Game-Ready Inventory Systems for Virtual Reality

Marc Mußmann, Samuel Truman, and Sebastian von Mammen
Julius-Maximilians University
Würzburg, Germany
marc.mussmann@stud-mail.uni-wuerzburg.de, {samuel.truman, sebastian.von.mammen}@uni-wuerzburg.de

***Abstract*—In this paper we present a requirement analysis of inventory systems for virtual reality (VR) games as well as three according, integrative prototypes, and we compare them to state-of-the-art approaches. Our key contribution integrates solutions for selection and manipulation of inventory items as required by various categories of adventure games. Therefore, we follow a user-centered engineering approach to VR inventory systems design. We further improved the prototypes implementing proprioceptive interactions and adapting established user interface design practices such as the application of layout patterns.**

I. Introduction

VR games enable entirely new ways of interaction and new possibilities and challenges arise in the implementation of long-established systems in games [1]. For instance, an item in a traditional 2D adventure game is picked up by moving the mouse pointer atop and clicking, whereas current VR hardware allows to naturally grab objects based on 3D controllers or hand tracking. In order to allow the player to carry and use more than one such item, inventory systems have been designed and developed in various manners. Inventory systems are commonly used in games to expand the ways the player can interact with the world and enable more complex gameplay mechanics. And just like a grabbing interaction task, the design of inventory systems offers new opportunities but also has to consider new constraints in VR. An inventory system is a seminal part of most adventure games, which are characterized by the combination of storytelling, exploration, and solving puzzles. Adventure games are one of the oldest genres in video game history with the first published game Colossal Cave Adventure in 1976 [2]. Although the adventure game genre as a whole is agnostic towards input and output modalities as examples exist as pure text-based, graphical text, graphical, and point-and-click adventures, the concrete interaction mechanics and the collection of items in the environment play an elementary role in progressing in the games.

In this paper, we contribute to the resulting two questions: (1) What are the possibilities to integrate an inventory system as intuitively as possible in a VR game? (2) How can inventory systems be designed with the requirements of adventure games in mind? To tackle these questions, we will first consider a relatively broad spectrum of related works—from basic item selection in VR, over different menu designs, to the requirements that arise from the play mechanics in adventure games. Next, we detail a requirements analysis of inventory systems for VR with respect to play and interaction mechanics. We present three inventory system prototypes that fulfill the identified requirements in different ways and compare them with inventory systems from the literature. Finally, we discuss and classify the results before concluding the paper with an outlook on potential future work.

II. Related Works

A. VR Interactions

The user can interact in VR by means of various input methods. Most VR systems on the market use a head-mounted display (HMD) and a 6 degrees of freedom (6DoF) tracking controller for each hand [3]. The user can select and pick up virtual objects via gestures with the controllers, as opposed to a keypress in non-VR games. In an according study, Nukarien et al. [4] evaluated three techniques in terms of user experience and selection speed: point and keypress to confirm (HandTrigger), look at and keypress to confirm (GazeTrigger), or look at and dwell to confirm (GazeDwell). The HandTrigger technique achieved the best results which also relates best to our natural way of picking up objects. VR allows us greater freedom in interaction than reality, e.g. to pick up remote objects. To clarify the possible interaction processes, a taxonomy for object selection and manipulation techniques in VR was proposed [5] relying on the three interaction categories classes grasping, pointing, and hybrid [6]. Grasping encompasses techniques where the user picks up an object with their hand. A commonly used interaction technique within this class is the Simple Virtual Hands technique, where a virtual proxy object imitates the hand, and a target object can be picked up by pressing a button when the proxy object touches the target object. The pointing class is divided into two subclasses: vector-based and volume-based [5]. Vector-based techniques include, for example, the HandTrigger technique [4]. The Flashlight technique is an example of volume-based interaction. It allows selecting objects within a cone, starting from the hand. The hybrid class combines techniques from the other two classes. For example, the HOMER technique [7] uses a virtual laser beam for selection and the Simple Virtual Hands technique for manipulation. There is a relationship between modalities of interaction and problems like the Heisenberg effect (unintentional controller movement on keypress), fatigue, and cognitive load [5]. Interaction techniques differ in usability, user experience, and their suitability for different scenarios. In general, interaction techniques based on proprioception, which describes the perception of the position

978-1-6654-3886-5/21/$31.00 ©2021 IEEE

and location of the body and limbs relative to each other, bear many benefits [8]. These body-relative interactions provide real-world physical context, a more direct and precise sense of control, and enable "eyes off" interactions, that do not require continuous, proactive supervision of one's input.

B. VR Inventory Systems

Based on these interaction techniques, items can be used with inventories in VR for which different designs have been proposed in the literature. In [9] an abstract and a metaphor-based inventory system were presented. The abstract inventory is realized as a 2D menu, which is attached to the player's hand and can be switched on and off by pressing a button. The items are represented as icons arranged in a 3x3 grid and can be selected with the other hand. The abstract icon view allows the player to see all available items, which in turn requires the user to switch it off when not used to avoid cluttering the view. As an alternative, the metaphorical inventory was implemented as a revolving virtual belt featuring nine holders, similarly also featured in [10]. The common use of belt holders and this implementation's integration into the game environment promise greater immersion but the need for rotating the belt to reach items in combination with the fact that they are usually not in view might render it more difficult to use, at the same time. Both systems were evaluated for usability and spatial presence in a study with 24 participants. No significant difference was found between the two systems. In [11] three more concepts for inventory systems in VR were presented, i.e. the Flat Grid, the Virtual Drawers, and the Magnetic Surface techniques. Items in the Flat Grid inventory were arranged as icons in a 2D overlay at a static position $1.5m$ away from the player. The items can be manipulated using the HandTrigger technique. The Virtual Drawers system holds the items themselves normalized scale on a shelf that can be freely placed in space. The items can be arranged and used by means of the Simple Virtual Hands technique. Finally, the Magnetic Surface method allows the player to place objects on a plate that can be moved with a handle. Again, the player can interact with objects using the Simple Virtual Hands technique. According to the authors, the usefulness of these inventory systems highly depends on the application context and, for instance, on the need for well-aligned representations or clustered collections of items [11]. The same authors also suggested a classification for inventory systems in accordance with its interface (type, position, style), its representation (2D, 3D), its level of integration (abstract, integrated), item arrangement (layout/arrangement, capacity, improvements), and means of interaction (open/close the inventory, add/manipulate items). While the first few dimensions are self-explaining or immediately relate to the examples above, we briefly elaborate on the last two: For instance, as part of the layout, a linear quick bar can be offered for frequently accessed items [12] and grids for storing high numbers of items [13]. While it is clear that high capacities can quickly worsen the overview and comprehensibility of an inventory, automatic or manual arrangements of items can, to some extent, counterbalance this effect. Further improvements can be achieved, e.g. by introducing hierarchies, categories, or stackable items. These enhancements have rarely been implemented in VR games and need to be further investigated in terms of their usefulness. In terms of interactions, there are different approaches to accessing an inventory in VR. For example, the inventory can be opened and closed by pressing a button on the controller. However, there is also the possibility of integrating the inventory into the world, such as in the aforementioned virtual belt or by reaching into a virtual backpack behind one's head to add or remove items for a queued data structure [14].

For our survey, we also searched for complementary inventory implementations in commercial VR games. In [15], for example, items can be placed at a normalized scale at specific positions on one's palm. Ammunition can be stored away in a virtual backpack and are served contextually, dependent on a currently held weapon. Hand-held items can be selected by opening a menu upon pressing a controller button. It offers the available items as icons arranged on a cross. Releasing the button selects the item whose icon touches the player's hand. In [16], items are placed in a predetermined location in the inventory when collected. Each item has an individual maximum capacity. Furthermore, there is a certain area in the inventory where items can be moved to combine, or craft, them into a new item.

C. VR Menus

Due to similar interaction requirements, linear and radial menu systems in VR can be used as a basis for designing inventory systems in VR as well: Linear menus arrange items in rows or columns, oftentimes attached to the sides, e.g. the bottom, of one's view. Radial menus temporarily align the elements along a circle around the cursor or the controller. Such temporarily shown menus need to be activated—releasing a button pressed to activate the menu oftentimes confirms the selection of a touched item. Alternatively, turning the controller can be directly mapped to rotating the selection dial along a circle [17]. It takes longer to select an item on linear menus than on radial menus. However, no significant difference was found with respect to their general usability [18]. In [19], several item selection mechanisms for radial menus were compared: grabbing by hand (here, testing for collision with the index finger of Simple Virtual Hand), by ray (i.e. the HandTrigger method), or rotating by hand or by stick, similar to [17]. All the menus displayed a maximum of eight items and were attached to one of the two 3D controllers. In the grabbing-oriented designs, the menu was located at the non-dominant hand, allowing the items to be selected with the other hand. The rotation menus were located on the dominant hand. Small usability enhancements were introduced such as providing physical feedback on virtual touch by indentation of the respective button, or by augmenting the rotation of the controller to reduce the stress on the wrist. Despite overall scores of grabbing by hand were the highest, the rotation-based approaches dominated the personal preferences. In [20], another radial menu, the spin menu, was introduced, which is attached to the dominant hand and can be accessed by pressing a button on the controller. Here, rotating the controller rotates the items whereas the selection slot remains fixed. The items can be arranged in a quickly traversable hierarchy. To this end, a crossed layout displays each level in the hierarchy orthogonally to the previous one, a concentric layout displays different levels in parallel rings next to each other, with the innermost ring always representing the active level, and a stacked layout variant displays the levels on a stack, with the manipulable level at the top.

D. *Adventure Games Genre*

The selection of individual objects/items and their recovery from menus cover seminal aspects of inventory systems. But the analysis of inventories of adventure games reveals additional aspects. These story-based games typically feature at least one player character and focus on exploration and puzzle-solving, which are performed by means of interactions based on object manipulation and use, dialogues with non-player characters, and spatial navigation [2]. Adventure games have continuously been adapted to technological advancements to improve playability: Completely text-based (input commands and output feedback) adventures were complemented by illustrations as soon as the hardware permitted to do so. Typed text as input was largely replaced by point-and-click technology using a mouse, whereas clickable words in the UI were mostly replaced by icons, with only dialogue choices remaining in mainstream titles. Concerning the concrete interaction tasks of inventory systems, they include adding items, item manipulation (also introspection), the application of the items to objects outside the inventory [2, p. 19]. Usage of items requires an arbitrary number of targets, typically only up to one. Depending on the vocabulary of an offered UI, a concrete usage might have to be specified by the player, e.g. realized by a UI leaning towards a text-based approach, where first a transitive verb is selected and then an object, as in [21]. However, there are also implementations as in [22], where the object is selected first and then a viable action can be selected. The choice of action can also be realized by means of gesture-based controls [2]. Clearly, more choices expand the puzzle space potentially making the game harder to play but also more rewarding. While simply using a certain action with a certain item may already solve a puzzle, e.g. consider a key-and-lock puzzle where a key card needs to be found and applied as in [23]. Others might be combinations of simpler puzzles and can consist of, e.g., reconstructing a map from pieces or repairing a machine by correctly arranging spare parts as in [24].

III. Requirements Analysis

In the following, the dominating user requirements (UR) as well as the respective sub-requirements (SR) are presented. The requirements for game-ready inventory systems in VR that we inferred from the related works outlined in the previous section were refined in two unstructured review sessions. One of the authors who has ample experience in the design, development, and evaluation of games, including VR adventures, provided the feedback. In the first review meeting, the overarching UR and the SR to meet UR2 were approved. In this meeting, the wording, order, and hierarchy of the requirements were adjusted. In the second meeting, two weeks later, the remaining requirements were reviewed which resulted in the introduction of a new, dedicated user requirement to consider visual feedback (UR9). In Figure 1, we show UR1 as an example of the established the dependency hierarchies between UR and SR. At the end of the review process, the requirements were captured in nine UR (see Table I) and 116 SR.

At the very minimum, any inventory system must allow the user to select and manipulate items (UR1), add (UR2), remove (UR3) them from the inventory, and present all stored items to the user (UR4). To facilitate common game mechanics in VR adventure games, additional requirements must be fulfilled. In particular, the user must be able to select one or more items in the inventory (UR5) and perform actions on the selected items (UR6). Additionally, the user must be able to perform actions on an item utilizing another item in the inventory (UR7) and combine multiple items to create new ones (UR8). Furthermore, the player must receive feedback to stay informed about the status of the interactions (UR9). In the following paragraphs, we detail these UR in prose, also summarizing the vast collection of 116 SR.

A. *Interaction with Items*

Selection and manipulation, especially movement, of items are of central importance to adventure game mechanics. In this context, interactions can be implemented in VR using various techniques. It is essential, that visual feedback is provided to show the currently selected item to the player. In order to select and manipulate items, the user must either be able to grab items, e.g. by using a controller or use pointing techniques. Both variants relate to the classification of interaction techniques described in Section II. In order to grab items using a controller, it must be possible to select an item when it is within reach. Also, it must be possible to align the item's transformation with the controller's transformation while pressing a button. Fulfilling these requirements realises the Virtual Hands technique (see Section II). Alternatively, selecting items via pointing can be implemented by means of a ray-casting technique or by overlapping an item with a cone-shaped volume, starting from e.g. the controller. Since pointing techniques are rarely used for manipulation [5], they are listed here only as a requirement for selection.

B. *Adding Items to the Inventory*

Selecting an item and pressing a button is a simple interaction task to add the item to the inventory. An alternative is explicitly picking up the item and placing it in a specified "drop" area. This area can either be a position on the user's body, e.g. a holder on a virtual belt, the drop zone of a virtual backpack, the hips or backside of one's hands, etc. or an empty space in the inventory. These examples clarify

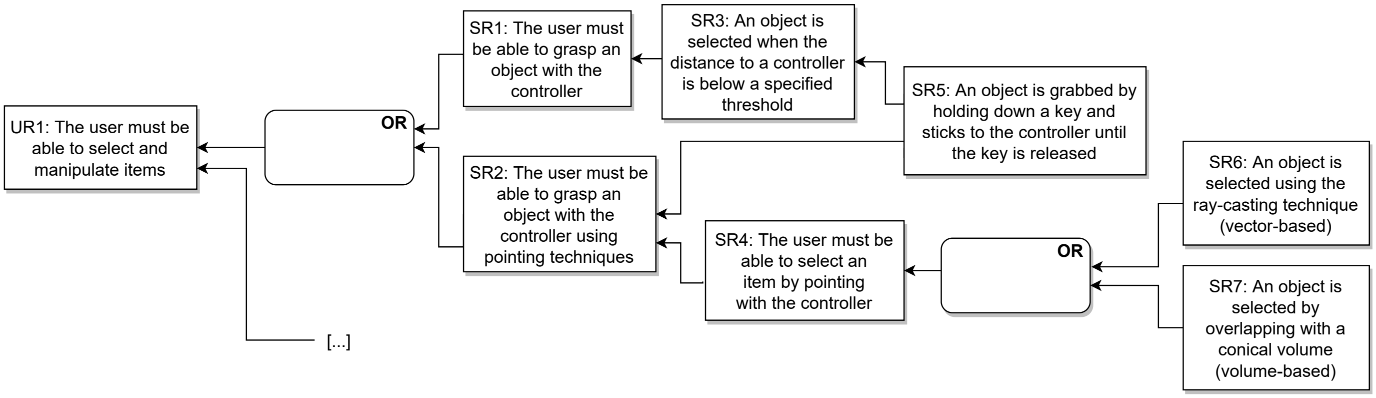

SR1: The user must be able to grasp an object with the controller
SR3: An object is selected when the distance to a controller is below a specified threshold
SR5: An object is grabbed by holding down a key and sticks to the controller until the key is released
UR1: The user must be able to select and manipulate items
OR
SR2: The user must be able to grasp an object with the controller using pointing techniques
SR6: An object is selected using the ray-casting technique (vector-based)
SR4: The user must be able to select an item by pointing with the controller
OR
SR7: An object is selected by overlapping with a conical volume (volume-based)
[...]

Fig. 1. The requirements hierarchy for UR1. Lower-level requirements must be met to fulfill higher-level requirements. Sometimes there are alternatives ("OR"). Another UR (UR9) that is related to UR1 is hinted at by "[...]".

that if the area is on the user's body, the item can either be removed from the scene and displayed in the inventory instead, or it simply remains in the area and is considered an entry of the inventory. Such drop areas allow for body-relational interactions, which are more effective than interactions based purely on visual information [8] and thus can be achieved quickly through natural movements, as demonstrated in [15] (also see Section II). In order to place items in the inventory, empty slots for items must exist. This requirement is, for instance, satisfied by the Virtual Drawers design, where a limited set of compartments exist in which the user can place items (see Section II). An alternative would be free space that is dynamically created as soon as the held item overlaps the inventory. In this case, predefined fixed positions are unnecessary like in the Magnetic Surface inventor (Section II).

C. Removing Items from the Inventory

To remove an item from the inventory, upon selection, the user must be able to either press a button to drop it or select the drop action from the list of available actions, as can be seen in [25]. Especially in the case of body-relative inventories, gestures are an apt method for item removal. Here, the user can take out items by grabbing and pulling them from their designated area, e.g. from behind one's head as in [15] (see Section II). Alternatively, all items in the inventory could be visible and grabbable by the user, possibly by means of scaled-down clones of the items. To take an item out of the inventory, the user could then directly grab it and place it somewhere else. This can also be seen in the Virtual Drawers inventory design [11] (see Section II).

D. Inventory Representation

Following the taxonomy of inventory systems as outlined in Section II, the means to bringing up the inventory, positioning it in the virtual environment, as well as representing and arranging its items are categorical dimensions of inventories. Some inventories are constantly visible, some can be opened and closed. The interaction for opening and closing the inventory can be triggered either by a keystroke/press of a button or by performing a gesture. For instance, the inventory can be brought out from behind the head by grabbing it with a controller and closed again by letting go of the inventory at that spot. Another possible gesture for calling up the inventory would be to move the hand as if looking at a wristwatch. An inventory must either have a fixed position or can be freely positioned by the user. To enable free positioning, the user can use some kind of handle to move the inventory, as in the Magnetic Surface inventory [11]. If the position is fixed, the inventory can either be placed at a position relative to the user, as with the Flat Grid Inventory [11], or attached to the user's body. If the inventory is attached to the body, it can e.g. be positioned at either controller, as in the Spin Menu [20]. Certain use cases might require additional information displayed with the items. This can be done by displaying a text field for each item, as it is done for example in the 2D inventory in [9]. Items can be represented in the inventory either in an abstract or realistic manner. For abstract representations, icons and a 2D overlay could be used. To represent items realistically, their copies can be rendered at a different, normalized scale as in the Virtual Drawers inventory. Finally, the order and arrangement of items must be specified for each inventory system. Items in the inventory can either be sorted according to certain properties or can be reordered manually by the user. In addition, the items must be arranged according to a certain layout. The items can be arranged uniformly or, like in [15], with different layouts according to this category. Many different layout designs exist. Items can be arranged in a pie or spin menu, or in a linear or grid layout as described in Section II. Other examples are the virtual belt inventory and the Magnetic Surface inventory (see Section II). Certain game mechanics require the inventory to have a specified capacity. The inventory can have an infinite number of slots, a fixed number of slots, or a dynamic, often increasing number of slots. Additional interactions like scrolling might have to be implemented for a large number of item slots. Items of the same type can either be stacked in one place or they can have their own place in the inventory. In case items are stacked, their numbers must be displayed to the user.

E. Selecting Items in the Inventory

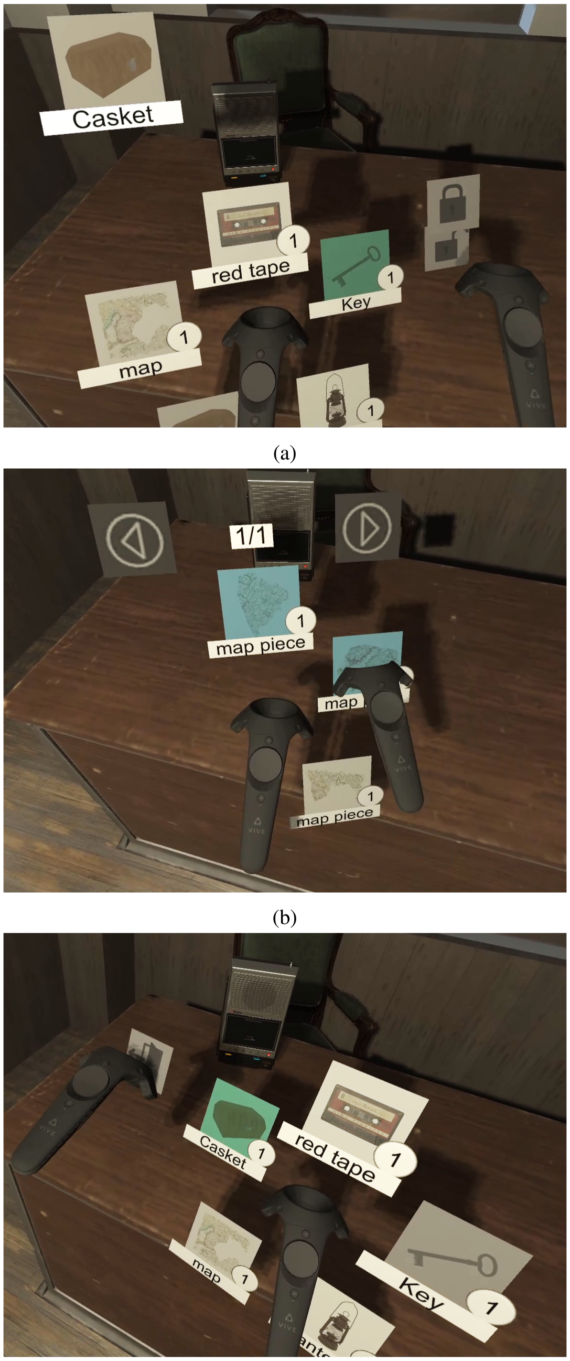
The ability to select one or more items in the inventory is required for several features such as reordering, combining, or removing items from the inventory. In order to confirm such a multi-selection, the user might press a button, as in the VR menu that realises grabbing by means of raycasting [19] (see Section II). However, the selection can also be confirmed by touching the representation of the item, as implemented in the hand-based grabbing version of the VR menu [19]. There are several concepts on how to select an item. An item in the inventory can be determined by an always active selection, which can be moved either with the thumbstick (stick rotation [19]) or by rotating the controller (hand rotation [19]). Alternatively, the selection is only active when the user selects an item, e.g. by pointing or by means of the Simple Virtual Hands technique (see Section II). The selection technique must be augmented to allow the user to select multiple items. For example, the user could hold down a button and then continue to select an arbitrary number of items or use gestures such as a rubber band selection. It is important to provide feedback about which items are selected, e.g. by highlighting them. Finally, it must be possible to deselect selected items.

F. Performing Actions on Items

In order to perform an action on one or a set of items, it must be possible to select and confirm the action. One approach is to only show the available actions once an item has been selected. This way, only those actions are displayed that are available for the item. Another approach is to show all possible actions before an item has been selected. Available actions can have 3D representations, or be visualized as 2D icons like in [22], or as text like in [21]. Some actions require the consideration of another item. To invoke such actions, it must be possible to determine which second item can be selected, e.g. by highlighting and providing feedback when an incompatible item has been selected. It must be possible to cancel the selection of a second item. Some adventure game mechanics like State of Affairs puzzles require the user to combine multiple items in the inventory to create a new item. First, possible combinations must be defined by developers. A simple implementation allows the user to combine selected items by pressing a button. An alternative approach requires the user to place the items at a specific location in the inventory, as in [16]. The user must be able to confirm the combination process, e.g. by means of pressing a button. It is vital to provide feedback on the state of interactions to the user. Visual feedback can be provided to the user by means of a 3D cursor, which is, e.g. represented as a virtual hand or laser beam. The cursor can vary its visual properties depending on the status of the interaction. Visual feedback can also be provided by highlighting the interaction target, changing its color, or displaying a shadow object based on the interaction status [5].

IV. Prototypes

In this section, we present three integrative prototypes based on the requirements detailed in the previous section, with a special focus on the selection and manipulation of items, as well as proprioceptive interactions. Afterwards, their implementation is discussed.



(a)

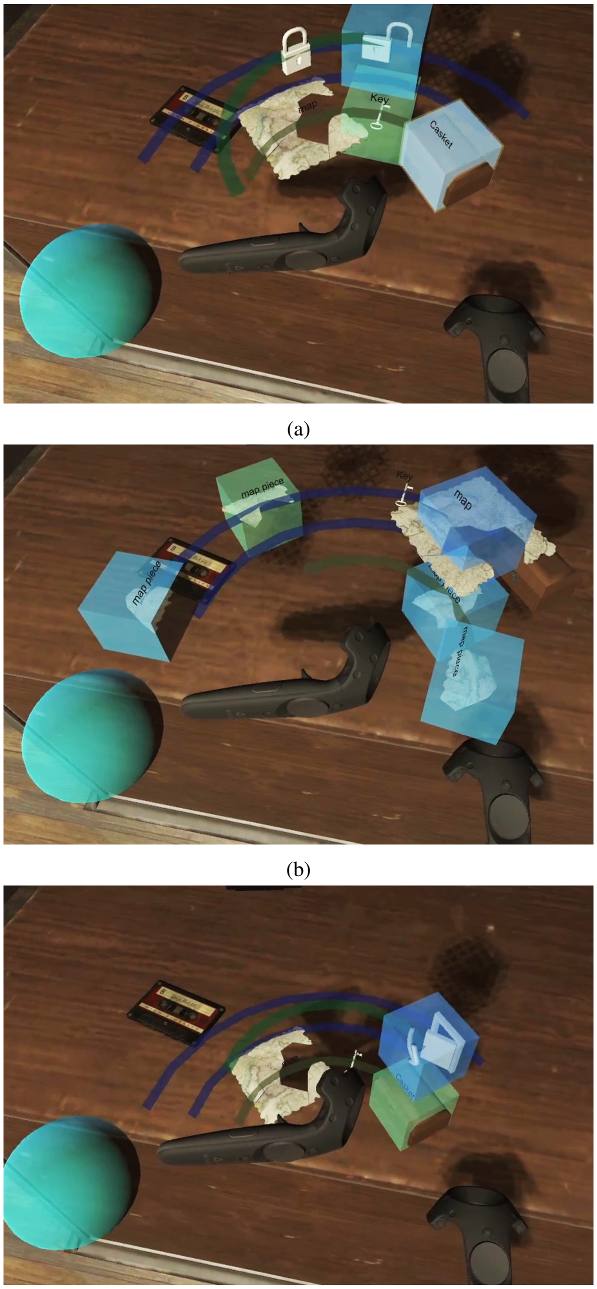
(b)

(c)

Fig. 2. The first prototype is based on a pie menu layout.

Fig. 2 shows the first prototype concept that is based on a pie menu layout [1]. Items can be grabbed using the Simple Virtual Hands technique, placed there by overlapping with the inventory, and retrieved by grabbing the representation. The inventory is accessed by pressing a button and is attached to the non-dominant hand. The items are displayed as 2D icons unsorted in a 2D overlay. Furthermore, items are displayed in a pie menu layout with multiple pages, each displaying a fixed number of entries. Items in the inventory can be selected by pressing on them (green background in Fig. 2(a)), and actions are displayed on the selected item as 2D icons, e.g. the locks hovering above the key in Fig. 2(a). Actions that require a second item to be selected are displayed in an extra menu, e.g. the casket in Fig. 2(a). In addition, a selectable representation of the possible combination, based on the selected items, is displayed in the inventory (multiple map pieces in Fig. 2(b)). For convenience, a button to close the menu is offered, seen in Fig. 2(c). Visual feedback is implemented by changing the color of the target item. Advantages of this inventory design are low cognitive complexity and its familiarity due to its similarity with traditional, non-VR inventory systems. This is our only prototype that represents items via abstract, 2D icons. Selecting items via touch feels intuitive. This inventory design is easy to understand and use by inexperienced users. A large number of items can be stored by utilizing multiple pages and stacking items of the same type.

The second prototype concept shown in Fig. 3 is based on the Spin Menu layout and combines the stacked item hierarchies with the concentric layout. The items can be grasped using the Simple Virtual Hands technique and can be placed or removed from the inventory by releasing them in an area on the non-dominant hand. The inventory can be opened using a gesture and is attached to the non-dominant hand. The items are displayed as thumbnails with a 3D representation. In addition, the items are sorted alphabetically by name (casket, key, map, from right to left in Fig. 3(a)) and there is a maximum number of non-stackable items. The selection is done by means of a moveable selection via the thumbstick. Actions are displayed on the selected item, e.g. the key in Fig. 3(a). When an item is selected, an additional item can be selected the same way as the first. Fig. 3(a) shows how a casket is locked using a key. A selectable representation of the possible combination is displayed when the required items are selected. For example, two map pieces are combined into a map in Fig. 3(b). Visual feedback is implemented by a shadow, i.e. transparent representation of the original item, on the target. The inventory can be closed by selecting the according thumbnail (Fig. 3(c)). Although both the pie menu and this spin menu-based inventory are both radial, following the best practice results presented in Section II, they differ in several ways, concerning the performance of access and selection, dealing with hierarchies and item compositions, and visualisation of the items themselves.



(a)

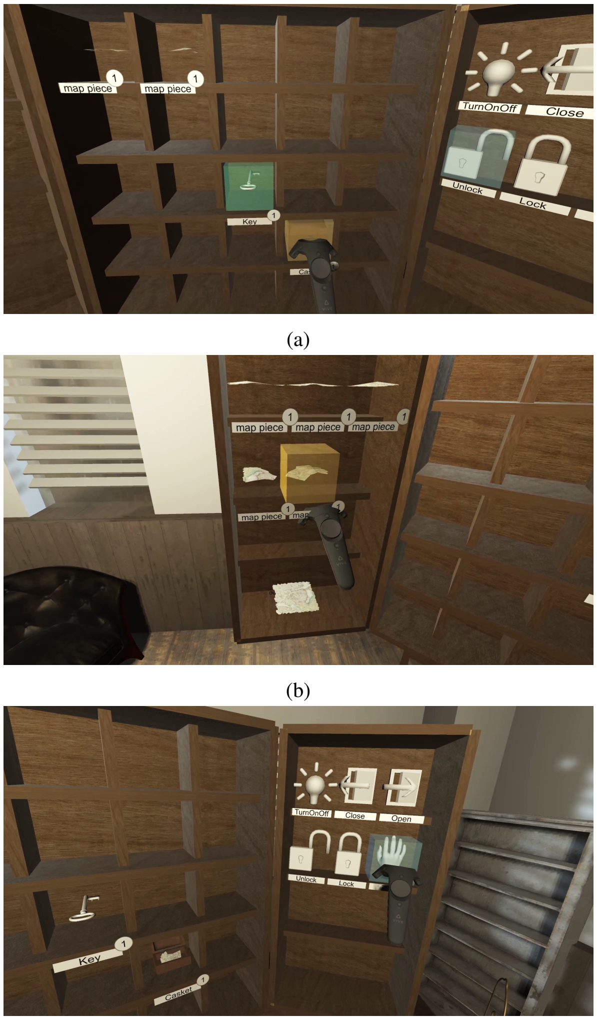
(b)

(c)

Fig. 3. The second prototype based on the Spin Menu layout.

The third prototype concept is based on the Virtual Drawers inventory (see Fig. 4). Items can be grabbed using the Simple Virtual Hands technique and placed in or removed from the individual compartments of the inventory (Fig. 4(a)). Displaying the inventory is implemented as in the pull-down menu, where the inventory can be activated by means of a pull-down gesture above one's field of view. The items are displayed in the compartments with normalized scaling as a 3D representation. In the inventory, the items are unsorted with a maximum number of individual items, though they can be stacked. Selection is done by pressing a button when the controller overlaps with an item. Actions are displayed in a 3D representation on a shelf to the right of the items (Fig. 4(c)). All actions are displayed regardless of the selected item. A second item can be selected

[1] We applied the Photoshop auto improve brightness and contrast function to all figures.



(a)

(b)

(c)

Fig. 4. The third prototype based on the Virtual Drawers inventory.

in the same manner as the first one. Items can be combined by placing them on an additional shelf to the left (Fig. 4(b)). Possible item combination are highlighted and the combination process is confirmed by grasping the respective item. Visual feedback is implemented by a shadow item at the target item. Anecdotally, this design induced greater immersion compared to the two radial prototype designs as all the items, actions and compositions are explicitly represented. It also gives the user to most control regarding the order of items. Yet, the number of distinct types of items is fixed and the required view real estate is rather high.

All three prototypes were implemented in the Unity engine. The XR Interaction Toolkit is used to integrate the VR devices. The implementation of the prototypes was tested with the HTC Vive. All implementations of the prototypes use a common database in which items with associated actions and the possible combinations are stored. In addition, the prototypes can be tested in a demo level, where the requirements for interaction with items in the adventure games genre have to be used to solve a State of Affairs puzzle.

V. Results & Discussion

In this section, we compare our prototypes with VR inventory systems from the literature on the basis of the formulated requirements. The systems to be compared are the Magnetic Surface, Virtual Drawers, and Flat Grid inventory [11] and the Virtual Belt and hand-held inventory menu inventory [9]. The features of the hand-held inventory menu are partly inferred from an image. The results can be seen in Table I. All but one inventory systems rely on 3D grasping techniques, using the Simple Virtual Hands technique. The exception is the Flat Grid inventory, where a hybrid technique is used. There are few differences when placing items in the inventories. All systems allow the user to add and remove items from the inventory. The flat Grid inventory utilizes a pointing technique to select items. Almost all systems from the literature allow the user to open and close the inventory using a button, whereas our prototypes also explore gesture-based interactions. The previously published systems as well as our prototypes are attached to the user's body or can be freely positioned in the virtual world by the user. All systems use different display strategies. Two of the systems from the literature display additional information about the items in the inventory. In contrast to the existing systems, our prototypes implement extensions such as stacking, sorting, and scrolling. All presented systems use different layouts, whereby the arrangement of the items in a grid layout is used most frequently in the comparison. With exception of the hand-held inventory menu, the systems from the literature do not explicitly allow the user to select items in the inventory. Instead, the user picks up items by grabbing the representation in the inventory. The hand-held inventory menu allows the user to select an item, but not multiple items at once. All systems but the Magnetic Surface and Virtual Drawers provide some kind of visual feedback to the user.

Although the given implementations are similar regarding item interactions, our prototypes extend the functionality of existing inventory systems for the application in adventure games. In particular, they provide the infrastructure to select and manipulate multiple items. By resorting to concrete VR menu approaches, our prototypes also expanded the design space of VR inventory systems in general. Our comparison highlights how inventory systems can be enhanced by body-relational interactions especially when activating an inventory, as well as when placing and removing items. Such proprioceptive approaches result in better integration of the inventory into the virtual environment.

VI. Conclusion & Future Work

In this paper, we presented a survey of item selection approaches in VR including according menu designs, inventory systems for VR and the use of inventories in adventure games. These findings were the foundation to blast off a user-centred engineering process during which we refined the requirements and developed according solutions that, for now, were only tested in terms of their immediate functionality and not subjected to extensive user studies, yet. In particular, we presented a radial VR inventory system augmented with

TABLE I
THE COMPARED INVENTORY SYSTEMS AND THE USER REQUIREMENTS THEY FULFILL

	Magnetic Surface	Virtual Drawers	Flat Grid	Virtual Belt	Hand-held	Our Prototypes 1-3
Select and manipulate items (UR1)	✓	✓	✓	✓	✓	✓
Add items to inventory (UR2)	✓	✓	✓	✓	✓	✓
Remove items from inventory (UR3)	✓	✓	✓	✓	✓	✓
Show stored items (UR4)	✓	✓	✓	✓	✓	✓
Select items in the inventory (UR5)					✓	✓
Perform actions on selected items (UR6)						✓
Use items on other items (UR7)						✓
Combine items (UR8)						✓
Provide feedback to the user (UR9)			✓	✓	✓	✓

actions and multi-selection mechanics, a novel VR inventory system that is based on a VR spin menu, that allows for item interaction also considering hierarchical organisation based on stacking by means of a concentric layout, and an extension of the Virtual Drawers inventory. At this point, a mere requirements-based comparison with previously published VR inventories underlined a heightened readiness of the presented prototypes for the deployment in games. To this end, VR inventory systems can be enhanced by various aspects, such as body-relative interactions, layouts from menu systems, or stackability of items.

Analyzing and comparing the technical and computational impact of these enhancements, also considering the number of items, would yield further useful insights. While these enhancements are useful for many adventure games in VR, additional extensions for more specialized use cases could be considered for future work. To account for recent improvements in controllerless gesture recognition, interactions purely based on hand tracking should be considered. Our proprioceptive, gesture-based interactions provide a good starting point for further research. To account for growing demand and increasing hardware capabilities for comprehensive game experiences, which might lead to more complex adventure games, a search function could be added to allow for a larger number of items. Inventory systems with higher capacities could benefit from input modalities like voice and eye-tracking.

ACKNOWLEDGEMENTS

This research was partially funded by the German Federal Ministry of Education and Research (BMBF) as part of the research initiative for Human-Technology Interaction ”Digital Platforms: Interactive Assistance Systems for Humans“.

REFERENCES

[1] S. Truman, N. Rapp, D. Roth, and S. von Mammen, “Rethinking real-time strategy games for virtual reality,” in *Proceedings of the 13th International Conference on the Foundations of Digital Games*, pp. 1–6, 2018.

[2] C. Fernández Vara, *The tribulations of adventure games: integrating story into simulation through performance*. PhD thesis, Georgia Institute of Technology, 2009.

[3] C. Anthes, R. J. García-Hernández, M. Wiedemann, and D. Kranzlmüller, “State of the art of virtual reality technology,” in *2016 IEEE Aerospace Conference*, pp. 1–19, IEEE, 2016.

[4] T. Nukarinen, J. Kangas, J. Rantala, O. Koskinen, and R. Raisamo, “Evaluating ray casting and two gaze-based pointing techniques for object selection in virtual reality,” in *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*, pp. 1–2, 2018.

[5] M. Weise, R. Zender, and U. Lucke, “How can i grab that?: Solving issues of interaction in vr by choosing suitable selection and manipulation techniques,” *i-com*, vol. 19, no. 2, pp. 67–85, 2020.

[6] J. J. LaViola Jr, E. Kruijff, R. P. McMahan, D. Bowman, and I. P. Poupyrev, *3D user interfaces: theory and practice*. Addison-Wesley Professional, 2017.

[7] D. A. Bowman and L. F. Hodges, “An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments,” in *Proceedings of the 1997 symposium on Interactive 3D graphics*, pp. 35–ff, 1997.

[8] M. R. Mine, F. P. Brooks Jr, and C. H. Sequin, “Moving objects in space: exploiting proprioception in virtual-environment interaction,” in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pp. 19–26, 1997.

[9] K. Wegner, S. Seele, H. Buhler, S. Misztal, R. Herpers, and J. Schild, “Comparison of two inventory design concepts in a collaborative virtual reality serious game,” in *Extended Abstracts Publication of the Annual Symposium on Computer-Human Interaction in Play*, pp. 323–329, 2017.

[10] Rocksteady Studios, “Batman: Arkham vr.” https://store.steampowered.com/app/502820/Batman_Arkham_VR/, 2016.

[11] S. Cmentowski, A. Krekhov, A.-M. Müller, and J. Krüger, “Toward a taxonomy of inventory systems for virtual reality games,” in *Extended Abstracts of the Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*, pp. 363–370, 2019.

[12] Mojang, “Minecraft.” https://www.minecraft.net, 2011.

[13] Blizzard Entertainment, “World of warcraft.” https://www.worldofwarcraft.com, 2005.

[14] Cloudhead Games, “The gallery.” https://store.steampowered.com/app/270130/The_Gallery__Episode_1_Call_of_the_Starseed/, 2016.

[15] Valve, “Half-Life: Alyx.” https://www.half-life.com/en/alyx, 2020.

[16] Endnight Games, “The forest.” https://store.steampowered.com/app/242760/The_Forest/, 2018.

[17] D. B. Chertoff, R. W. Byers, and J. J. LaViola Jr, “An exploration of menu techniques using a 3d game input device,” in *Proceedings of the 4th International Conference on Foundations of Digital Games*, pp. 256–262, 2009.

[18] A. Santos, T. Zarraonandia, P. DÍaz, and I. Aedo, “A comparative study of menus in virtual reality environments,” in *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*, pp. 294–299, 2017.

[19] M. Mundt and T. Mathew, “An evaluation of pie menus for system control in virtual reality,” in *Proceedings of the 11th Nordic Conference on Human-Computer Interaction: Shaping Experiences, Shaping Society*, pp. 1–8, 2020.

[20] D. Gerber and D. Bechmann, “The spin menu: A menu system for virtual environments,” in *IEEE Proceedings. VR 2005. Virtual Reality, 2005.*, pp. 271–272, IEEE, 2005.

[21] Terrible Toybox, “Thimbleweed park.” https://thimbleweedpark.com/, 2017.

[22] Funcom, “The longest journey.” https://store.steampowered.com/app/6310/The_Longest_Journey/, 1999.

[23] Sierra On-Line, “Space quest: Chapter i – the sarien encounter.” , 1986.

[24] LucasArts, “Indiana jones and the fate of atlantis.” , 1992.

[25] Infocom, “Zork i.” , 1980.