DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
THE UNIVERSITY OF TEXAS AT ARLINGTON

SYSTEM REQUIREMENTS SPECIFICATION
CSE 4316: SENIOR DESIGN I
FALL 2018

TEAM 5 – JOVIAL MCNULTY
JEN - UR5 JENGA-PLAYING ROBOT ARM

JOE CLOUD
GABRIEL COMER
CARLOS CRANE
SAMMY HAMWI
MAXWELL SANDERS

Revision History

Revision	Date	Author(s)	Description
0.1	10.19.2018	SH	document creation
0.2	10.05.2015	JC, GC, CC, SH, MS	complete draft

CONTENTS

List of Figures

1 Product Concept

The objective of this project is to implement a Jenga-playing robot system involving a human opponent. Our solution will utilize 3D computer vision to develop scans of the tower's state, and a vacuum gripper to grip blocks. The human opponent will take turns with the robot performing block pulls and placing them at the top of the stack. The goal isn't necessarily to defeat the human opponent, but build a system capable of make multiple moves prior to the tower collapsing.

1.1 Purpose and Use

This project serves as an interactive demonstration of computer science concepts with the goal of inspiring the next generation of scientists & engineers. This product is designed to engage K-12 students at showcases and university outreach events. Students are expected to engage with the product by taking turns with the robot in a fun, friendly game of Jenga.

1.2 Intended Audience

This product is intended to be used by the university, College of Engineering, or the Computer Science & Engineering Department as a recruiting tool for future engineering students. Making this product available commercially to other universities and companies is possible. Though, as this is an end-to-end system designed to demonstrate the possibilities as a STEM student at UT Arlington, it is intended to be administered on behalf of the university. The end-user of this system is a member of the general public. Mattel® states that Jenga is appropriate for players ages six and up.

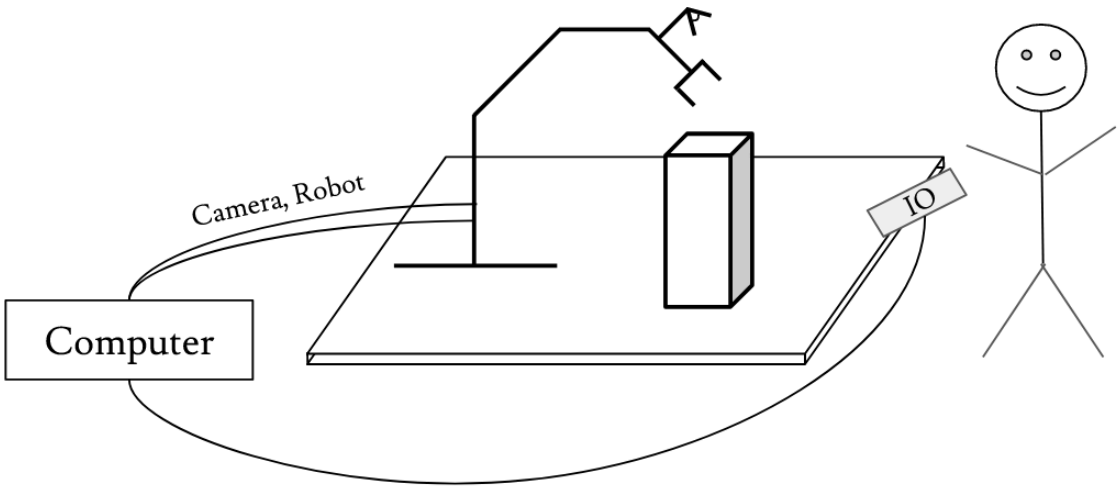

Figure 1: System concept drawing

2 Product Description

This section provides the reader with an overview of The UR5 Jenga Playing Robot. The primary operational aspects of the product, from the perspective of the end users and maintainers, are defined here. The primary features and functions found in the product, as well as user interfaces are described in this section. Reference Figure 1 for a conceptual drawing of the product.

2.1 Features & Functions

The UR5 Jenga Playing Robot will have the following features & functions, referencing Figure 1:

- Robotic arm for movement and 3D scanning, a Universal Robotics 5 robotic arm
- Camera, a 3D camera array
- Computer package, likely an Intel NUC
- Gripping component
- IO device, a physical button with wires connecting it to the computer package

2.2 External Inputs & Outputs

The only external data flow used in The UR5 Jenga Playing Robot will be the single IO device pressed by the user to inform the system that their turn is complete.

2.3 Product Interfaces

There will be no graphical user interface for this product. See Figure 1 for a mock-up of the physical user IO device interface that will be used.

3 Customer Requirements

This section covers the features and functions that can be expected from the UR5 Jenga-Playing Robot. Every requirement in this section is associated with a specific customer need that we need to satisfy. Our customers include: ourselves, our sponsor, our class, and the college of engineering. The features covered in this section are directly observable and should be transparent to the customer who expects the requirement. Any of the requirements below should not be changed without the agreement of the affected customer.

3.1 Input Interface

3.1.1 Description

The user interface of the UR5 will be simple and intuitive. It shall be a button that determines when the player move is completed, and the robot move is ready to begin.

3.1.2 Source

Maxwell

3.1.3 Constraints

The button should not take up a considerable amount of space, and should not interfere with the game. Pressing the button should not shake the tower.

3.1.4 Standards

N/A

3.1.5 Priority

High

3.2 Output Interface

3.2.1 Description

The output interface will be a LED that indicates whether or not the robot has completed its turn.

3.2.2 Source

Maxwell

3.2.3 Constraints

The LED will not be blinding, and should not interfere with the vision system of the robot as well as the vision of the player. The LED should not affect the gameplay.

3.2.4 Standards

N/A

3.2.5 Priority

Low

3.3 Custom Jenga Tower

3.3.1 Description

The UR5 Jenga-Playing Robot shall include with it for use in demonstrations, one custom Jenga tower. The tower shall be comprised of 54 3D printed blocks, each block.

3.3.2 Source

Gabriel Comer

3.3.3 Constraints

3D printing takes a lot of time, so having to print a new tower if our first attempt is too small, too heavy, or has too much static friction, could add significant delays. Also, printing 54-block towers could become costly as we try different block types and iterate through block size and material options.

3.3.4 Standards

N/A

3.3.5 Priority

Medium

3.4 Turn Autonomy

3.4.1 Description

The UR5 Jenga-Playing Robot shall pull blocks from the tower and place them on the top without intervention by the user. Once the users has signalled the end of their turn through the button, the user will not be required to touch the tower, or interact with the display in any way, until their next turn. User intervention might improve the UR5 Jenga-Playing Robot's performance, but it would also decrease the excitement of the game being played.

3.4.2 Source

Gabriel Comer

3.4.3 Constraints

Lack of user intervention will likely decrease the block pulling success rate of the UR5 Jenga-Playing Robot. In cases where the tower is crooked, or leaning to one side, the UR5 Jenga-Playing Robot may have difficulty obtaining accurate scans of the tower, and pulling blocks without knocking the tower down.

3.4.4 Standards

N/A

3.4.5 Priority

High

4 Packaging Requirements

The UR5 Jenga-Playing Robot's software will be delivered via download and the software packages used will be downloaded through a package manager. Different software packages will be used to create the UR5 Jenga-Playing Robot. Python will be used for programming the UR5 Jenga-Playing Robot, using the software packages OpenCV and ROS. OpenCV will be utilized in Python for all computer vision, 2D or 3D, tasks the Robot will use to play Jenga. ROS will be used for all communication with the UR5 Robot Arm; it will act as a middle-ware to control the robots components.

4.1 Software Packaging: OpenCV 3.4.3

4.1.1 Description

The system will use computer vision to successfully play Jenga. Using the open source software package OpenCV, the system will be able to utilize computer vision. The installation will be done through a package manager.

4.1.2 Source

https://opencv.org/

4.1.3 Constraints

There is weak documentation regarding OpenCV with Python, making a constraint in learning how to use the technology. Pythons run-time with OpenCV is proven to be slower than with C++, making a constraint on how fast our program can run. OpenCV is written in C++, so any changes to OpenCV itself must be done in C++ and not Python making a constraint on how customized our computer vision system can be.

4.1.4 Standards

https://opencv.org/platforms/ https://opencv.org/about.html

4.1.5 Priority

Critical

4.2 Software Packaging: ROS

4.2.1 Description

As the software largely depends on the ROS ecosystem, software installation and configuration shall be consistent with ROS conventions. i.e. custom software shall be written as a Catkin package which in turn manages dependencies.

4.2.2 Source

https://wiki.ros.org/Packages

4.2.3 Constraints

Modularity in implementation is desirable, meaning, the project should be decomposed such that software not included in a package either as methods or dependencies is not needed at runtime, though may enable complementary functionality if present.

4.2.4 Standards

ROS Standards can be found in the ROS Enhancement Proposals (REP)
REP-0136: http://www.ros.org/reps/rep-0136.html
REP-0144: http://www.ros.org/reps/rep-0144.html
REP-0149: http://www.ros.org/reps/rep-0149.html

4.2.5 Priority

High

4.3 Software Packaging: Python

4.3.1 Description

As individual methods and topics will be based in Python, it is important to follow Python conventions and packaging where possible.

4.3.2 Source

4.3.3 Constraints

A strict Python style guide shall be adhered to (refer to Standards) to improve maintainability, readability, and support cross-compatibility with Python 3 to ease porting to new versions of ROS.

4.3.4 Standards

ROS Standards can be found in the ROS Enhancement Proposals (REP)
Additional Python requirements can be found in the references
REP-0008: http://www.ros.org/reps/rep-0008.html

4.3.5 Priority

Critical

5 Performance Requirements

Since the UR5 Jenga-Playing Robot is intended for live demonstrations, performance requirements related to the game-playing ability of the final product are very important. Our goal of inspiring users to pursue the field of Engineering requires that the UR5 Jenga-Playing Robot provide an exciting display that will hold the users attention.

5.1 Minimum Rounds per Game Against Self

5.1.1 Description

In order to ensure that the UR5 Jenga-Playing Robot can play games for long enough to stay interesting for users, the UR5 Jenga-Playing Robot shall be able to play against itself for no fewer than 10 turns. Since user skill levels may vary greatly, the performance of the UR5 Jenga-Playing Robot playing turns against itself should provide more consistent and reliable feedback regarding its block-pulling abilities.

5.1.2 Source

Gabriel Comer

5.1.3 Constraints

The UR5 Jenga-Playing Robot's performance will be effected be abnormal changes in tower state, such as twisting, shifting, and leaning, occurring at different levels within the tower. In order to maximize the number of turns the UR5 Jenga-Playing Robot can play against itself, these tower states will either have to be accounted for while choosing which block to pull, or its block-pulling technique will have to be good enough to remove blocks from the tower without dramatically effecting the balance of the tower.

5.1.4 Standards

N/A

5.1.5 Priority

High

5.2 Time Limit for Turns

5.2.1 Description

The UR5 Jenga-Playing Robot shall take no longer than 60 seconds for each of its turns. This ensures that users will be able to maintain interest during the down-time between their own turns. It also ensures that any line to play against the UR5 Jenga-Playing Robot in action at a demonstration will move at a steady pace.

5.2.2 Source

Gabriel Comer

5.2.3 Constraints

Our ability to meet this requirement depends on the speed at which the UR5 Jenga-Playing Robot is able to rotate around the tower and develop a scan of the tower's state. If we move the arm too quickly, it may not obtain reliable vision data from its visual sensors. Furthermore, quickly swinging the arm around the tower may present a safety risk to the user standing nearby, waiting for the UR5 Jenga-Playing Robot's turn to finish. Then, the time it takes for the UR5 Jenga-Playing Robot to pull a block will effect similarly effect the turn time. We can not pull blocks to fast, as it will destabilize the tower. But moving to slowly will be boring for the users.

5.2.4 Standards

N/A

5.2.5 Priority

Medium

6 Safety Requirements

The UR5 Jenga-Playing Robot will be semi-autonomous, which can lead to some safety issues. It is very important to properly follow safety protocols to prevent injuries from occurring. The UR5 Jenga Playing Robot must follow the RIA Robotic Manipulator Safety Standards for the minimization of hazards. Laboratory safety protocols must also be followed while using any fabrication equipment. Safety is very important, making these requirements critical to follow.

6.1 Laboratory equipment lockout/tagout (LOTO) procedures

6.1.1 Description

Any fabrication equipment provided used in the development of the project shall be used in accordance with OSHA standard LOTO procedures. Locks and tags are installed on all equipment items that present use hazards, and ONLY the course instructor or designated teaching assistants may remove a lock. All locks will be immediately replaced once the equipment is no longer in use.

6.1.2 Source

CSE Senior Design laboratory policy

6.1.3 Constraints

Equipment usage, due to lock removal policies, will be limited to availability of the course instructor and designated teaching assistants.

6.1.4 Standards

Occupational Safety and Health Standards 1910.147 - The control of hazardous energy (lockout/tagout).

6.1.5 Priority

Critical

6.2 National Electric Code (NEC) wiring compliance

6.2.1 Description

Any electrical wiring must be completed in compliance with all requirements specified in the National Electric Code. This includes wire runs, insulation, grounding, enclosures, over-current protection, and all other specifications.

6.2.2 Source

CSE Senior Design laboratory policy

6.2.3 Constraints

High voltage power sources, as defined in NFPA 70, will be avoided as much as possible in order to minimize potential hazards.

6.2.4 Standards

NFPA 70

6.2.5 Priority

Critical

6.3 RIA ROBOTIC MANIPULATOR SAFETY STANDARDS

6.3.1 DESCRIPTION

Robotic manipulators, if used, will either housed in a compliant lockout cell with all required safety interlocks, or certified as a "collaborative" unit from the manufacturer.

6.3.2 SOURCE

CSE Senior Design laboratory policy

6.3.3 CONSTRAINTS

Collaborative robotic manipulators will be preferred over non-collaborative units in order to minimize potential hazards. Sourcing and use of any required safety interlock mechanisms will be the responsibility of the engineering team.

6.3.4 STANDARDS

ANSI/RIA R15.06-2012 American National Standard for Industrial Robots and Robot Systems, RIA TR15.606-2016 Collaborative Robots

6.3.5 PRIORITY

Critical

7 Maintenance & Support Requirements

As most of the components in this product are Commercial Off The Shelf (COTS) components, a repository of relevant documentations will be curated to help a maintainer troubleshoot issues that may arise. As this project largely depends on third-party software with specific version constraints, it is important that software version requirements are specified in adherence to REP and PEP protocols. This helps ensure that the product continues to work when the software is re-installed or migrated to a new computer. In the case major hardware components fail they will need to be serviced by an authorized repair center or repurchased if outside warranty/support period.

7.1 Third Party Software Documentation

7.1.1 Description

In order to make maintaining the software involved with the UR5 Jenga-Playing Robot, we shall compile documentation for the software packages that we use.

7.1.2 Source

Gabriel Comer

7.1.3 Constraints

We will only be able to compile documentation that already exists. There isn't an expectation to write documentation for COTS products.

7.1.4 Standards

N/A

7.1.5 Priority

Medium

7.2 Legacy Communications

7.2.1 Description

We will maintain communication with customers who choose to keep supporting the product. If any problems arise with the product after full delivery, depending on availability we will assist in patching or resolving the issue.

7.2.2 Source

Maxwell Sanders

7.2.3 Constraints

Theses communications will only last as long as the product support and will also be constrained to our personal time after graduation.

7.2.4 Standards

N/A

7.2.5 Priority

Future

8 Other Requirements

As realistically, the robotic arm will be re-purposed at conclusion of the senior design project. It is important that the team considers the modularity of the system so that all components can easily be detached/re-attached with minimal effort. This is important as the customer may want to use the components in this product for other projects yet retain the ability to take the polished project to showcases and other demonstration events where quality of presentation is critical.

8.1 Linux

8.1.1 Description

The control computer shall run Ubuntu Linux 16.04 LTS

8.1.2 Source

https://wiki.ros.org/universal_robot

8.1.3 Constraints

As the robot's drivers depend on a specific version of ROS, the Linux version and distribution must be selected to accommodate it.

8.1.4 Standards

N/A

8.1.5 Priority

High

8.2 Robot Operating System

8.2.1 Description

The control computer shall run ROS Kinetic Kame

8.2.2 Source

https://wiki.ros.org/universal_robot

8.2.3 Constraints

The robot's ROS driver package is only supported by ROS Indigo and Kinetic.

8.2.4 Standards

N/A

8.2.5 Priority

High

9 Future Items

In this last section, we reiterate all requirements that are listed as priority 5. The requirements listed here have been considered and documented within this System Requirements Document, but are unlikely to be implemented in the prototype version of the UR5 Jenga-playing Robot due to various constraints, such as budget, time, and skills.

9.1 Legacy Communications

9.1.1 Description

We will maintain communication with customers who choose to keep supporting the product. If any problems arise with the product after full delivery, depending on availability we will assist in patching or resolving the issue.

9.1.2 Source

Maxwell Sanders

9.1.3 Constraints

Theses communications will only last as long as the product support and will also be constrained to our personal time after graduation.

9.1.4 Standards

N/A

9.1.5 Priority

Future

References