

SOFTWARE METAPAPER

pyObs: Open-Source Software for Computer-Assisted Behavioral Observation Coding

William A. Griffin^{1,2}

¹ Center for Social Dynamics and Complexity, Arizona State University, Tempe 85287-2701, Arizona, US

² The Sanford School of Social and Family Dynamics, Arizona State University, Tempe, Arizona, US

william.griffin@asu.edu

pyObs is a computer-assisted behavioral observation coding system used to record social interaction among children in classroom and playground settings. Its flexible architecture allows it to be useful for capturing behaviors occurring in settings that vary across age group or context. Written in Python, pyObs has a cross-platform GUI using drop down menus and check boxes along with audio and visual cues that increase coder reliability. Time-stamped GIS maps record subject location. And data are output into standard formats, such as csv. pyObs is extensible, making it simple to change the interface depending on the research objectives.

Keywords: observational coding; behavior; ethology; classroom; playground; children; autism; Python; wxPython

Funding statement: This material is based upon work supported by the National Science Foundation under Grant Nos.: 0338864 & 0324208.

(1) Overview

Introduction

Computer-assisted behavioral observation systems

With the advent of hand held computer devices, behavioral observation coding evolved from checklists, clipboards and stopwatches to slick GUIs with real-time data storage. Used by ethologists in the field, educators in the classroom, and behavioral scientists across diverse environments including hospitals, homes, and playgrounds these contemporary codings systems have the ability to cover more behaviors with greater specificity and higher reliability [1].

Existing coding systems are either one of two types: open-source or proprietary. Within each type, the scope of the program ranges from a specific topic area addressing a narrow research question to a more general omnibus system, built to cover a wide range of behavioral areas. Proprietary systems tend to be well designed, more often general in their scope, and often expensive. For example, the Noldus Observer XT [2] is typically used to observe human interaction across disparate settings, ranging from parent-child interaction [3] to patient-doctor conferences [4].

Conversely, there are ample, typically open-source, one-off computer programs that address a narrow band of research objectives focusing on specific populations. These are well suited for pilot studies, limited budget projects, or

niche research situations. Because of their limited focus, these programs tend to be simple, built with accessible code, and have minimal bloat, either in the code or user interface (UI). pyObs is in this latter genre – built to capture time-stamped behavior, affect, and physical movement in preschool children, this software was a critical component in a multi-year NSF funded project, entitled Peers Across Landscapes (PALs), that sought to describe the genesis of friendships in 3–5 year olds (Li & Griffin, 2013; Torrens & Griffin, 2012).

Implementation and architecture

Although pyObs typifies software of the type described above (i.e., specialized needs for a specific project), from the beginning, the design focused on encompassing the critical characteristics associated with a general behavioral observation coding system (see [5] for a recent review). During its evolution, from conception to implementation, each feature, consistent with the design architecture, emphasized speed and accuracy with minimal error. Specifically, the following outline covers the design prescript used during piloting and guided minor changes over the life of the project:

1. Graphical User Interface:
 - Uncluttered User Interface (UI)
 - Logical menu arrangement and flow

- 2. Data Entry:
 - Immediate feedback on entry mistakes
 - Summary error checking
 - Data cannot be stored until all errors are cleared
- 3. Project Specific Needs:
 - Temporal cues for the coder
 - Clock with visual and audio feedback
 - Data are time-stamped with geospatial reference
 - X, y location
 - Farthest distance
- 4. Data output:
 - Local storage on the hard drive
 - Standard format (e.g., csv)
 - Human readable
 - Configurable for immediate statistical analysis and database storage
- 5. Extensible:
 - Scheme and UI can be changed as project needs change
 - New codes can be added with minimal programming
 - Allow rapid reconfiguration when subject and geospatial features change

team of undergraduates using tablet PCs collected the data; to reduce the likelihood entry mistakes, we focused on making the flow of the Graphical User Interface (GUI) intuitive. Likewise, data entry had to be simple with robust error checking methods.

During the PALs study, assessing interrater reliability commenced from the initial piloting until data collection ended, 2.5 years and 178,565 observations later; Kappa scores ranged from .70 to .99 for all coding structures. Details of the behavioral coding associated with the PALs study are elsewhere [7–8]. The general scheme was as follows, while watching a randomly chosen child for 10 s, the observer coded the child’s behavior in a naturalistic setting, recording both social and geographic information. Specifically, a coder recorded whether the child was alone, with a teacher, playing with other children, or passively engaged in group behavior through parallel play – children are playing in proximity to each other, but not with each other.

After determining the appropriate code, observers next recorded the associated affect: positive, negative, or neutral while in the presence of social peers – those involved in direct interaction, or area peers – those in the physical vicinity but not interacting with the target child. The specificity of the data, along with its breath, enabled us to analyze how activity type and physical location commingled to influence micro-social processes among preschool children. See **Figure 1** for a schematic of this procedure.

Structure

pyObs captures naturally-occurring free-play interactions among preschool children [6]. Over the 3 years of the PALs study, from piloting to implementation, the emphasis was on coder accuracy, error-less data entry, and speed. A large

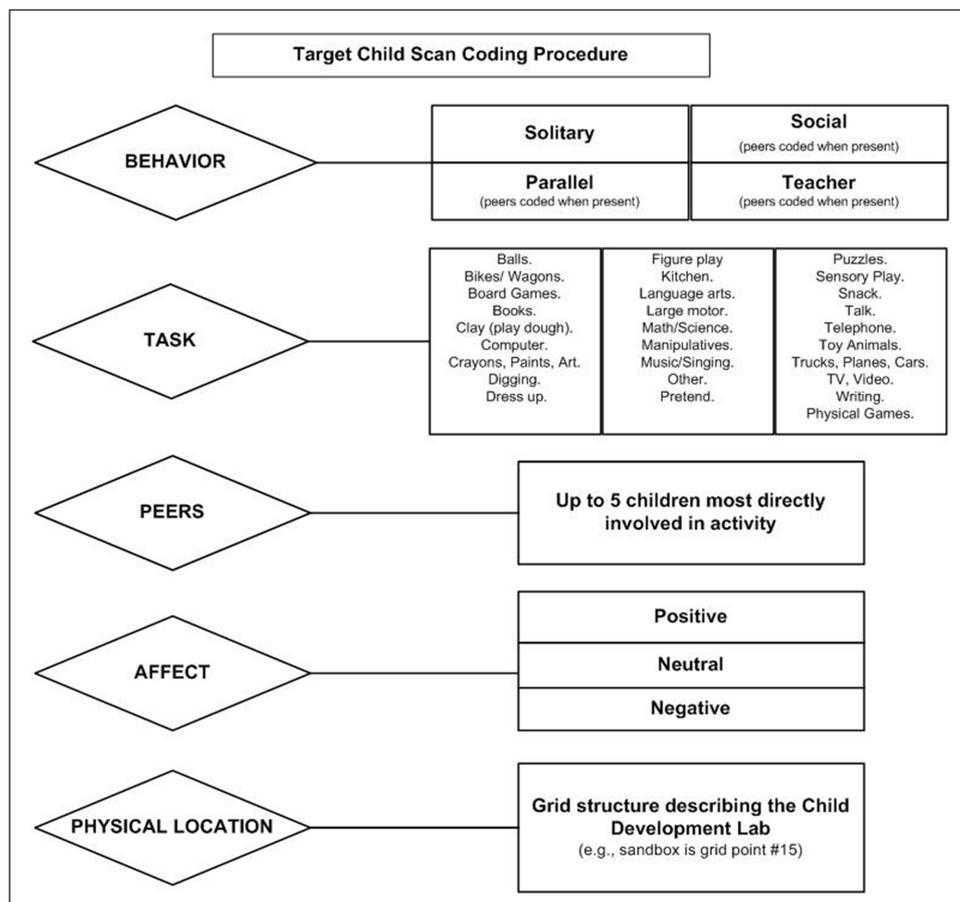


Figure 1: pyObs Target Child Scan for PALs.

Data Flow

As shown in **Figure 1**, pyObs has 5 primary components relating to the target child: Behavior, Task, Peers, Affect, and Physical Location. Within pyObs' architecture, assuming the selected child is present and available, relevant interconnected menu boxes require completion before submission for data storage. **Figure 2** illustrates the GUI data entry flow. Elaboration of the study and the coding procedure is available elsewhere [7–8].

GUI

Using pyObs requires minimal computer programming skills. As shown in **Figure 3**, data gets entered via drop down menus or check boxes. pyObs populates the selection options using either external or internal lists, depending on researcher preference. Output goes to the local directory unless directed elsewhere via an external configuration file. Data are saved as simple ascii text in csv format; in turn, these output are easily converted to a pandas DataFrame, a SQL database or a spreadsheet for subsequent manipulation and statistical analysis.

Depending on specific operating platform requirements, it may be necessary to make minor adjustments (e.g., directory dependences, color schemes) but the code should transfer without modification. Python programming etiquette encourages small modules, each reflecting a different function; conversely, pyObs is a single module. Because the PALs project required frequent updating across hand-held devices (e.g., Windows based tablets with a digitizer pen), to minimize the likelihood of introducing errors, we decided in the development phase that a large single module was preferable to multiple smaller

modules. Although not ideal from the perspective of a programmer, this strategy ensures rapid and error free updating across machines. pyObs' GUI uses wxPython [9], a wrapper for the cross-platform GUI API wxWidgets, written in C++. Coders used the digitizer pen to move through the appropriate sequence of drop down menus and check boxes (see **Figure 3**). Instead of a digitizer pen, current technology allows data entry using a touchscreen.

Geospatial Reference

Actuate observation of behavior in naturalistic settings require context; pyObs does this by embedding into the GUI a gridded bitmap image (BMP) file to assist geocoding of the target child's location. Each relevant geographic area shows on the GUI as a cartographic representation of the environment, with cues for location and relative distances between structures and objects in the mapped area. Grids on the BMP map help the observer specify location; grid size, of course, can vary in size to best meet the research requirements.

In the PALs study, for example, we used 3 ft grids on maps of the outside and classroom environments (see **Figure 3**). Embedded images are either raw BMP files or first converted to python code with the wxPython img2py tool; using the latter method decreases loading time.

To enhance geospatial accuracy, and increase reliability, a tap on the image generates an exploding map. **Figure 4** shows an example of enlarged classroom map. Tapping a location on the enlarged map returns a time-stamped x, y coordinate reference, along with a visual confirmation (see **Figure 5**).

In the PALs implementation, we used three different geodata points: the start location (a [x, y, t] point in

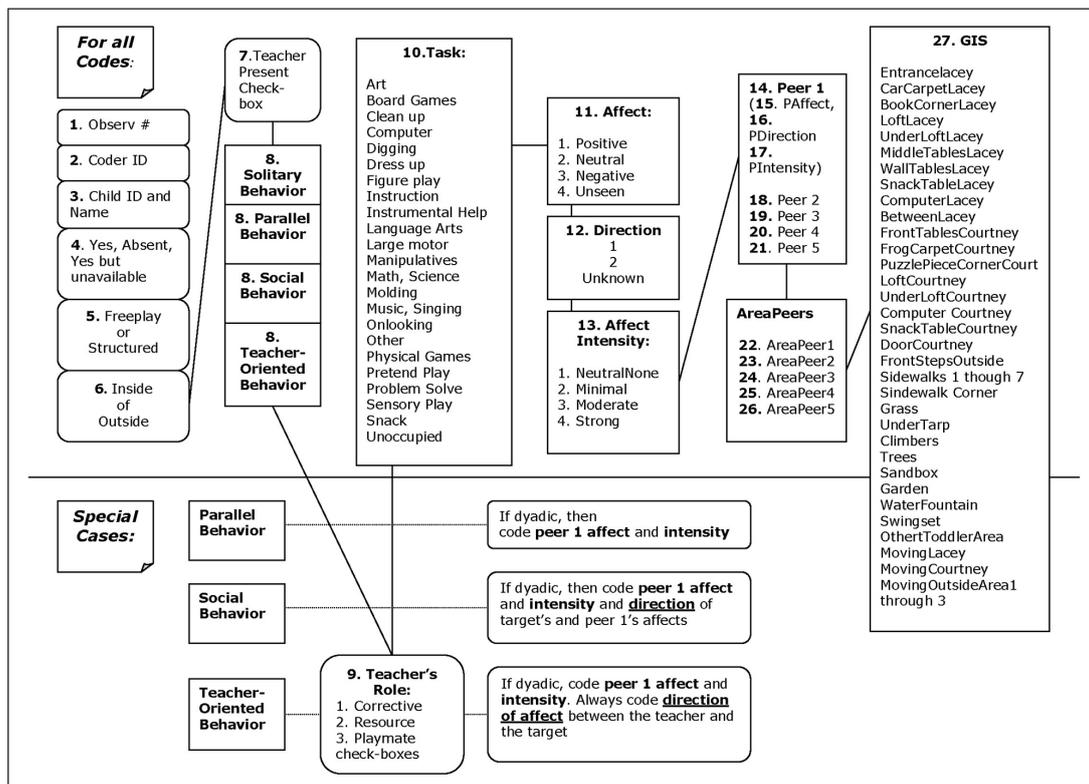


Figure 2: pyObs Flow Chart of Data Entry for PALs.

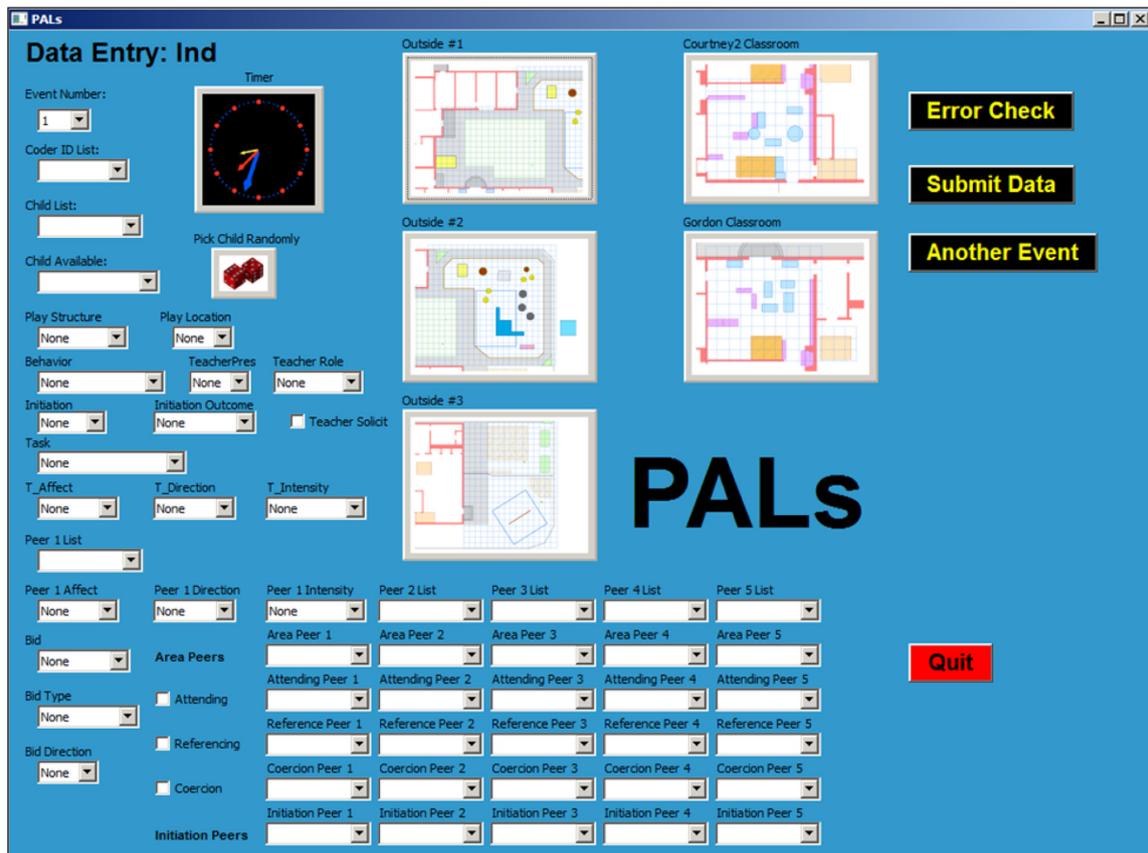


Figure 3: pyObs User Interface for the PALs project (Windows Platform).

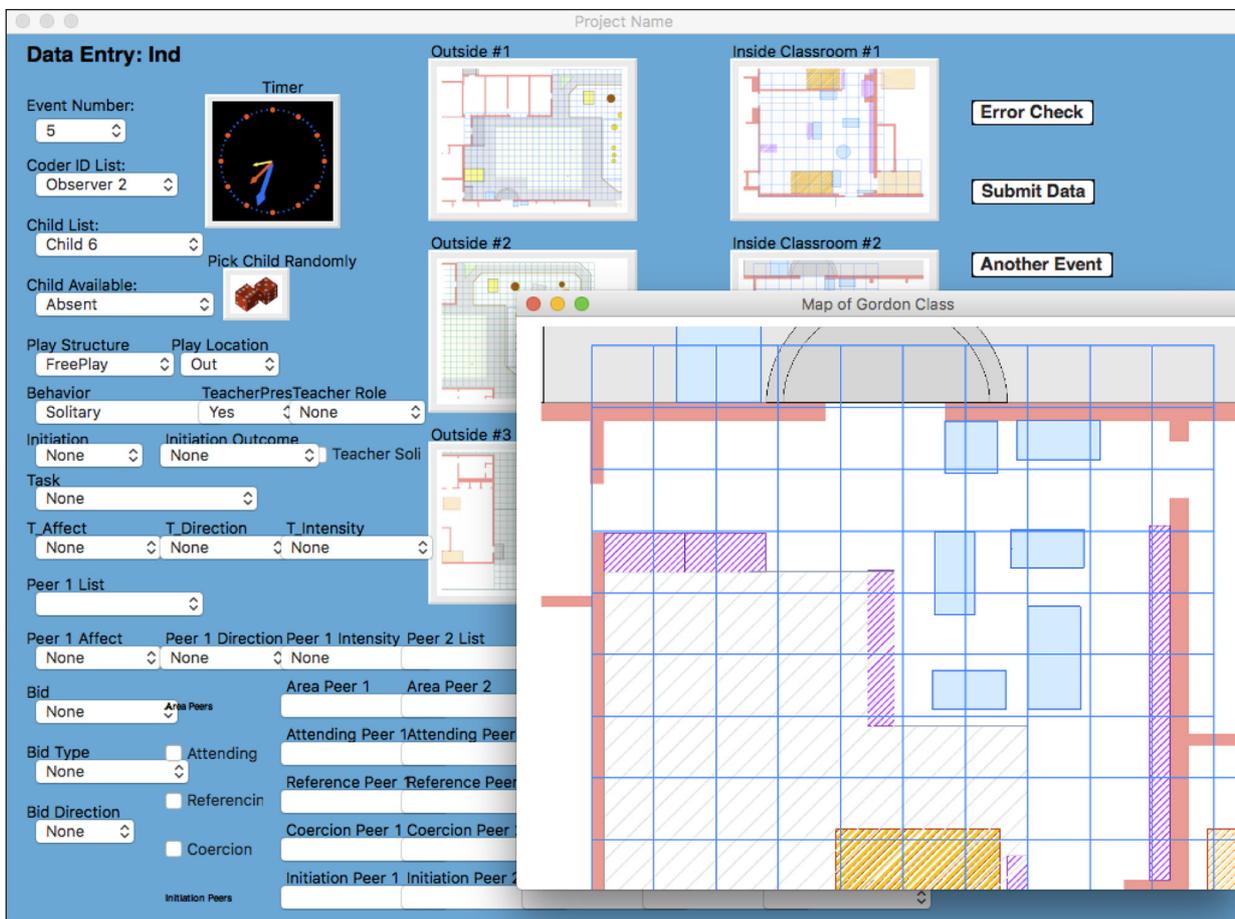


Figure 4: Exploding map ready of coder input (OS X).

space and time), stop location, and farthest distance traveled. Including the farthest distance allowed us to trace the typical amount of movement within the 10 s window.

Variants

As pyObs evolved, novel research opportunities arose, resulting in two variants that capture selected micro-features of social interaction among children. Building a typical variant of pyObs, such as those described below, requires modifying small sections of original code. Because pyObs was built as an omnibus behavioral observation coding system, with its myriad components, changes made to the pyObs code usually are done by either removing or commenting out irrelevant functions and, if needed, modifying the appropriate GUI boxes locations and titles. This takes minimal time and programmer experience.

Additionally, depending on the researcher's preference, populating the choices shown in the GUI are done using either an internal list or an external text configuration file. Although in pyObs the child's range of activities are hard coded in the python module it is sometimes easier to use an external file. For example, the following function, taken from the VisualAttention variant described below, uses the activityList.txt file. Contained within the file is a list showing the range of possible child activities (e.g., nap,

playgroup). To modify the range, the researcher simply alters the list.

```
def GetActivityList (self):
    activityList = []
    f = open ('activityList.txt')
    line = f.readline ()
    while (len (line)>0):
        line = line.strip ().split (',')
        if len (line) == 1:
            activityList.append ("")
        else:
            activityList.append (line[1])
        line = f.readline ()
    f.close ()
    return activityList
```

1. Visual Attention

One variation of pyObs was the foundation coding system used to capture visual attention among preschoolers [10–11]. To gather these data, observers watched a target child for a 6 s interval and recorded the identity of peers receiving a unit of visual regard, along with affect, from the target; visual regard, defined as the orientation of head and eyes toward the peer recipient, measures social engagement.

Figure 6 shows the GUI for this variant. Note that the interface has no GIS map, instead the observer identifies

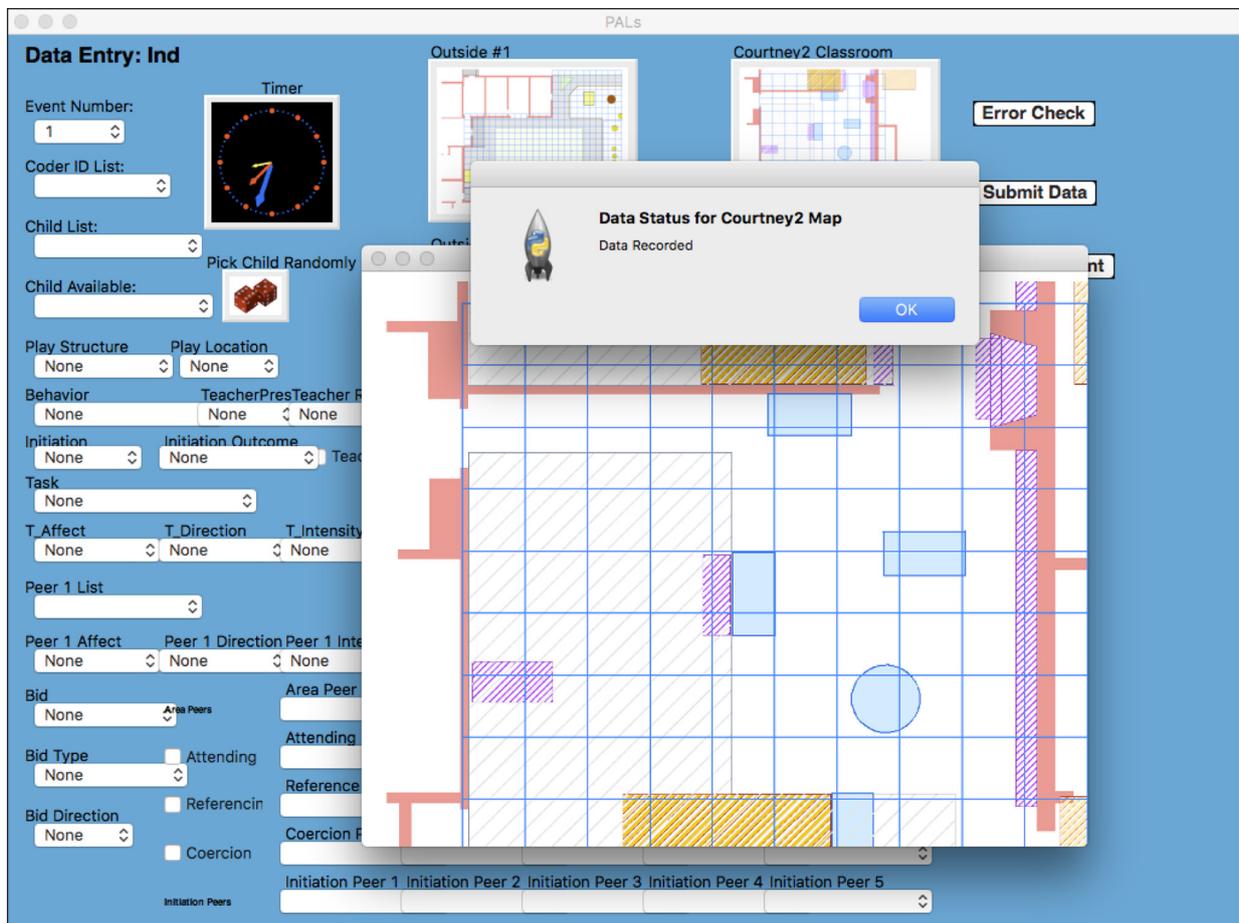


Figure 5: Exploding map with entry confirmation (OS X).

Visual Attention

Data: 30/05/17 Observer: Other 1

Class: 103 Round#(today): 1 # Children present: 2

Site: room Activity setting: Snack

child2 (not present) Current observation #: 1

Visual Attention To: VR Code: Affect:

Visual Target 1: child3 N/A Glance Look N/A Positive Neutral Negative

Visual Target 2: N/A Glance Look N/A Positive Neutral Negative

Visual Target 3: N/A Glance Look N/A Positive Neutral Negative

Visual Target 4: N/A Glance Look N/A Positive Neutral Negative

6.0 seconds timer:

Continue Recording Save Record Discard Record

Note: You are to observe the target child for a period 6-sec. Note the identifier of EACH person (peer or adult) to whom the observed child directs visual regard. Units of VR are coded as "G" if the child looks briefly (<2-sec) at a peer or adult and coded as "L" if the child looks for a longer (>2-sec).

current time: 15:02:23

Figure 6: Visual Attention (OS X).

the location of the interaction using a box for classroom, and then a site within the room.

2. Affect Expression

In a more complicated GUI, we developed an interface that covered a larger space – the ground floor of a building with separate classrooms (see **Figure 7**). This code configuration captures a target child's affect expression during an interaction with another child while engaged in a specific activity; the observation occurs during a 15 s window of time [10–11]. With this interface, the observer codes the initiator and target of the interaction plus the observed affective valence (positive, neutral and negative). Like the example shown in **Figure 5**, each section of the map explodes to enhance coder accuracy.

Quality control

pyObs runs on operating systems that support Python. It has been tested on Linux, Mac OS X, and Windows.

(2) Availability

Operating system

pyObs runs on operating systems that support Python; for example, the Anaconda Python distribution by Continuum Analytics. It has been tested on Linux (Ubuntu, Mint); Mac OS X, and Windows (XP, 7, 10).

Programming language

Python 2.7; pyObs has not been tested for Python 3.5+ compatibility using wxPython 4.0.

Additional system requirements

Requires input method: mouse, digitizer pen, or touchscreen – depending on the operating system.

Dependencies

wxPython 3.0+; it has not been tested with Python 3.5 using recently release wxPython 4.0; it should work with minimal code changes. wxPython wraps the wxWidgets C++ toolkit and provides access to the user interface portions of the wxWidgets API. Numpy.

List of contributors

William Griffin, Xun Li

Software location

Archive

Name: Zenodo

Persistent identifier: <https://doi.org/10.5281/zenodo.804072>

Licence: MIT

Publisher: William Griffin

Version published: 2.2.1

Date published: 07/06/17

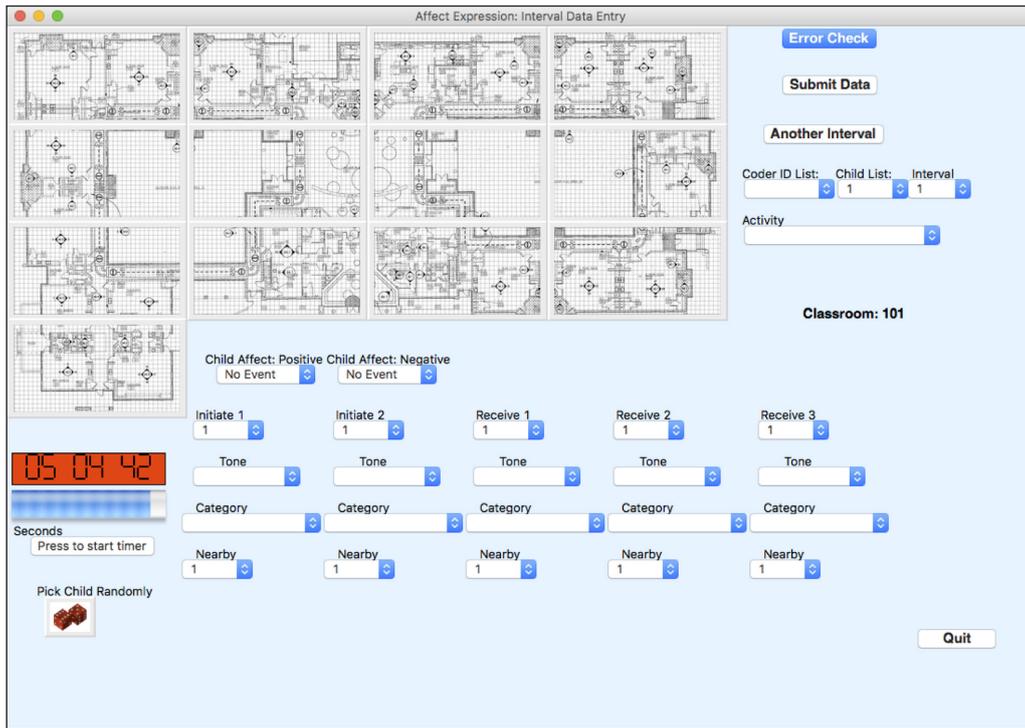


Figure 7: Affect Expression (OS X).

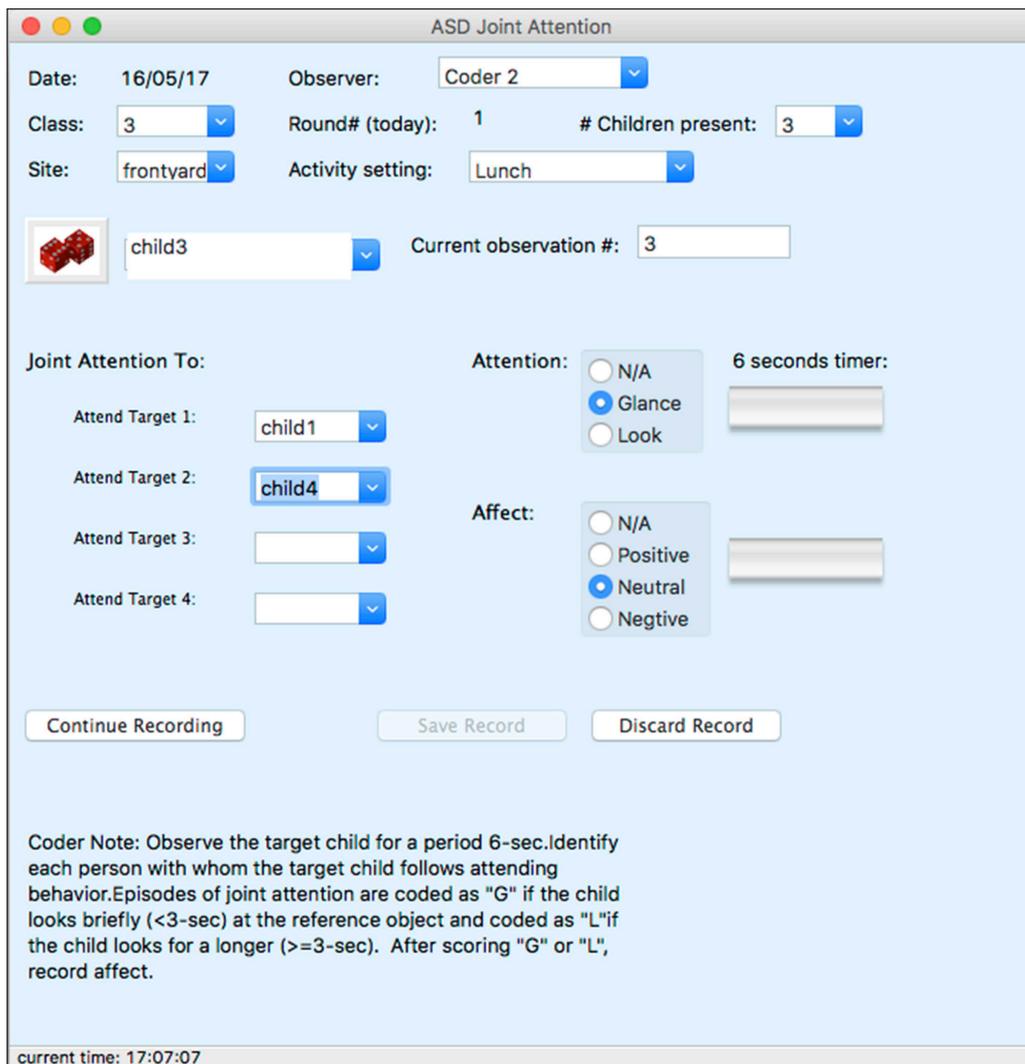


Figure 8: Modified pyObs showing a relevant ASD joint attention collection GUI (OS X).

Code repository**Name:** Github**Persistent identifier:** <https://github.com/billgriffin/pyObs>**Licence:** MIT**Date published:** 07/06/17**Language**

English

(3) Reuse potential

As shown in the Variants section of this paper, pyObs adapts to the needs of the researcher. At its core, pyObs is a behavioral observation toolkit – all critical components are available, leaving the investigator to assemble them according to the research question. This requires some minor computer programming skills. The bulk of the assemblage is deciding which menus to retain, appropriate labeling through internal or external lists, setting timers and audio cues, installing maps if needed, and modifying data output locations. After making these decisions, the researcher organizes the codes, places appropriately labeled boxes on the GUI, and tests the error checking algorithm.

Illustration

For at least a decade scholarly publications have reported an increase in autism, or more formally, the diagnosis of Autism Spectrum Disorder (ASD) [12]. Recent reviews suggest that observation data is critical for accurate diagnosis and evaluating treatment outcome [13–14]. After reviewing this literature, to illustrate the versatility pyObs, I took the basic pyObs code structure, borrowed the interface from the Visual Attention variant, and modified it to capture joint attention – the coordinating and sharing of attention between an object and another person – an assumed critical component in identifying autism [15–17]. Individuals with ASD lack the aptitude for responding to gestural or physical cues that enhance social coordination.

With a few small computer code changes, I generated a simple ASD observation coding program, see **Figure 8**. This joint attention interface illustrates that pyObs contains the key components needed to build a specific observational coding instrument; it captures the target individual, the coder, the location, time references, the observed behavior, and data storage.

Competing Interests

The author has no competing interests to declare.

References

1. **Gravlee, C, Zenk, S, Woods, S, Rowe, Z and Schulz, A** 2006 Handheld Computers for Direct Observation of the Social and Physical Environment. *Field Methods*, 18(4): 382–397. DOI: <https://doi.org/10.1177/1525822X06293067>
2. **Noldus Observer XT** 2017 <http://www.noldus.com/human-behavior-research/products/the-observer-xt>.
3. **Smith, J, Dishion, T, Shaw, D, Wilson, M, Winter, C and Patterson, G** 2014 Coercive family process and early-onset conduct problems from age 2 to school entry. *Development and Psychopathology*, 26(4pt1): 917–932. DOI: <https://doi.org/10.1017/S0954579414000169>
4. **Smets, E, Hillen, M, Douma, K, Stalpers, L, Koning, C and de Haes, H** 2013 Does being informed and feeling informed affect patients' trust in their radiation oncologist? *Patient Education and Counseling*, 90(3): 330–337. DOI: <https://doi.org/10.1016/j.pec.2012.05.001>
5. **Chorney, J, McMurtry, C, Chambers, C and Bakeman, R** 2014 Developing and Modifying Behavioral Coding Schemes in Pediatric Psychology: A Practical Guide. *Journal of Pediatric Psychology*, 40(1): 154–164. DOI: <https://doi.org/10.1093/jpepsy/jsu099>
6. **Griffin, W A, Schmidt, S K, Nara, A, Torrens, P M, Fewell, J H and Sechler, C** 2007 Integrating ABM and GIS to model typologies of playgroup dynamics in preschool children. In: North, M, Macal, C and Sallach, D (Eds.), *Proceedings of Agent 2007: Conference on Complex Interaction and Social Emergence, Chicago, IL, 7–9 October*. 17–24. <http://www.agent2007.anl.gov/>. Evanston, IL: Argonne National Labs and the University of Chicago.
7. **Li, X and Griffin, W A** 2013 Using ESDA with Social Weights to Analyze Spatial and Social Patterns of Preschool Children's Behavior. *Applied Geography*, 43: 67–80. DOI: <https://doi.org/10.1016/j.apgeog.2013.06.003>
8. **Torrens, P M and Griffin, W A** 2012 Exploring the micro-social geography of children's interactions in preschool: a long-term observational study and analysis using Geographic Information Technologies. *Environment & Behavior*, 45(5): 584–614. DOI: <https://doi.org/10.1177/0013916512438885>
9. **wxPython** 2017 <https://wxpython.org/>.
10. **Santos, A, Peceguina, I, Daniel, J, Shin, N and Vaughn, B** 2012 Social Competence in Preschool Children: Replication of Results and Clarification of a Hierarchical Measurement Model. *Social Development*, 22(1): 163–179. DOI: <https://doi.org/10.1111/sode.12007>
11. **Shin, N, Kim, M, Goetz, S and Vaughn, B** 2013 Dyadic Analyses of Preschool-aged Children's Friendships: Convergence and Differences between Friendship Classifications from Peer Sociometric Data and Teacher's Reports. *Social Development*, 23(1): 178–195. DOI: <https://doi.org/10.1111/sode.12043>
12. **Masi, A, DeMayo, M, Glozier, N and Guastella, A** 2017 An Overview of Autism Spectrum Disorder, Heterogeneity and Treatment Options. *Neuroscience Bulletin*, 33(2): 183–193. DOI: <https://doi.org/10.1007/s12264-017-0100-y>
13. **Parsons, L, Cordier, R, Munro, N, Joosten, A and Speyer, R** 2017 A systematic review of pragmatic language interventions for children with autism

- spectrum disorder. *PLoS ONE*, 12(4): e0172242. DOI: <https://doi.org/10.1371/journal.pone.0172242>
14. **Sam, A, Reszka, S, Odom, S, Hume, K and Boyd, B** 2015 The Use of Coding Methods to Estimate the Social Behavior Directed toward Peers and Adults of Preschoolers with ASD in TEACCH, LEAP, and Eclectic “BAU” Classrooms. *Behavioral Disorders*, 40(2): 91–101. DOI: <https://doi.org/10.17988/BD-13-47.1>
 15. **Hooker, J** 2013 *Observing Joint Attention in Children with Autism: The Development of a Video Coding Protocol*. Undergraduate Honors Theses. Paper 383. Available online at: http://scholar.colorado.edu/honr_theses.
 16. **Dawson, G, Toth, K, Abbott, R, Osterling, J, Munson, J, Estes, A and Liaw, J** 2004 Early Social Attention Impairments in Autism: Social Orienting, Joint Attention, and Attention to Distress. *Developmental Psychology*, 40(2): 271–283. DOI: <https://doi.org/10.1037/0012-1649.40.2.271>
 17. **Leekam, S and Ramsden, C** 2006 Dyadic Orienting and Joint Attention in Preschool Children with Autism. *Journal of Autism and Developmental Disorders*, 36(2): 185–197. DOI: <https://doi.org/10.1007/s10803-005-0054-1>

How to cite this article: Griffin, W A 2018 pyObs: Open-Source Software for Computer-Assisted Behavioral Observation Coding. *Journal of Open Research Software*, 6: 5. DOI: <https://doi.org/10.5334/jors.183>

Submitted: 11 June 2017 **Accepted:** 21 November 2017 **Published:** 26 January 2018

Copyright: © 2018 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/4.0/>.