Game Save Manager Proposal

CMPS 4910, Fall 2017-Winter 2018
Nick Polach

Background - Why I chose this project

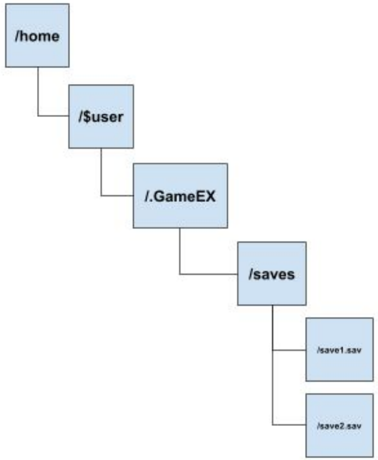
- Inspired by the “GameSave Manager” (www.gamesave-manager.com)
    - Game save manager program targeting Windows.
- My program will target GNU/Linux operatings systems

Core components

Database

- Game entries will be stored in database
- Fields include:
    - name - name of game
    - notes - information the end user should know
    - location - a list of paths that contain save files
- An example using the directory tree:

Name	Notes	Location
GameEX	None	/home/$user/.GameEX/saves



/home

/$user

/.GameEX

/saves

/save1.sav

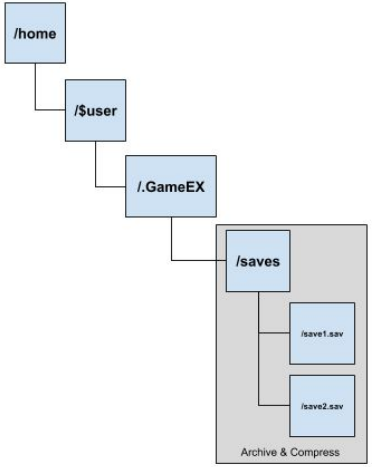
/save2.sav

Database Cont.

- Too many games to store all of them in database
- Users can add custom entries to the database
  - Add games that don’t exist in the official database yet
  - Can overwrite official game entries

Backup

- Look in game database for game entry
- Get location for game saves
- Grab files at that location
- Compress files into a single backup file

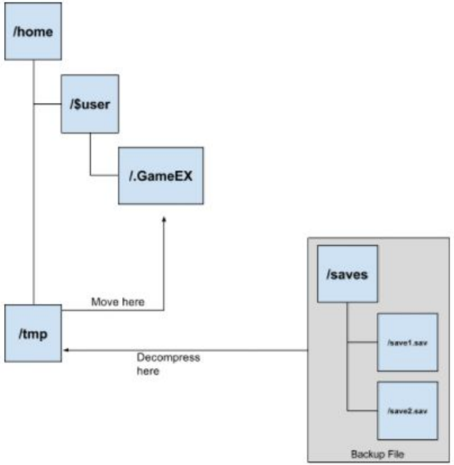


Backup Cont.

- Along with save files, include an information file in the backup
    - Information file will include data from the game’s entry (name, notes, and location)
- Information file will be used to verify that backup was made with the program
- Information will checked for save location information if the official game database doesn’t contain an entry for the game
    - This could happen if someone creates a custom game entry and distributes the backup file

Restore

- Extracts backup file to temporary location
- Check for an information file to verify backup was created with the program
- Get game information from database
  - If entry doesn't exist, use information file
- Move save files from temporary location to location from game entry



Small Issue & Solution

Issue

- File paths are unique for different systems
  - Ex. Home directories have different names depending on username
- Can’t store these unique paths in database

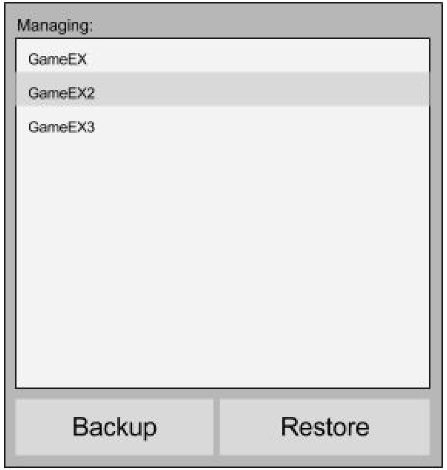
Solution

- Store paths with placeholders
  - Ex. “/home/nick/.GameEX/saves” -> “/home/{user}/.GameEX/saves”
- Python’s string method “format()” can insert unique text into those placeholders

Managing

- Keep track of games that currently have save files on system
- Stores those game in list and display them on the GUI
- User clicks on game in list and presses “Backup” button to backup game

- Program will check if files exist that are in the database to see which games should be managed



Managing:

GameEX

GameEX2

GameEX3

Settings

- Users can set custom backup directory where backup files will be placed
- User can set the naming format for backups
  - Ex. gameex.backup, gameex-092117.backup, etc.
- Users can toggle whether the program will search for new games at startup.

Libraries

Compression

Tarfile

- Compresses and decompresses files
- Can compress using gzip or bz2

Database

sqlite3

- Handles SQLite databases
- SQLite chosen because it is lightweight and embedded
    - Uses local database
    - Stored in single file

GUI Framework

PyQt5

- Provides Python bindings for the Qt framework
- Creates nice looking graphical interfaces

Others

- **os** - provides operating system dependent functionality
  - Will use it to check for file existence
- **shutil** - provides high-level file operations
  - Will use to move/copy files
- ...