Instructions for using the Garden Game task

Bonn Spatial Memory Lab

Laura Nett, Tim A. Guth, Philipp K. Büchel, Lukas Kunz

Department of Epileptology, University Hospital Bonn, Bonn, Germany

lukas.kunz@ukbonn.de

January 17, 2025

Overview

This document provides detailed information on running and adjusting the Garden Game, which is a task for cognitive and neuroscientific studies. We created the Garden Game using the Unity engine (version 2021.3.14f1) under Windows 11 with the *C#* programming language. In the GitHub folder, you can find the directories "Source," "Build," "TrialConfiguration," and "MatlabEmbedding."

The "Source" folder contains a Zenodo link to the Unity source version of the task. The "Build" folder contains a Zenodo link to the exported task, which you can run on Windows by double-clicking the "GardenGame.exe" file. The "TrialConfiguration" folder contains a Matlab script for creating participant-specific trial sequences. The "MatlabEmbedding" folder contains a Matlab script that you can use for synchronizing the Garden Game with other types of data (e.g., neural recordings). Please see below for detailed information on each folder.

Unity source version of the Garden Game

The source version of the Garden Game lets you modify the task according to your needs. You can open the Garden Game source version using the Unity Hub (`https://unity.com/unity-hub`), after installing the appropriate Unity version within the Unity Hub (2021.3.14f1). The source version is fully modifiable, so you can change whatever you want. For example, you may want to change the layout of the scenes, add or remove any scenes, or change what is written into the log file. You can export the Garden Game to various operating systems including Windows, Linux, and macOS. Please note the license under which the Garden Game is published.

In brief, the Garden Game source version is organized into seven scenes (under `Project\Assets\Scenes`): "1_OriginScene," "2_EncodingScene," "3a_AllocentricMapScene," "3b_EgocentricMapScene," "4_ScoreScene," "X_GUIScene," and "Y_ExplanationScene." Each scene contains a similarly named C# script controlling the game logic within that scene (under `Project\Assets\Scripts`). The script "Origin.cs" contains important parameters, such as the minimum duration of the starting position, the maximum distance during egocentric recall, and participants' movement speed. You can modify these parameters to adjust the Garden Game according to your needs.

Unity build version of the Garden Game

The build version of the Garden Game lets you run the task out of the box. Download the folder from Zenodo, unzip it, and double-click the "GardenGame.exe" file. The task will start in full-screen mode and take you to the start screen.

Start screen
At the beginning of the task, the Garden Game displays a start screen that allows the selection of various options. First, enter the participant's ID in the field "Subject ID." This ID has to consist of three digits in the range between 001 and 999. Based on this ID, the game reads a specific trial-configuration file and tree-configuration file for this participant (from `Project\Assets\StreamingAssets\Config`). Next, choose the participant's language. Currently, the Garden Game supports English and German (but, for example, German can easily be replaced by other languages). Then, choose whether synchronization triggers shall be logged using external triggers or phototransistor triggers (see detailed information below). Next, decide whether to enable eye tracking (if you have connected a Tobii eye tracker), and whether the participant shall complete an explanation/practice trial. Then press "Start," which will turn green if the task is ready to start. If you have not entered sufficient information on the start screen, the "Start" button will turn red and the task will not start.

Player input
Throughout the task, participants can use a keyboard or a gamepad (e.g., Logitech F310) to interact with the task. During the encoding period, participants can move forward (up arrow), turn right (right arrow), and turn left (left arrow). They can use the "Y" button to move faster. During retrieval periods, participants can move the response cross up (up arrow), down (down arrow), to the right (right arrow), and to the left (left arrow). When prompted, participants need to press the "X" button to proceed to the next task phase (e.g., during the score screen).

Explanation/practice trial
By default, the first trial of the Garden Game, defined as trial 0, is a practice trial with verbal instructions explaining the task to the participant. This ensures standardized instructions across participants. The practice trial guides participants through the different aspects of an example trial. Currently available languages for the practice trial are English and German (you can add other languages through the source version). The explanation trial can be skipped by choosing the "Skip training" option on the start screen.

In the practice trial, the player starts from the center of the environment, facing the north fence. The animals from the practice trial are not otherwise used in this session and are positioned 5 virtual units ahead of and 5 virtual units to the left of the player's starting position, respectively. While participants listen to the explanations, their input is disabled (they thus cannot move during these periods). After the practice trial, you should give participants the option to ask questions if anything is unclear.

Encoding
The build version of the Garden Game contains 60 trials, with each trial consisting of two parts, encoding and retrieval.

During the encoding periods, the Garden Game features a virtual environment that extends into infinity (no distal cues). The navigable area of the environment is limited by a square boundary with an edge length of 20 virtual units (vu). Virtual units roughly correspond to virtual meters given the height of the player and its movement speed. The square boundary consists of four fences at each cardinal direction. The north fence is highlighted in a different color (black) than the other three fences (brown), thus serving as a major orientation cue. The minimum distance between the participant's position and a fence is 0.1 vu. Three trees serve as intramaze landmarks and are placed at the centers (plus some jitter) of three quarters of the environment (randomized across participants).

At the beginning of each encoding period, participants are located at a random starting position (x and z coordinates ranging from -9 to 9 vu) within the environment and face a specific allocentric direction. Participants are required to stay still at this starting position for a minimum duration of 3 seconds, and they have to press the "X" button to start navigating. Participants are instructed to pay attention to this starting position as they are required to perform egocentric retrieval relative to this starting position. The starting position is at least 5 vu away from both animals and at least 1 vu away from the trees. Next, they can begin navigating through the virtual garden environment and encounter, in sequence, two animals at different locations within the environment (x and z coordinates ranging from -8.5 to 8.5 vu). When reaching an animal (cutoff distance between the player and the animal of 1 vu), participants remain at this position for a duration of 2 seconds, during which the name of the animal is displayed (e.g. "Bird") and its animal sound is played. The animal then disappears and the next animal appears at a different location in the environment. Participants are instructed to memorize the positions of the two animals relative to their starting position and orientation (egocentrically) and relative to the environment (allocentrically).

Three of the animals have the same encoding location across trials (plus a random jitter of ±1 vu), which we refer to as "stable animals". The other three animals have different (i.e., random) encoding locations across trials, and we refer to these animals as "unstable animals". During each encoding, one stable and one unstable animal is encountered. In the build version of the Garden Game, animals are placed in the environment at least 1.5 vu away from the nearest tree. Stable animals are placed at least 10 vu away from other stable animals and at least 5 vu away from the center of the environment. Unstable animals are placed at least 10 vu away from other animals on the same trial and at least 1 vu away from all other animals on all trials. These settings can be modified by changing how the trial-configuration files of the participants are created (see below).

Retrieval

During the retrieval periods, the Garden Game displays two abstract coordinate systems for allocentric and egocentric retrieval. There are four individual recalls per retrieval period (one allocentric recall for each animal and one egocentric recall for each animal).

During allocentric retrieval, participants view an allocentric map of the garden environment from a bird's eye view (the black north fence is shown at the top) and a question at the top of the screen indicates which animal the participant is asked to retrieve. Participants then move a red cross across the allocentric map and press a button to indicate the remembered allocentric location of the animal. Once participants have pressed the response button, a blue cross appears at the correct location on the allocentric map (duration of 2 seconds) and participants' total score displayed in the lower left corner is updated (0–10 points per retrieval) depending on the accuracy of their response.

During egocentric retrieval, participants view an egocentric map (i.e., a polar coordinate system with its angles representing egocentric directions and its radii representing egocentric distances) that is centered on their starting position and orientation (ahead is shown at the top) and a question at the top of the screen indicates which animal the participant is asked to retrieve. Participants then move a red cross across the egocentric map and press a button to indicate the remembered egocentric location of the animal relative to their starting position and orientation. Once they have pressed the response button, a blue cross appears in the correct location on the egocentric map and participants' total score is updated (0–10 points per retrieval) depending on the accuracy of their response.

The order of allocentric and egocentric retrieval alternates between trials and is randomized across participants. The order of retrieval for the firstly and secondly encoded animal is random across all retrieval periods. For each participant, the Garden Game features six different animals, randomly chosen from a pool of 14 possible animals (bird, camel, cat, chicken, dog, elephant, horse, leopard, penguin, pig, pug, rhino, sheep, and tiger). Across all 60 trials of the task, participants encode each animal 20 times and recall each animal 40 times (20 times allocentrically and 20 times egocentrically).

Score screen

Following each fourth round, a score screen is displayed, providing participants with feedback on their points per trial and the number of trials completed.

Performance scores

For both types of retrieval, memory performance is calculated based on the Euclidean distance between the participant's response location and the correct location of the animal in the abstract allocentric and egocentric coordinate systems. This Euclidean distance is then ranked within 1 million surrogate distances to account for the fact that different animal locations have different chance levels, following established procedures (Kunz et al., Neuron, 2021; Nature Neuroscience, 2024).

To keep participants motivated during the task, participants receive points for each response that are added to their total score. The number of points depends on the response-specific memory performance (minimum of 0 points; maximum of 10 points). Points are given based on an adaptive algorithm so that participants generally receive between 7 and 8 points per retrieval.

This algorithm calculates the mean allocentric/egocentric memory performance throughout the task and updates an allocentric/egocentric adjustment value after each trial. The adjustment values are initially set to zero and undergo incremental changes of ±1 based on the mean allocentric/egocentric memory performance. If the mean performance is below 7, the adjustment value becomes more positive, while it becomes more negative if the performance is above 8. If the mean memory performance falls between 7 and 8, the adjustment score remains unchanged. Each retrieval type (allocentric and egocentric) has its own adjustment value, ranging from -10 (because of good performance) to +10 (because of poor performance). If you wish, you can change these settings through the source version of the Garden

Game (these parameters are set in the "Origin.cs" file).

Ending the Garden Game
The Garden Game can be interrupted at any time by pressing "Esc" on the keyboard. A screen will appear asking whether you want to continue or quit the game. Quitting the game will shut down the task and end writing the log file.

Log file
During the task, the Garden Game generates a log file containing a variety of task and behavioral information. The temporal resolution of this information depends on the frame rate of the experimental laptop, as it is updated with every new frame. Here is a list of the main types of information included in the log file:

- Encoding
  - the current trial index,
  - the current period index,
  - the participant's starting position and orientation,
  - the participant's positions and orientations during navigation,
  - the object's name, position, and orientation (always facing the center of the environment),
  - the tree positions,
  - eye-tracking information (if used; see Eye Tracking).
- Allocentric retrieval
  - the current period index,
  - the object to be retrieved,
  - the participant's response positions,
  - the participant's final response location,
  - the correct location,
  - the drop error,
  - the ranked drop error,
  - average memory points,
  - the memory adjustment,
  - the memory points,
  - the player score,
  - eye-tracking information (if used; see Eye Tracking).
- Egocentric retrieval
  - the current period index,
  - the object to be retrieved,
  - the participant's response positions,
  - the participant's final response location on the egocentric map,
  - the egocentric distance of the response location,
  - the egocentric direction of the response location,
  - the participant's final response location converted into allocentric coordinates,
  - the drop error,
  - the ranked drop error,
  - average memory points,
  - the memory adjustment,
  - the memory points,

- the player score,
- eye-tracking information (if used; see Eye Tracking).

Trial configurations

Together with the Garden Game, we provide a script in MATLAB (The MathWorks, Natick, MA, US) for creating custom trial-configuration files. These text files are read by the Garden Game during runtime and determine the structure of the task. They are placed within the `Project\Assets\StreamingAssets\Config` folder of the task. Users can flexibly adjust this MATLAB script (or change the trial-configuration files by hand) to modify the Garden Game structure according to their needs. The MATLAB script has several adjustable parameters, such as the number of trials, objects, and trees, as well as the possible tree, object, and player positions. For sanity checks, the script generates several figures, for example regarding the tree positions, object positions, player starting positions, and ego- and allocentric starting positions.

Synchronization with other recordings

We provide two options for synchronizing the behavioral Garden Game data with other types of data such as neural data during runtime.

Matlab embedding for external triggers
The first synchronization option makes use of a MATLAB script (`LK_GardenGame_MatlabEmbedding_20230706.m`) that sends USB triggers to, for example, a DAQ (e.g., Diligent MCC USB-1208FS-Plus) of a neural recording system using functions from Psychtoolbox (Kleiner et al., Perception, 2007). In synchrony with these USB triggers, the MATLAB script simulates button presses ("T" and "R") that the Garden Game logs as triggers, using the markers "external trigger" and "external backup trigger."

When using this synchronization option, you will need to run this MATLAB script, which in turn invokes the Garden Game. You will be asked if Predator Sense, a performance optimization software for overclocking the CPU and GPU on certain Acer computers (Acer Inc., New Taipei City, Taiwan), is turned on. Next, you will be taken to your file explorer where you can select the "GardenGame.exe" you want to use.

When using the MATLAB embedding, the input device (gamepad or keyboard) must be specified in the MATLAB script, otherwise an error will occur. Furthermore, if no DAQ is connected a message will appear: "Did not find a device, should I clear PsychHID and try again?" Then ensure that a DAQ is connected, otherwise no USB triggers will be sent to the recording device. If the DAQ is connected and you still get this error, restart MATLAB. At the beginning, an initial set of 10 triggers with a duration of ∼0.5 seconds is sent to the recording device (these triggers will not appear in the Garden Game log file). The Garden Game will then start automatically and the start screen will appear. The triggers that MATLAB will send while the Garden Game is running will be written to the Garden Game log file and to the recording device. After quitting the Garden Game, 20 additional triggers (duration of ∼0.5 seconds) are sent to the recording device to indicate the end of the task (these triggers will not appear in the Garden Game log file).

Phototransistor triggers
The second synchronization option displays a small rectangle at the lower right corner of the laptop screen whose changes from black to white can be detected using a phototransistor. Using an appropriate electric circuit, you can feed the phototransistor output into the analog channels of another recording device, such as a neural recording device. Whenever the rectangle changes from black to white color, the marker "photo trigger" is written into the Garden Game log file.

Eye tracking

To perform eye tracking during the Garden Game, you may connect different types of eye trackers

by Tobii (Tobii AB, Stockholm, Sweden) to the experiment laptop. This option must be selected on the initial start screen. The eye tracker must be installed and configured for the laptop in the Tobii Pro Eye Tracker Manager application (`https://connect.tobii.com/s/etm-downloads?language=en_US`).

The practice trial is always conducted without eye tracking. After the practice trial, i.e., before the onset of trial 1, a head tracking box is displayed to ensure that the participant's angle and distance to the eye tracker are correct. Then, when the study coordinator presses "C" on the recording laptop, a six-point calibration (which can be adjusted to have fewer or more points in the calibration prefab in the Garden Game source version) is performed in which participants are instructed to maintain steady fixation on dots displayed at various locations on the screen. After calibration, the participant's 3D gaze point is visualized as red dots in the virtual environment, which disappears once the participant presses "X" to start the trial. If desired (e.g., for double-checking), the participant's 3D gaze point within the virtual environment can be displayed during the trial by pressing "E" on the keyboard. Eye movements are recorded during both encoding and retrieval. Eye-tracking data are included in the behavioral log file and updated with every new frame (the temporal resolution of the eye-tracking data thus depends both on the eye tracker and on the frame rate). This data includes the left and right gaze ray on the screen, gaze points in the virtual world and on the screen, left and right pupil diameters, and the last object viewed.

Contact

If you have any questions, please contact us at `https://spatialmemorylab.com/contact/`.