Putting the ghost in the machine: exploring human-machine hybrid virtual patient systems for health professional education

Topps D, Cullen M, Sharma N, Ellaway RH

University of Calgary

Corresponding author: David Topps, 2808 11 Ave NW, Calgary, AB. T2N 1H9.
topps@ucalgary.ca

Abstract

Background: Virtual patient authoring tools provide a simple means of creating rich and complex online cases for health professional students to explore. However, the responses available to the learner are usually predefined, which limits the utility of virtual patients, both in terms of replayability and adaptability. Using artificial intelligence or natural language processing is expensive and hard to design. This project description lays out an alternative approach to making virtual patients more adaptable and interactive.

Methods: Using OpenLabyrinth, an open-source educational research platform, we modified the interface and functionality to provide a human-computer hybrid interface, where a human facilitator can interact with learners from within the online case scenario. Using a design-based research approach, we have iteratively improved our case designs, workflows and scripts and interface designs. The next step is to robustly test this new functionality in action. This report describes the piloting and background as well as the rationale, objectives, software development implications, learning designs, and educational intervention designs for the planned study.

Results: The costs and time required to modify the software were much lower than anticipated. Facilitators have been able to handle text input from multiple concurrent learners. Learners were not discouraged waiting for the facilitator to respond.

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1659v1 | CC-BY 4.0 Open Access | rec: 21 Jan 2016, publ: 21 Jan 2016

Discussion: The implementation and use of this new technique seems very promising and there are a great many ways in which it might be used for training and assessment purposes. This report also explores the provisional implications arising from the study so far.

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1659v1 | CC-BY 4.0 Open Access | rec: 21 Jan 2016, publ: 21 Jan 2016

Introduction

Virtual patients (VPs) are computer-based simulators of patient encounters for the purposes of instruction, practice, and assessment in health professional education.(Ellaway et al., 2006) Their strengths come from their qualities as medical simulators and from being accessible through almost any kind of computing device.(Huwendiek et al., 2009) Their principle limitation however is that, without using complex and expensive artificial intelligence (AI), they need to be defined in advance of their use and as a result they are relatively limited in their ability to provide flexible and emerging representations of patient encounters or to incorporate adaptive instructor feedback.(Huwendiek et al., 2009) Since all possible responses are pre-defined, there can be substantial cueing effects which may also be disadvantageous.(McCarthy, 1966)(Tabbers, Martens & van Merrienboer, 2004)

Although there has been some exploration of the use of text recognition and speech recognition in virtual patients, it has proved to be complex, time consuming, and limited in its adaptability and range (Danforth et al., 2009; Rizzo, Kenny & Parsons, 2011). Alternative approaches to online case simulators, such as standardized patients and mannequin simulators can be more effective in this regard but are resource intensive, lack scalability, and are of limited use in a distributed or networked educational environment. Thus, although the use of virtual patients has been associated with critical learning outcomes such as clinical decision-making (Cook & Triola, 2009), their utility has been limited by the technologies that underpin them. In particular, we note the poor return on investment associated with using certain kinds of advanced technologies, such as AI (Nirenburg et al., 2009), to provide natural language processing in a scalable and easy-to-use manner.

Rather than continuing to pursue a purely computer-based solution to this problem, we decided to explore the possibilities of human-computer hybrids as the basis for virtual patient activities. Humans are far better at interpreting short written phrases than computers and responding in kind, particularly within loosely defined or ambiguous contexts, as well as when the language involves idiom, humour, colloquialisms, and local variations of medical jargon and abbreviations. However, one-to-one interactions, usually face-to-face, present problems of scalability and access, particularly for distributed educational programs. We need to make use of web-based media to provide access and scalability, while engaging human agents to manage episodes of semantically rich communication between learner and activity. The challenge then is how to

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1659v1 | CC-BY 4.0 Open Access | rec: 21 Jan 2016, publ: 21 Jan 2016

leverage human pattern recognition and language processing abilities across multiple concurrent users in a distributed, networked environment.

There are examples of this type of resource leveraging in the commercial world with customer relationship management systems that use remote operators, along with chat-based interaction and decision support trees. A single operator can concurrently manage multiple clients, sometimes up to 12 at a time, depending on the tolerable delay in response and the cognitive load for the operator.(Steindler, 2014; Sherwood, 2015; Support Team, 2015) This reflects the story of the Mechanical Turk where a human performed tasks in the guise of a mechanical computer.(Levitt, 2000) Amazon made reference to this in the naming of its Mechanical Turk service [www.mturk.com], something they describe as 'artificial artificial intelligence'. Amazon's service works by humans undertaking semantically rich tasks, involving judgement and wisdom that machines cannot do, with the requests and responses being mediated through an online system as if it were a machine responding. Building on this principle, we sought to develop and test ways of using human agents to respond to learners within virtual patient activities. This paper sets out our progress to date and plans for the development and testing of human-driven natural language functionality and its utility in the design of virtual patient cases, using the free and open-source OpenLabyrinth platform (Ellaway, 2010).

Methods

Overview: the aim of the study is twofold:

1. to develop OpenLabyrinth to support Turk Talk, so that learners undertaking a virtual patient activity can respond to challenges and tasks within it (by typing answers, questions, hypotheses, suggestions or other inputs) with one or more facilitators receiving these responses and either replying via a live chat discussion or moving the learner to the next place in the VP activity. The facilitator should be able to see all of their learners, undertaking the same activity at the same time, so that they can field responses from each learner as they see fit. Additional functionality where learners can collaborate through sharing their responses will also be explored.

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1659v1 | CC-BY 4.0 Open Access | rec: 21 Jan 2016, publ: 21 Jan 2016

2. to test these new tools and techniques (for functional quality, performance, usability, and utility) across a wide range of settings and with different activities and to feed the findings back in to the software quality assurance and improvement processes.

Because there are so many intersecting variables and unknowns to such a development, we needed an iterative cycle of design and testing that looks at how teachers and learners are guided by and interact with this software adaptation. We have therefore adopted a design-based research approach.(Design-Based Research Collective, 2003)

Setting: The study will be conducted at the University of Calgary with testers, students, and faculty members from the School of Nursing and the Cumming School of Medicine. We have also been collaborating with fellows of the Royal College of Surgeons in Ireland, who are interested in exploring the educational opportunities afforded by this approach.

Participants: We plan to work with fourth year nursing students from the University of Calgary School of Nursing Bachelor of Nursing program, where participation in the program evaluation will be required as part of the course curriculum. Using a case based on interaction with a patient with a mental illness, the facilitators for the Turk Talk activity will be experienced nursing faculty members with a background in caring for mental illness, and familiarity with the mental health learning outcomes for the BN program.

Intervention: The OpenLabyrinth platform is able to support multiple concurrent users in a networked environment, presenting them with complex branching clinical pathways, along with enquiry and feedback mechanisms, linked discussion forums and externally linked educational resources. Virtual patient cases can be made available to groups of users in a password-controlled manner, using the Scenario Manager. Rapid reporting and simple messaging systems are already inherent in the software design of OpenLabyrinth.

The learning design and concept mapping tools that are used to construct a set of pathways and responses can also be used to generate a standardized set of scripts and decision support trees for the facilitators. For a sample script, see our example case.(Topps, Cullen & Sharma, 2015)

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1659v1 | CC-BY 4.0 Open Access | rec: 21 Jan 2016, publ: 21 Jan 2016

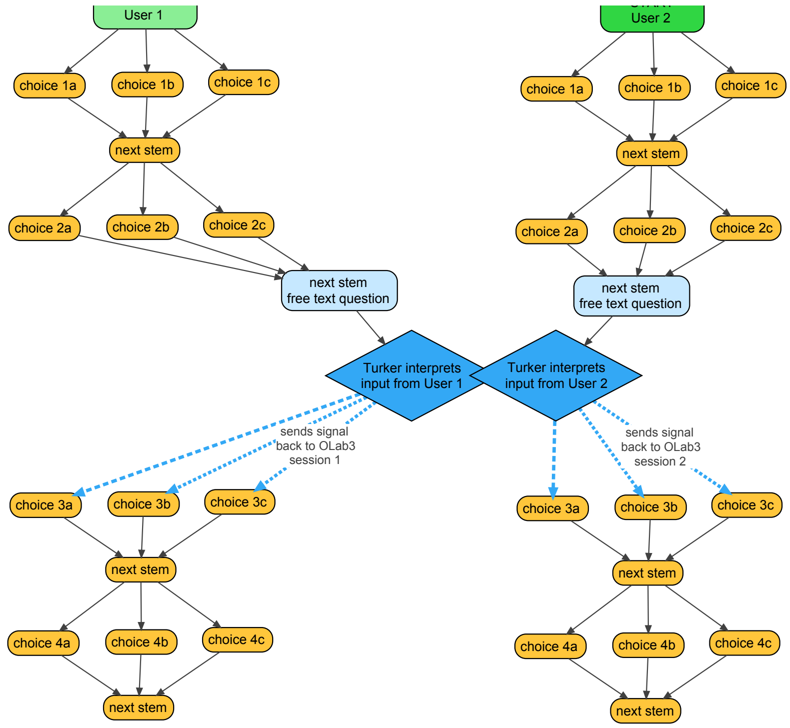

Figure 1: workflow or information flow in a Turk Talk case

During play, learners will independently navigate their way along a narrative pathway, with interactions with the facilitator interspersed at intervals through the case. Figure 1 illustrates two user pathways, running in parallel through a set of nodes. The facilitator (Turker) provides live interaction with the users at Turk Talk nodes (marked blue above), and then sends learners on their way along further nodes in their narrative paths. Although the two user pathways are concurrent and independent of each other, they each interact with the same Turker, who has an overview of the progress and responses for each user.

OpenLabyrinth tracks all mouse clicks and responses for each user, with timestamps for every action taken. These metrics are used to track performance and decision making amongst learners. They can also be used to track performance of the Turkers or of the decision trees and scripts that they utilize. Examination of these analytics affords the case designers with an internal quality improvement and audit mechanism. These metrics lend themselves both to quantitative analysis of numeric or categorical inputs, and to qualitative analysis of the text responses gathered.

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1659v1 | CC-BY 4.0 Open Access | rec: 21 Jan 2016, publ: 21 Jan 2016

OpenLabyrinth provides some simple analytics tools through its own reporting functions, but data can also be exported to Excel or queried directly using standard SQL database tools.

OpenLabyrinth was extended to provide the additional functions required with this approach. More specifically, new mechanisms were needed to support:

1. concurrent display of multiple learner session pathways to Turkers
2. rapid collating of new user-generated input from selected Turk Talk input nodes
3. ability for Turker to respond to each learner by text chat message
4. ability for Turker to direct each learner on to next appropriate step in pathway by node jump

Extending OpenLabyrinth

The pilot patient case designs in OpenLabyrinth employed a relatively linear case design, with simple branching around a central pathway. OpenLabyrinth provides multiple ways of presenting choices to users within an individual page or Node, including multiple choice questions, drag and drop lists etc. However, one particular case design that is popular amongst our case authors is using dandelions (multiple options (petals) returning to the same central point) and branched sets (sets? Not sure what these are). This allows the learner to choose between complex related concepts, receiving detailed feedback as they proceed. For an illustration of how this works, see Figure 2.

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1659v1 | CC-BY 4.0 Open Access | rec: 21 Jan 2016, publ: 21 Jan 2016

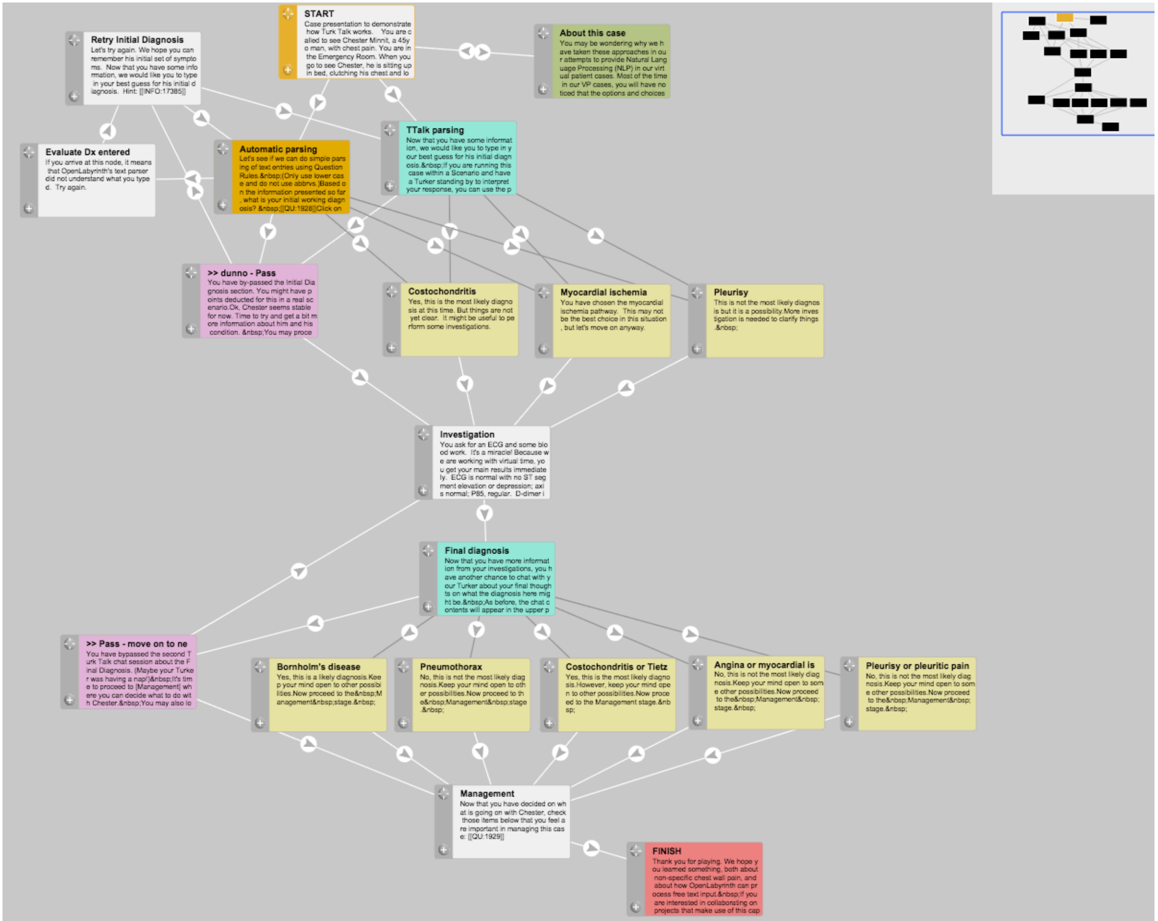

Figure 2 legend: In this concept map, extracted from our first Turk Talk case design, the flow is from top downwards. The aquamarine nodes are Turk Talk nodes where the learner interacts with the Turker. The light yellow nodes are the possible destination points that the Turker can send the learner on to. The dark grey links signify hidden paths that are not immediately apparent to the learner as a choice but that the Turker can direct them along. Using dandelions and branched sets in this pathway design has the advantage that OpenLabyrinth can track user activity in much greater detail, examining such parameters as time taken, node order precedence, complexity of the navigational pathway chosen by the learner and efficiency of case navigation. We were able to take advantage of these features by providing real-time feedback to Turkers as to where in the case the learner had progressed to, and also about what choices were available at any given Turk Talk decision point. For those who are interested in how the user interface and learning design works, we refer them to our online tutorial, embedded in the first of our Turk Talk cases.(Topps, Cullen & Ellaway, 2016)

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1659v1 | CC-BY 4.0 Open Access | rec: 21 Jan 2016, publ: 21 Jan 2016

We designed two separate virtual patient cases for our initial two trial runs with interested Turkers and faculty members:

1. Our first case design was quite short and simple so that we could run several iterations of the case and allow faculty members to act as Turker. We incorporated some very simple rule-based text processing into the case as a comparator so that Turkers and testers could get a feel for the expected responses when machine-generated or human-generated. We recorded these sessions, along with field notes, and debriefed after each run, seeking group input on changes needed to the software interface and to the virtual patient case designs.

2. In our second trial, we made the virtual patient case design much more complex with the intention of “stress testing” the Turker. The interface was set up to accommodate up to eight concurrent learners. We set up our session so that all eight learner slots were taken, and we had two Turkers. We also set up the activity to be more complicated so as to challenge the text-interpretive skills of the Turker in handling eight concurrent text conversations. We intentionally placed the most complex interaction nodes towards the end of the case so that the learners would arrive at this point in a staggered manner. This was done to decrease the likely lag time while learners waited for the Turker to respond, and to determine how confusing it would be for the Turker to have learners interacting with them at many different stages within the case.

We adopted an agile design process, rapidly incorporating feedback and suggestions from the testers into the software and case designs. We analyzed the case activity metrics, using the built-in timestamps and pathway metrics available in OpenLabyrinth, and by directly querying the underlying SQL tables, as noted above. This allowed us to further refine our case design performance in an iterative manner.

Results

So far, we have built the basic functionality in OpenLabyrinth, and we have formally tested this in two trial runs with faculty members from Medicine and Nursing. The programming costs of implementing this design were 74% less than projected, and the development schedule was largely complete two months before anticipated delivery.

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1659v1 | CC-BY 4.0 Open Access | rec: 21 Jan 2016, publ: 21 Jan 2016

Participants in the trials commented favourably on how easy the interface was to use. Small refinements to enhance efficiency and responsiveness have been introduced but the overall design has proven sound. Our test "learners" (faculty members acting in learner mode) have commented that they were very favourably impressed with how quickly they received a Turker response. They were expecting delays of 1-2 minutes. Indeed, because we were concerned that learners would become bored while waiting for their turn with the Turker, we designed several distractors into our virtual patient cases for the second trial run, providing the learners with other opportunities to learn, comment about the case, or even play tiny games. During our testing, we found that none of these distractors were used for boredom abatement. Some testers explored them simply because they knew they were there but commented that they did not feel a need for them. Our next steps will be to employ this learning design in real courses with real learners.

The scalability of this approach will be crucial to many centres. For this particular study with the School of Nursing, the numbers look to balance out very well. A single teacher should be able to handle 6-8 concurrent users, which is the size of the small groups who participated in the previous paper-based practice exercises.

Our labyrinth authors learned to be sparing in their use of this Turk Talk interface. While it was tempting to make a large number of nodes into free-text nodes, this starts to resemble the text-based navigation of Interactive Fiction.(Derksen & Zundark, 2011) Most users find such a navigation interface to be slow and frustrating. By focusing on the approximately 10% of choices where it is important that the learner is not cued to the possible responses expected, the navigation flow can be kept efficient and yet challenging for the learner where needed.

Our initial trial runs also highlighted how important it is to properly prepare the Turkers or Scenario Directors. We provided them with concept maps illustrating the allowed pathways within the map, along with scripts and tables of acceptable and unacceptable learner responses.(Topps, Cullen & Sharma, 2015) See http://demo.openlabyrinth.ca/renderLabyrinth/index/606 for a live example of a Turk Talk case and associated files. We also worked with Turkers to create shortcut text macros for frequently used responses such "tell me more" or "you need to read the question again". The current format of these Turker scripts is cumbersome and we anticipate further refinements to this process as we iteratively improve these in real learner sessions.

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1659v1 | CC-BY 4.0 Open Access | rec: 21 Jan 2016, publ: 21 Jan 2016

Although the original design had intended a one-to-many relationship between the Turker and their group of learners, during our second trial run, we were pleased to discover that it was possible to run two Turkers concurrently on separate terminals, each monitoring the same group of learners. Each Turker could see what the other was doing, and what all learners were typing, and learners could receive responses from either Turker. This happened quite smoothly, with no apparent lag in the software. Functionally, it became clear that workflow was more effective if the Turkers shepherded particular learners, rather than having overlapping responses from one or other Turker. The user interface enabled this by providing a visual grouping of learners according to each Turker's preferences. As well as allowing for larger groups with multiple Turkers, this also allows for a much more complex learning design with more than one tier or layer of Turker, with a more experienced teacher being able to supervise novice or front-line Turkers. As part of our future testing, we will be exploring just how scalable this approach is and what complexities arise as additional layering into the supervisory structures is introduced.

Discussion

We anticipate being able to use this approach in a variety of learning designs in our upcoming studies:

Learning Designs

1. Two to eight concurrent learners all tackling the same OpenLabyrinth scenario or set of linked VP cases. Key sections in the case map are selected where the Scenario Director (SD) is notified of entry into the section of interest and of completion of the section of interest. Scenario Director can monitor data from Turk-Talk-type questions embedded in Nodes in the active section. The SD reads and interprets the text response, compares it with a list of possible responses (provided by the OpenLabyrinth decision support pathways); if an appropriate match is found then the SD sends back to the learner a pre-defined message or directs them to the appropriate Node. The learner continues with the case map from that Node in the usual manner of virtual patient case navigation, until the case ends or the next round of Turk Talk interpretation is required. If the SD does not find a match amongst the available listed response scripts, the SD asks the learner to rephrase the response or to provide other information or sends them on a no-match

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1659v1 | CC-BY 4.0 Open Access | rec: 21 Jan 2016, publ: 21 Jan 2016

pathway. SD requests are largely pre-scripted macros (in a manner similar to canned responses to SMS text messages) to save time in typing responses.

2. Bookending over time, with early simple cases, progressing to more complex and intense interactions, possibly leading towards a more sophisticated full Standardized Patient-based OSCE. If there is a time spread over which parallel participants are playing the cases asynchronously, what is an acceptable delay for learners when warned to expect these? We have observed with simple semi-synchronous word games played on mobile phones, like "Words with Friends", that delays of minutes or even hours are tolerated, depending on the context and expectations of the players.

3. Communicating with minority ethnic or cultural groups or in mental health situations. Different communications conventions can apply (eg less or no eye contact, different phrasing or idioms or concepts of time or land ownership etc). It can be hard to create learning experiences that simulate this, especially with a shortage of skilled facilitators or SPs. We intend to explore for signs of the Proteus Effect (first demonstrated in the work from Stanford (Yee & Bailenson, 2007), where you start to act in accordance with your appearance, avatars, etc). Converting a large, ebullient, domineering alpha-male preceptor into a shy retiring female patient by use of avatars plus pre-recorded response phrases was shown to create interesting changes in participant behaviour in their studies. Our design affords a simpler, cheaper approach to explore such educational facets in cultural sensitivity training.

4. Scaling up to larger numbers of concurrent learners will be explored, using a Tier 1 and Tier 2 support model, typically seen in the Customer Relationship Management systems in a tech-support office. Tier 1 simple responses can largely be patterned into expected common groups after 2-3 iterations of the Scenario. This allows the facilitators to be relatively unskilled. Geographic location is irrelevant, allowing home-based workers or outsourced labour to be employed. More complicated questions can then be escalated to Tier 2 clinical supervisors but with less need for such skilled personnel.

We are interested in hearing from others who may wish to explore this approach in their virtual patient scenario designs. The source code will be incorporated into upcoming versions of OpenLabyrinth, which is free and open-source on GitHub (Topps et al., 2015). But for those teams who wish to try this out in a collaborative fashion, we may be able to provide guidance on

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1659v1 | CC-BY 4.0 Open Access | rec: 21 Jan 2016, publ: 21 Jan 2016

case and scenario writing challenges. Please contact us via info@openlabyrinth.ca and we can set you up with a demo login.

Acknowledgements

Support for development of OpenLabyrinth as an educational research platform, exceeding its original roots as a virtual patient authoring application, has been provided through a variety of research and development grants.

OpenLabyrinth is free open-source, open-standard, web-based software. Its development is guided by the OpenLabyrinth Development Consortium, including the University of Calgary, St George's University London, Karolinska Institut, Aristotle University Thessaloniki, Queens University Ontario, the University of Alberta and the Alberta International Medical Graduate program.

Conflict of interest

Two of the authors (Ellaway and Topps) have been involved in the development of OpenLabyrinth. However, as the software is free and open source, neither author receives or has received any material benefit from its development or use.

References

Cook DA., Triola MM. 2009. Virtual patients: A critical literature review and proposed next steps. *Medical Education* 43:303–311.

Danforth DR., Procter M., Chen R., Johnson M., Heller R. 2009. Development of Virtual Patient Simulations for Medical Education. *Journal For Virtual Worlds Research* 2:4–11.

Derksen B., Zundark A. 2011.Interactive Fiction. *Available at http://www.webcitation.org/6eSiQY1XA* (accessed January 11, 2016).

Design-Based Research Collective. 2003. Design-based research: An emerging paradigm for educational inquiry. *Educational Researcher* 32:5–8.

Ellaway R., Candler C., Greene P., Smothers V. 2006. An architectural model for MedBiquitous virtual patients. *Baltimore, MD: MedBiquitous*.

Ellaway RH. 2010. OpenLabyrinth: An abstract pathway-based serious game engine for professional education. In: *Digital Information Management (ICDIM), 2010 Fifth International Conference on*. 490–495.

Huwendiek S., De leng BA., Zary N., Fischer MR., Ruiz JG., Ellaway R. 2009. Towards a typology of virtual patients. *Med Teach* 31:743–748.

Levitt GM. 2000. The Turk, Chess Automation.

McCarthy W. 1966. An assessment of the influence of cueing items in objective examinations. *J Med Educ* 41:263–266.

Nirenburg S., McShane M., Beale S., Jarrell B., Fantry G. 2009. Integrating cognitive simulation into the Maryland virtual patient. *Studies In Health Technology And Informatics* 142:224–229.

Rizzo A., Kenny P., Parsons TD. 2011. Intelligent Virtual Patients for Training Clinical Skills. *JVRB - Journal of Virtual Reality and Broadcasting* 8(2011).

Sherwood R. 2015.The CRM Directory. *Available at http://www.webcitation.org/6eUoZt4zk*

Steindler Z. 2014.Pair Live Chat with Helpdesk/CRM Software. *Available at http://www.webcitation.org/6eUpSyivN*

Support Team L. 2015.LiveChat CRM Agent’s Handbook. *Available at http://www.webcitation.org/6eUptID5e*

Tabbers HK., Martens RL., van Merrienboer J. 2004. Multimedia instructions and cognitive load theory: Effects of modality and cueing. *Br J Educ Psychol* 74:71–81.

Topps D., Ellaway R., Simpson M., Corral J. 2015.OpenLabyrinth: source code. *Available at*

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1659v1 | CC-BY 4.0 Open Access | rec: 21 Jan 2016, publ: 21 Jan 2016

https://github.com/olab/Open-Labyrinth

Topps D., Cullen M., Ellaway R. 2016.Turk Talk tutorial. *Available at http://www.webcitation.org/6eSz3f2wL* (accessed January 11, 2016).

Topps D., Cullen M., Sharma N. 2015.Turker Script for Scenario 92. *Available at http://www.webcitation.org/6eSzjYfsK*

Yee N., Bailenson JN. 2007. The Proteus Effect: Self transformations in virtual reality. *Human Communication Research* 2007:271–290.

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1659v1 | CC-BY 4.0 Open Access | rec: 21 Jan 2016, publ: 21 Jan 2016