Import Modules

```
import tkinter as tk
from tkinter import ttk, messagebox, simpledialog
```

Importing the tkinter module for creating the GUI.

Importing additional components like ttk for modern widgets, messagebox for information/error messages, and simpledialog for simple dialogs.

Game Data

```
game_list = [
    {
        "ps4": [
            {"name": "God of War", "price": 49.99, "stock": 10},
            ...
        ]
    },
    ...
```

A list of games stored as a list of dictionaries. Each dictionary represents a console with its list of games and related information.

Function to Display Console List

```
def display_console_list():
    ...
```

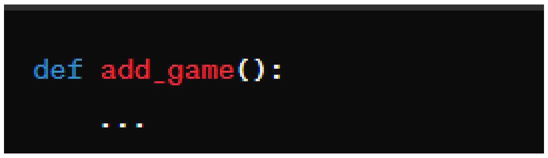
This function retrieves the set of all consoles from game_list and displays them in console_list (Listbox).

Function to Display Games by Console

```
def display_game_by_console(event=None):
    ...
```

This function displays the list of games based on the selected console from console_list.

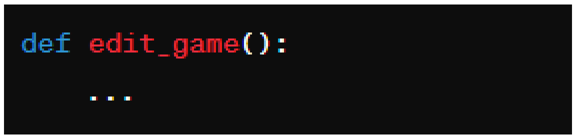
Function to Add Game



```
def add_game():
    ...
```

This function adds a new game to game_list based on user input

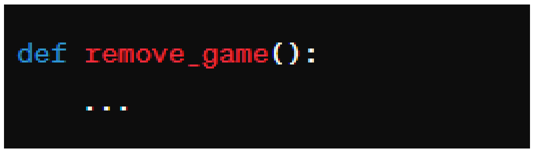
Function to Edit Game



```
def edit_game():
    ...
```

This function edits the selected game's information based on user input.

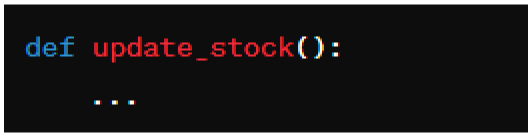
Function to Remove Game



```
def remove_game():
    ...
```

This function removes the selected game from game_list.

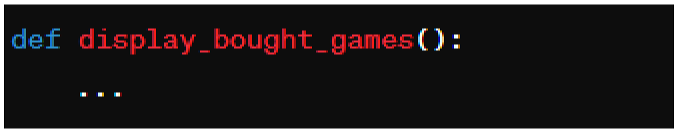
Function to Update Game Stock



```
def update_stock():
    ...
```

This function updates the selected game's stock based on user input.

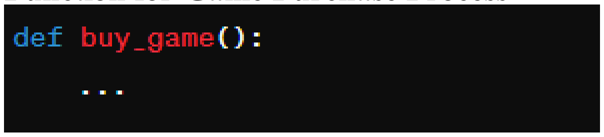
Function to Display Bought Games



```
def display_bought_games():
    ...
```

This function displays the list of purchased games in bought_game_tree (Treeview).

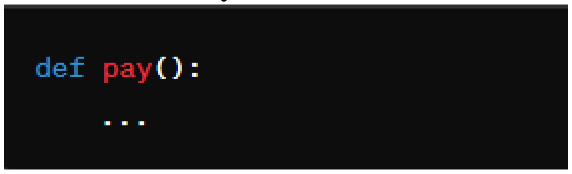
Function for Game Purchase Process



```
def buy_game():
    ...
```

This function processes the game purchase based on user input and reduces the available game stock.

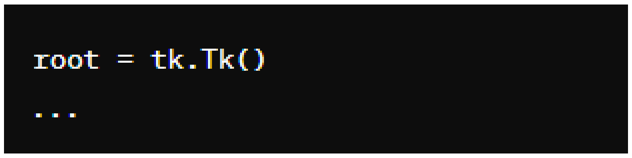
Function for Payment Process



```
def pay():
    ...
```

This function calculates the total payment, prompts the user for the payment amount, and displays the change message.

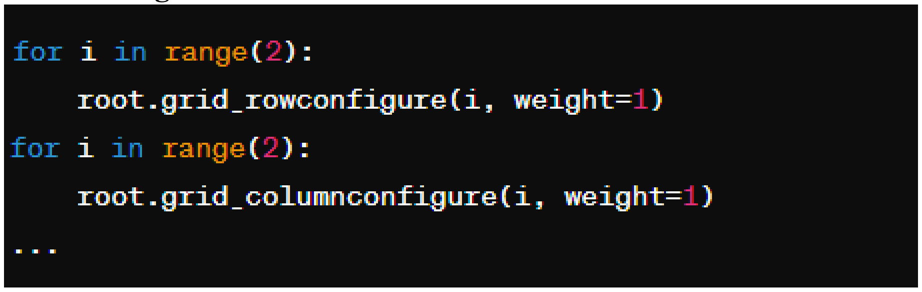
Creating the GUI



```
root = tk.Tk()
...
```

Creating the main window with the title "Game Store" and preparing GUI components such as Listbox, Treeview, Button, Entry, etc.

Grid Configuration and Frames



```
for i in range(2):
    root.grid_rowconfigure(i, weight=1)
for i in range(2):
    root.grid_columnconfigure(i, weight=1)
...
```

Setting grid configurations for the main window.

Console Frame and List

```
console_frame = ttk.LabelFrame(root, text="Consoles")
console_frame.grid(row=0, column=0, padx=10, pady=5, sticky=tk.W)

console_scroll = ttk.Scrollbar(console_frame, orient=tk.VERTICAL)
console_list = tk.Listbox(console_frame, yscrollcommand=console_scroll.set)
```

ttk.LabelFrame =>This creates a labeled frame to group related widgets.

ttk.Scrollbar =>This creates a scrollbar for the console list.

tk.Listbox =>This creates a listbox to display the available consoles.

yscrollcommand=console_scroll.set: This connects the scrollbar to the listbox for vertical scrolling.

Displaying Consoles

```
def display_console_list():
    consoles = set()
    for console in game_list:
        consoles.update(console.keys())

    console_list.delete(0, tk.END)
    for console in consoles:
        console_list.insert(tk.END, console)
```

display_console_list() => This function populates the listbox (console_list) with available consoles by iterating through game_list.

Binding Console Selection

```
console_list.bind("<<ListboxSelect>>", display_game_by_console)
```

This line binds the display_game_by_console function to the listbox selection event. When a console is selected, it triggers the display of games for that console.

Game Frame and Treeview

```
game_frame = ttk.LabelFrame(root, text="Games")
game_frame.grid(row=0, column=1, padx=10, pady=5, sticky=tk.W)

game_tree = ttk.Treeview(game_frame, columns=("Name", "Price", "Stock"), show="headir
```

ttk.LabelFrame => Another labeled frame for grouping game-related widgets.

ttk.Treeview => This creates a treeview widget to display games with columns for game name, price, and stock.

Displaying Games

```
def display_game_by_console(event=None):
    ...
```

display_game_by_console() => This function displays the list of games for the selected console in the game_tree treeview widget.

Menu Frame and Actions

```
menu_frame = ttk.LabelFrame(root, text="Actions")
menu_frame.grid(row=1, column=1, padx=10, pady=5, sticky=tk.W)
```

ttk.LabelFrame => A labeled frame for grouping action-related widgets.

Add, Edit, Remove, and Buy Game Frames

```
add_frame = ttk.LabelFrame(root, text="Add Game")
edit_frame = ttk.LabelFrame(root, text="Edit Game")
remove_frame = ttk.LabelFrame(root, text="Remove Game")
buy_frame = ttk.LabelFrame(root, text="Buy Game")
```

ttk.LabelFrame => These frames group widgets related to adding, editing, removing, and buying games.

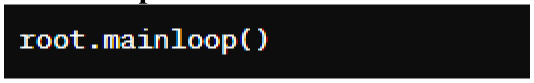
game_tree.bind("<<TreeviewSelect>>", on_game_select)

This line binds the on_game_select function to the treeview selection event. When a game is selected, it triggers actions related to that game.

pay_button = tk.Button(root, text="Pay", command=pay)

This button triggers the pay() function to handle the payment process for purchased games.

Main Loop



```
root.mainloop()
```

This starts the main event loop of the application, which listens for user interactions.

This code is a simple program that uses Tkinter to create a game store management application. The code has several main components and different functions. Here's the analysis:

- **Main Components**

Tkinter => The code uses the Tkinter module to create the GUI (Graphical User Interface) of the application.

Treeview => To display data in the form of a table.

Listbox => To display the list of consoles.

Combobox => To select the console name when adding a game.

Button => To execute specific functions like adding, editing, or deleting games.

Entry => To input text or numbers like game name, price, and stock quantity.

LabelFrame => To group specific widgets into one frame with a title.

Scrollbar => To provide scrolling functionality for specific widgets.

- **Main Functions**

display_console_list() => Displays the list of available consoles.

display_game_by_console(event=None) => Displays the list of games based on the selected console.

on_game_select(event=None) => Displays action options that can be performed on the selected game.

add_game() => Adds a new game to the list based on the selected console.

edit_game() => Edits the information of the selected game.

remove_game() => Removes a game from the list.

update_stock() => Updates the stock of the selected game.

display_bought_games() => Displays the list of purchased games.

buy_game() => The game purchasing process.

pay() => The payment process for purchased games.

show_frame(frame) => Displays the selected frame.

- **Application Flow**

1. Display the list of available consoles.
2. When selecting a console, display the list of games for that console.

- Game Interaction

Selecting a game will display options to buy, edit, or delete the game.

- Purchase Function

Allows users to buy games by entering the desired quantity.

After purchasing, the game will be displayed in the purchase list.

- Payment

Displays the total price for the purchased games.

Users are prompted to enter the payment amount.

If the payment is sufficient, display the change and clear the purchase list.