Democratizing Psychophysiological Research: Leveraging Open Resources for Experimental Settings and Data Analysis of Flow, Engagement, and Performance.

Elena Sajno, Samantha Chen, Federico Dagostin,
Stefano De Gaspari, Chiara Pupillo and Giuseppe Riva

July 27, 2024

Democratizing Psychophysiological Research: Leveraging Open Resources for Experimental Settings and Data Analysis of Flow, Engagement, and Performance.

Elena SAJNO [a,b,c,f,g], Samantha CHEN [c] Federico DAGOSTIN [d]
Stefano DE GASPARI [a,b] Chiara PUPILLO [a,b] and Giuseppe RIVA [b,e,h]
[a] *University of Pisa, Pisa, Italy*
[b] *HTLab, Università Cattolica del Sacro Cuore, Milan, Italy*
[c] *MIT MediaLab, Massachusetts Institute of Technology, Cambridge, USA*
[d] *BMI Labs "Mario Stefanelli", University of Pavia, Italy*
[e] *ATNPLab, IRCCS Istituto Auxologico Italiano, Milan, Italy*
[f] *Corresponding Author*
[g] *Special thanks to Matteo "Dalfo" Dalfonso*
[h] *Research partially supported by the Italian Ministry of Health*
ORCiD ID: Elena Sajno https://orcid.org/0000-0002-9621-8981, Samantha Chen https://orcid.org/0000-0003-1159-0467, Federico Dagostin https://orcid.org/0009000694066343, Stefano De Gaspari https://orcid.org/0009-0006-6083-2134, Chiara Pupillo https://orcid.org/0009-0007-5764-1211, Giuseppe Riva https://orcid.org/0000-0003-3657-106X

Abstract. Contemporary research faces challenges due to economic constraints and a lack of specialized software. However, advancements such as affordable wearable sensors and collaborative platforms like GitHub are revolutionizing this landscape. This work proposes the development of an experimental setup that integrates game stimuli presentation and synchronized recording of facial movements and cardiac data, facilitating resource sharing and collaboration. The objective of this framework is to elicit and detect states of flow and engagement along with associated physiological activations. Using different difficulty levels of a Tetris-style game as stimuli, the framework synchronizes players' activity with Interbeat interval (IBI) data from the Polar H10 cardiac chest band and webcam videos. Key to the approach is precise data synchronization using timestamps and established libraries like LabStreamingLayer (LSL). Three Python scripts manage Tetris presentation, Polar H10 Bluetooth connection for cardiac data collection, and video frame tagging for facial expression analysis. These scripts adhere to FAIR principles and provide a practical implementation for studying psychophysiological responses. By leveraging open resources, this framework addresses economic and software limitations and allows for adaptation to various research questions or theoretical constructs.

Keywords. flow, video game, open source, ECG, emotions, FAIR

1. Introduction

Contemporary research often grapples with economic constraints and a shortage of specialized and updated software tailored to specific research needs, or cheap and reliable hardware. The landscape is, however, evolving: the emergence of affordable wearable sensors, increasingly often considered the gold standard for research due to their reliability and the high sampling frequency is a resource for the researcher in psychology. Meanwhile, platforms such as GitHub, Hugging Face, PhysioNet, and Kaggle facilitate the democratization of science by enabling the sharing of scripts and datasets, thereby fostering collaboration and accelerating progress. Yet, assembling resources for studies, particularly those integrating psychophysiology, remains complex, especially for the crucial need for synchronized data. Within this context, this work proposes the development of an experimental setup stemming from shared resources, that allows the presentation of game stimuli and the synchronized recording of facial movements and cardiac data, with the aim of creating a new sharable instrument. The full code is available on GitHub at the address: https://github.com/elenasajno/TetrisInterbeats .

Objectives: The experimental goal for which this framework is designed is to elicit and detect states of flow and engagement and the physiological activations associated with them, as described in [1,2,3]. As stimuli, several levels of different difficulty of a Tetris-style video game are employed; the player's activity needs to be saved and synchronized with data coming from the cardiac wearable device Polar H10 and videos received from the webcam. The final framework can also be adapted to other research questions or theoretical constructs.

Why using videogames: Video games are seen as effective tools for evoking various specific emotions [4], which can be measured through psychophysiological signals [5,6]. Additionally, different difficulty levels within the same video game, such as Tetris, have been found to provoke distinct mental states: for instance, varying levels of difficulty in tasks can evoke feelings of boredom, anxiety, and engagement [4]. Video games that offer various levels, similar in structure but different in difficulty, enable a more tailored match to the abilities of each participant [4,7]. Specifically, Tetris was used for studies inherent in the states of flow [8,9] and engagement [4,11].

Cardiac Psychophisiology: Psychophysiological signals, such as the cardiac one, offer insight into the sympathetic and parasympathetic activities, emotions, and mental activation [12,13]. Moreover, the use of wearable devices to keep track of ECG/BVP signals allows a comfortable for the participants' data collection, and research which can be set also outside the laboratories. Specific features calculated from heart beating are also particularly informative: Hearth Rate Variability (HRV), for example, can be decomposed into frequencies, each one more related to distinct activation of the autonomic nervous system or mental/affective state [15]. HRV values are usually computed by selecting the R peaks in ECG, and calculating the time between an R peak and the following [15]; to reach a "gold standard" HRV an at least 250Hz ECG is required [14]. Another effective way to derive HRV from Interbeat Intervals (IBIs), that already measure the millisecond from a beat and the following [15]: in this work, we propose to extract directly the IBIs, calculated by the Polar H10 band in milliseconds (so at 1000Hz), which is reported as a reliable instrument for the R-R detection [22]. Since the recording of a baseline is necessary for good experimental design involving HRV [16], we have included a specific function for selecting the chosen duration for a baseline acquisition, timestamping its start and end moments.

Facial expressions & Emotions: Since Ekman's studies [17] facial expressions, through microexpressions, are one of the more direct ways to understand which kind of emotion is perceived by a person. Attention [18] and mental workload [19] can also be inferred through eye movement parameters. Software such as Noldus' FaceReader, GazeRecorder, or other AI video applications can detect from videos of a face, the microexpressions, and ocular movements linked to emotions, affect, and level of attention. Recording the participant's facial activity can be easily achieved with a laptop camera: we propose a code for this function, putting in overlay the timestamp of each frame.

Need for Synchronization: The synchronization of signals from different sensors is a crucial step in signal pre-processing, particularly in the design of multimodal affective computing experiments [20]. Typically, synchronization is achieved by collecting markers when stimuli are presented [21]. This process allows for precise alignment of data streams, enabling the investigation of the temporal relationships between stimuli presentation and physiological responses. The objective of this work is to provide scripts to create a complete experimental setting, in which each section can be synchronized with the others: these Python 3 codes are meant to be run from the terminal on the same machine to ensure local timing consistency. Key moments of stimulus presentation will be recorded, video acquisition will be timestamped for each frame, and each IBIs will be recorded with its corresponding timestamps. We have decided to use the LSL timestamp, which calculates the timestamp from the machine's boot time down to the microsecond, the machine's proper tick, and the Unix Time Stamp, which records down to the microsecond referring to the reference timestamp (from which the precise date and time of the recording can be extracted). This approach ensures that it is possible to trace back the various moments of the data collection and that they can be synchronized to the millisecond.

2. Framework Design

Key to our approach is the punctual synchronization of data streams, through precise timestamps and the utilization of established libraries, such as LSL. By recording and storing essential metadata alongside physiological and behavioral data, our framework enables the reconstruction of temporal sequences, crucial for in-depth analysis and interpretation. Drawing from shared codes on GitHub, our framework is divided into three Python 3 scripts, all intended to run on the same machine, ensuring consistency in local timestamp management. 1) The Tetris component allows the presentation of the stimuli and the collection of gaming performances and timestamps for various in-game events; it is also possible to insert a countdown of desired length to record an initial physiological baseline. 2) The PolarH10 module enables the Bluetooth connection with the chestband, plus the detection and saving of cardiac IBIs with a 1000 ms precision for a reliable HRV calculations [22]. Additionally, the following synchronization with game-activities and facial expressions enables the assessment of both the resting baseline state and the reactions to specific stimuli/events. 3) The video component ensures that each frame is tagged with multiple timestamps, facilitating the synchronization to perform analysis through facial expression recognition softwares. All three scripts allow for including the subject number (*subject_number = input("Enter the subject number: ")*), task number (*task_number = input("Enter the task number: ")*), and date of data collection (*date_str = input("Enter the date (YYYY-MM-DD): ")*) calling before the main function. Those data are used to save a file named after them (shaped like *filename = f"tetris_data_{date_str}_subject{subject_number}_task{task_number}.csv*). In the same function are defined the rules for writing the .csv file, defining multiple types of timestamps (LSL, tick, and Unix's: *timestamp = datetime.datetime.now().timestamp()*; *tick = lib.GetTickCount64()*; *local_time = pylsl.local_clock()*), that are reported for each saved data.

Tetris Component: The Tetris framework was started forking tetris.py from https://github.com/ Datamine/Tetris, a custom Tetris-style game based on pygame. A countdown function was defined, to allow a custom-length baseline recording, that can be selected before the main function (*t = input("Enter baseline length (in seconds): ")*):

```
def countdown(t):
    print('start_baseline')
    write_to_csv("start_baseline")
    while t:
        mins, secs = divmod(t, 60)
        timer = '{:02d}:{:02d}'.format(mins, secs)
        print(timer, end="\r")
        time.sleep(1)
        t -= 1
    print('end_baseline')
    write_to_csv("end_baseline")
```

The commands *print(text)* and *write_to_csv(text)* were added in multiple positions in the script, in the functions defining the gameover, the pause, the resume, the proper gaming phases (the start of the new level and which level was selected), when a complete line is achieved (and if it is single one or more), the exit to menu, and the actual start of the application. For example, the first part of the function defining the game is here reported:

```
def game(screen,startinglevel):
    print("inizio livello")
    write_to_csv("inizio livello")
    cleared = 0
    tetrimino = newtetrimino()
    bestscore = int(getmaxlines())
    board = [['']*20 for n in range(10)]
    background = pygame.image.load("Grid.PNG")
    backgroundcolor = getrgb(gridline)
    timestep = time.time()
    bottom = pygame.font.Font('BebasNeue.ttf',20)
    white = getrgb("#FFFFFF")
    print("starting level", startinglevel)
    write_to_csv("starting level:"+ str(startinglevel))
```

Polar H10 IBI Component: The project started from https://github.com/markspan/ PolarBand2lsl/blob/main/Polar2LSL.py . It allows connecting a PolarH10 chest band to the machine running the Python code, receiving the ECG signal (130Hz), and streaming it in a way compatible with softwares able to detect LSL streams. The device connection is established using its Bluetooth Low Energy (BLE) framework through the Bleak library, which utilizes Predefined UUIDs (Universal Unique Identifiers) - based on the Heart Rate GATT (Generic Attribute Profile) service protocol - to access its services. This allows for obtaining a targeted response input from the device, essentially serving as an API. The code establishes a connection to the device, retrieves device information (such as model number, manufacturer name, and battery level), and starts the data stream. Here an extract is reported:

```
import asyncio
```

```
from bleak import BleakClient
uuid16_dict = {v: k for k, v in uuid16_dict.items()}
# UUIDs for device information
MODEL_NBR_UUID = "0000{0:x}-0000-1000-8000-00805f9b34fb"
    .format(uuid16_dict.get("Model Number"))
MANUFACTURER_NAME_UUID = "0000{0:x}-0000-1000-8000-00805f9b34fb"
    .format(uuid16_dict.get("Manufacturer Name"))
BATTERY_LEVEL_UUID = "0000{0:x}-0000-1000-8000-00805f9b34fb"
    .format(uuid16_dict.get("Battery Level"))
# UUIDs for connection and data streaming
PMD_SERVICE = "FB005C80-02E7-F387-1CAD-8ACD2D8DF0C8"
PMD_CONTROL = "FB005C81-02E7-F387-1CAD-8ACD2D8DF0C8"
PMD_DATA = "FB005C82-02E7-F387-1CAD-8ACD2D8DF0C8"
PMD_CHAR1_UUID = "fb005c81-02e7-f387-1cad-8acd2d8df0c8"
PMD_CHAR2_UUID = "fb005c82-02e7-f387-1cad-8acd2d8df0c8"
U1_CHAR2_UUID = "6217ff4d-91bb-91d0-7e2a-7cd3bda8a1f3"
```

Due the need to obtain data with a higher sampling rate, to follow HRV guidelines [14], we decided to retrieve the IBIs signal. To achieve this, we modified the script with information coming from https://github.com/kbre93/dont-hold-your-breath/blob/master/PolarH10.py. The UUID for the RR interval is UUID 00002a37-0000-1000-8000-00805f9b34fb [23]) and, the raw signal retrieved from the Polar H10 needs to be converted from byte to float [23]. The various UUIDS information was added to our script, alongside the conversion parameters:

```
IBI_WRITE = bytearray([0x02, 0x00, 0x00, 0x01, 0x37, 0x00, 0x01,
   0x01, 0x0E, 0x00])
ibi_SAMPLING_FREQ = 1000
HEART_RATE_MEASUREMENT_UUID = "00002a37-0000-1000-8000-00805f9b34fb"
byte0 = data[0] # heart rate format
uint8_format = (byte0 & 1) == 0
energy_expenditure = ((byte0 >> 3) & 1) == 1
rr_interval = ((byte0 >> 4) & 1) == 1
if not rr_interval: return
first_rr_byte = 2
    if uint8_format:
        hr = data[1]
        pass
    else:
        hr = (data[2] << 8) | data[1] # uint16
        first_rr_byte += 1
    if energy_expenditure:
        ee = (data[first_rr_byte + 1] << 8) | data[first_rr_byte]
        first_rr_byte += 2
    for i in range(first_rr_byte, len(data), 2):
        ibi = (data[i + 1] << 8) | data[i]
        ibi = np.ceil(ibi / 1024 * 1000)
        ibi_stream_values.extend([ibi])
        ibi_stream_times.extend([time.time_ns()/1.0e9])
```

In the same function in which the IBIs are calculated, the corresponding timestamps are retrieved, and each IBI and its related timestamps (through *local_time = pylsl.local_clock()* e *current_datetime = datetime.datetime.now()*) are written to a row of a CSV file (nominated as already described).

Video Acquisition Component: The Video acquisition module started forking https://github.com/maksimKorzh/open-cv-tutorials/blob/main/src/record-webcam/webcam.py based on the cv2 library, which permits to display and save videos recorded through the webcam. To ensure that the video can be aligned with the rest of the data streams, we chose to record the video by superimposing on each frame the timestamp at which they were recorded. To achieve this, the libraries and functions for displaying the various timestamps were added, as in the previous modules, and a script that superimposes the various timestamps *(timestamp = time.time()*, *current_time = datetime.datetime.now()*, *tick = lib.GetTickCount64()*, *local_time = pylsl.local_clock()*, *tick = int(str(tick))* in a precise position, each frame, before saving the video was created.

```
superimposes
cv2.putText(frame, timestamp_str+" "+tick_str, (10, 30),
    cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 0, 255), 2)
cv2.putText(frame, formatted_time_with_ms+" "+local_time_str, (10, 50),
    cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 0, 255), 2)
video.write(frame)
```

3. Conclusions

The software here presented provides a practical implementation of shared resources for the elicitation of flow and engagement. It is coupled with the synchronized collection of cardiac data (IBIs) and facial movements for the study of psychophysiological responses by means of a PolarH10 band and a webcam. Feelings of boredom, anxiety and engagement were provoked by tweaking the difficulty settings within the game of Tetris. The code is divided into three separate scripts: the game component, the PolarH10 module and the video recording component, in part adapted from other open source software projects available on GitHub. According to FAIR [10](Findable, Accessible, Interoperable, Reusable) principles, the scripts have been bundled into a GitHub repository (https://github.com/elenasajno/TetrisInterbeats) and are available along with all subsequent versions and evolutions of the toolset, under GPLv3 license terms to all parties, to replicate results or adapt the developed framework to their own research needs. This degree of availability addresses some issues pertaining to replicability, in that it removes the variable of tools used to collect the data, allowing for a more direct comparison of the harvested values. In conclusion, leveraging open resources for experimental setups and data analysis not only addresses resource and software limitations but also promotes collaboration and accelerates scientific progress in psychophysiological research.

References

[1] Sajno E, Riva G. Follow the Flow: Artificial Intelligence and Machine Learning for Achieving Optimal Performance. Cyberpsychology, Behavior, and Social Networking. 2022 Jul;25(7):476-477.

[2] Sajno E, Beretta A, Novielli N, Riva G. Follow the Flow: A Perspective on the On-Line Detection of Flow Mental State through Machine Learning. In: Proceedings of the 2022 IEEE International Conference on Metrology for Extended Reality, Artificial Intelligence and Neural Engineering (MetroXRAINE); 2022 Oct 26. IEEE; p. 217-222.

[3] Sajno E. Affective Computing for Detecting Psychological Flow State: A Definition and Methodological Problem. In: Proceedings of the 2023 11th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW); 2023 Sep 10. IEEE; p. 1-5.

[4] Chanel G, Rebetez C, Bétrancourt M, Pun T. Emotion assessment from physiological signals for adaptation of game difficulty. IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans. 2011 Nov;41(6):1052-63.

[5] Kivikangas JM, et al. A review of the use of psychophysiological methods in game research. Journal of Gaming & Virtual Worlds. 2011;3(3):181-199.
[6] Granato M, Gadia D, Maggiorini D, Ripamonti LA. An empirical study of players' emotions in VR racing games based on a dataset of physiological data. Multimedia Tools and Applications. 2020 Dec;79(45):33657-33686.
[7] Landowska A, Wróbel MR. Affective reactions to playing digital games. In: Proceedings of the 8th International Conference on Human System Interaction (HSI); 2015 Jun. pp. 264-270.
[8] Keller J, Bless H. Flow and Regulatory Compatibility: An Experimental Approach to the Flow Model of Intrinsic Motivation. Personality and Social Psychology Bulletin. 2008 Feb;34(2):196-209. https://doi.org/10.1177/0146167207310026.
[9] Harmat L, de Manzano Ö, Theorell T, Högman L, Fischer H, Ullén F. Physiological correlates of the flow experience during computer game playing. International Journal of Psychophysiology. 2015;97(1):1-7. https://doi.org/10.1016/j.ijpsycho.2015.05.001.
[10] Wilkinson MD, Dumontier M, Aalbersberg IJ, Appleton G, Axton M, Baak A, Blomberg N, Boiten JW, da Silva Santos LB, Bourne PE, et al. The FAIR Guiding Principles for scientific data management and stewardship. Scientific data. 2016;3(1):1-9, doi: 10.1038/sdata.2016.18
[11] Chanel G, Rebetez C, Bétrancourt M, Pun T. Boredom, engagement and anxiety as indicators for adaptation to difficulty in games. In: Proceedings of the 12th International Conference on Entertainment and Media in the Ubiquitous Era. MindTrek '08. Tampere, Finland: Association for Computing Machinery; 2008. p. 13–17. https://doi.org/10.1145/1457199.1457203.
[12] Sajno E, Bartolotta S, Tuena C, Cipresso P, Pedroli E, Riva G. Machine learning in biosignals processing for mental health: A narrative review. Frontiers in Psychology. 2022 Jan 13;13:1066317. https://doi.org/10.3389/fpsyg.2022.1066317.
[13] Cacioppo JT, Berntson GG, Larsen JT, Poehlmann KM, Ito TA. The psychophysiology of emotion. In: Lewis M, Haviland JM, editors. Handbook of Emotions. New York, NY: Guilford Press; 2000.
[14] Malik M, Bigger JT, Camm AJ, Kleiger RE, Malliani A, Moss AJ, Schwartz PJ. Heart rate variability: Standards of measurement, physiological interpretation, and clinical use. European Heart Journal. 1996 Mar;17(3):354–381. https://doi.org/10.1093/oxfordjournals.eurheartj.a014868.
[15] Shaffer F, Ginsberg JP. An overview of heart rate variability metrics and norms. Frontiers in Public Health. 2017;5:258. doi: 10.3389/fpubh.2017.00258.
[16] Laborde S, Mosley E, Thayer JF. Heart Rate Variability and Cardiac Vagal Tone in Psychophysiological Research – Recommendations for Experiment Planning, Data Analysis, and Data Reporting. Frontiers in Psychology. 2017 Feb 20;8:213. https://doi.org/10.3389/fpsyg.2017.00213.
[17] Ekman P. An argument for basic emotions. Cognition & Emotion. 1992 May 1;6(3-4):169-200. https://doi.org/10.1080/02699939208411068
[18] Hoang Duc AK, Bays P, Husain M. Eye movements as a probe of attention. Progress in Brain Research. 2008;171:403-411. https://doi.org/10.1016/S0079-6123(08)00659-6
[19] de Greef T, Lafeber H, van Oostendorp H, Lindenberg J. Eye Movement as Indicators of Mental Workload to Trigger Adaptive Automation. In: Schmorrow DD, Estabrooke IV, Grootjen M, editors. Foundations of Augmented Cognition. Neuroergonomics and Operational Neuroscience. FAC 2009. Lecture Notes in Computer Science, vol 5638. Berlin, Heidelberg: Springer; 2009. https://doi.org/10.1007/978-3-642-02812-0_26
[20] Bota PJ, Wang C, Fred ALN, Da Silva HP. A review, current challenges, and future possibilities on emotion recognition using machine learning and physiological signals. IEEE Access. 2019;7:140990–141020. doi:10.1109/ACCESS.2019.2944001.
[21] Koelstra S, Muhl C, Soleymani M, Lee J-S, Yazdani A, Ebrahimi T, et al. DEAP: a database for emotion analysis using physiological signals. IEEE Transactions on Affective Computing. 2012;3:18–31. doi:10.1109/T-AFFC.2011.15.
[22] Gilgen-Ammann R, Schweizer T, Wyss T. RR interval signal quality of a heart rate monitor and an ECG holter at rest and during exercise. European Journal of Applied Physiology. 2019;119(7):1525–1532. doi:10.1007/s00421-019-04117-1.
[23] Pramukantoro, E.S. A Study of Continuous Heartbeat Monitoring System Using The Internet of Things Approach. PhD dissertation. Okayama University; September 2022.