

UNIVERSITY OF TARTU
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
Institute of Computer Science
Computer Science Curriculum

Kälver Kilvits

Toolbox for creating virtual reality
environments for memory research

Bachelor's Thesis (9 ECTS)

Supervisor: Jaan Aru, PhD

Supervisor: Raul Vicente, PhD

Toolbox for creating virtual reality environments for memory research

Abstract:

Our memory is the ability to store and recollect vast amount of information over long time periods. Current memory research methods are quite limited and virtual reality can offer new approaches. For example, it is suggested that activity of our memory in virtual reality might be more lifelike than in case of laboratory experiments since more cognitive skills are provoked simultaneously. The main purpose of this thesis is to create a toolbox that enables to create memory experiments in virtual reality. The toolbox will have a random game room generator, level editor and an experiment creator. It will support four different types of memory experiment methods.

Keywords: recognition, recall, episodic memory, memory training, method of loci, virtual reality

Tööriistakast mälu katsete läbiviimiseks virtuaalses reaalsuses

Lühikokkuvõte:

Mälu on omadus säilitada informatsiooni pikaajaliselt. Tänapäeva mälu uuringu meetodid on suhteliselt piiratud. Virtuaalse reaalsuse kasutamine mälu uuringutes võib aidata meetodeid laiendada ning leida uusi lähenemisi. Näiteks on välja toodud, et mälu tegevus virtuaalses reaalsuses võib peegeldada igapäevast mälu kasutust rohkem kui laboriekspriimendid. Seda eelkõige seetõttu, et sarnaselt reaalsele elule rakendab ka virtuaalne reaalsus töösse erinevaid kognitiivseid protsesse. Käesoleva lõputöö eesmärgiks on luua tööriistakast, mis võimaldab teha mälu uuringuid virtuaalses reaalsuses. Tööriistakastiga saab eksperimente luua, mänguruumi genereerida, muuta ja ükshaaval kokku panna. Toetatud on neli erinevat mälu uuringu meetodit.

Märksõnad: äratundmine, meeldetuletamine, episoodiline mälu, mälu treening, aukohameetod, virtuaalne reaalsus

Contents

1	Introduction	4
2	Background and related work	5
2.1	Virtual reality	5
2.2	Basic concepts of memory research	5
2.3	Methods for memory research	6
2.4	Prior experiments using virtual environments	7
2.5	Goals of the thesis	7
3	The toolbox	9
3.1	Used technologies	9
3.1.1	Oculus Rift	9
3.1.2	Unity	10
3.2	Programs structure	10
3.3	System requirements	11
3.3.1	Mutual requirements	11
3.3.2	Thesis specific requirements	14
3.4	Use cases	21
4	General Discussion	27
4.1	Future works	28
4.2	Conclusion	29
A	Appendices	30
A.1	The toolbox	30

1 Introduction

Our memories make up who we are and guide our everyday life. Current research methods to assess memory are quite limited as it is hard to create and manage real life environments so that the subject is not influenced by uncontrolled factors. Hence typical experiments consist of test subjects memorizing lists of randomly arranged words presented on a computer screen and measuring how many items each individual can recall at a later time point [1]. A possible solution to expand methods of research are virtual environments.

Even though virtual reality (VR) has been around for a long time, new achievements in technology allow it to mimic real life more than ever. Thus, virtual environments offer new ways to approach memory research and make it ecologically more realistic. For example, researchers can use 3D objects instead of 2D words or manipulate with objects' parameters so they would become abnormal and see what effects it has on subjects' performance.

Conducting memory experiments is quite difficult and expensive, especially resource and time wise. Subject's environment and outside interference play an important role and may have a big impact on test results. This is where the main advantages of virtual environments come to play. When using virtual environments subject's surroundings are completely managed by the researchers, meaning there is no unnecessary interference. Since virtual environments are fairly easy to create and duplicate, using them also helps to decrease the cost of an experiment.

By combining Virtual Reality products such as Oculus Rift with virtual environments, researchers can make the subject's test experience more lifelike and unique. Oculus Rift is virtual reality head-mounted display with a wide field of view and low latency head tracking. It gives the user more immersive feeling about the environment and surrounding objects, for example looking down a steep cliff makes the stomach hollow.

As a result of this thesis a Virtual Reality toolbox will be created in collaboration with the psychology students of the University of Tartu. It will allow researchers to easily conduct many different types of memory experiments with Oculus Rift. The toolbox will have a set of predefined memory experiments and a "make your own" option. "Predefined" meaning that it will have everything except for subjects to carry out an experiment.

2 Background and related work

2.1 Virtual reality

Virtual reality enables physical presence in computer generated environments. Its aim is to make the experience as authentic as possible. By using modern technology even three of the traditionally recognized five human senses can be extended to the virtual world. For example, extensions of sight and hearing will also be used in this thesis.

When it comes to VR headsets they have to meet quite high standards for the result to seem natural to humans. Firstly, there are many technical difficulties. When trying to accomplish a natural experience in a virtual world, input and output methods have to mimic human perception closely. Due to that quite a high frame rate and low latency must be achieved which requires a lot of computing power. Currently, due to technological limits the VR experience is not as good as it could be, but with new technological achievements it is bound to improve. For example, the average field of view of a human is approximately 180 degrees, but most of the VR headsets offer a maximum of 100 degrees [2].

A lot of people who come in contact with VR undergo a virtual reality sickness. This phenomenon has been researched in-depth but has little closure on why it occurs. Its symptoms are quite similar to motion sickness [3], but they differ in a sense that in case of virtual reality sickness real self-motion is not necessary. Visual perception of self-movement, even when it is not really present, is sufficient for the sickness to set in [3].

New approaches to methodology are introduced due to many freedoms of virtual environments. When it comes to using a VR headset such as Oculus Rift for memory research it enables to enrich the experience even more. Combination of these two enable to carry out otherwise expensive and time-consuming experiments quite easily.

2.2 Basic concepts of memory research

Memory is one of the base components for intelligent life. Memory research involves a wide field of different research groups such as social psychologists, physiologists, neuropsychologists etc.

As it is hard to define what memory is we can only give a brief overview about what it does. It is a property of a living organism that enables information storage, encoding, retainment and recollection.

Memory is our ability to store previous experiences and remember them. It gives us the capability to learn and adapt from previous experiences as well as to build new relationships. It is the capability to recall previously acquired knowledge and skills. Our memories do not have a formal structure, they are rebuilt on request from abstract elements located in our brain [1].

Memory is roughly divided as short-term and long-term memory. As the name might

indicate short-term memory is the ability to reproduce acquired information immediately after learning it. Short-term memory can store a limited amount of information until it is constantly repeated and the information is deleted fairly rapidly once the revision has stopped.

Long-term memory is for the long-haul, data can be stored for long periods of time even indefinitely.

Short-term memory is heavily bound to long-term memory. Every time an item in short-term memory is recalled, its imprint in long-term memory strengthens [1, 4].

Even though they are quite similar most of the factors that affect long-term memory do not influence short-term memory. This indicates that long-term memory uses different parts of brain for its functionality than short-term memory [1].

2.3 Methods for memory research

Memory research aims to understand and decode processes which are connected to assimilating new knowledge, maintaining it for retrieval and recollecting or recognizing previously acquired information.

Most common method of research is to test subject's ability to memorize lists of randomly arranged objects presented at regular pace. Ebbinghaus (1885) was one of the pioneers of this technique. Researchers would analyze how long it took the subject to memorize a list of objects and how quickly one would forget it. Recognition and recall are the most known methods to investigate memory.

Recognition is the ability to reckon whether or not the object was previously on the list. This is usually accomplished with a multiple choice quiz or a "yes or no" response. There are two main methods for recognition memory research, forced-choice and free-choice.

Forced-choice is similar to multiple choice tests. Subject is given a set of N objects from which only some belongs to target category and others are distractors. Memory performance is measured by the amount of correct answers.

In case of free-choice, objects are presented one by one and the subject has to indicate whether the object belongs in the target category or not. Memory performance is evaluated by the rates of correct "yes" responses related to the rates of incorrect "yes" responses. Although in this case subjects performance cannot be derived straight from the results and researchers have to use statistical analysis to assess it [1].

Free recall has participants study a list of objects which they later have to recall either in a specific order or randomly. In these experiments researchers are commonly interested in the data that is derived from the order in which the objects were recalled and the inter-response times. Results are used to decode memory operations during storage and retrieval [1, 5].

2.4 Prior experiments using virtual environments

Many studies have used virtual environments as a part of their memory experiment and it is proven to widen classical methodology [6, 7].

Schultheis et al. [8] have found that virtual reality based tools have the ability to capture multiple aspects of cognition which might make it a good solution to rehabilitate or diagnose patients with cognitive disorders.

Gaën et al. [7] studied the age effect components on episodic memory in virtual reality by using recognition and free recall as their performance testing solutions. One of the aims of the study was to analyze what virtual environment tests have in common with classical neuropsychological tests. Their results suggested that performance of the subject appeared to be connected to general cognitive tests. Study concluded that virtual environments could be proven to be a sound tool to research age effect on episodic memory.

Method of loci (MOL) is a memorization technique that has been researched extensively using virtual environments. MOL is a good example to observe studying long-term memory and recall.

Legge et al. [6] predicted that the MOL should be most effective when used with rich and highly detailed environments that have been learned via real navigation e.g home. As a result they found that the virtual environment was not significantly more difficult to use with MOL than a personally familiar environment. Additionally the results suggested that high-imageability words are better recalled than low-imageability words. Thus, using virtual environments for memory research should not affect subjects memorization in any major way.

This rapidly developing area opens up a lot of possibilities, methodologies and new hypothesis for future investigation. Many of previous studies [6, 7] have used virtual environments as a part of their memory experiments, but none of them have included virtual reality headsets which enrich the experiment process even more. It can provide many additional features to memory research such as imitation of a more realistic environment and extend some of the human senses to virtual reality.

2.5 Goals of the thesis

Main goal of the thesis is to produce a ready to use toolbox that enables researchers to create memory experiments in virtual reality. To serve this purpose many subgoals must be addressed.

Following is an overview of the subgoals:

1. Random room generator.
2. Random object generator.
3. Level editor.

4. Implementation of different memory experiment methods.
5. User interface for the toolbox.

First of all a game room is needed. To solve this problem a random game room generator is implemented. It takes different room models as an input and outputs a randomly placed bundle of rooms that form a level.

To serve the purpose of object memorization research a random object generator is introduced. Researcher assigns objects to be used and these will be randomly placed across specific room models.

Random generation has quite low precision thus level editor is needed. It is as a post processor for game room generation. Level editor enables to make manual changes to the generated room.

Toolbox will support four different methods of memory research that were explained in section 2.3.

Toolbox will have a user interface that does not expect programming knowledge from the user to create basic memory research experiments.

3 The toolbox

Toolbox is being developed in collaboration with Markus Kängsepp [9].

3.1 Used technologies

3.1.1 Oculus Rift

Oculus Rift is a head mounted display used for immersive 3D experiences. Currently there are two development kits available, DK1 and DK2. DK2 can be seen from figure 1. As introduced on its official web page the rift has unique head tracking technology that enables to experience virtual environments more realistically [2]. Also it is claimed that their stereoscopic 3D view perceives images similarly to human eyes which make the environment more natural. Compared to computer monitors it also has quite a large field of view (100-degrees). Although there is still much to work on as humans native field of view is almost 180-degrees [10].

Oculus is currently the most popular and fast-growing head mounted display. It has an active community and quite a lot of demos and tutorials. Thus it was a good candidate to use for developing the toolbox. Also, Tartu University already had multiple head sets which made the decision making even easier.

But of-course the toolbox is not bound to work only with Oculus. When other VR headsets arise transfer from one to another is quite easy.



Figure 1: Oculus Rift head mounted display, Development Kit 2 [11]

3.1.2 Unity

Unity is a game engine that enables to develop small to mid size games quite effectively. It has easy to grasp interface, quite a lot of premade assets and an active community. Unity is also cross-platform ready, meaning that its easy to develop and migrate applications to other operating systems.

Unity was chosen as a base game engine mainly because it has a free personal edition, it is quite easy to work with and has Oculus integration. Quite a lot of Oculus experiences are done with Unity. It supports three scripting languages C Sharp, Javascript and Boo script. C Sharp was chosen as the primary scripting language for the toolbox as this seemed to fit the most.

3.2 Programs structure

The program itself has a relatively simple structure, see figure 2. All components are accessible from the main menu. There are five possible routes to take from main menu which do not involve closing the program. A more detailed structure of the program can be seen from figure 3. In the diagram rectangles represent views that the user can interact with, multiple cut rectangles mean a choice from list has to be made to continue, diamonds serve as decision nodes, circles are the endpoints and hexagons mark generation processes.

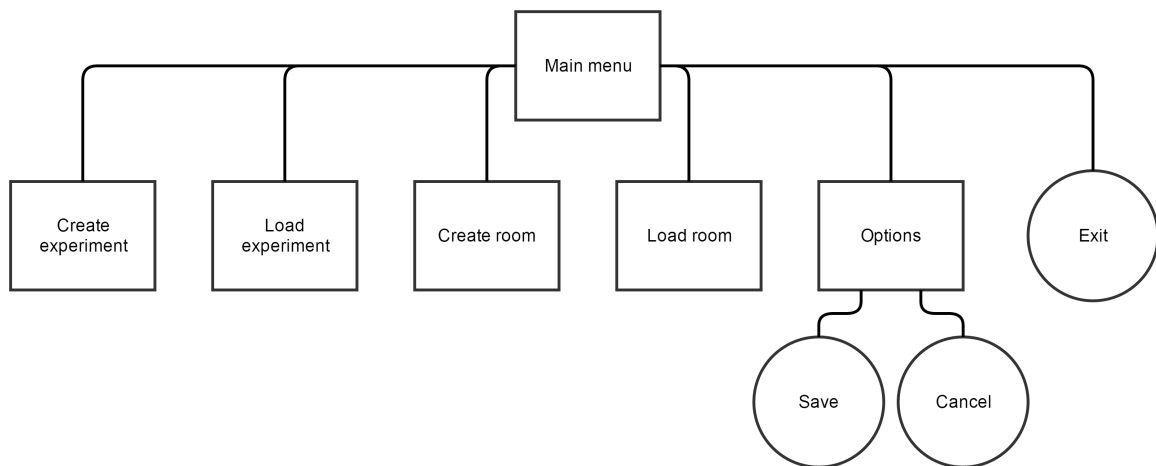


Figure 2: Simplified structure of the program (In the online version click for bigger image)

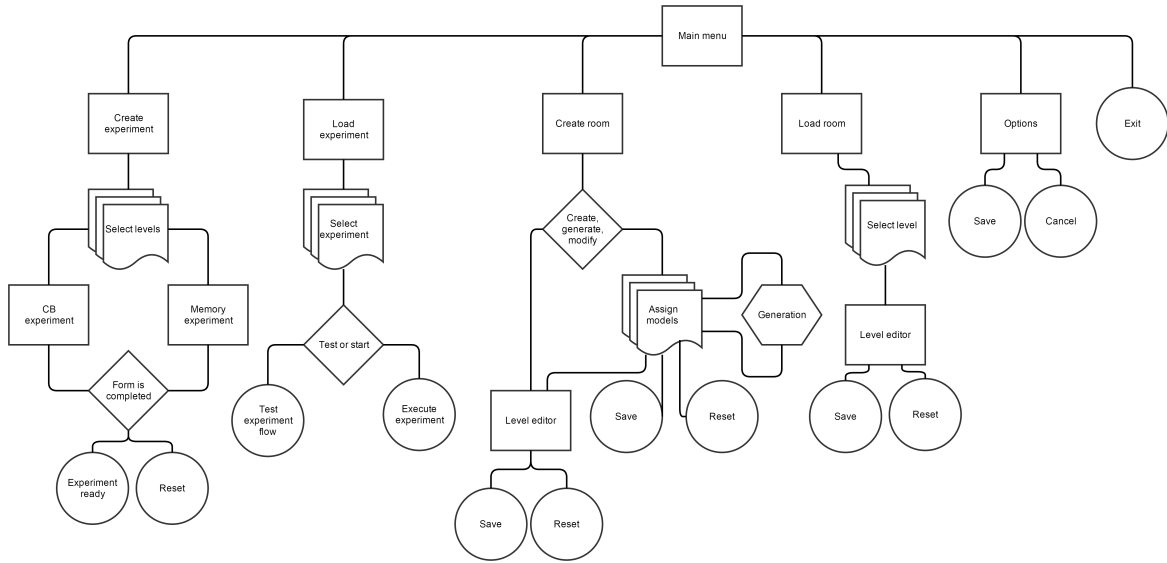


Figure 3: Programs structure (In the online version click for bigger image)

3.3 System requirements

This chapter discusses functional and non-functional requirements of the toolbox.

3.3.1 Mutual requirements

Requirements that overlap with Markus Kängsepp's thesis are developed jointly.

Non-functional requirements

Availability. Requires Unity Free and Oculus SDK if used with Oculus character controller.

Performance. Programs performance is mainly dictated by the graphics used. Scripts are made as frugal as they can be to author's knowledge.

Scalability. Program requires good graphics card and a moderate PC. Also Unity and Oculus SDK must be present.

Usability. Program is made to be easy to use. This is achieved by developing it in collaboration with psychology students. Main experiment cases are pre-programmed, see figures 11 to 14. If the user has previous experience using Unity, 90% of them should be ready to use the program without any basic training and 60% of users should find the program easy to use.

Technologies. Unity game engine and Oculus Rift virtual reality headset.

Functional requirements

1. System enables the user to manually manipulate the prefabricated objects (prefabs) in the game room.
2. System has a predefined player prefab.
3. System comes with a set of pre-configured prefabs.

Level editor. In case of memory experiments level editor acts as a post processor which enables to enhance and fine-tune the generated room. With level editor user is able to manipulate rooms' interior manually. This includes deletion, placement, rotation and color of level objects. For example, user can add more interior objects to the rooms if generation output was unsatisfactory or change the position of objects that were placed illogically. After desired changes are applied, there is also a possibility to test out the level with standard character controller to see the result in-game [9].

Player. Player is a standard Oculus character controller prefab which has been improved significantly. User can define which input devices are being used in the experiment, enable and disable subjects capabilities such as using free movement versus a statical position. See figure 4 for an example view.

Following features have been added to the standard character controller:

1. Fixed view of the room.
2. Experiment time limitation.
3. Experiment methods.
4. Object detection.
5. Ability to capture images from camera.

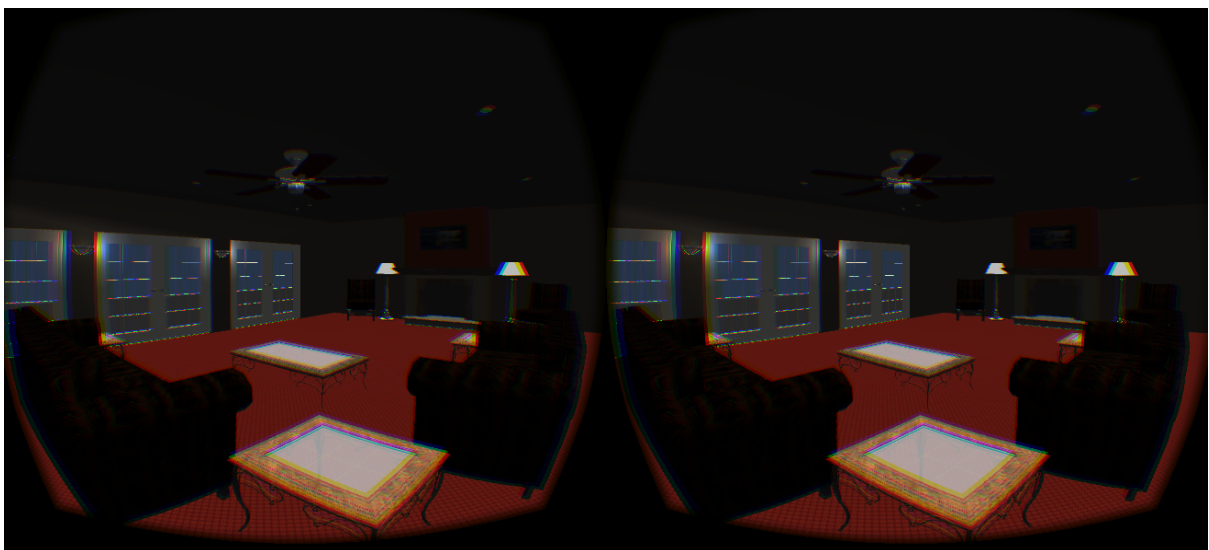


Figure 4: An example of a Oculus player view on 2D screen

Prefabs. Toolbox comes with a set of predefined and processed prefabs that are compliant with programs requirements. Predefined prefabs involve room and object models. Before defining rooms and objects an explanation about exits and inner connections is due. These two classes are basically identical, but separated in the toolbox for simplicity. Exits and inner connections are used to mark and connect different nodes in game room. Exits define to what can its parent connect to (rooms etc.) and inner connections define what modules can connect to it. Graphical examples can be seen from figures 5 and 6.

Room. Room is a large area which has one to N exits. Configurable properties of a room are its tags and amount. Tags define what the module is e.g. room or a corridor and the amount fixes the number of modules available for the generation algorithm to initiate. Main requirement for using a module is that it has to have exit markers which have predefined properties.

Currently new room models can be added through Unity, toolbox does not have this functionality although it provides needed scripts and instructions. Figure 5 displays an example room model.

New models can be added with three simple steps:

1. Add exit connections to doorways.
2. Add Unit script to the model.
3. Define tag values and amount.

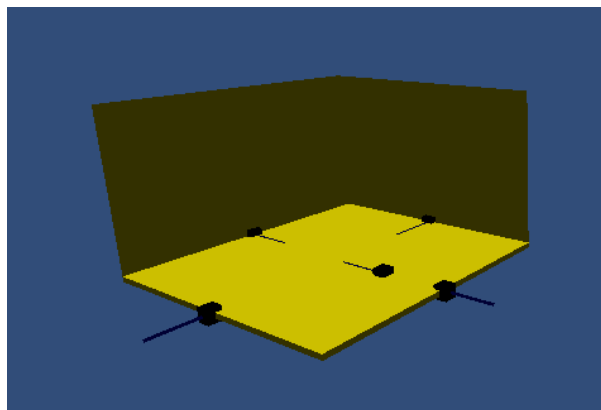


Figure 5: An example of room model with inner connections

Object. Object is an interior decoration model that is categorized by tag values. Figure 6 shows two types of objects that are currently available in the toolbox - ground objects and small objects. Ground objects have an inner connection by what it can be connected to the room and multiple small inner exits which enable to generate objects on top of each other.

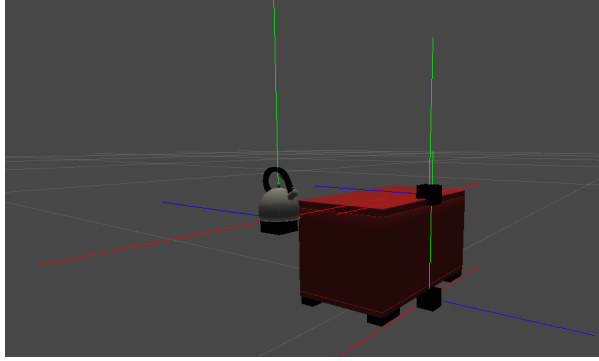


Figure 6: An example of a ground and a small object

New objects can be added through Unity. Toolbox does not have this functionality although it provides needed scripts and instructions.

Objects can be introduced with three simple steps:

1. Add inner connections to the object.
2. Add Unit script.
3. Define tag values and amount.

3.3.2 Thesis specific requirements

Functional requirements

1. Managing and creating memory experiments.
2. System enables random game room generation.
 - (a) Random multilevel room.
 - (b) Random object placement inside the rooms.
 - i. Objects that are predefined by the user are placed randomly.
3. Testing memory performance.
 - (a) Recognition.
 - i. Subject has to choose from a pile of objects (free choice).
 - ii. Objects are shown one by one and subject has to indicate whether it was in the room or not (yes or no).
 - (b) Recall.
 - i. Free recall.

Managing and creating memory experiments The main view for creating memory experiments can be seen from figure 7.

First, it is wise to select the test type (see below) according to the experiment in making. Next, experiment time limit is the amount of seconds that the subject is allowed to roam

the experiment room before the transition to the next level. Between time is the amount of seconds that the subject must waste in a level defined by the experimenter before the memorization control can start. Answer time limits the amount of seconds subject can answer provided questionnaire. Answer instructions enable to show a custom message to the subject on control canvas. If randomize objects field is checked control objects are picked randomly until there are no more objects in the collection. Otherwise objects are picked linearly meaning that the ones subject saw first will be shown first.

Lastly there is a possibility to include distractors. Distractors should be objects that were not included in object generation, otherwise conflicts arise. The aim of adding distractors is to control whether the subject is not bluffing. If there are no distractors added all the objects shown are correct. In case of location recognition experiment distractors may be added to a specific folder. When finished with filling out the fields and the create button is pushed, then the experiment is saved and ready to be used. It can be loaded from the main menu.

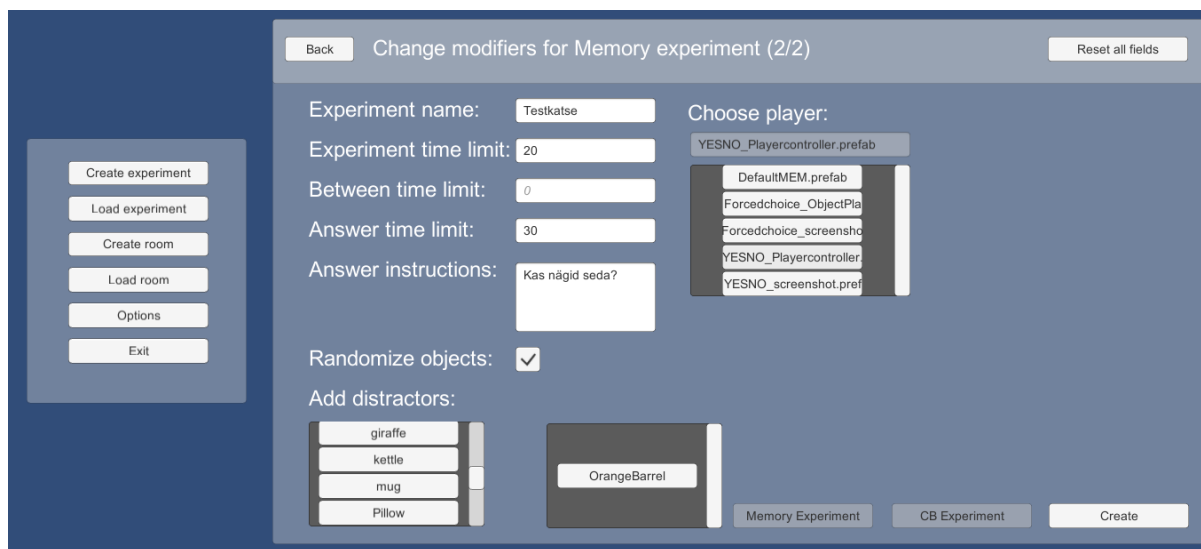


Figure 7: An example of memory experiment creation

Creating the game room. There are multiple algorithmic approaches for random game room generation [12]. For the purposes of memory experiments it was decided to use a treelike generation with prefabricated models.

When using the room generation algorithm, the user has to specify rooms used, their amount and connection rules. It should be noted that each generation process must have enough rooms with only one exit to complete the tree, otherwise the generation will go into infinite loop. Rooms with only one exits could be thought of as tree leaves.

The algorithm is capable of generating a multistorey game room with a variety of different room models. As an additional feature rooms can be assigned a script which generates objects inside the room. Objects and their count in a room group is also predefined by the user.

To create the game room successfully it has to have a set of rules. In case of room generation algorithm there are two types of modules defined.

- A room is a large area which has one to N exits.
- A corridor is a narrow section that has exactly two exits. These models are mainly sloped to create multiple levels.

Additional modules can be introduced by the user but it is essential for the algorithms work that objects are compliant with prefabs requirements.

Integration of the modules has to have a set of rules as well. These rules are held in exits. Exits main properties are its location and labels to what it can connect to. In this case, the set of rules does not have to be strict, although they still have a big impact on the outcome. To some extent integration rules are configurable by the user.

By default every model can connect to every other model. This makes the possible outcome of the generation more vast. However it comes with a cost. First, it was noted during testing that if there were a large number of corridors, the random model picker might select the same type of model multiple times in a row which creates a very long and valueless branch. This applies mainly to corridors. It could be fixed by introducing a restriction that no more than N corridors can be joint. On the other hand, it seems that a better solution to limit this behaviour is to change the integration rules such that corridors cannot connect to each other. Also limiting the amount of corridors to a smaller number would help. Figure 8 displays three consecutively generated rooms.

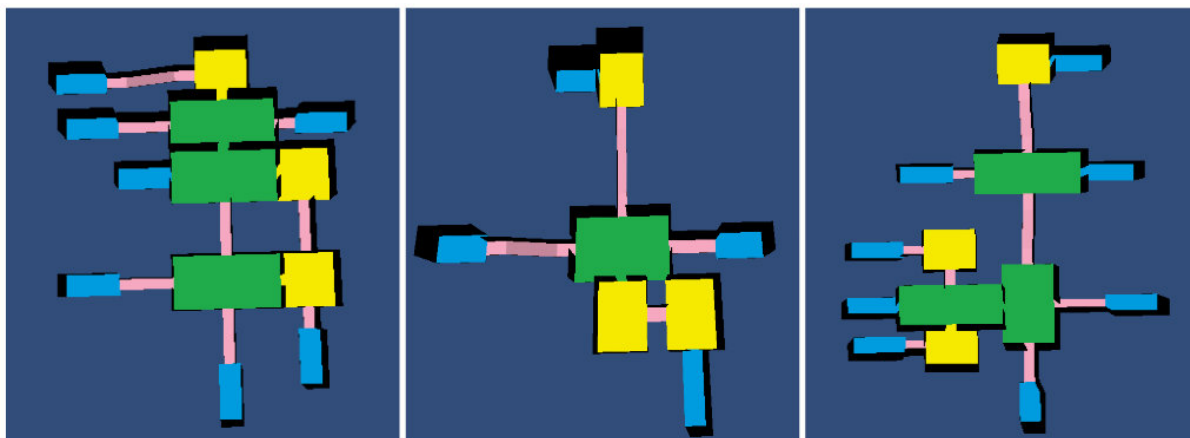


Figure 8: An example of three consecutively generated rooms

Game room building process is similar to breadth first search but it is used for N-ary tree initiation.

1. Start component is chosen randomly from user defined models and a recursive function is called.
2. Node's all available exits are filled with randomly chosen children.

3. New children are added to a collection.
4. Second point is called recursively for each child in the collection until it is a dead end module.

Connection of models makes use of exits requirement that its Z axis in Unity's world is pointed outwards from its parent, see first picture of figure 9. When two modules must be connected, then the one which is being added to the tree has to rotate around its Y axis with the rotation origin at its exits position to match the rotation of the exit to match. As a result two exits have opposite Z axes and matching Y axes, see third picture of figure 9.

After the rotation, translation of the new modules position must be done so that the new exit and old exit positions overlap and as a result of this process two modules are connected, see fourth image of figure 9.

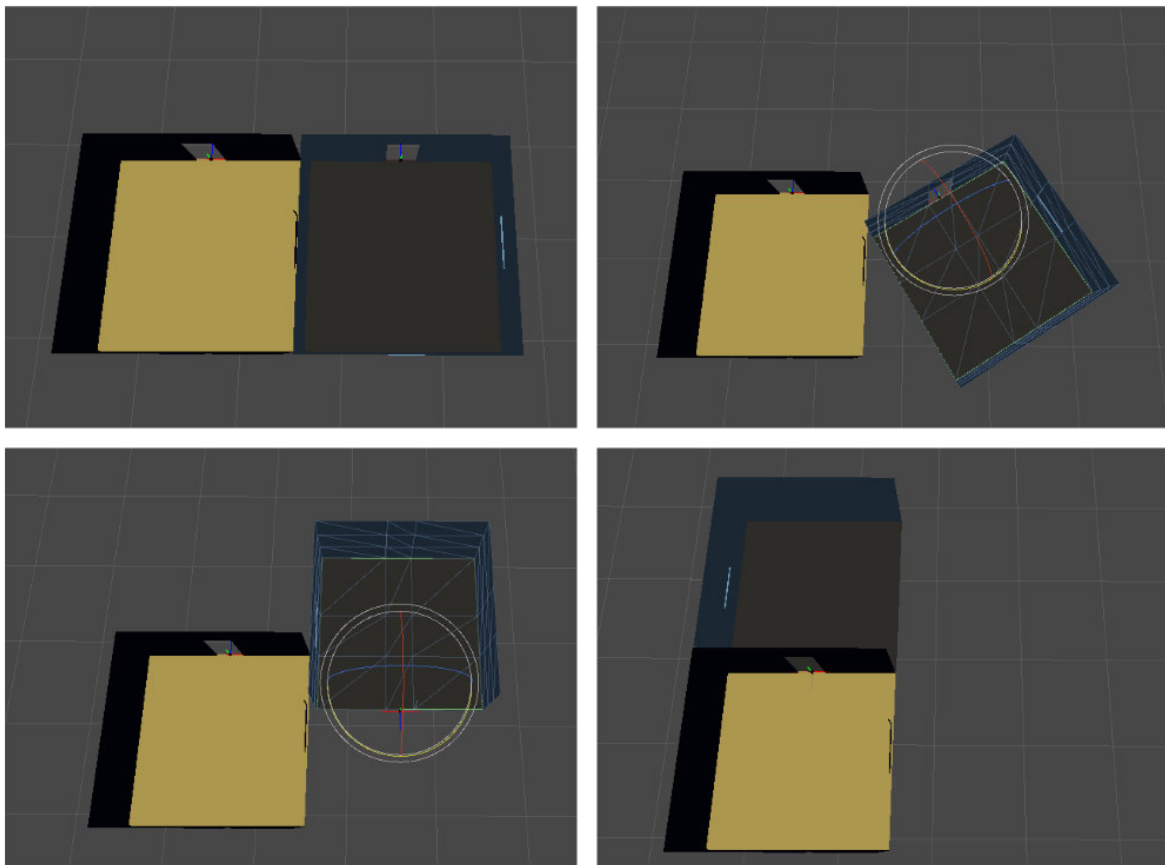


Figure 9: An example of connecting two rooms

Current implementation introduces some specific behaviours that should be taken into account. Since modules are always chosen randomly the actual amount of objects used in generation can differ from user set preferences. There is a chance that all exits are filled with a deadlock on the first iteration of the algorithm. Due to that user set amount should be interpreted as the maximum number of units that can be used. It also should

be noted that connection restrictions only work downward. For example, when parent models exit has a tag "room", meaning that only another room can connect to this exit the algorithm picks a random model with a matching tag and connects them. There is no check whether the random model's exit tags match to its parent.

When generations get substantial and multidimensional, collision of the modules becomes the main source of issues. Problem arises from the way collisions are detected in Unity. Since module initiation is done relatively fast, collision detection for the objects is triggered quite a lot of time after. Usually the colliding objects already have many children attached to them which means the algorithm has to undo all the work it has done with both colliding branches.

To address this problem modules have collision detector scripts which in case of collision delete all of the colliding objects children and in some cases itself. After that newly available exits are replaced with a door. A more proper solution would have been to fit different pieces to extend the game room before adding a door, but this is currently not essential to start with memory experiments as the generated worlds are big enough. Thus, the implementation is left for future research.

Since the generation is implemented as a tree it means loops are not tolerated, which is usually not a case in real buildings. Loops may occur by chance, but they are shut off with a door because collision is triggered. If that is the case then the doors can be removed manually, which allows to create loops artificially. In order to properly solve this problem additional features to the algorithm should be introduced.

Object Generation. For the purposes of object generation two approaches were implemented. First one did not use additional restrictions such as inner connections. Objects were randomly placed within the bounds of the parent. For this to work objects were marked to belong in a group. For example ground objects, wall objects and ceiling objects. Then the algorithm calculated size of the parent to be and picked a random position within the parent to place the object. A transparent indicator in the middle of the room marked to which side of the wall object should be placed. This approach was quite successful in a rectangular room with primitive objects such as cubes, cylinders and spheres. But as soon as more complex rooms and objects were included it was certain that current approach will not work. There were many problems with object scaling which ruined positioning of the objects.

Second approach to object generation is more fixed in terms of placement and implemented in a similar way to module generation. There are predefined inner connections in the module and on the objects which dictate what type of objects can connect to it. Currently the generation is done on two levels, meaning that there can be a maximum of two things on top of each other, but with little code addition this can be extended infinitely. During the first iteration of object generation ground objects are initiated and joined on the same principles as the room modules. Usually these objects have a high volume and might contain additional connections to where smaller objects may be placed. If this is the case, then newly created connections are listed as connections for small objects. Then during the second iteration of object generation the algorithm fills these newly created connections with user defined small objects. Figure 10 demonstrates

object generation inside three room models.

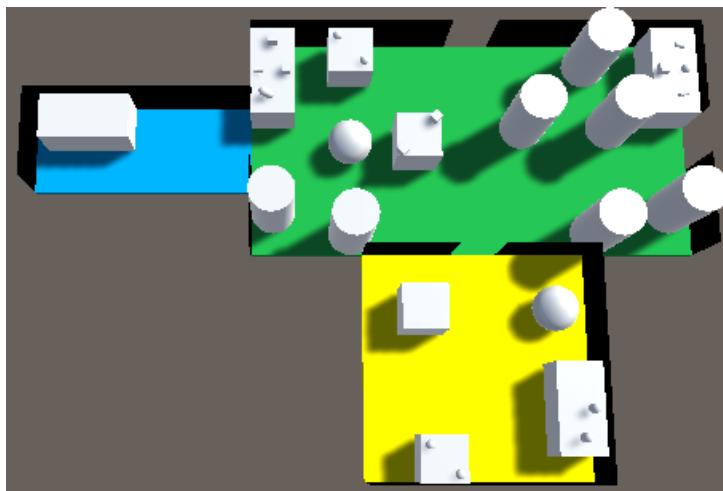


Figure 10: An example of object generation inside the modules

Testing memory performance. There are four types of memory tests that can be done using recognition. First one, shown in figure 11, is used to study the memorization of different objects that the subject encounters during the experiment. Objects are registered by raycasting from the center of users view. During the experiment constant rays are sent out and if the ray hits an interior object it is registered as seen. When the experiment is finished and it is time to test the memorization, registered objects are re-initiated and displayed to the subject. Thereon subject has to indicate whether the objects were seen or not.

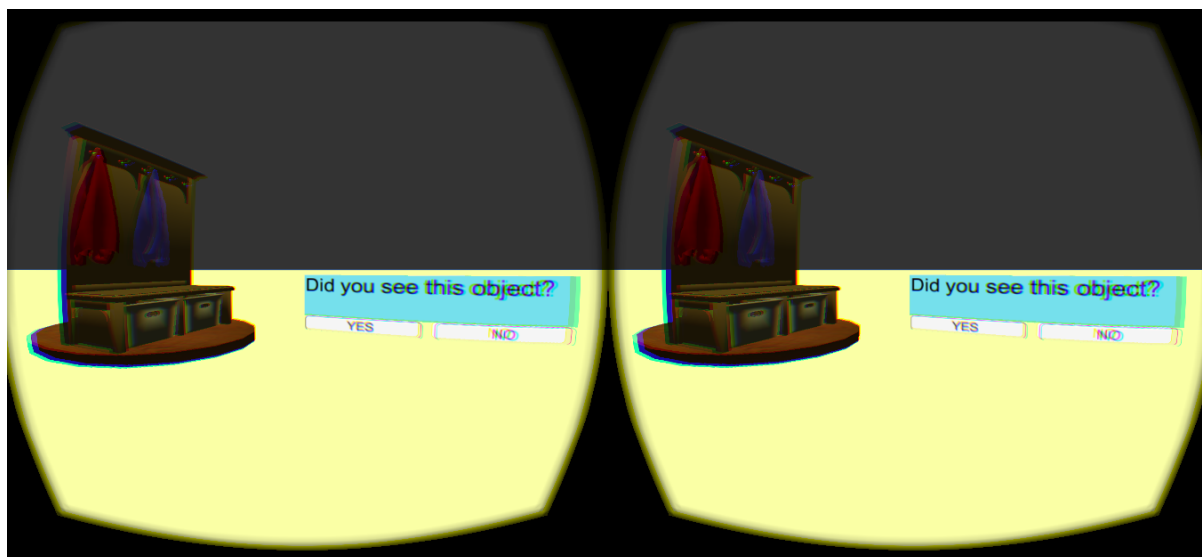


Figure 11: An example of object recognition player controller

Second one also studies the memorization of object, but in this case the test is done using forced choice. As a result all the objects the subject saw plus distractors, if some were

given, will be placed to test room. Then the subject has to choose from N objects which ones were seen. Figure 12 demonstrates testing phase using forced choice method. On the right image object is marked as seen and on the left it is not.

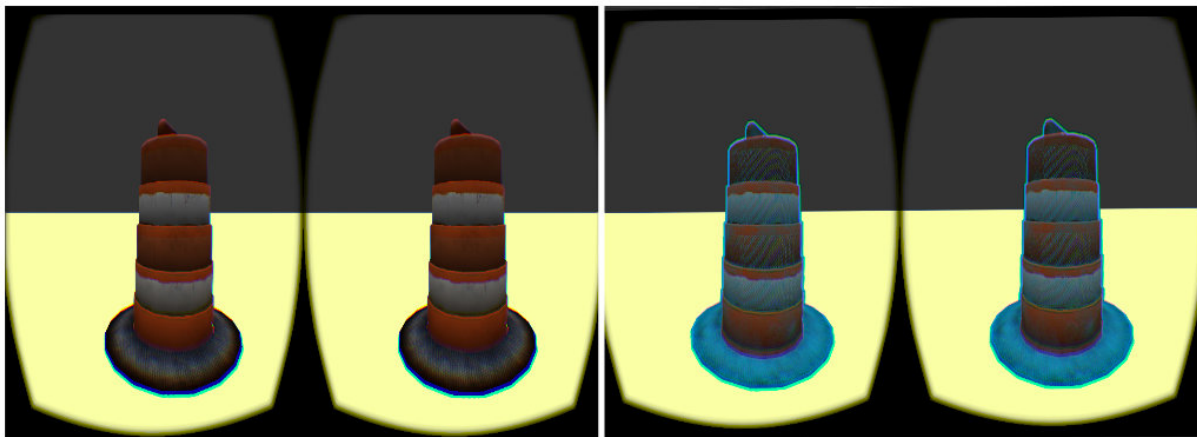


Figure 12: An example of forced choice player controller

Another research method, an example in figure 13, studies the memorization and recognition of visited locations. During the experiment subject is free to explore the rooms but unknowingly to the subject researcher is taking pictures of visited places. Oculus has a camera for each eye and similarly to human eyes both show a little different picture. To capture pictures from point of view of the subject left or right camera is used. During testing subject is shown different pictures of locations and asked whether one has been there.

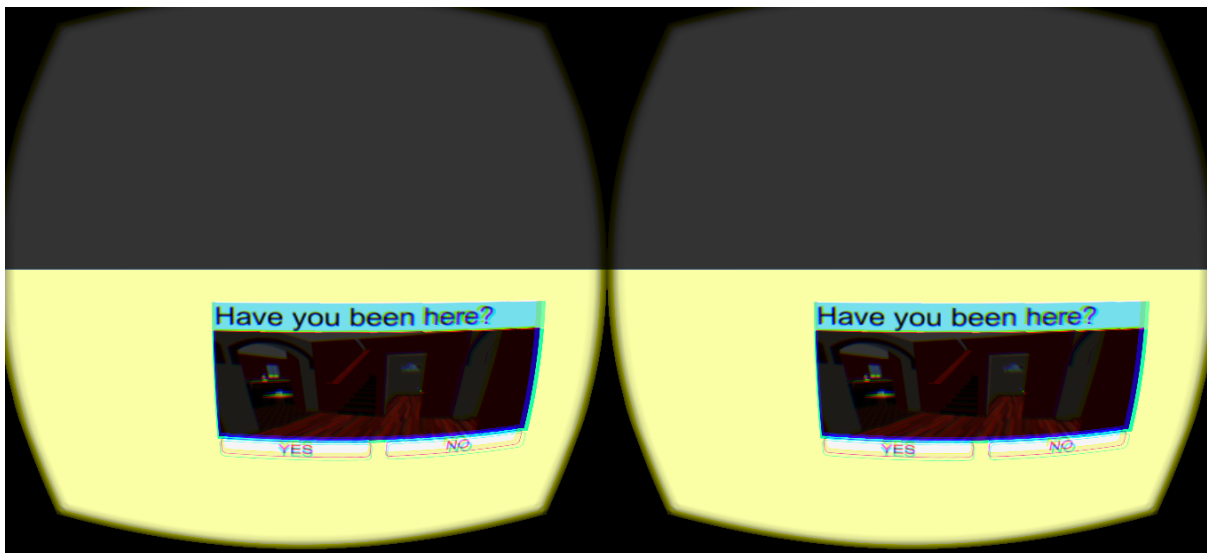


Figure 13: An example of location recognition player controller

As in the previous case location memorization also has a forced choice character controller. While testing pictures of locations are formed to a multiple choice form. This method

is deprecated, but still available as the user interface of performance testing might not always work as intended. Testing phase shown in figure 14.

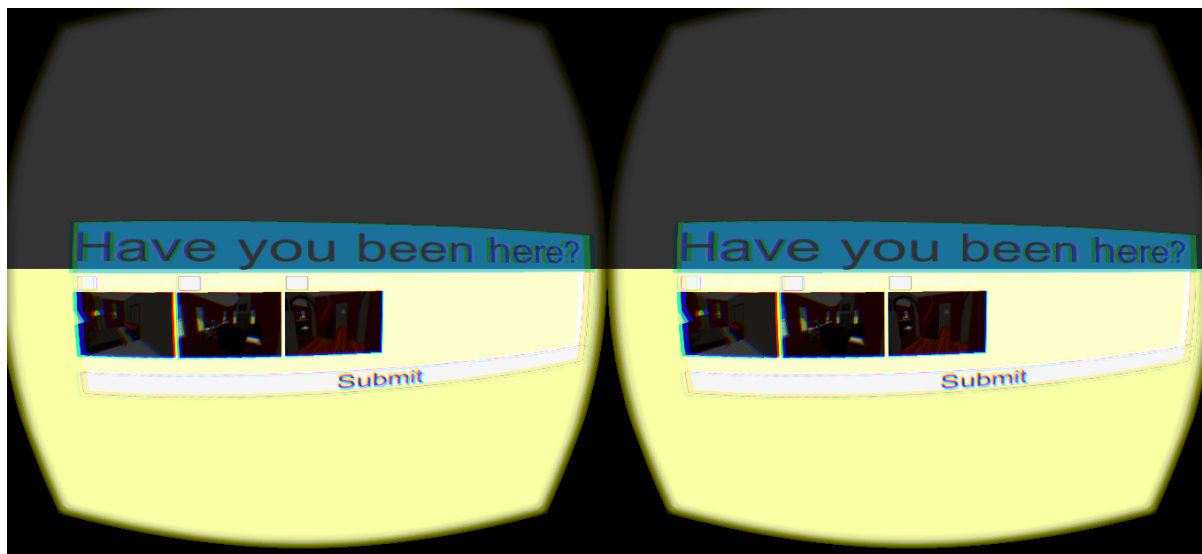


Figure 14: An example of location forced choice player controller

When using standard input devices such as a mouse and a keyboard the subject can answer the questionnaire by clicking appropriate buttons. But standard input devices are not always handy to navigate in 3D space comfortably. To operate in VR more naturally it has to have similar degrees of freedom to real life interactions. On this ground more experimental way to answer questionnaires is introduced. Katsuomi Kobayashi has come up with a solution to detect gestures with Oculus in VR environments [13]. This capability is added to the recognition player controllers as an experimental way to make VR experience even more immersive.

Recall is strongly tied to standard input methods and there is little to none automatic control. Subject can give feedback via keyboard or a microphone. When using the microphone recording time is assigned by the experimenter and recording can be ended on keypress. Files are saved to disk and named as dd/mm/yy hh:mm:ss according to the creation date and time.

Recall is implemented in a more classical experiment that is currently not part of the toolbox, but was initially developed for it. It will be available as an extra and is fully functional just lacks the user interface and must be used through Unity. User can configure word amount, pace by which new words are displayed and whether words are picked randomly or successively from the file. There is also a possibility to configure whether the display moves in sync with subjects head or is fixed in the environment.

3.4 Use cases

Creating a memory experiment

Actor: User

Precondition: There is at least one game room generated previously and current view is main menu.

Basic flow: User clicks create experiment button and a level choice view is displayed. User selects the experiment level and the waiting level. Memory experiment button is clicked and experiment configuration view is laid out. User chooses the testing method, fills in the appropriate fields and adds distractors. Create button is clicked thus the experiment is saved to disk and user is notified that creation was successful.

Alternate flow: User selects only one level and clicks memory experiment button. Fills in the appropriate fields, does not select any distractors and clicks create. By doing this memory experiment is created but there is no waiting level between the experiment and memory testing. Also there are no distractors included meaning that all the shown objects are correct ones.

See figures 7 and 15 for an example view.

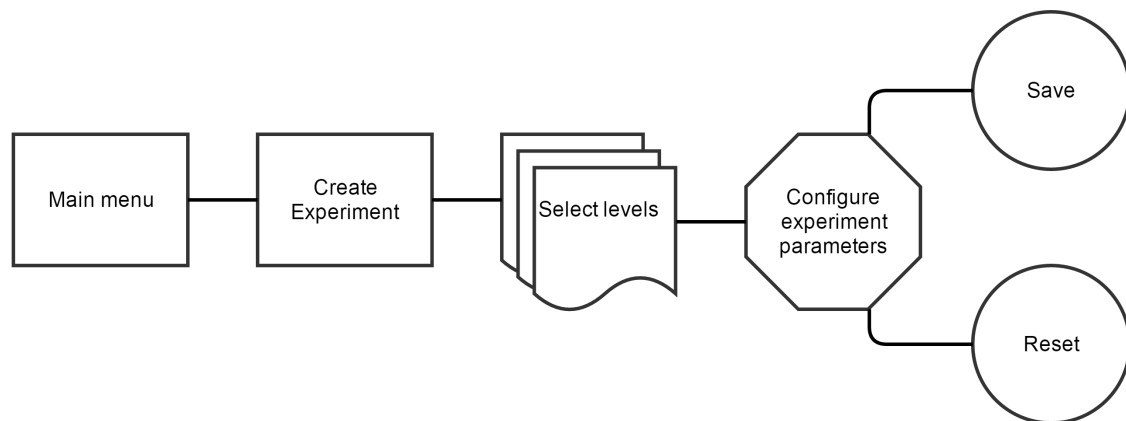


Figure 15: Steps to create an experiment (In the online version click for bigger image)

Loading memory experiment

Actor: User

Precondition: There is at least one experiment created previously and current view is main menu.

Basic flow: User clicks load experiment button and all created experiments are listed. User clicks on one of the experiments and detailed info about the experiment is shown. User clicks start button and experiment is loaded.

Alternate flow: After selection user clicks on test button and can test the experiment flow.

See figure 16 for a graphical example.

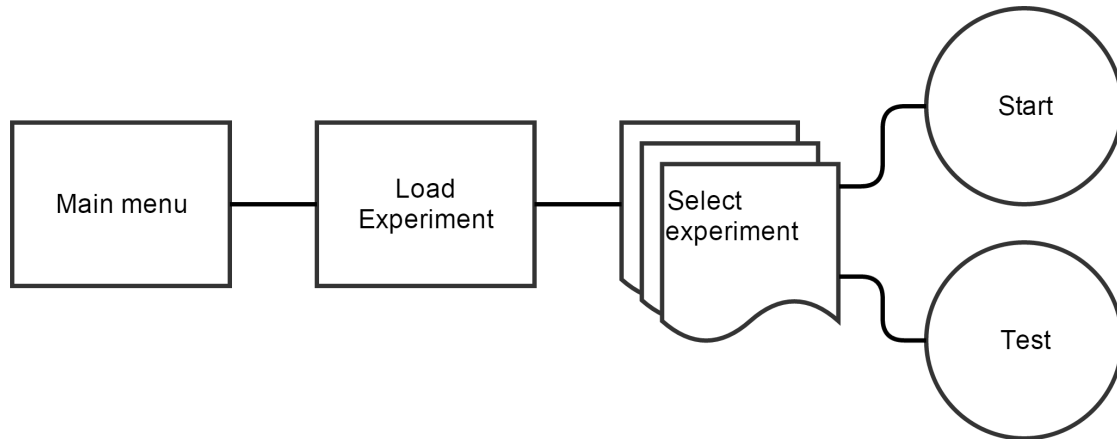


Figure 16: Steps to load an experiment (In the online version click for bigger image)

Creating a room for memory experiment

Actor: User

Precondition: Current view is main menu.

Basic flow: User clicks on create room button and then on generate room button. User enters the amount of models that can be used and clicks on open file button to select desired models. A model is displayed and can be configured using model settings. After the model is configured, user can add interior objects by clicking on add objects button. Objects are chosen using the open file button and configured through settings. When all required models have been chosen and configured generate room button is clicked. This generates a random game room and shows it to user. User can regenerate if needed. When room model is satisfactory interior is generated using the object generation button. Objects can also be regenerated if needed. When the whole game room is ready, user clicks on edit button which loads the level editor. Level editor enables to fine-tune the generated level. After the required changes have been made, user clicks on test button which enables to test out the environment in-game. If the results are good game room can be saved.

Alternate flow: After object generation is done the user finds that game room is not like it should be. Delete button is clicked and game room is deleted. User can return to main menu or regenerate.

Alternate flow: User clicks on save button instead of edit button. Thus the level is saved to disk and can be accessed later through the level editor.

See figures 17 and 18 for a graphical example.

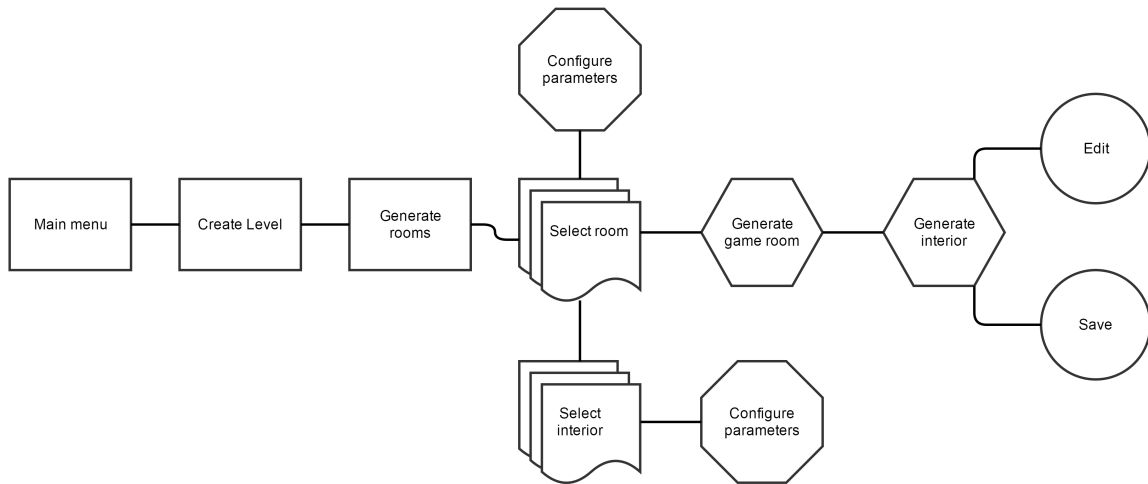


Figure 17: Steps to generate a game room (In the online version click for bigger image)

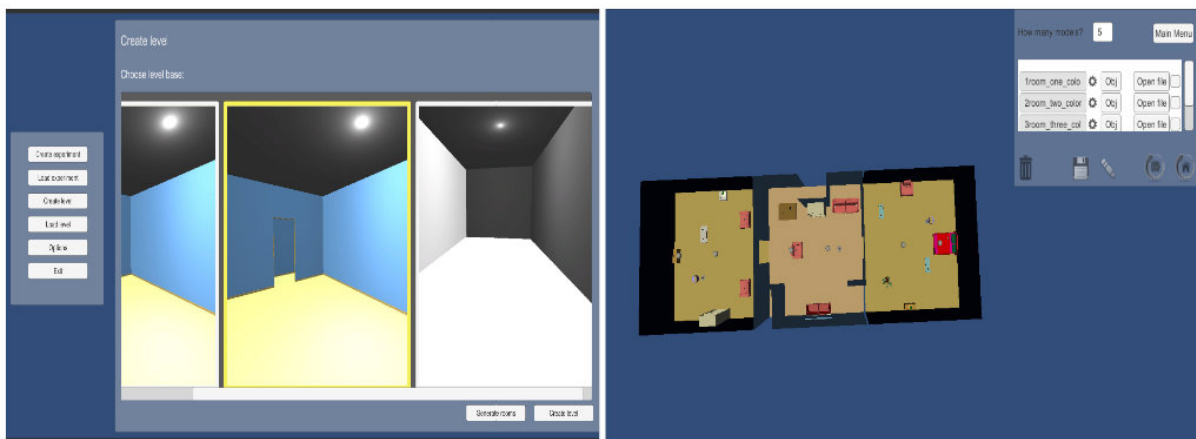


Figure 18: An example of room generation views

Load game room

Actor: User

Precondition: There is at least one room previously generated and current view is main menu.

Basic flow: User clicks on load room button and all created rooms are listed. User chooses the desired level and clicks on modify button. Level editor is loaded and user can fine-tune the generated level. User saves the modified level.

Alternate flow: User does not like the modifications and discards them.

See figure 19 for a graphical example.

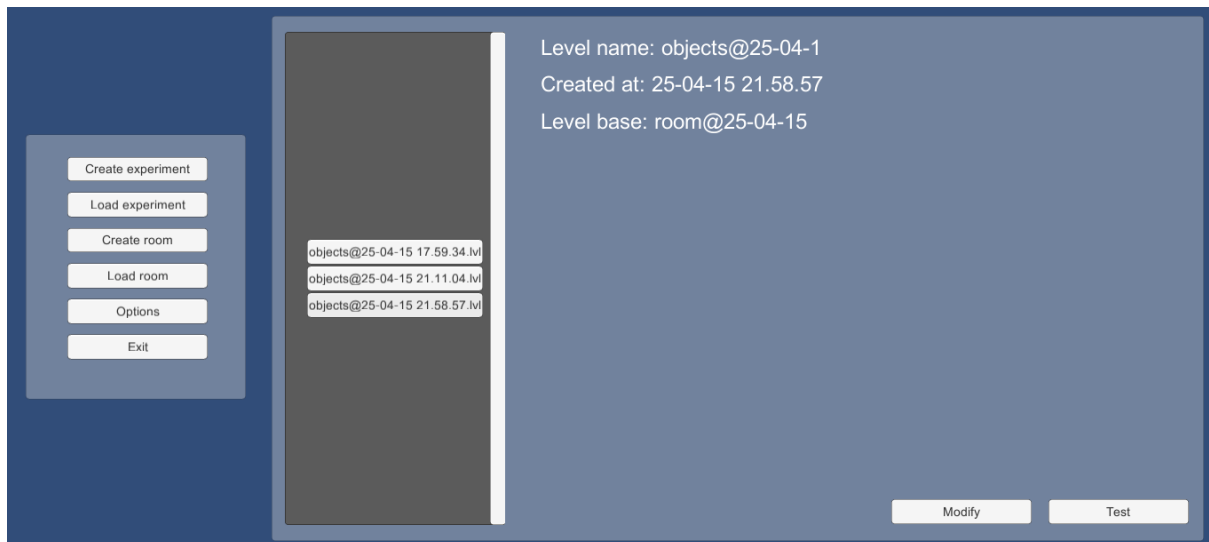


Figure 19: An example of room loading view.

An example of recognition experiment with objects

Subject is placed to a virtual environment that is specifically created for memory experiments. Participant is able to move within the game room and interact with different objects that one may encounter. For the purposes of object memorization all objects that fall into the center of the field of view are registered and saved for testing. When the timer has run out or the next level button is pressed, memory performance test will be displayed. During the test the subject is shown randomly picked objects one by one and asked to indicate by clicking yes or no whether it was there or not. When the test is finished results are calculated and saved to a file.

Alternatively testing can be done using forced choice method. In this case all encountered objects plus distractors are re-initiated and placed to the testing scene. Subject has to walk around the scene and mark which objects were present during the experiment.

An example of recognition experiment with location memorization

Subject is placed to a randomly generated virtual environment. Participant is able to move within the game room and interact with different objects during the experiment time or until next level key press. Unknowingly to the subject, during the roaming, researcher takes pictures of visited locations for testing purposes. Later the test will be displayed. While testing user is shown one by one randomly picked pictures of locations and asked to indicate by clicking yes or no whether one has been there or not. When the test is finished results are calculated and saved to a file.

Alternatively testing can use forced choice method. In this case all pictures are bundled to a multiple choice questionnaire and the subject has to toggle the fields which include the picture of a visited location.

An example of recall experiment

Good example of a recall experiment is the classical method of research - verbal memory test. User chooses a text file that contains words which should be memorized by the subject. Program parses the file and starts displaying words at a regular pace. After the testing, dependant on the chosen examination form the program analyzes given answers and saves results to disk.

4 General Discussion

As a result of this thesis a toolbox for creating memory experiments from ground up has been made. It should be noted that the output of the program is completely ready-made experiment it only lacks the subjects. It enables to create four different types of memory experiments.

These are:

1. Object recognition using forced choice method.
2. Object recognition using free choice method.
3. Location memorization using forced choice method.
4. Location memorization using free choice method.

Object recognition tests subject's ability to memorize and recognize encountered items usually in a fixed time frame. When testing the memorization, objects are shown one by one and the subject has to indicate whether it was seen or not. Thereupon researchers analyze the results.

Free choice method also tests subject's ability to memorize and recognize encountered items, but the testing method differs. While testing subject is shown all seen items at once and one has to mark which objects were seen. In both cases there is also a possibility to add distractors to testing phase which helps to ensure genuinity of the answers.

In case of location memorization subject can roam the game room freely. During the experiment researcher is able to capture pictures from subject's point of view which will be used for memory testing. In testing subject is presented with pictures of locations and asked to answer "yes, I have been here" or "no, I have not been here".

Free choice method bundles taken pictures and forms a multiple choice questionnaire. Subject has to toggle the fields that have a recognized location attached. And there is also a possibility to add location distractors which should be pictures of rooms not visited. Again, this is to ensure genuinity of the answers.

Additionally the toolbox includes a graphical user interface that helps to:

- Create experiments.
- Load experiments.
- Create rooms.
- Load rooms.
- Change settings.

Experiment creation form allows to configure necessary parameters and is relatively simple to use. It was displayed in figure 7.

When loading an experiment a list of experiments is populated. The user has to choose desired experiment and press start.

Room creation can be done automatically through room generator or manually using level editor. Room generator enables to connect random rooms and place interior objects across the generated rooms. Level editor enables to fine-tune generated rooms or create a completely new one.

There is also a number of pre-configured prefabs that can be used for room creation. But adding new prefabs is also quite easy.

The toolbox also provides means to create change blindness experiments. This part was developed by Markus Kängsepp [9].

4.1 Future works

User interface. Current user interface is very basic and does not please the eye. It serves the purpose of being a simple middleman between the user and toolbox's code. It is sufficient for creating simple memory experiments, but does not allow much custom configuration. In the future if the aim is highly configurable memory experiments, then the user interface becomes a liability and it may be reasonable to use Unity's editor instead of building a custom UI.

Random game room generation. First off it was noted that connection restrictions only work in downward direction. For greater lengths of modification it should be implemented such that restrictions would work both ways. Thus, it would be possible to configure connection rules per room group bases. For example, a room with just one exit can connect to only other rooms even if all other rooms that have more than one exit can connect to every other model. Possible usage scenarios would be to isolate special room types and avoid valueless branches.

Creating loops. As previously stated creating loops is restricted in the current implementation. This is quite a large setback in emulating real life architecture. One of the proposed solutions would be to select close-set exit pairs and try to build a path between the two. This requires a lot of additional complexity and a big assortment of different connection models.

Collision detection. On a related note collision detection of room models should be enhanced to use doors as a last resort. Instead, it should try to fit different models to extend the game room. To address this problem a different approach to collision detection is necessary since Unity's natural collision detection works with a delay. One solution would be to calculate whether the area to be occupied is free before initiating the model. Unfortunately this requires quite a lot of computing power.

Another solution would be to use previously mentioned loop creation solution. In this case, it is already known that colliding models or their parents are relatively close to each other thus they could be connected to avoid collision. Although this solution would only solve half of the problem as it does not enable to create bigger game rooms.

New methods. In collaboration with instructors new ways to study different memory related phenomenons using virtual reality have come forth. For example, it would be interesting to research the phenomenon of misplaced wallet or car keys using virtual reality. The task would be that the subject has three things to put away when arriving at a friend's house and after some extraneous activities subject has to recollect those items. Another intriguing subject is that doorways cause forgetting [14]. It would be interesting to see whether this phenomenon is reproducible in virtual reality and whether it can be countered by doing memorization tests without leaving the experiment room.

Object detection. In order to register that the subject has seen an object it needs to fall right in the center of subject's field of view. This implementation has a few setbacks since smaller objects might not fall into the center very often. This could be solved by casting multiple object detection rays from the player. Currently there is only one.

Natural movement. There is a possibility to direct humans' natural movement to virtual environment using a script made by Filipp Keks [15]. This would be a very interesting feature for memory player controller as it makes the experience even more realistic and unique. However it might distract the subject from performing the memorization.

New technology. Leap motion is quite popular technology that could contribute vastly to make the VR experience more immersive and realistic. Adding this gives the feeling of really being part of the virtual world. Leap enables to display virtual hands that mimic the movements of users hands. This could be used to improve 3D space navigation and memorization testing.

Graphics. There is also a lot of work that can be done to make the experiments graphically more advanced and realistic. For example, creating more detailed models.

4.2 Conclusion

It was brought forth that virtual environments make experiments ecologically more realistic and lifelike compared to standard research methods. There is better isolation and control over the subject's environment and it is relatively cheap resource and time wise. The product of this thesis enables researchers to carry out four different memory related experiments in virtual reality. It is a necessary step towards developing new and expanding existing memory research methods.

A Appendices

A.1 The toolbox

The toolbox is included as an online supplementary material. The material will be withheld from publication until 26.06.2016. Access to the material can be gained from [here](#).

The online supplementary contains the following files and folders:

1. "Read me" file to guide the users.
2. "toolboxfull.7z" contains the latest Unity archive.
3. "toolboxbuild.zip" contains a fully functional build.
4. "Use Cases.pdf" describes toolbox's use cases.
5. "User Manual.pdf" is a manual for the program.
6. "Recall experiment" contains:
 - (a) "Read me" contains info about the program.
 - (b) "recallexperiment.7z" contains the latest Unity archive.

References

- [1] E. Tulving, *Mälu*, M. Rauk, Ed. University of Tartu Press, 2002, translation to Estonian by Jüri Allik, 2002.
- [2] (2015, March) Oculus rift. Oculus Rift. Visited 18.03.2015. [Online]. Available: <https://www.oculus.com/dk2/>
- [3] J. J. J. LaViola, “A discussion of cybersickness in virtual environments,” *ACM SIGCHI Bulletin*, vol. 32, pp. 47–56, 2000.
- [4] (2015, February) Long-term memory. Wikipedia. Visited 17.02.2015. [Online]. Available: http://en.wikipedia.org/wiki/Long-term_memory
- [5] (2015, February) Methods used to study memory. Wikipedia. Visited 17.02.2015. [Online]. Available: http://en.wikipedia.org/wiki/Methods_used_to_study_memory
- [6] E. N. J. C. ELG Legge, CR Madan, “Building a memory palace in minutes: Equivalent memory performance using virtual versus conventional environments with the method of loci.” *Acta psychologica*, vol. Volume 141, p. 380 to 390, November 2012.
- [7] S. P. P. Gaën; Gyselinck, Valerie; Nicolas, “Age effect on components of episodic memory and feature binding: A virtual reality study,” *Neuropsychology*, vol. Vol 24(3), pp. 379–390, May 2010.
- [8] H. J. . R. A. R. Schultheis, M. T., “Virtual reality and neuropsychology: Upgrading the current tools.” *Journal of Head Trauma Rehabilitation*, vol. 17, pp. 379–394, 2002.
- [9] M. Kängsepp, “Virtual reality environment for change blindness experiments,” 2015.
- [10] (2015, May) Wikipedia. Wikipedia. Visited 01.05.2015. [Online]. Available: http://en.wikipedia.org/wiki/Human_eye#Field_of_view
- [11] (2015, May) Oculus. Oculus Rift. Visited 01.05.2015. [Online]. Available: <https://www.oculus.com/order/>
- [12] (2015, May) Tutplus. Tutplus. Visited 01.05.2015. [Online]. Available: <http://gamedevelopment.tutplus.com/tutorials/bake-your-own-3d-dungeons-with-procedural-recipes--gamedev-14360>
- [13] (2015, May) Riftgesture. Git. Visited 01.05.2015. [Online]. Available: <https://github.com/KatsuomiK/RiftGesture>
- [14] S. A. K. Gabriel A. Radvansky and A. K. Tamplin, “Walking through doorways causes forgetting: Further explorations,” *THE QUARTERLY JOURNAL OF EXPERIMENTAL PSYCHOLOGY*, 2011.
- [15] (2015, May) Oculus natural movement. Github. Visited 07.05.2015. [Online]. Available: <https://github.com/taphos/oculus-natural-movement>

Non-exclusive licence to reproduce thesis and make thesis public

I, Kälver Kilvits (date of birth: 29th of October 1992),

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:

1.1 reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

1.2 make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, as of 26.06.2016,

Toolbox for creating virtual reality environments for memory research
supervised by Jaan Aru and Raul Vicente.

2. I am aware of the fact that the author retains these rights.

3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, 14.05.2015