PyMOL360: Collaborative Gamepad Control of Molecular Visualization Software

Patrick V. Holec and Benjamin J. Hackel

Correspondence to: Benjamin J. Hackel (E-mail: *hackel@umn.edu*)

ABSTRACT

Interactive visualization of biomolecular structure is a powerful tool for scientists, students, and audiences from the laboratory to the classroom and beyond. However, existing platforms lack the ability to generate a collaborative environment to engage multiple users simultaneously. To address this need, PyMOL360 is described, which enables control of PyMOL molecular visualization software from multiple gamepad controllers. Extensive modularity enables user-defined configurations while default settings ease adoption by new users. A walkthrough of the software demonstrates its ability to select relevant molecules and modify viewing perspectives, molecular styles, colors, highlighted residues or chains. Provision of source code encourages use and further development.

Introduction

Structural biology has seen an exponential growth in the number of structures available through public portals, with the Protein Data Bank (PDB) surpassing 100,000 entries in 2015.[1] As studies across the field continue to fuel the expansion of these databases of structural data, the need for effective tools to interpret and communicate results becomes increasingly important. A number of molecular visualization tools – including PyMOL[2], UCSF Chimera[3], VMD[4], and JMol[5] – have filled this gap, providing a diverse set of frameworks to navigate biological structures in the standardized Protein Data Bank file format. While software like this can frame particular discussions or highlight features in either pictorial or video form, dynamic interaction with a structure is most frequently achieved through a keyboard and mouse. Alternative control schemes have been proposed such as smartphone controllers[6] and gaming controllers[7], but most of these technologies are intended to support only their particular niches and rarely emphasize truly cooperative environments. Herein, we present PyMOL360, software to enable collaborative control of molecular visualization software using gamepads traditionally designed for video game play.

Many molecular viewers are well-suited for functional extensions through the use of scripting. Namely, PyMOL, a leading molecular graphics system, supplements its capabilities by providing full control via Python. This connection to well-established languages opens the door for integration with thousands of existing modules. With this in mind, we can rapidly develop new applications for molecular viewers should a module already partially fill the necessary code architecture. One particular library of interest is Pygame, a module intended for game design in Python. Traditionally distanced from academic applications, this tool provides two critical features that extend to PyMOL: the ability to robustly connect to any number of gamepad devices, and the ability to easily create windows. This makes this package particularly appropriate for introducing new control devices and displaying information describing the connection between controller and software.

In order to advance the state of molecular visualization into interactive and multiuser experiences and facilitate broader usage

through familiar controllers, we have developed a plugin to PyMOL using the Pygame library to supplement multiuser sessions through gamepad support. This framework provides high user manipulability, presenting a broadly applicable extension to the communication of structural biology.

Methods

Interface Design

All scripts were developed in Python (v2.5). With the exception of the Pygame module, only default libraries were used to ease portability between systems. The only utilized non-standard Python module, Pygame, provides the integral relay between gamepad inputs and Python interpreter. Additionally, Pygame provides packages for window creation while operating out of an input driven standby mode. Normal PyMOL360 usage exchanges between two modes, governing user settings or PyMOL control. Both divisions utilize the window creations features included in the Pygame display module. Text generation, orientation and application was completed through a module provided through Pygame.

Control Interpretation

Upon execution of PyMOL360, any devices registered as gamepads on the machine will be detected and mapped as objects in the PyMOL instance using the Pygame controller module. Commands are classified into two categories: static inputs and dynamic inputs. Static inputs are only detected if the control mapped to that function is changed, such as a button being mapped to changing the color of the selection. Dynamic inputs constantly monitor the function they are mapped to and relay this state specific information to be executed. An example of this would be rotation being mapped to a joystick on the gamepad. In this way, the map between functions and controls detects events from any

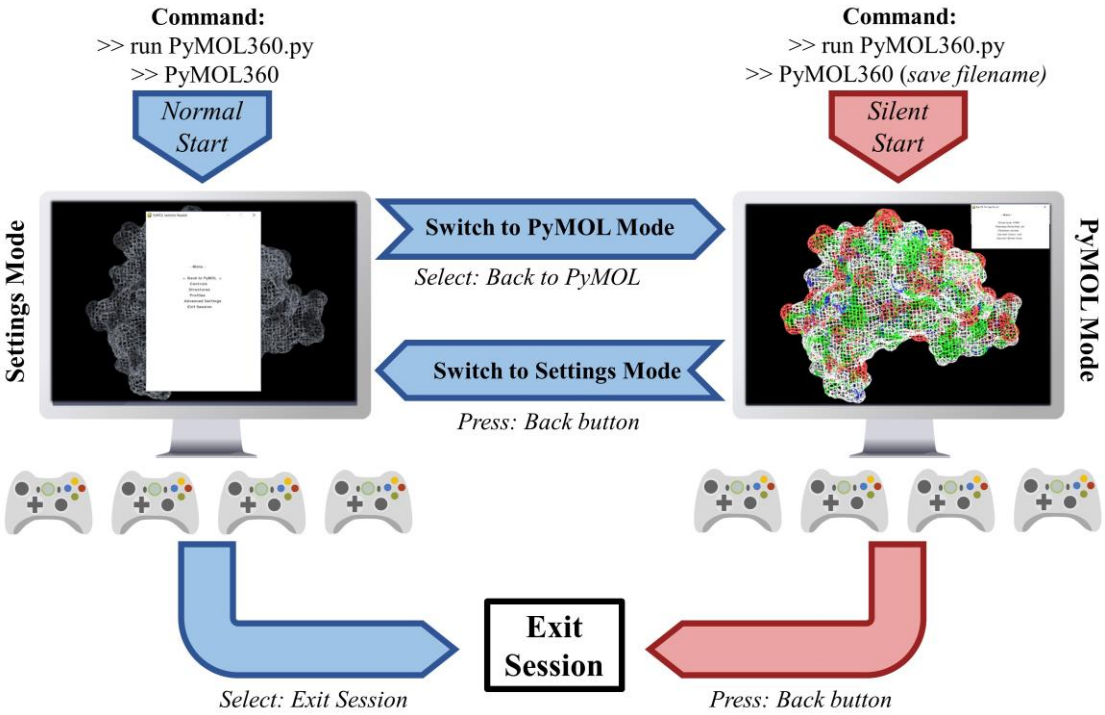

Figure 1. Overview of PyMOL360 usage. Normal start provides control over both the user settings mode and PyMOL usage mode, while the silent start bypasses user settings and directly starts in PyMOL mode.

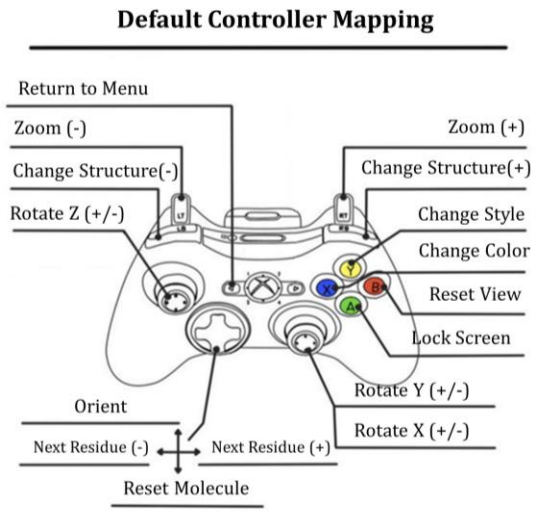

Figure 2. Demonstration of default mapping for an Xbox360 controller. All presented controls can be bound to new buttons, and any buttons can be mapped to new functions not included in the default package (see ReadMe.txt for more details on how to add new functions).

joystick object and converts them to PyMOL function (Figure 1). This mapping can be modified through the "Controls" section of the main menu.

Supporting Functions

Emphasis was placed in ensuring PyMOL360 included many supportive functions to extend utility to a wide range of potential applications. Through the settings menu, structures may be activated or deactivated based on the subset of PDB files relevant to the intended presentation. Cycled colors and representation types (*e.g.*, surface, stick, and cartoon) can be controlled similarly. If a particular representation is not included on the default list, custom settings can be appended via manual name declaration. Additionally, we acknowledge the inability to effectively map the full collection of PyMOL formatting functions that may be required to compose a presentation-ready image from a raw PDB file. To address this, current PyMOL states and control maps can be saved through the profile menu. This allows users to format proteins prior to initialization and save configurations for quick access in future use and readily change profiles throughout an implementation. Additionally, the interface can be bypassed for quick implementation through the use of the silent call feature for rapid deployment.

Compatibility

Scripts are compatible with all current versions of educational PyMOL 1.3 for Windows. There are no known backwards compatibility issues for previous PyMOL versions. Installation requires the inclusion of the Pygame module into the site packages of PyMOL. If PyMOL runs out of a copy of Python not registered in the directory, users can follow installation instructions found in the package README file. Otherwise, Pygame can be installed using the pip install functions (see Python documentation).

Implementation

A detailed walkthrough of the software (Supporting Information) illustrates some of the capabilities of PyMOL360 by exploring the active site of hen egg-white lysozyme enzyme. Briefly, the walkthrough demonstrates opening relevant PDB files, viewing the molecule(s) from multiple perspectives (rotation, zoom, and translation),

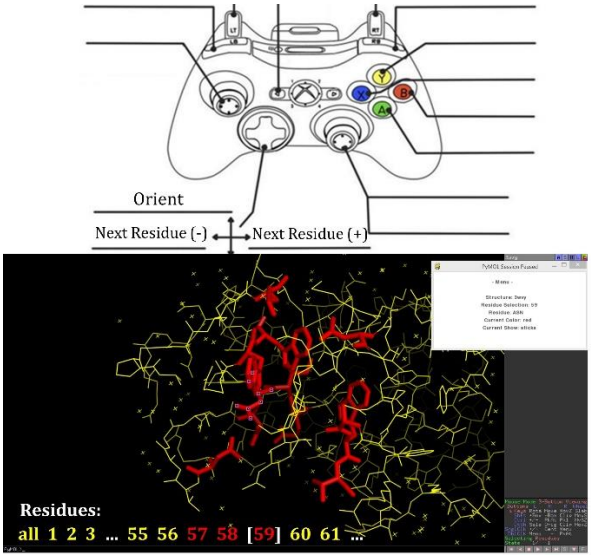

Figure 3. Selecting and highlighting residues (by color and perspective) in PyMOL360.

viewing various molecular styles (*e.g.* stick, cartoon, sphere, and surface), modifying color, highlighting particular molecular regions (residues or chains; Figure 2) including color and perspective, as well as toggling across multiple molecules. Moreover, it discusses the ability to modify control configurations and expand functionality.

Conclusions

Tools for scientific communication are crucial to the structural biology community. PyMOL360 was designed for extensive modularity to fit the users' needs while remaining accessible for uninitiated users. Thus, the extension of PyMOL described here has the ability to facilitate multiuser discussions regarding protein structure as well as provide an interactive basis for those new to structural biology. A software walkthrough and source code are freely provided to encourage use and further development.

Acknowledgments

This work was supported by the University of Minnesota.

Keywords: PyMOL, collaboration, gamepad, molecular visualization

Additional Supporting Information, including a software walkthrough and software code, may be found in the online version of this article.

References and Notes

(1) Rose, P. W.; Bi, C.; Bluhm, W. F.; Christie, C. H.; Dimitropoulos, D.; Dutta, S.; Green, R. K.; Goodsell, D. S.; Prlic, A.; Quesada, M.; Quinn, G. B.; Ramos, A. G.; Westbrook, J. D.; Young, J.; Zardecki, C.; Berman, H. M.; Bourne, P. E. *Nucleic Acids Res.* **2013**, *41* (D1), 475–482.

(2) DeLano, W. L. In *CCP4 Newsletter on Protein Crystallography*; 2002; pp 44–53.

(3) Pettersen, E. F.; Goddard, T. D.; Huang, C. C.; Couch, G. S.; Greenblatt, D. M.; Meng, E. C.; Ferrin, T. E. *J. Comput. Chem.* **2004**, *25* (13), 1605–1612.

(4) Humphrey, W.; Dalke, A.; Schulten, K. *J. Mol. Graph.* **1996**, *14* (1), 33–38.

(5) Hanson, R. M. *J. Appl. Crystallogr.* **2010**, *43* (5 PART 2), 1250–1260.

(6) Yiu, C.-P. B.; Chen, Y. W. *Methods Mol. Biol.* **2014**, No. 1091, 343–352.

(7) Moritz, E.; Hagen, H.; Wischgoll, T.; Meyer, J.; Acm. *Vrst 2007 Acm Symp. Virtual Real. Softw. Technol. Proc.* **2007**, *1* (212), 243–244.

GRAPHICAL ABSTRACT

Patrick V. Holec and Benjamin J. Hackel

PyMOL360: Collaborative Gamepad Control of Molecular Visualization Software

PyMOL360 is Python-based software that provides multi-user control of PyMOL molecular visualization software using video game controllers. The code is written to facilitate easy adoption by uninitiated users while providing modularity for advanced scientists.