Block Building Coaching Agent in Virtual reality

Vidar Häggström Fordell

M.Sc Artificial Intelligence. 2020

vidarhf@gmail.com,

This project report is submitted on 10 June as part of the course Trends in Interactive Intelligent Environments in 2021.

1. INTRODUCTION

Driven by success within private sector firms, governmental agencies in various domains (e.g. health and education) have started to adopt Artificial Intelligence (AI) technologies. Recent progress in AI subdomains, namely natural language processing and machine learning, has enabled advanced exploitation of the specific AI technology chatbots in communication between companies/agencies and customers/citizens [1]. A conversational AI assistant is much like a chatbot and could have the ability through dialogue to train, coach and manage users in various scenarios [2].

Virtual Reality (VR) puts users into a computer generated 3D environment, shielded from external distractors, facilitating presence. With VR becoming a low-cost accessible mainstream technology there is a new digital channel of communication untapped by many companies and government agencies. Although VR is a fully immersive experience, user interface (UI) is most often limited to the traditional 2D WIMP-interface (Windows, Icons, Menus and Pointers). A virtual assistant in this channel enables a richer and more expressive interaction compared to a traditional chatbot since while dialogue with a bot most often is limited to the acoustic channel, VR and embodiment enables both interaction partners to convey extra-linguistic information through the visual channel. Additionally, an expressive embodied interface has the ability to improve user experience and engagement while also encouraging the user to mimic human-human interaction [3, 4].

In this project an embodied conversational agent, *Block Coach*, is developed to guide a user in the use case of building with blocks - all while immersed in virtual reality.

2. AIM AND RESEARCH QUESTIONS

The aim of this project is: ***the implementation of an Embodied Conversational Agent that from an initial block building instruction, can find a sequence of actions that lead to the goal state and guide a user through that process via voice-to-voice dialogue. Additionally, to explore user experience (UX) opportunities that VR brings to the communication with a virtual assistant.***

- **Given an initial construction, can the agent suggest a possible construction?**
- **Given an intermediate construction and a final goal, can the agent recognize if there is a wrong construction?**
- **Will the agent be able to guide the building process voice-to-voice in a virtual reality environment?**

3. METHOD

To answer the first two research questions the agent requires a logic-based action representation and reasoning of the block-building. Action reasoning is appropriate for this scenario [5] and its implementation is described in 3.2.

The dialogue interaction with the agent requires both an automated conversation system as well as voice-recognition of the user and giving the agent its voice of its own. The conversation is automated by the

Rasa open source machine learning framework [1] in which the assistant is built (described in 3.3) and speech is powered by Azure cognitive services [2] (described in 3.4).

The project thus consists of three primary components: VR environment, Action Reasoning, and voice communication. To tie all three parts together in an application a game engine is a suitable tool for its ability to create 3D-environments as well as external calls to other tools. Therefore, Unity game engine [3] was chosen for the project as it has become the largest VR content creation platform in the world with 60% of the market share in 2020 [4].

3.1 Material and software components

Itemized list:

- Unity 3D rendering engine, for the VR interactive experience.
- Rasa open source machine learning framework, for automated conversation.
- DLV-k[5] , logic-based planning system for generating action recommendations.
- Azure Cognitive Services Speech API, for voice recognition and speech synthesis.
- Oculus Avatar SDK[6], for expressive lip-sync.
- Oculus Quest 2[7], VR headset of choice.

3.2 Action reasoning (DLV-k)

The theory of action reasoning is for the agent to reason about actions and their effect independent of the domain in which it'll function [6]. The language K is well-suited for representing incomplete knowledge and an existing front-end integration of DLV-k in Unity [8]provided a good starting point to build upon. DLV is an artificial intelligence system based on disjunctive logic programming.Via the integration of DLV in Unity the game engine makes external calls to a deductive database system for knowledge representation of the world and action planning.

The world plan in the scenario is the (in)famous *Blocks World* planning problem. It contains Block(s) *B* and Location(s) *L* where L can be on-top of a block or the ground/table where blocks are placed. Blocks and Locations themselves are seen as statics and do not change over time. A block is *occupied* and can't be moved if another block is on top of it.

The domain-specific block's world representation was described as follows:

World plan

```
%% "Block B is on location L" is a property that can change

fluents: on(B,L) requires block(B), location(L).

    occupied(B) requires location(B).

actions: move(B,L) requires block(B), location(L).
```

[1]Rasa, accessed 10 June, 2021
https://rasa.com/docs/rasa/
[2] Speech Services, Microsoft, accessed 10 June, 2021,
https://azure.microsoft.com/sv-se/services/cognitive-services/speech-services/
[3]Unity, accessed 10 June, 2021
https://unity.com/
[4]Peckham, "Unity IPO aims to fuel growth across gaming and beyond", accessed 10 June, 2021,
https://techcrunch.com/2020/09/10/how-unity-built-a-gaming-engine-for-the-future
[5] DLV-k. accessed 10 June 2021
http://www.dlvsystem.com/k-planning-system/
[6] Oculus Avatar SDK, Oculus, accessed 10 June 2021,
http://developer.oculus.com/documentation/unity/as-avatars-sdk-intro/
[7]Oculus, *"Quest 2"*, accessed 10 June, 2021
https://www.oculus.com/quest-2/
[8]Brännström, DLV_Unity, accessed 10 June 2021
https://github.com/AndreasbCS/DLV_Unity

```
always:

        %% It is possible to move non-occupied blocks

        executable move(B,L) if not occupied(B), not occupied(L), B <> L.

    inertial on(B,L).

    caused occupied(B) if on(B1,B), block(B).

    caused on(B,L) after move(B,L).

    caused -on(B,L1) after move(B,L), on(B,L1), L <> L1.

noConcurrency.

---
```

There are two configurations, seen as collections of blocks and respective positions, the *Initial* construction and the *possible* construction seen as a *Goal state*.To be able to guide the user during building, the agent's knowledge of the world needs updating during run-time as the user executes actions on the world. From the point of efficiency, a new observation of the world is only needed when a block has changed it's location. To account for this each 3D block object has colliders for each side that triggers an observation of the world when moved.

Example configurations

```
initially: on(b0,table),on(b1, b0), on(b2, table).

Goal: on(b2, table), on(b1, b2), on(b0, b1).
---
```

After user executes action: `move(b1, table)`

```
initially: on(b0,table),on(b1, table), on(b2, table).

Goal: on(b2, table), on(b1, b2), on(b0, b1).
---
```

To be able to recognize a wrong construction or if the goal is met each configuration is represented as dictionaries of text value pairs in Unity. The two dictionaries are then able to be compared to tell if the goal construction has been met. This can be seen as a perk of the VR scene and its full knowledge of the environment as if performed in a real setting this comparison would not have been possible without additional sensors e.g. vision.

The configurations are loaded dynamically at start of run-time depending on the available blocks and their location while default goal is a tower construction of said blocks. Both the amount of blocks and goal state can be changed during gameplay as a result of this dynamical implementation.

3.3 Automated conversation (Rasa)

The Rasa framework is powerful and suitable in this scenario as it can be built on top of pre-trained language models and only needs around 15 pre-fed examples to recognize custom intents. By taking advantage of the NLU pipeline to recognize intent instead of specific voice commands the user is allowed to use natural dialogue with the agent that also became less error prone e.g. ignoring wrongful wording or minor recognition problems.

Important intents and example input

`ask_for_next_step`

> *"-What should I do next?*

`ask_for_goalstate`

“-Tell me my end construction”

`ask_if_correct`

“-Have I succeeded?”

`set_goalstate`

“-I want this construction as a goal”

A story is a representation of a conversation between the assistant and the user. Recognized intents results in a custom response *Action* from the assistant. If the action requires action reasoning the flag *\COACH_COMMAND_[action]* is passed with the message to trigger a method defined in Unity.

Example story

```
- story: asking for help
  steps:
  - intent: ask_for_next_step
  - action: utter_next_action
```

Finally, the conversations can be saved as the idea is to continuously improve the assistant by learning from interactions with the user.

3.4 Embodiment and speech

The Unity application utilizes Microsoft cognitive services[9] to enable voice-to-voice communication with the assistant. Calls are made to the service API for recognizing user speech to text as well as synthesizing assistant text into audio. An option to run locally using the Azure SDK is included and can be toggled with a checkbox at will.

Lip movement is enabled by changing the texture of the mouth on the 3D-model depending on recognized syllables in the audio output. This also works for the player avatar, synchronizing his/her lip movements to microphone input.

[9] Speech Services, Microsoft, accessed 10 June 2021, https://azure.microsoft.com/sv-se/services/cognitive-services/speech-services/

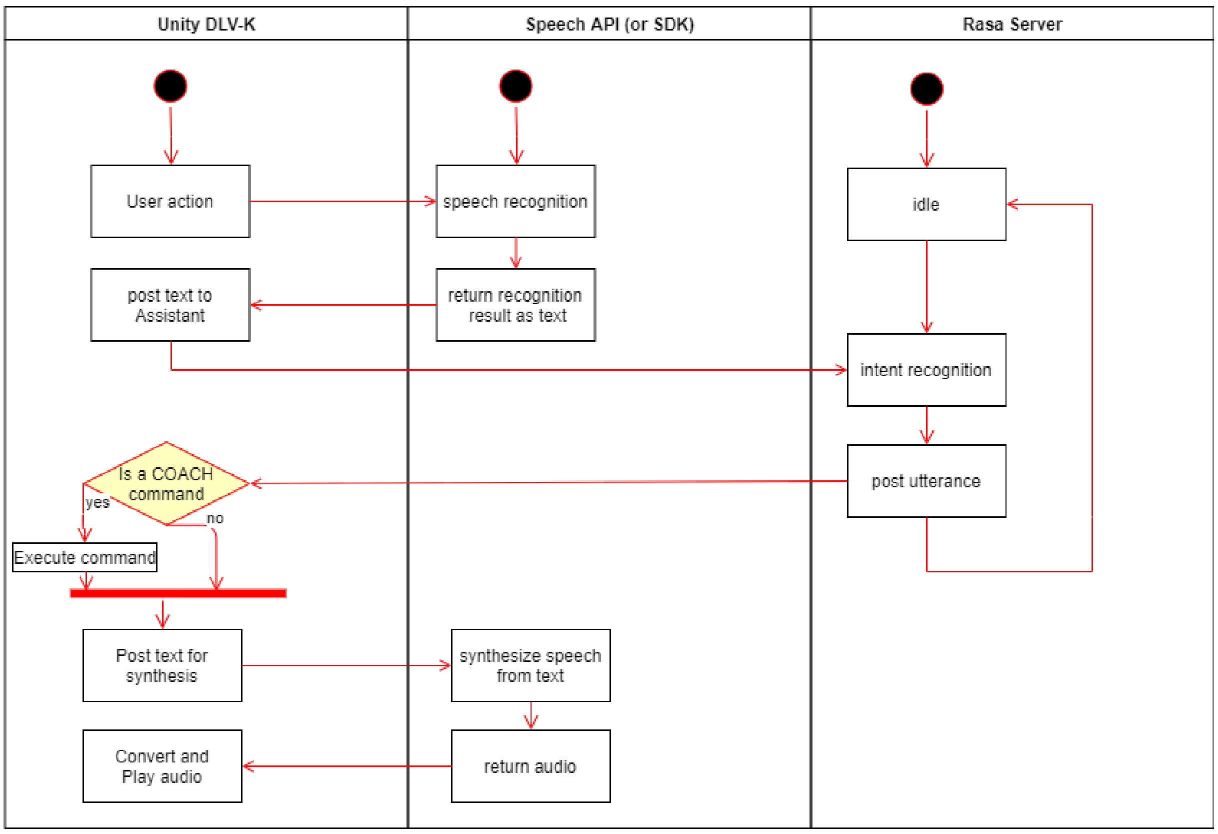


Fig 1. Activity flow of the user talking to the assistant.
The user triggers an action by pressing a button on the controller and speaking. The speech API then recognizes text from the recorded audio. This is then posted to the assistant which classifies the users intent and answers appropriately. If the assistant response calls for an DLV-k command (with \COACH_COMMAND_ [ACTION] flag) it's executed then the response is synthesized into a voice response that plays in unity.

4. RESULTS

The project results are as follows:

A proof of concept virtual reality application where an embodied conversational agent, *Block Coach*, through voice-to-voice dialogue with a user guides the process of building a construction with wooden blocks. Through continuous observations of the blocks the agent modifies it's instruction to the user in the process of reaching the goal state. The Block Coach suggests a goal construction as default but the user has the ability to set his/her own goal.

4.1 Example Interactions

The interaction between Coach and a user typically proceeds as shown in the following dialogue (baseline case).

1. *User: Hi*
2. Hey! I'm the Block Coach. Are you ready to build?
3. *User: yes*
4. Great!
5. Move(blue on top of table) then move (green on top of red)...
6. [User succeeds in building the goal construction and audio feedback is played]

Example 1: User asks for help

1. *User: what is my goal state?*
2. The goal state is: blue on top of table, red on top of blue, green on top of table, yellow on top of green..

3. *User: what should I do next then?*
4. move(orange on top of red), then move (green on top of table)...

Example 2: User asks sets a new goal construction

1. *[A construction is built]*
2. *User: Ok coach, set this as my goal state.*
3. Ok, new goalstate set! [audio feedback is played]

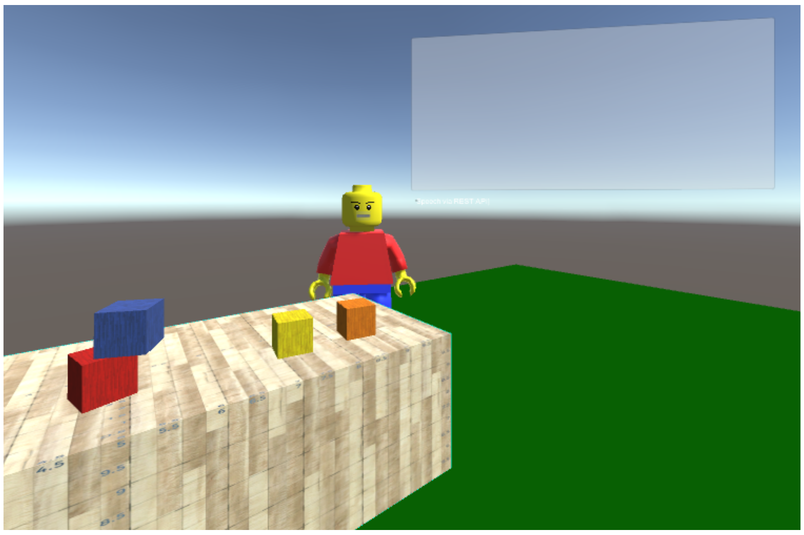


Fig 2. View from inside the VR environment with Block Coach standing behind the table and a dialogue window on the right side.

5. DISCUSSION

By using dialogue the embodied conversational agent has the ability to establish a social relationship to the user apart from solving the task goal at hand. Additionally, it creates an illusion of will that facilitates human-like interaction. By incorporating humoristic dialogue responses the agent would appear more likable, competent and cooperative [7] and while the coach has some social dialogue options small talk as in [8] initiated by the agent could further increase user engagement.

As for the case scenario in this project, the NLU pipeline succeeded with it's task using only 10-15 pre-fed examples but more complex tasks would require more conversational data for training. Virtual Reality has the potential to provide a safe training environment where users can interact with the agent in order to produce training data and evaluation of the agent until it´s performance meets requirements for deployment.

The digital 3D-scene domain can provide full knowledge of the world and actors. This information can be gathered to provide entities and fill Slots (anecdotally bots memory) that influence the assistants behaviour. In [9] a multimodal input of touch and speech is merged to reach target intent, by the same approach entities such as *\HoldingBlock[Red], \GazeTarget[Blue]* or *\WorldState[o]* could be provided to the block building coach for added intent recognition.

Despite all of the Coach's abilities, the spoken interaction between a human and an embodied conversational agent is still a very unusual way of interacting with machines. In this paper some benefits are highlighted, but whether or not the interaction approach chosen in this project truly is beneficial and enhances quality, usability and an acceptance on the part of the human user - that can only be answered based on a thorough evaluation. This evaluation should be of performance and quality of the assistant according to a standardized questionnaire supported by the Intelligent Virtual Agent community [10] and one that distinguishes the attributes of performance and quality [11].

6. REFERENCES

[1] Androutsopoulou, A., Karacapilidis, N., Loukis, E., & Charalabidis, Y. (2019). Transforming the communication between citizens and government through AI-guided chatbots. Government Information Quarterly, 36(2), 358-367.

[2] Nasirian, F., Ahmadian, M., & Lee, O. K. D. (2017). AI-based voice assistant systems: Evaluating from the interaction and trust perspectives.

[3] Foster, M. E. (2007, July). Enhancing human-computer interaction with embodied conversational agents. In International Conference on Universal Access in Human-Computer Interaction (pp. 828-837). Springer, Berlin, Heidelberg.

[4] Fraser, J., Papaioannou, I., & Lemon, O. (2018, November). Spoken conversational ai in video games: Emotional dialogue management increases user engagement. In Proceedings of the 18th International Conference on Intelligent Virtual Agents (pp. 179-184).

[5] Baral, C., & Gelfond, M. (2000). Reasoning agents in dynamic domains. In *Logic-based artificial intelligence* (pp. 257-279). Springer, Boston, MA.

[6] Gelfond, M., & Kahl, Y. (2014). *Knowledge representation, reasoning, and the design of intelligent agents: The answer-set programming approach*. Cambridge University Press.

[7] Morkes, J., Kernal, H. K., & Nass, C. (1998, April). Humor in task-oriented computer-mediated communication and human-computer interaction. In CHI 98 Conference Summary on Human Factors in Computing Systems (pp. 215-216).

[8] Bickmore, T., & Cassell, J. (2000, November). how about this weather?" social dialogue with embodied conversational agents. In Proc. AAAI Fall Symposium on Socially Intelligent Agents.

[9] S. Link et al., "An intelligent multimodal mixed reality real-time strategy game," 2016 IEEE Virtual Reality (VR), 2016, pp. 223-224, doi: 10.1109/VR.2016.7504734.

[10] Fitrianie, Siska, Bruijnes, Merijn, Richards, Deborah, Abdulrahman, Amal and Brinkman, Willem-Paul (2019). What are We Measuring Anyway? - A Literature Survey of Questionnaires Used in Studies Reported in the Intelligent Virtual Agent Conferences. Proceedings of the 19th ACM International Conference on intelligent virtual agents, p.159-161.

[11] Weiss, Benjamin, Wechsung, Ina, Kühnel, Christine and Möller, Sebastian (2015). Evaluating embodied conversational agents in multimodal interfaces. Computational cognitive science, Vol.1 (1), p.1-21.

7. Appendix (Screenshots)

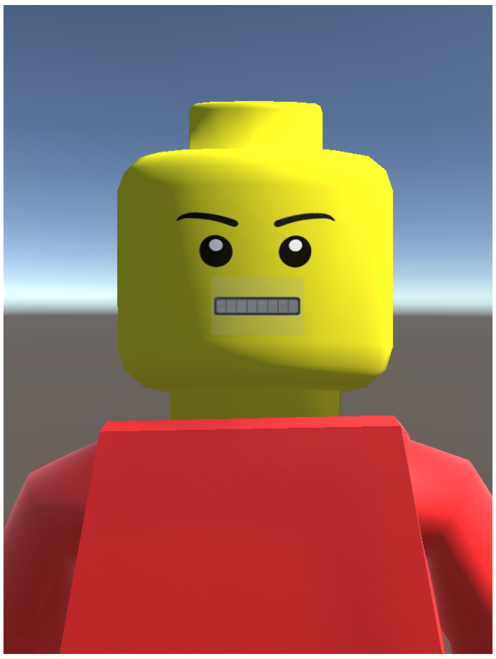
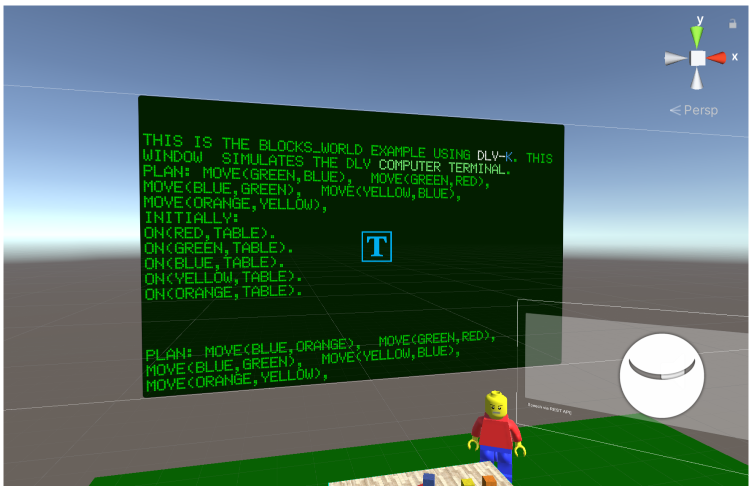


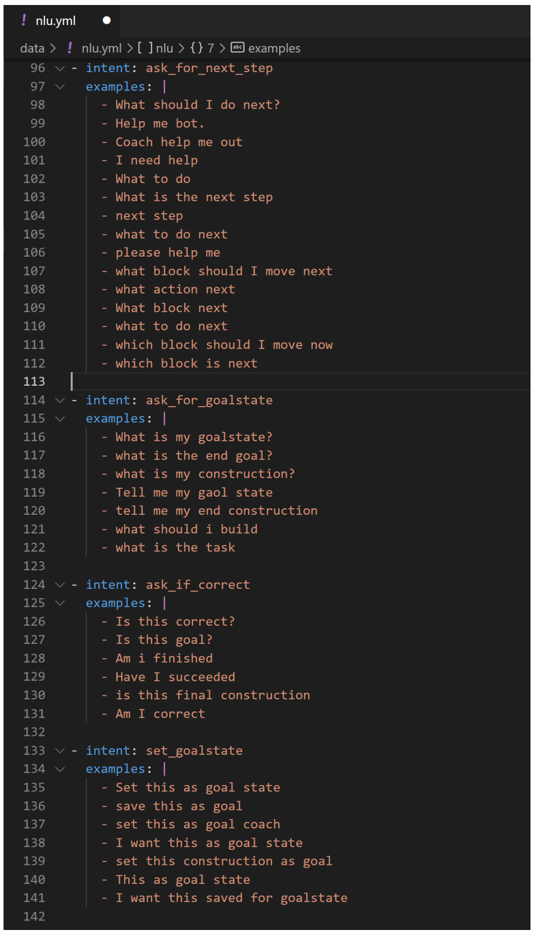
Fig 3. Block Coach



```
THIS IS THE BLOCKS_WORLD EXAMPLE USING DLV-K. THIS
WINDOW  SIMULATES THE DLV COMPUTER TERMINAL.
PLAN: MOVE(GREEN,BLUE),  MOVE(GREEN,RED),
MOVE(BLUE,GREEN),  MOVE(YELLOW,BLUE),
MOVE(ORANGE,YELLOW),
INITIALLY:
ON(RED,TABLE).
ON(GREEN,TABLE).
ON(BLUE,TABLE).
ON(YELLOW,TABLE).
ON(ORANGE,TABLE).


PLAN: MOVE(BLUE,ORANGE),  MOVE(GREEN,RED),
MOVE(BLUE,GREEN),  MOVE(YELLOW,BLUE),
MOVE(ORANGE,YELLOW),
```

Fig 4. Terminal created for debugging purposes. Simulating DLV-K communication.



```
nlu.yml
data > nlu.yml > [ ] nlu > { } 7 > examples
- intent: ask_for_next_step
  examples: |
    - What should I do next?
    - Help me bot.
    - Coach help me out
    - I need help
    - What to do
    - What is the next step
    - next step
    - what to do next
    - please help me
    - what block should I move next
    - what action next
    - What block next
    - what to do next
    - which block should I move now
    - which block is next

- intent: ask_for_goalstate
  examples: |
    - What is my goalstate?
    - what is the end goal?
    - what is my construction?
    - Tell me my gaol state
    - tell me my end construction
    - what should i build
    - what is the task

- intent: ask_if_correct
  examples: |
    - Is this correct?
    - Is this goal?
    - Am i finished
    - Have I succeeded
    - is this final construction
    - Am I correct

- intent: set_goalstate
  examples: |
    - Set this as goal state
    - save this as goal
    - set this as goal coach
    - I want this as goal state
    - set this construction as goal
    - This as goal state
    - I want this saved for goalstate
```

Fig 7. Screenshot of important intents and pre-fed examples