py
│   │   └── item.py
│   ├── systems/
│   │   ├── __init__.py
│   │   ├── stealth.py
│   │   ├── detection.py
│   │   ├── combat.py
│   │   ├── pathfinding.py
│   │   └── inventory.py
│   ├── ui/
│   │   ├── __init__.py
│   │   ├── renderer.py
│   │   ├── input_handler.py
│   │   ├── screens/
│   │   │   ├── __init__.py
│   │   │   ├── game_screen.py
│   │   │   ├── inventory_screen.py
│   │   │   ├── menu_screen.py
│   │   │   └── tutorial_screen.py
│   │   └── components/
│   │       ├── __init__.py
│   │       ├── minimap.py
│   │       ├── chat_log.py
│   │       └── status_panel.py
│   ├── generation/
│   │   ├── __init__.py
│   │   ├── network_generator.py
│   │   ├── room_generator.py
│   │   ├── enemy_placer.py
│   │   └── loot_generator.py
│   ├── data/
│   │   ├── __init__.py
│   │   ├── data_manager.py
│   │   ├── save_system.py
│   │   └── achievements.py
│   └── utils/
│       ├── __init__.py
│       ├── math_utils.py
│       ├── pathfinding.py
│       └── debug.py
├── data/
│   ├── exploits.json
│   ├── enemies.json
│   ├── items.json
│   ├── classes.json
│   ├── networks.json
│   ├── balance.json
│   └── memory_fragments.json
├── assets/
│   ├── sprites/
│   │   ├── main_tileset.png
│   │   ├── ui_elements.png
│   │   └── effects.png
│   ├── fonts/
│   │   ├── consolas.ttf
│   │   └── dejavu_mono.ttf
│   ├── audio/
│   │   ├── sfx/
│   │   └── music/
│   └── shaders/
├── tests/
│   ├── __init__.py
│   ├── test_stealth.py
│   ├── test_detection.py
│   ├── test_generation.py
│   └── test_data.py
├── tools/
│   ├── sprite_packer.py
│   ├── data_validator.py
│   ├── balance_analyzer.py
│   └── level_viewer.py
└── build/
    ├── windows/
    ├── linux/
    └── mac/
```

Initial Setup

1. Environment Setup

```bash
# 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
Thumbs.db
```

Build Scripts

build.py

```
python
```

```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

```
bash

# 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

python

```python
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)

yaml

```yaml
# .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
```