

python 刺激提示

DelayedMatch

ver 1.0.2

1. 動作環境

- Windows 10 pro/home または、 Windows11 pro/home
- 非 s-mode (s-mode でないこと)
- 64bit OS
- Intel 64bit CPU
- メモリー 8 G–16Gbyte

2. 動作確認ハードウェア

- SurfaceGO (intel モデル 2022 購入)
- national instruments 製 USB-6211 ハードウェア
(ボタン制御 TTL は、Active-HI でコントロール)
- national instruments 製 USB-6501 ハードウェア 但し、TTL 入力のみ
(ボタン制御 TTL は、Active-HI でコントロール)

3. 関連ファイル

3.1. Psychopy3 本体

本システムでは、Psychopy が必要です、

現時点では(2022/09/20) では Psychopy 2022.2.1 を使用しています。

Psychopy で、”Builder”を使用して刺激提示プログラムを作る方法と”Colder”を利用する方法があります。本プログラムでは、”Colder”を使用しています。

尚、Psychopy についてはインターネット上の情報をご覧ください。

3.2. プログラム本体/python スクリプト

- PyBTTsRunOne.exe : ランチャープログラム本体
- BTTs16DMExtraStateDelayTTLLoop.py : python 刺激提示プログラム本体

3.3. python 刺激提示プログラム(PyBTTsDelayedMatchV16 フォルダ内)

- CommonV010011 : python ライブラリフォルダ 各種ライブラリが格納されている。
- BTTs16DMExtraStateDelayTTLLoop.py (プログラム本体 メインプログラム/刺激提示)
- bttsTrialDM.py (プログラム本体 トライアル本体)

3.4. python 刺激提示プログラム関連実行ファイル(PyBTTsParamList フォルダ内)

ランチャーから呼び出すプログラムの選択を行う時、このフォルダ内のファイルから選択します。

- BTTsParamDMV16-20220928a.py
良く使う変数はこのファイルを編集することでデフォルト値を変更できます。

3.5. フォルダ位置

各ファイルは、コンピュータの各ユーザーの'Documents'の『PyBTTs』に入っています。

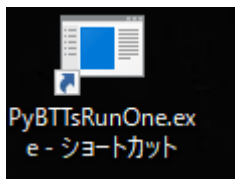
- 『bin』 ランチャープログラム本体(PyBTTsRunOne.exe)が入っています。
- 『Env』 各コンピュータ個別の設定を格納します。
- 『Run』 各『python 刺激提示プログラム』を格納するフォルダです。

4. タッチパネル関係の Windows 上での設定について

4.1. Windows10/11 でのタッチパネルによるマウスイベント

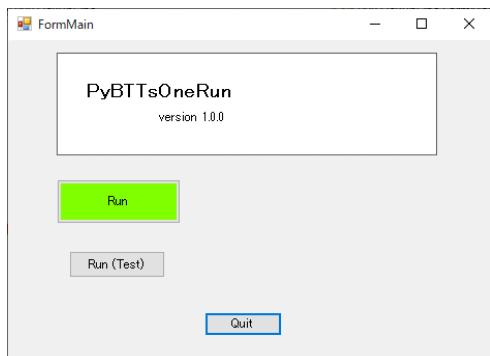
- Windows10 では、タッチパネルのイベントはマウスのイベントに変換されます。『タッチパネルの長押し』についてはタッチパネルの反応を遅くするため、Windows10 の設定で、『タッチパネル』『長押し』による右クリックは機能をオフにしてください。

5. 実行の流れ



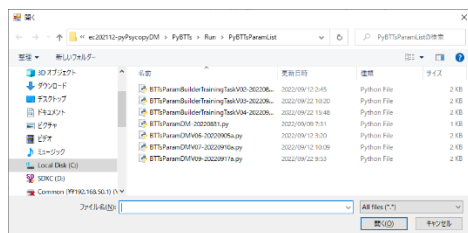
ランチャーソフトウェア

PyBTsRunOne を起動します。

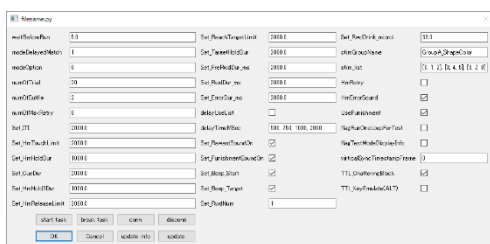


Run ボタン：通常の実験ではこちらをクリックします。

Run(Test)ボタン：動作を確認する場合はこちらをクリックします。ウィンドウが小さく開き、マウスカーソルもオンとなります。



パラメータリストから1つファイルを選択し実行します。

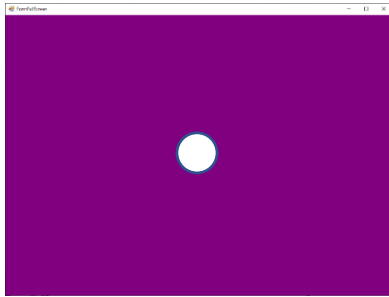


選択したタスクのパラメータ設定が表示されます。

『start task』 ボタンを押すとタスクが開始します。この時、『Wait Before Run』の項目の時間だけタスクの開始が遅れます。

この間に給餌装置本体のセットアップを終わらせます。

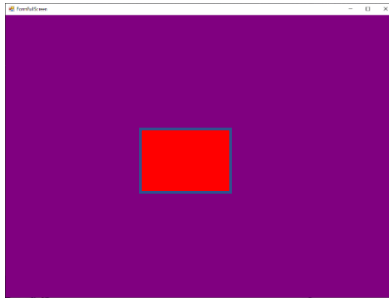
注意：実際にタスクが開始するまでソフトウェア起動に10数秒かかります。



[fixation-Phase]

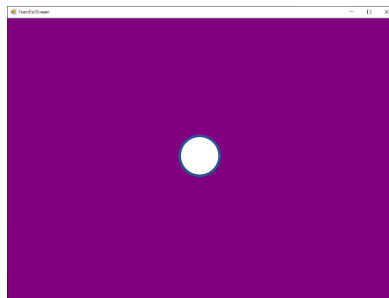
タスク用のウィンドウが開き、タスクが起動します。中央 fixation が表示されたら、ホームボタンを押します。

(ハードウェアボタンが無い場合はキーボード左側の ALT キーで代用することができます。)。ホームボタンは押し続ける必要があります。



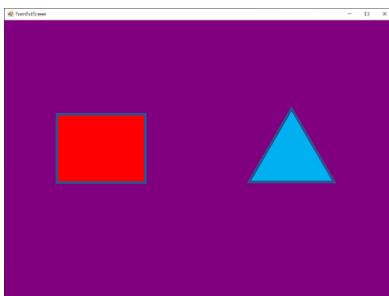
[Sample-Phase]

ホームボタンを押し続けると基本図形 Cue が表示されます。色または形あるいはその両方を記憶します（色・形のモード）。このままホームボタンを押し続けます



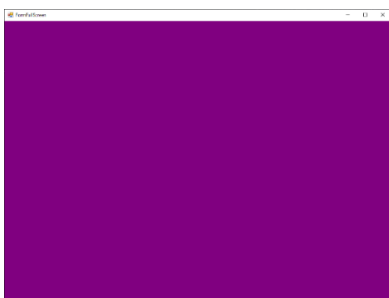
[Delayed-Phase]

Cue の表示が終わり、再び fixation が表示されます。指定時間分この状態が続きます。



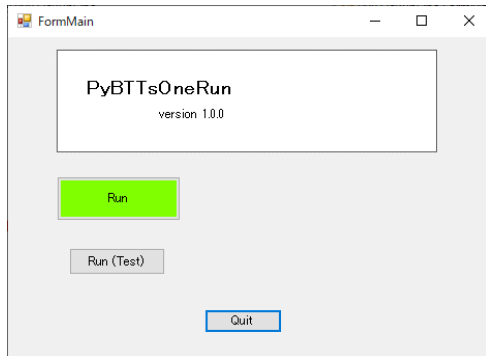
[Choice-Phase]

2つの図が表示されたらホームボタンを離します。記憶している形・色などを参考に2つの図形の内どちらかをタッチします。正しい図形を選択し、それをタッチしつづけた場合報酬が支給されます。





マウスの右クリックで、『右上端』『左下端』の順で3回繰り返すと終了します。



起動が終わりしたら、『Quit』で終了して下さい。

6. 環境設定ファイル

6.1. 環境設定ファイルの内容

パソコンにより異なる設定が必要な場合、『python 刺激提示プログラム』に埋め込むのは不便です。例えば給餌装置のシリアル通信ポート番号がパソコンにより異なります。環境設定ファイルはこのような場合パソコン内に記述することで『python 刺激提示プログラム』で利用できるようにしています。

6.2. 環境設定ファイル／フォルダ位置

各ユーザーの'Documents'の『PyBTs』『Env』フォルダ内に『envPyBTs.txt』ファイルがあります。

6.3. 設定内容

```
[default]
#screen
screen.background.color.rgb255.r=128
screen.background.color.rgb255.g=0
screen.background.color.rgb255.b=128
screen.background.color.r=0
screen.background.color.g=-1
screen.background.color.b=0

# feeder COM
feeder.device.comport=COM4
feeder.device.use=false
feeder.device.drink=false

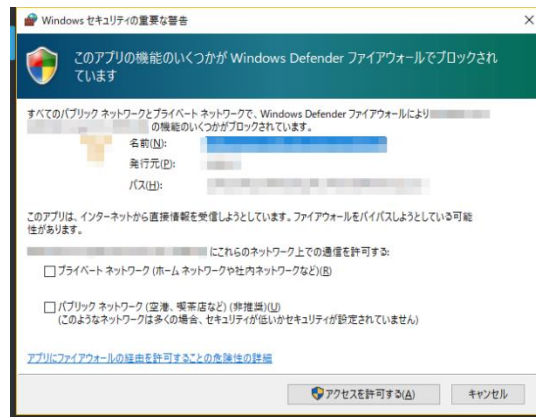
# DIO/Aout
hard.ttlout.1.device.use=false
hard.ttlout.1.device.devicename=dev1
hard.ttlout.1.device.nidaq.use=true
hard.ttlout.1.device.nidaq.ttlonly=false
```

- [default] このファイル内の設定の宣言
このファイルは、必ず [default]で始まります。
- コメントは行の頭に『#』印となります。
- screen.background.color.rgb255
バックグラウンドカラーを示します。RGB 各種 (0-255)の数字を指定します。
刺激提示の python スクリプトで色の指定が(0-255)の指定する場合はこちらを利用します。

- `screen.background.color.`
バックグラウンドカラーを示します。RGB 各種 (-1.0 から 1.0)までの数字を指定します。
刺激提示の python スクリプトで色の指定が(-1.0 から 1.0)を指定する場合こちらを利用します。
- `fedder.device.comport`
給餌装置のシリアル COMPORT を指定します。
- `fedder.device.use`
給餌装置が接続されていない場合 `false` を指定します。使用する場合は `true` にします。
- `fedder.deivce.drink`
- 給餌装置が drink 系の場合 `true` を指定します。そうでない場合 `false` を指定します。
- `hard.ttlout.1.device.use=false`
TTL 入力(ホームボタン)、Analog 出力ハードウェアを使用するかどうか。True の時は使用する。
- `hard.ttlout.1.device.devicename=dev1`
- ハードウェアのデバイスの指定。例えば'dev1'はデバイスの指定を表す。この内容は実装するハードウェアで変わります。この例では national instruments 製 USB-6211 ハードウェアでの例を示しています。
- `hard.ttlout.1.device.nidaq.use=true`
ハードウェア『nidaq』(national instruments 製)のデバイスの使用するかどうかを指定します。
`true` の時は使用する。
TTL 入力は NI デバイス'port0'の下位 2 ビット中の最下位 1 ビットを使用しています。
Analog 出力は NI デバイス'ao'のゼロチャンネル目(ao0)を使用しています。
- `hard.ttlout.1.device.nidaq.ttlonly=false`
ハードウェア『nidaq』(national instruments 製)が TTL のみサポートしている場合 `true` に設定。Analog 出力もサポートしている時は `false`。例えば、
national instruments 製 USB-6211 時は、`ttlonly=false`
national instruments 製 USB-6501 時は、`ttlonly=true`
と指定して下さい。

7. ネットワーク通信

PyBTTsRunOne.exe : ランチャープログラム本体と、BTTs16DMExtraStateDelayTTLLoop.py :
python 刺激提示プログラム本体 (トレーニングタスク) とは、ネットワーク通信を行っています。
このため初期起動時には以下のようなメッセージが出力されることがあります。



この場合、『プライベートネットワーク』にはチェックを入れ、『パブリックネットワーク』にはチェックを外してアクセス許可を行ってください。

8. サンプルプログラム『DelayedMatch』

8.1. サンプルプログラム『DelayedMatch』は給餌装置用のサンプルとなっています。

このプログラムは Psychopy の Coder を使用して作成されています。Psychopy の Coder の使い方については web サイト Psychopy.org などをご覧ください。

8.2. 『DelayedMatch』パラメータ

『DelayedMatch』は Psychopy を利用しているため大半の変数はプログラム内に埋め込まれています。しかし一部のパラメータは外部のダイアログからコントロールにすることができます。リモートシステムでは、これを利用し外部からコントロールします。

waitBeforeRun	5.0	Set_ReachTargetLimit	2000.0	Set_RwdDrink_micro	33.0
modeDelayedMatch	1	Set_TargetHoldDur	2000.0	stimGroupName	GroupA_ShapeColor
modeOption	0	Set_PreRwdDur_ms	2000.0	stim_list	[0, 1, 2], [3, 4, 5], [6, 7, 8]
numOfTrial	20	Set_RwdDur_ms	2000.0	HmRetry	<input type="checkbox"/>
numOfSuffle	2	Set_ErrorDur_ms	2000.0	HmErrorSound	<input checked="" type="checkbox"/>
numOfMaxRetry	0	delayUseList	<input type="checkbox"/>	UsePunishment	<input checked="" type="checkbox"/>
Set_JTI	2000.0	delayTimeMsec	500, 750, 1000, 2000	flagRunOneLoopForTest	<input type="checkbox"/>
Set_HmTouchLimit	2000.0	Set_RewardSoundOn	<input checked="" type="checkbox"/>	flagTestModeDisplayInfo	<input type="checkbox"/>
Set_HmHoldDur	1000.0	Set_PunishmentSoundOn	<input checked="" type="checkbox"/>	virtualSyncTimestampFrame	0
Set_CueDur	2000.0	Set_Beep_Start	<input checked="" type="checkbox"/>	TTL_ChatteringBlock	<input checked="" type="checkbox"/>
Set_HmHold2Dur	1000.0	Set_Beep_Target	<input checked="" type="checkbox"/>	TTL_KeyEmulate(ALT)	<input type="checkbox"/>
Set_HmReleaseLimit	2000.0	Set_RwdNum	1		

start task break task conn disconn
OK Cancel update info update

- waitBeforeRun : タスク起動前に指定時間秒を待たせます。給餌装置のシステムの設置には時間がかかるためこの時間を利用します。
- modeDelayedMatch: DelayedMatch のモードを指定します。
 - 0: 場所 ランダム
図形は同じものを表示します。場所が『DelayedMatch タスク』の対象となります。
 - 1: 図形 ランダム
ターゲットの図形がランダムとなります。図形が『DelayedMatch タスク』の対象となります。
 - 4: 場所 ランダム(正解場所場所固定)
図形は同じものを表示します。 場所が固定されます。場所の指定は modeOption に
modeOption: 0 の時、場所 0 で固定されます。
modeOption: 1 の時、場所 1 で固定されます。
 - 5: 図形ランダム (正解図形固定)
正解図形が固定されます。正解図形は、
- modeOption: DelayedMatch のモードのオプションを指定します。

- numOfTrial : トライアルの回数を指定します。
- numOfShuffle : タスクに対して刺激提示の条件をランダム化する時、出現回数を同一にするためシャッフルでランダム化します。例えば、色と形と場所の条件がある場合、
色(3色) 形(3つ) 場所(2)の時、正解不正解の組み合わせは、正解が9種類の図形(色3種、形3種)、不正解が8種類、場所が2種類となるため $9 \times 8 \times 2 = 144$ 通りとなります。これをランダム時シャッフルすることで出現回数を同一化します。
numOfShuffle は、このとき上記シャッフル時指定分だけかけてランダム化します。例えば上記例では、numOfShuffle = 2 となり、 $144 \times 2 = 288$ をシャッフルランダムします。
- numOfMaxRetry : 不正解の時、条件を変えずにリトライさせることができます。
numOfMaxRetry の値が 0(ゼロ)の場合不正解の時も次のトライアルでは新しい条件となります。もしも numOfMaxRetry が複数の値をとるとその回数分不正解が続いてもトライアルは間違ったときの条件を繰り返します。
- Set_ITI : タスク開始前期間の設定時間(msec)
- Set_HmTouchLmit(ms): ホームボタンにタッチしなければならない制限時間(msec)
- Set_HmHoldDur(ms): ホームボタンにタッチし続ける期間の設定時間(msec)
- Set_CueDur(ms): 手がかり刺激提示期間の設定時間(msec)
- Set_HmHold2Dur(ms): 遅延期間の設定時間(msec)
- Set_HmReleaseLmit(ms) ホームボタンから前足を離さなければならない制限時間(msec)
- Set_ReachTargetLimit(ms): ホームボタンから前足を離してから、ターゲットにタッチしなければならない制限時間(msec)
- Set_RwdDur(ms): 報酬時間の設定時間(msec)
- delayUseList: デイレイ時間のランダム化。Set_HmHold2Dur(ms)の代わりにこのリストの値を使ってデイレイ時間をランダム化することができます。ランダム化したい場合はこのチェックボックスをチェックします。
- delayTimeMSec: ランダム化するデイレイ時間の指定。この値でシャッフルランダム化します。
デフォルト値は、500, 750, 1000, 1250, 1500, 1750, 2000 ms
- Set_RewardSoundOn: リワード時に音を鳴らします。
- Set_PunishmentSoundOn: パニッシュメント時に音を鳴らします。
- Set_Beep_Start: トライアルの最初で音を鳴らします。
- Set_Beep_Target: トライアルのターゲット表示時に音を鳴らします。
- Set_RwdNum: 一回コレクトした時のリワードの回数を設定します。
- Set_RwdDrink_microl: 給餌装置がドリンク時、1回当たりのドリンクの量を指定します。デフォルトは 33 マイクロリットル
- stimGroupNmae: 刺激画像のグループ名の指定：
プログラム内に以下の3つのグループが定義されています。各グループの画像はゼロから始まる番号が付けられます。

GroupA_ShapeColor : 色(red,green,blue), 形(circle, rectangle, Cross)の組み合わせ

0: circle/red	1: circle/green	2: circle/blue
3: rectangle/red	4: rectangle/green	5: rectangle/blue
6: cross/red	7: cross/green	8: cross/blue

GroupB_ShapeColor: 色(red,green,blue), 形が(triangle, Inverted triangle, Star)の組み合わせ

0: triangle /red	1: triangle /green	2: triangle /blue
3: inv-triangle /red	4: inv-triangle /green	5: inv-triangle /blue
6: Star /red	7: Star /green	8: Star /blue

GroupA_image_signal: PNG 形式のファイルで作成された図形。

0: imgStart.png	1: imgStartY.png	2: imgStartRed.png
3: imgStopBlue.png	4: imgStopY.png	5: imgStop.png

尚、GroupA_image_signal は、6 つまでですのでご注意ください。

- stim_list: 図形を指定します。使用する番号は刺激画像のグループ名で異なります。例えば GroupA_ShapeColor の場合番号は、0 から 8 までの数字となります。動作するモードによって list での指定を工夫することで方向性を持たせることができます。
これについては、最終章の Appendix A をご覧下さい。
- HmRetry: Cue が表示される前、あるいは Cue の表示時、リトライとして繰り返す時指定を行います。トライアルの条件を更新されません。
- flagRunOneLoopForTest: 普段はこのスイッチは Off のままご使用下さい。条件を変える時このチェックボックスにチェックを入れると繰り返しループさせることができます。
- flagTestModeDisplayInfo: 普段はこのスイッチは Off のままご使用下さい。チェックを入れると画面に『状態』が表示されます。テストする場合に便利です。
- vertialSyncTimestampFrame: 画面の表示に同期してタイムスタンプを残す場合、この値のフレーム分だけ遅れてタイムスタンプを打つ時刻が調整されます。調整するには、使用機器の測定が必要となります。
- TTL_ChatteringBlock: TTL のチャタリングをソフトウェア的に防止します。但し入力が遅れるため、外部の回路でホームボタンのチャタリングを防ぐことをお勧めします。
- TTL_KeyEmulate(ALT): ホームボタンの代わりにキーボードの左側の ALT キーを使用することができます。

8.3. 『DelayedMatch タスク』フォルダ位置

フォルダ位置は各ユーザーの 'Documents' の 『PyBTTs』 『Run』 フォルダの下、 『PyBTTsDelayedMatchV16』 になります。

8.4. Psychopy 起動後の途中終了について

以下のどちらかで終了できます。

- キーボードの ESC キー

- マウスの右クリックで、『右上端』『左下端』の順で3回繰り返すと終了します。

8.5. タッチパネルへの対応

Psychopy では、現時点(2022/09/10)では基本的にはタッチパネルには対応していません。このプログラムでは以下工夫をしています。

- タッチパネルへのタッチをマウスの右クリックにエミュレートしている。
- Windows10 の設定で、『タッチパネル』『長押し』による右クリック

8.6. 各ファイル

8.6.1. bttTrialDM.py (DelayedMatch クラス/定義ファイル)

主なクラスを示します。

8.6.1.1. class BTTsDefineDM

クラス『BTTsDefineDM』は定数の定義を行っています。例えばエラーコードやアナログ出力値など固定値を定義しています。

8.6.1.2. class BTTsStateDM

1つのトライアルの流れが進むに合わせて『状態(state)』変数が変化します。クラス『BTTsStateDM』はこの『状態』を管理しています。この変数の変化により刺激提示を変化させています。例えばクラスのインスタンス変数、state や state1 は

- self.state 大まかな状態を表す。例えば Cue の表示フェーズ、ターゲットの表示フェーズなど
- self.state1 各フェーズ中の状態を示す。例えばターゲット表示フェーズ中のタッチ待ちなどを表しています。

8.6.2. BTTs16DMExtraStateDelayTTLLoop.py (DelayedMatch のメインファイル)

DelayedMatch のメインファイルです。

8.6.3. PyBTTsSound.py (PyBTTsEnv 内、サウンド環境ファイル)

各コンピュータでのサウンド環境の設定を行います。

8.6.3.1. Psychopy 環境の指定

サウンド環境の設定する項目と SurfaceGO でのデフォルト値を示します。詳しくは、Psychopy のドキュメントをご覧ください。その他のコンピュータではこれらを調整が必要な場合があります。

- 'audioLib' - Psychopy オーディオライブラリの指定。'pyo'を指定
- 'audioDriver' - オーディオドライバーの指定。'Primary Sound'を指定
- 'audioLatencyMode' - オーディオレイテンシモードの指定。'0'を指定。

8.6.3.2. サウンド環境の変数値

class BTTsSoundInfo にはサウンド環境の変数値が設定されます。

- sound_sampleRate : サウンドのサンプリングレート。
- useReload : play 時設定のリロードが必要な場合。False で不必要

- hardsnddelay : play までのディレイが必要な場合。秒単位

8.6.4. LoadPyBTTsTouchOne.py (PyBTTsTouch 内、タッチパネルサポート)

Psychopy で、タッチパネルへの反応をイベントとして得るために、この機能によって『タッチパネルイベント』=>『マウスの右クリックイベント』への変換を行っています。

8.6.4.1. PyBTTsTouchOne.dll (64bit ライブラリ マウスイベント変換)

この機能によって『タッチパネルイベント』=>『マウスの右クリックイベント』への変換を行っています

8.6.5. wrapperPyDIOBase.py (PyDIO フォルダ内、TTL/Analog-Out サポート)

ホームボタンの入力 (TTL in) と アナログ値出力 (Analog out)のサポートを行います。

8.6.5.1. class OneShotDAQTLL

TTL 入力とアナログ出力のメソッドの基底クラスです。実ハードウェアへの実装を別で行われます。

8.6.6. wrapperPyDIONIDAQ.py (PyDIONIDAQ フォルダ内、NI-DAQ TTL/Analog-Out サポート) national instruments 製 ハードウェアのサポート。

8.6.6.1. class NIDAQOneShotDAQ

NI 社のハードウェアの OneShotDAQTLL の実装クラス。

- NIDAQOneShotDAQ コンストラクタ (deviceName, mode)
deviceName : NI 社ハードウェアのデバイス名。例えば'Dev1'など。ハードウェアの実装によって変わります。

8.6.7. TextOpenGLObj.py (TextGL フォルダ内、テキスト文字サポート)

画面描画で使用しています。インフォメーションの表示のみに使用し、通常は使っていません。

8.7. 共通ライブラリ (CommonV010011 フォルダ内)

タスクを定義する時に使用しているライブラリ群です。

8.7.1. bttEnv.py (PyBTTs フォルダサポート)

各ユーザーのドキュメントフォルダ内の PyBTTs フォルダが使用されます。

8.7.1.1. class BTTsEnv

PyBTTs フォルダのサポート。

8.7.2. BTTsEnvDefine.py (定数サポート)

プログラム内で使用される定数を定義します。

8.7.2.1. class BTTsEnvDefine

- **sizeBinaryOnePacket** : BTTsParam で通信/保存/読み込みで使用する 1 つのパケットの大きさ。バイト数。

8.7.3. bttEnvFile.py (環境設定ファイル の読み込み)

環境設定ファイル『envPyBTTs.txt』の読み込み。

8.7.3.1. class BTTsEnvFile

環境設定ファイル『envPyBTTs.txt』の読み込み。クラスの BTTsEnvFile のコンストラクタ呼び出しと共に、『envPyBTTs.txt』の読み込む。主に、

- バックグラウンドカラーの指定
 - 給餌装置ハードウェアの指定
 - TTL-in/Analog-out ハードウェアの指定
- を読み込みます。

8.7.4. bttRandom.py (PyBTTs システムのランダム指定/定義)

PyBTTs システムのランダム指定/定義を行っています。

8.7.4.1. class BTTsRandom

ランダムの生成方法の定義。

- genRandomList1Array(list):
リストを指定するとそのリストを『シャッフル』を行います。

8.7.4.2. class RandomList

ランダムの生成とランダム値取り出し。class BTTsRandom を使用しランダムを生成します。また生成したランダムの取り出しを行います。生成した1つのランダムリストを全て使用すると新たにランダムを作り直します。この操作により、『一定の割合のランダム』を作り出しています。

- setListValue(list)
ランダムリストの設定。
- init():
ランダム化の初期化。リスト取り出しのインデックスの初期化。
- getValue():
ランダムリストより1データを取り出す。
- nextIndex():
ランダムリストの更新。

8.7.5. bttValue.py (PyBTTs システムのパラメータ定義/レコード定義用値クラス群)

パラメータ定義時に使用する各値のクラス群。ネットワーク通信時、データ保存/読み込み時に値をシリアル化することができる。またダイアログ GUI のアイテム GUI とのやりとりを行う。各値クラスの主なメソッドを以下に示す。

- コンストラクタ(__init__)
vname: 値の名前
value: 値(初期値)
- toKeyValue():
戻り値は、GUI で使用するキーと値。各値クラス毎に値の型が違う。
- setValue(value):
各値クラスの値設定。
- setValueGUI(value):
各値クラスの値の GUI からの設定。value 値は string のため、各値に変換する。例えば float32

ならば文字列数値から float32 値へ。int32 ならば文字列数値から int32 値へ。

8.7.5.1. class ValueString

ストリングを扱う。但し扱えるのは英字/数字のみとなる。ネットワーク通信時、データ保存/読み込み時には ASCII コードに変換される。GUI 使用時は、python-string で扱われる。

8.7.5.2. class ValueInt32Array

整数値(int32)を扱う。ネットワーク通信時、データ保存/読み込み時には binary 4 バイト。リトルエンディアンでエンコード／デコードされる。GUI 使用時は、python-int で扱われる。

8.7.5.3. class ValueFloat32

単精度浮動小数点(float32)で扱う。ネットワーク通信時、データ保存/読み込み時には binary 4 バイト。リトルエンディアンでエンコード／デコードされる。GUI 使用時は、python-float で扱われる。

8.7.5.4. class ValueBool8

ブール値(1 バイト)で扱う。ネットワーク通信時、データ保存/読み込み時には binary 1 バイト。GUI 使用時は、python-string で扱われる。

8.7.5.5. class ValueInt32Array

整数値(int32)の配列リストで扱う。ネットワーク通信時、データ保存/読み込み時には配列の個数を binary 4 バイト、配列を 4 バイト×個数の binary で扱う。リトルエンディアンでエンコード／デコードされる。GUI 使用時は、python-string で扱われる。

8.7.5.6. class ValueFloat32Array

単精度浮動小数点(float32)の配列リストで扱う。ネットワーク通信時、データ保存/読み込み時には配列の個数を binary 4 バイト、配列を 4 バイト×個数の binary で扱う。リトルエンディアンでエンコード／デコードされる。GUI 使用時は、python-string で扱われる。

8.7.5.7. class ValueInt32Array2D

整数値(int32)の 2 次元配列リストで扱う。ネットワーク通信時、データ保存/読み込み時には配列の個数を binary 4 バイト、1 次元配列を配列の個数を binary 4 バイト、配列を 4 バイト×個数の binary で扱う。リトルエンディアンでエンコード／デコードされる。GUI 使用時は、python-string で扱われる。

8.7.6. bttsWaitFor.py (PyBTTs システムの時間ウェイト機能)

BTTs システムの時間ウェイト機能。主な特徴として、

- callback 関数の呼び出し機能
- 外部からのウェイト中断機能
- 時間制度の向上

8.7.6.1. class BTTsWaitFor

- stampNextFor() 時間ウェイト時基準スタンプ
- nextFor(self, nextSec, callback=None, callbackArgs=None) : 時間ウェイト stampNextFor() で呼び出した時点を基準に指定時間 nextSec(秒)ウェイトする。オプションで callback 関数の指定、callbackArgs は callback の引数の指定となる。

9. 出力ファイル

『DelayedMatch タスク』の出力ファイルは、各ユーザーの'Documents'の『PyBTTs』『Run』フォルダーの下、『PyBTTsDelayedMatchV16』の下に『data』フォルダが作られます。この中に出力ファイルが書き込まれます。

9.1. 出力ファイルパラメータの一覧

stim_pos	ターゲットのポジション番号。
stim_shape	ターゲットの図形番号。
stim_correctIndex	正解側のインデックス。stim_pos,stim_shapeに適用される。
.thisRepN	psychopyの管理保存域。通常使用しない。
.thisTrialN	psychopyの管理保存域。通常使用しない。
.thisN	psychopyの管理保存域。通常使用しない。
.thisIndex	psychopyの管理保存域。通常使用しない。
DataNum	試行番号
SuccessTrialNum	正解試行数（1セッション中の積算数）
NonTargetTouchNum	不正解試行数
TrialNum	正解試行数 + 不正解試行数
ErrorCode	どこでエラーをしたのか
retryCount	リトライした回数
(space01)	
TrialOnT(s)	タスク開始合図の時間
HmHoldOnT(s)	ホームボタンにタッチした時間
HmHoldOffT(s)	ホームボタンから手を離した時間
CueOnT(s)	手がかり刺激提示の開始時間
CueOffT(s)	ホームボタンから手を離した時間
HmHold2OnT(s)	遅延時間の開始時間
HmHold2OffT(s)	ホームボタンから手を離した時間
TargetOnT(s)	ターゲット提示の開始時間
HmReleaseT(s)	ホームボタンから手を離した時間
TargetTouchT(s)	正解ターゲットにタッチした時間
NonTargetTouchT(s)	不正解ターゲットにタッチした時間
TargetHoldOff(s)	正解ターゲットから手を離した時間
PreRwdOnT(s)	報酬前期間の開始時間
RwdOnT_s(s)	報酬の開始時間
ErrorOnT(s)	反省期間の開始時間
ITIOnt(s)	タスク開始前時間の開始時間
(space02)	
expName	タスクの名前
psychopyVersion	psychopyのバージョン
frameRate	フレームレート

PreHmTouchDur(ms)	マーモセットがホームボタンにタッチするまでの時間
HmHoldDur(ms)	ホームボタンから前足を離すまでの時間
CueDur(ms)	手がかり刺激提示から、ホームボタンから前足を離すまでの時間
HmHold2Dur(ms)	遅延時間の開始から、ホームボタンから前足を離すまでの時間
HmReleaseDur(ms)	ターゲット提示から、ホームボタンから前足を離すまでの時間
TotalHmHoldDur(ms)	ターゲット提示後に、ホームボタンから前足を離すまでの時間
ResponseT(ms)	ターゲット提示から、ターゲットにタッチするまでの時間
TargetHoldDur(ms)	ターゲットにタッチし続けた時間
Corrected (space03)	コレクトした時、True
Set_HmTouchLimit	開始後ホームボタンにタッチしなければいけない制限時間(msec)
Set_HmHoldDur	ホームボタンにタッチし続ける期間の設定時間(msec)
Set_CurDur	手がかり刺激提示期間の設定時間(msec)
Set_HmHold2Dur	遅延期間の設定時間(msec)
Set_HmReleaseLimit	ターゲット提示後に、ホームボタンから離さなければいけない制限時間(msec)
Set_ReachTargetLimit	ホームボタンから前足を離してから、タッチしなければいけない制限時間(msec)
Set_TargetHoldDur	ターゲットにタッチし続けなければいけない時間(msec)
Set_PreRwdDur_ms	報酬前時間の設定時間(msec)
Set_RwdDur_ms	報酬時間の設定時間(msec)
Set_ErrorDur_ms	反省期間の設定時間(msec)
Set_ITI	タスク開始前期間の設定時間(msec)
Set_RwdDrink_micro l	ドリンク時、一回あたりドリンク量(micro l)
Set_RwdNum (space04)	1 回のリワードで与える餌の量
CueShape	手がかり刺激の形 (pngファイル時はそのファイル名)
CueColorNum	手がかり刺激の色の番号
CueColorName	手がかり刺激の色の名前
CueShapeNum	手がかり刺激の形の番号
CueShapeName	手がかり刺激の形の名前
CueSizeX	手がかり刺激のサイズ x
CueSizeY	手がかり刺激のサイズ y
CuePositionX	手がかり刺激の位置、X座標
CuePositionY	手がかり刺激の位置、Y座標
CueWindowSizeX	手がかり刺激のサイズ(タッチ時)
CueWindowSizeY (space05)	手がかり刺激のサイズ(タッチ時)

TargetShape	Targetの形 (pngファイル時はそのファイル名)
TargetColorNum	Target刺激の色の番号
TargetColorName	Target刺激の色の名前
TargetShapeNum	Target刺激の形の番号
TargetShapeName	Target刺激の形の名前
TargetSizeX	Target刺激のサイズ x
TargetSizeY	Target刺激のサイズ y
TargetPositionX	Target刺激の位置、X座標
TargetPositionY	Target刺激の位置、Y座標
TargetWindowSizeX	Target刺激のサイズ(タッチ時)
TargetWindowSizeY	Target刺激のサイズ(タッチ時)
(space06)	
NonTargetShape	NonTargetの形 (pngファイル時はそのファイル名)
NonTargetColorNum	NonTarget刺激の色の番号
NonTargetColorName	NonTarget刺激の色の名前
NonTargetShapeNum	NonTarget刺激の形の番号
NonTargetShapeName	NonTarget刺激の形の名前
NonTargetSizeX	NonTarget刺激のサイズ x
NonTargetSizeY	NonTarget刺激のサイズ y
NonTargetPositionX	NonTarget刺激の位置、X座標
NonTargetPositionY	NonTarget刺激の位置、Y座標
NonTargetWindowSizeX	NonTarget刺激のサイズ(タッチ時)
NonTargetWindowSizeY	NonTarget刺激のサイズ(タッチ時)
(space07)	
recDate	データ記録の時刻

10. Appendix A

10.1. 『DelayedMatch タスク』のポジションランダムで使用する時

『DelayedMatch タスク』をポジションで行う場合(modeDeletedMatch = 0)に指定します。
図形グループの指定例を示します。

例 1-1) 『GroupA_ShapeColor の図形を指定して、グループ内全ての図形を使用する場合』

- stimGroupName : GroupA_ShapeColor を指定
- stim_list : [0,1,2,3,4,5,6,7,8] のように指定します。

例 1-2) 『GroupA_ShapeColor の図形を指定して、グループ内 0,1,2 の図形のみを使用する場合』

- stimGroupName : GroupA_ShapeColor を指定
- stim_list : [0,1,2] のように指定します。

GroupA_ShapeColor の[0,1,2]は

0 : circle/red 1: circle/green 2: circle/blue

となり、形は同じだが、色違いでポジションの DelayedMatch が行われます。

10.2. 『DelayedMatch タスク』の色のランダムで使用する時

『DelayedMatch タスク』を色のランダムで行う場合(modeDeletedMatch = 1)に指定します。図形グループの指定例を示します。

例 2-1) 『GroupA_ShapeColor の図形を指定して、形が同じものを選択』

- stimGroupName : GroupA_ShapeColor を指定
- stim_list : [0,1,2],[3,4,5],[6,7,8] のように指定します。

このように指定すると、

[0,1,2]の組み合わせでランダム

[3,4,5]の組み合わせでランダム

[6,7,8]の組み合わせでランダム

となり、正解、不正解の組み合わせが形で固定されます。

例 2-2) 『GroupA_ShapeColor の図形を指定して、形が同じものを選択。数を減らす』

- stimGroupName : GroupA_ShapeColor を指定
- stim_list : [0,1],[3,4],[6,7] のように指定します。

このように指定すると、

[0,1]の組み合わせでランダム

[3,4]の組み合わせでランダム

[6,7]の組み合わせでランダム
となり、正解、不正解の組み合わせが形で固定され、組み合わせの数を減らすことができます。

10.3. 『DelayedMatch タスク』の形のランダムで使用する時

『DelayedMatch タスク』をポジションで行う場合(modeDeletedMatch = 1)に指定します。図形グループの指定例を示します。

例 3-1) 『GroupA_ShapeColor の図形を指定して、色が同じものを選択』

- stimGroupName : GroupA_ShapeColor を指定
- stim_list : [0,3,6],[1,4,7],[2,5,8] のように指定します。

このように指定すると、

[0,3,6]の組み合わせでランダム

[1,4,7]の組み合わせでランダム

[2,5,8]の組み合わせでランダム

となり、正解、不正解の組み合わせが色で固定されます。