Robo Replacements

This project is part of Udacity's VR Developer Nanodegree.

Design and Gameplay

Extensive design, testing, and scope tracking is documented in design docs.

Runtime

The application was designed and tested to run in both the Vive and Oculus environments. By default, the game will attempt to use the Oculus, then the SteamVR framework, but you can switch the VR engine by selecting other options in a combo-box in the top right. This strategy (powered by the VRTK engine even allows testing via keyboard with no VR engine at all.

A compiled version of the application is accessible in multi-part files `build_robo_replacements_parts.zip`. Standatd unzip software should correctly reassemble the files and unzip automatically. Additionally, the supplemental learner software (currently, it only collects samples) is included in the file `build_learner3d.zip`.

User Feedback

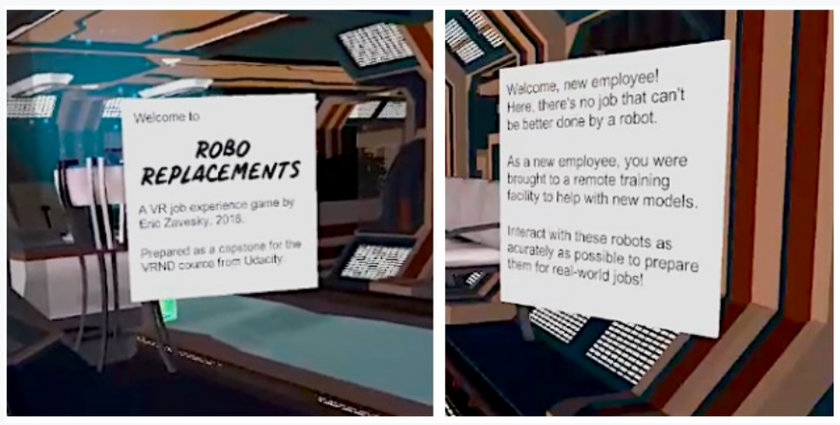
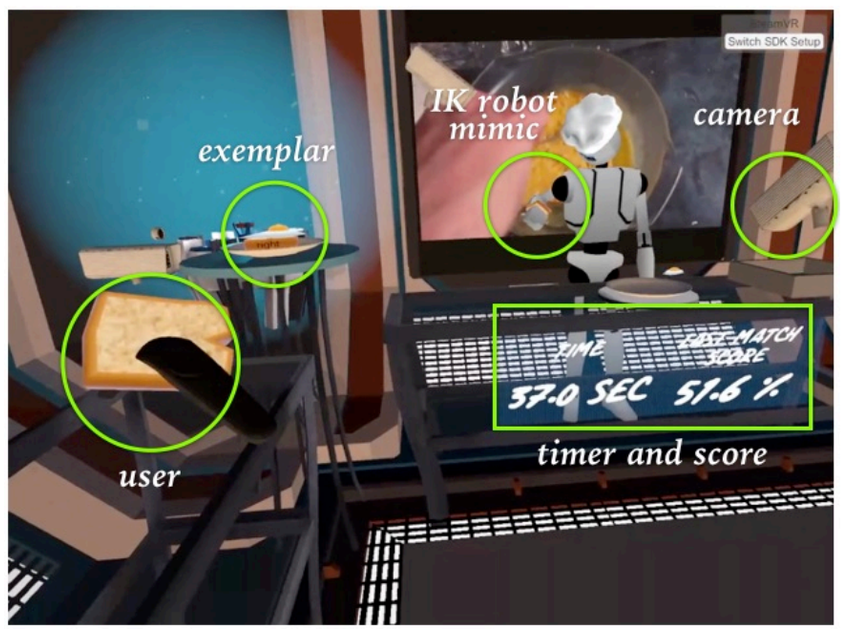
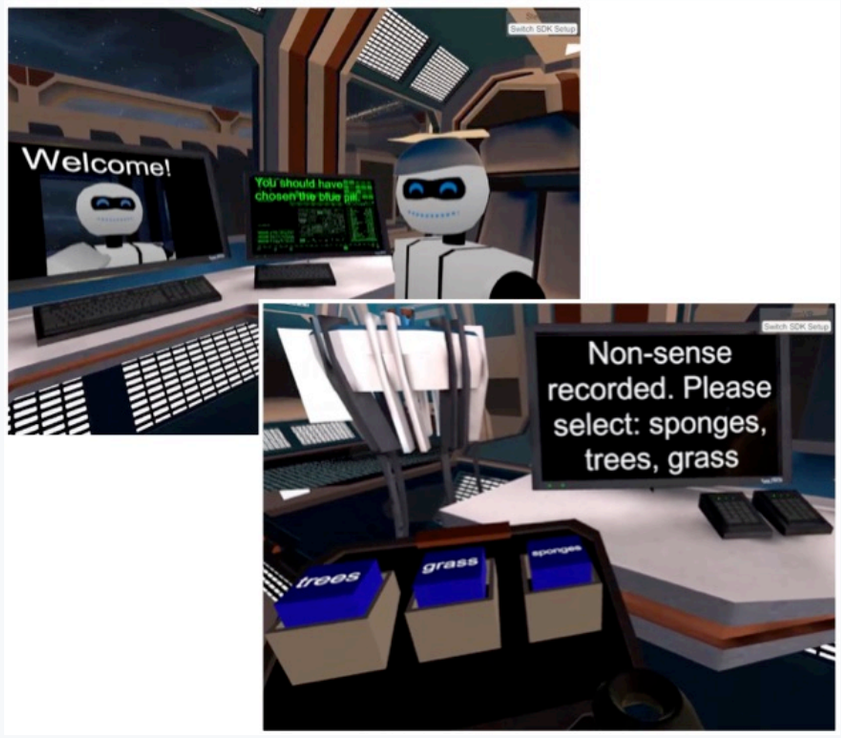
User feedback was sampled both for individual design elements as well as longer play sessions. Check out the design docs for more verbose logging and discussion.

Future Revisions

- Use of computer vision in seeing task for matching visual objects; using 2D companion app to compose
- Additional gamification stickers/badges for competing in the different levels
- Possibly additional content within each game (more food combos, more stories to tell, more shapes), but initial user feedback did not cite this as a shortcoming.
- A few interactivity components: properly size dialog panel for both VR environments, fix overlapping audio playback

Walkthrough

Snapshots and a video playlist of the overall and individual levels are included below. For more comprehensive discussions of each task and their design, please see the design docs and Rubric Checks below.

Showcase	Content Examples
Continuous Walkthrough [YouTube video] • Full walkthrough (original video) for sections below. • Demonstrates continuous flow through program with Oculus controllers. • Transition to each work task is also included, often with slightly fuzzy boundary conditions (by spatial box trigger).	
Introduction [YouTube video] • Upon entry, the user is introduced to the fictional company of Robo Replacements. • The company has different task stations that the user must help with as a new employee. • This environment has basic user interactions that change the narrative of each robot task.	
Doing/Cooking [YouTube video] • The user must simulate a cooked item by stacking food parts in the same fashion as the exemplar. • A timer and 3D placement and rotation matching scores are averaged and shown for motivation. • For flare, cameras hover and the exemplar food rotates. • The robot mimics then user actions via IK tracking of an object that is moved to a similar position as the user's target object. • Objects are cloned for the robot side and given perfect position symmetry with relative parent position correspondence. • Using IK targeting actions, the robot appears to mimic the user and is captured via "cameras" as in a TV display.	
Listening/Writing [YouTube video] • In the Listening and writing task, the robot prompts the user with a simple story to complete. • Instead of a full MadLibs implementation, a small set of words was hand picked for the user to choose from. • Physics-based push buttons with dynamically populated text are the primary means for interaction. • During playback, graphics associated with each user-selected word in the story are displayed. • In code, stories are easily created and linked to key words via simple templated data structures. • At time of writing this walkthrough, the stories were not auto-read through a text-to-speech system, but this could be a further	

refinement.

Seeing/Learning [YouTube video]

- The seeing task lets the user create basic pictures with a few shape primitives and evaluate them against in-app machine learning.
  Unfortunately, timing prevented this task from being fully completed. But some basic data was collected from a companion app and stored in the repo's /data directory. Proximal to the user's interaction was a logic and matching readout to help them understand what was happening in the algorithm itself.

Rubric Checks

Below, each item for the rubric check is documented below, but is also compiled in this fully annotated rubric video.

Rubric
Animation (100) animation was incorporated for different robot movements; while the quality and diversity will be improved, a starter for each station is in place; robots within the scene were animated both with look-based IK as well as motion/animation triggers throughout the level • animation is used for adding lifelike robot reactions - example from reading task, example walking robot • animation is explicitly used for a robot to mimic user actions through IK - example from doing task
Lighting (100) where insufficient with ambient lighting, real-time lamps were replicated and altered for hue; additionally, harnessing the directionality of the skybox a strong ambient light was created that has interesting shadow effects through windows. • lighting is added for mood with subtle particle effects where possible (and not computationally challenging) - example from reading corner, example from task corner
Locomotion (100) locomotion uses a combination of open teleporting along a main path through the output with specific regions of interest denoted with differently colored waypoints that will turn the user towards the location of primary action • teleportation is used as primary movement - example from introduction
Video (100) primarily used to motivate each task, background videos (open source) were opportunistically included; many static and moving image sources were utilized to enhance gameplay and lighten the mood; many custom objects were included • background video for mood and motivation - example from doing task • subtle screen video for robot/mood establishment - example from reading task
gamification (250) there are three mini-games within the space: one to teach (doing task), one to listen (speech recognition+ story reading task), one to challenge the AI (seeing task). The user can repeat each of them for a different gaming experience. Additionally, unique dialog entries are randomly chosen throughout the user's interactions with the robots.

- each task is divided into multiple sections (multiple locations)
- throughout the game, dynamic dialog interactions - example from doing task

Diegetic UI (250) where possible visuals and audio-based communication is given to the user; some text is still required, but attempts were made to condense them into short buttons **(non-text components are essential to gameplay)**

- illustrates use of buttons with pictures to assist in utility - example from seeing task
- illustrates use of exemplars (the physical objects) and a near-field camera (camera texture) to demonstrate utility - example of food from doing task

storyline (250) a lightweight dialog manager was hand-created with region triggers to start and end conversation with robots at different tasks; different conversations can be triggered for the same point using a coarse probabilty + random number selection

- spoken and textual prompts motivate each robot's task area - example from reading task

AI (250) a system for training and evaluating placement of blocks to guess visual patterns was implemented -- due to timing this was not hooked into the main VR experience -- instead, only a supplemental 2D learner application was created to help create training data for online models in the game; again though the models were not completed. **(partially completed)**

- method for feature exgtraction was created but actual classificaiton was not fully completed - example from seeing task, example supplementary data collection app

speech recognition (500) speech recongition was investigated but it could not be hooked in for streaming in such a short amount of time **(not completed)**

user testing (250) user testing was accomplished in two ways, early and quick user testing of individual ideas and visual concepts during design and implementation; a secondary, more complete test was also performed when significant milestones (like the completion of a task station); notes were taken along the way in the primary design document **(evaluation and improvements included)**

- design docs - user testing section describes revisions made based on user opinions of characters and interactions

Data Sources

Some assets were used from the open source community and are documented below.

Environment and objects

- Asteroids package
- Sci-Fi Environment
- toon robot
  - sitting and talking animations
  - simple animations
- space skybox
- howto cooking video
- comic panels font
- LeanTween package
- simple camera model
- food pack
- plate
- chefs hat

- graduation hat

Story Media

Specifically, in this section these (mostly image assets) were utilized for parts of the reading/listening task as background images that would be visualized on the robot's working terminal.

- reading task screen saver derived from this screen saver
- training messages
- monkey in mirror
- tiger
- children
- cheetah
- rainforest
- grassland
- Mars
- trees
- sponge
- eagle
- Jupiter
- phones
- kittens
- people
- pens
- sleeping baby
- pumpkin eater
- India Taj
- beach
- idea bulb
- library
- scrabble
- robots
- dog driving
- boxing
- ice cream
- stream light hoodie
- tooth brush

Versions

- Unity 2017.2.0f3
- Tested on Oculus RIFT and HTC Vive
- Leverages VRTK 3.3 for controller normalization (and testing) in Unity