Augmented Reality Sandbox

Bill of Materials

Group Members: N. Schram, R. Heifferon, R. Butler

Item Number	Part Number	Description	MFG	MFG PN	Supplier	Supplier PN	Quantity	Price	Total Price	
1	ARS-001	Laptop Computer	Apple	MacBook Pro	N/A	N/A	1	$0.00	$0.00	**Prototype Costs**
2	ARS-002	Digital Projector	BenQ	MX631ST	eBay	MX631ST	1	$799.99	$799.99	
3	ARS-003	RBG+D Camera	Microsoft	Kinect	N/A	N/A	1	$79.99	$79.99	
4	ARS-004	Silicone Sealant	N/A	N/A	Lowes	239983	2	$5.24	$10.48	
5	ARS-005	Plastic Cutting Tool	N/A	N/A	Lowes	163055	1	$3.98	$3.98	
6	ARS-006	Plexiglas 18"x24"	N/A	N/A	Lowes	78778	2	$12.98	$25.96	
7	ARS-007	Pine (Wood) 3/4"x 3/4" x 8'	N/A	N/A	Lowes	7871	1	$8.24	$8.24	
8	ARS-008	White Silicate Sand (10 lbs.)	N/A	N/A	eBay	N/A	1	$25.00	$25.00	
9	ARS-009	6x1/2" Wood Screws 100CT	N/A	N/A	Lowes	2391	1	$4.58	$4.58	
10	ARS-010	PyCharm Professional (Python IDE)	JetBrains	PyCharm	N/A	N/A	1	$0.00	$0.00	
11	ARS-011	8'x4' Plywood			Lowes		1		$0.00	**Final Product Costs**
12	ARS-012	2"x4"			Lowes		2		$0.00	
13	ARS-013	White Silicate Sand (50-75 lbs.)			eBay		1		$0.00	
14	ARS-014	1.5" Wood Screews (1 Box)			Lowes		1		$0.00	
15	ARS-015	1/16" Stainless Steel Sheet Metal (Mounting Bracket)					1		$0.00	
16	ARS-016	Flat White Paint (1 Gal.)			Lowes		1	$30.00	$30.00	
17	ARS-017	Matlab R2017b	Mathworks	N/A	Mathworks	N/A	1	$0.00	$0.00	
18	ARS-018	3D Printer (UCD Mechanical Lab)	N/A	N/A	N/A	N/A	1	$0.00	$0.00	
19	ARS-019	3D Printer Material (ABS), 1 Spool	N/A	N/A	N/A	N/A	1	$0.00	$0.00	
20	ARS-020	Anaconda Python Dependencies	N/A	N/A	N/A	N/A	1	$0.00	$0.00	
							Global Overall Cost		**$988.22**	

ELEC 4309 Senior Design I
Augmented Reality Sandbox
Software Requirements Document
Revision: 00
Revised: 2017-DEC-02

Definitions:

- CSV: a data format which is specified by comma separated values.
- Operating System: the base system running on the platform that allows users interface (Windows or Linux).
- Firmware: an embedded program on a platform that will not change, written in C or Python or both.
- Software: the program that is executed on the platform, written in C or Python or both.
- Hardware: devices that are controlled by software and interact with the physical world.
- RTOS: Real time operating system.
- Augmented Reality: is a live direct or indirect view of a physical, real-world environment whose elements are "augmented" by computer generated or extracted real-world sensory input.
- Media: the material that is used to partially fill the volume of the enclosure.
- BSP: Board Support Package

Problem Statement:

- To develop software that can capture topographical data by use of an RGB+D camera in real time.

Project Goals:

- To control various pieces of modern hardware using Python.
- To apply machine learning & computer vision concepts by exploiting the hardware/software interface of a host platform to analyze a system in real time.
- To aggregate data and display concurrently in a real time system.
- To develop a system whose latency is below the detectable threshold for human eyes, making it appear as if changes are happening simultaneously (I.E. augmented reality).

Constraints/Restrictions for Simplicity:

The following is a list of assumptions or constraints that we are using to implement our software in a controlled environment:

- Camera will be fixed and mounted above/over a region that can be manually changed (media) to show efficacy
- Media will be easy to modify and shape to show the changing contours in real time, simulating panning of a region
- There will be a secondary process to show the processed data, which will be displaying topographical contour lines on region from a projector mounted above/over the region
- Only the topographical data collected from within the sandbox will be analyzed, everything outside of this region will be ignored. This will either be implemented in software or by the placement of the hardware.

General Operation:

This device is an augmented reality sandbox used to demonstrate the capabilities of real time topographical mapping of a static environment captured by a camera. The structure will have a large internal volume that is partially filled with a fine sand-like material that is free from any/all debris. The structure will be supported by a mechanically sound "cart" with 4 wheels for easy moving of the entire setup (two caster, two rigid bi-directional). Above the sandbox will be a cantilevered structure that supports both a RGB+D camera and a digital projector both of which are fixed to the structure. Some minor built in adjustments may be implemented to allow for easier troubleshooting of the system during development but will be fixed and rigid for demonstrations. Cabling will be run from the sandbox to an adjacent laptop computer which will be running the software controlling the system. Once the system is powered on, the camera will take a depth measurement of the entire sandbox (z-axis height of the sand at all locations contained within the structure only, the program will ignore any data retrieved from outside of this area).

The data will then be parsed into levels of similar height which will be determined from the z-axis step size for grouping of similar data. The data will then be assigned a color based on its groupings and a line segment will be drawn linearly between each of the data points in the same group of that particular color. This will occur for all groupings of data for each color (red to blue). Once all the data has been parsed, the final contour image will be projected back down onto the media in the structure. This process will be in a loop routine and occur over and over again until user input is received to stop. The timing of this loop for processing the data is the most critical aspect since we require that this appear as if it is happening in real time. At any time during this process if the user wishes to capture the current environment (I.E. the current topology of the sand), a keypress can be made on the computer which will save the entire data set to a file. This will then be immediately processed using either Matlab or Python and will 3D plot the data onto the screen which the user can then save. From this point, the saved data set can then be sent over to a connected 3D printer where it will initialize a print of the saved workspace from ABS material. Due to useable printer platform sizes, the 3D print may be scaled to a fraction of the original data taken.

Requirement Number/Requirement Specification	Requirement Description
1. Hardware	
1.1 Platform	Platform must support required peripherals (see section X).
1.2 RGB+D Camera	Camera must be capable of taking depth data with a minimum resolution of 1cm.
1.3 Digital Projector	Digital projector must have HDMI connection and have a minimum display resolution of 1080p.
1.4 Cabling	Cabling will be purchased through an approved supplier and will not inhibit the overall data speed transfer rates set by the platform or other hardware.
1.5 3D Printer	3D printer must be capable of taking in X, Y, Z coordinates or point cloud information and printing in ABS material with a minimum resolution of 0.5 mm.
1.6 RTC	Real time clock will be on the platform to be used as a time stamp for any offloaded or saved data sets taken from the device.
2. Software	
2.1 Operating System of platform	Operating system will be up to date to manufacturer's current distribution.
2.2 Operating System of 3D Printer	3D printer will be up to date to manufacturers approved firmware version
2.3 BSP's	BSP's will be used to hasten development time if they are currently supported by the manufacturer.
2.4 3D Plotting Software	Matlab or Python will be used to plot the changes of the sandbox at set intervals determined in the software.
2.5 Software Include Files	Any utilized libraries will be declared at the beginning of the program. For Python, this will be in the form of *import x as y*. For C, this will be in the form of *#include xyz.h.*
2.6 Header Files	Header files are allowed but must have a descriptive header block at the beginning of the header which describes the following: header name, date of creation, author, and a verbose explanation of use.
2.7 Functions	Functions are allowed and highly encouraged. Every function must have a descriptive function block preceding the function itself and must at the minimum contain the following information: function name, date of creation, author, and a verbose explanation of use. All functions must have a defined function prototype which must be at the top of the main program, below the included files.
3. Language	
3.1 Embedded Software Language	All software will be written in either C or Python or a combination of the two.
4. Protocols	
4.1 HDMI	HDMI will be used to display the information from the platform to the projector and onto the surface.
4.2 USB 3.0	USB 3.0 will be used to transfer data from the RGB+D camera to the platform

4.3 SPI	SPI will be used if required to communicate to peripherals through full duplex.
4.4 I2C	I2C can be used as an alternative solution if there are a large number of hardware that are more easily addressable in this fashion.
5. Platform	
5.1 Power	Platform will have a power supply sufficient to power itself and must be able to plug into AC mains 110-125V.
5.2 Peripherals	The platform must have the following peripherals: HDMI, USB 3.0, SPI & I2C.
5.3 Performance	The platform must have at least one core.
5.4 Processor Speed	The platform must have a minimum processor speed of 1 GHz.
6. Overall Dimensions	
6.1 Exterior Dimensions (Prototype)	The overall outside dimensions of the prototype will be 12” x 12” x 6” ± 0.5” (L x W x H).
6.2 Interior Volume (Prototype)	The overall interior volume of the prototype will be no greater than 864 in^3 (12” x 12” x 6”).
6.3 Exterior Dimensions of the final design	TBD.
6.4 Interior Volume of final design	TBD.
7. Media	
7.1 Particle size	Material selected must be finer than 1mm.
7.2 Gradient	Particle size must be fine enough to produce a smooth gradient on the surface.
8. Speeds	
8.1 Refresh rate	The refresh rate is tied directly to the speed of which we can collect data from the Kinect and display the data with the projector. One cycle of this is our defined refresh rate.
8.2 Program execution time	A program timer will be implemented to determine how fast the program is executing. This will be the quantitative determinant for how to improve the programs speed. Optimization strategies will be developed if required based on that information.
8.3 Latency	There is not a quantitative value for this requirement. This will be more of a UX requirement as the system must update in real time to accomplish the augmented reality aspect. Therefore, there cannot be any detectable lag in the system from the user’s perspective.
9. Power	
9.1 Main power	Power will be 110-125 VAC provided from mains via a wall outlet.
9.2 Battery power	This revision of the device will not be designed to operate off of any battery power.
9.3 Device Power Supplies	Each of the 3 main pieces of hardware that is required for this project (Laptop, Xbox Kinect, Projector, 3D printer) all have their own provided power supplies which plug directly into a 3-prong wall outlet and provide the required DC voltage and current necessary to adequately power the

	devices. For additional information on the power consumptions of these devices see the power analysis document.
10. Z-axis Step Interval	
10.1 Z-axis step interval	Will be a #define or similar global variable that can be adjusted in one location and will change everywhere in the program.
11. Topographical line color palate	
11.1 Topographical line color palate	This will red to blue ranging from the minimum to the maximum and the color transition from one line to the next will be determined by the Z-axis step internal (Requirement #10).
12. Optimization	
12.1 Program optimization	Techniques such as loop unrolling and the use of intrinsics may be used to speed up the execution time of the program if required or desired for the appearance of real time.
13. 3D Printer	
13.1 Usable printer area	Due to the constraints of the 3D printer's usable workspace, the rendered print may need to be a scaled down version of the original data set. In this case, a secondary data set will be generated from the original which will be the reduced data necessary to fit on the printer's platform.
14. Cost	
14.1 Total Cost of the Project	TBD.
14.2 Total investment from the department of electrical engineering	$600.
14.3 Additional Funding	$100 (provided by the chair of the Department of Electrical Engineering).
14.4 Total prototype cost	$958.22
14.5 Total estimated production costs	TBD.
15. Data	
15.1 Topographical Data set	Each captured data set will contain all data points prior to being sorted. Each subsequent data captured will over-write the previous data. This will keep the database from "blowing up" in size and filling up digital space.
15.2 Topographical Data set to output file	When the user triggers the dataset to be saved to an output file, this will be time stamped and saved to the working directory. This data is never destroyed. The user could trigger many "save data to output file" and each will be saved uniquely to be reviewed or analyzed at a later time.
15.3 Output file naming convention	When the user triggers the program to save the current dataset to an output file, the filename will be the date and time the user triggered the save event.
15.4 Output file data format	CSV (comma separated values) or .xlxs (Excel file format).
16. User Interaction	
16.1 Start Program	When the program is initialized, to start the program press the "ENTER" key on the keyboard.
16.2 End Program	To end the program, press the "ESC" key.
16.3 Save data to output file	To save the last data to an output file, press the "S" key.

16.4 Send data to 3D printer	To send the data to a 3D printer, press the "P" key.
16.5 Send data to be 3D Plotted	To print the topographical map of the last data set, press the "M" key.
17. Software Best Practices	
17.1 Function naming convention	Functions will be named in a case sensitive manner in which the first word is lowercase and the following word first letter is uppercase. There will be no underscores used in function naming. Functions must be named such that their function is obvious. Example of correct syntax: void exampleFunction(void) Examples of incorrect syntax: void example_function(void) void ExampleFunction(void)
17.2 Variable naming convention	Functions will be named in a case sensitive manner in which the first word is lowercase and the following word first letter is uppercase. There will be no underscores used in function naming. Variables must be named such that it is easy to understand what purpose they serve. Ambiguous names are not allowed. Example of correct syntax: int testVariable Examples of incorrect syntax: int val1 int maximum_value_found
17.3 Function return values	If a function is going to return a numeric value for success or failure they will be done in the following way: For a success: return 0 RETURN_SUCCESS For a failure: return -1 RETURN_FAILURE
17.4 Main program organization	The main program will be in the following form (from top to bottom): Title block Include Files Preprocessor Directives Main function Supplemental Functions

17.5 If/then statements	If then statements are not allowed to have a full pass through. This means that if an if/then statement is used that the full if/then/else must be implemented to catch any issues that could arise from the conditions not being met in the if/then.
	Example:
	int i = 5;
	if (i == 1)
	{
	printf("IF CONDITION\n");
	}
	elseif (i == 2)
	{
	printf("ELSE CONDITION\n");
	}
	else

	{
	printf("CATCH CASE CONDITION\n");
	}

17.6 Case statements	Case statements are allowed.
17.7 While loops	While loops must be allowed to have one or more exit conditions. This means that the main program cannot run in a while(1) loop.
17.8 For loops	For loops are allowed but should be used with care as this program is time sensitive and these loops do not execute in one clock cycle.
17.9 Conditional expressions	Conditional expressions must have verbose parenthesis used to make it perfectly clear what is being compared or evaluated.
17.10 Ternary operator (?)	Ternary operators are not allowed.
17.11 Arrays	Arrays will be used to store the topographical data, it is recommended that numpy be used as it has some built in acceleration aspects already in place for optimization. All arrays must be correctly allocated for their size.
17.12 Memory Allocation	Memory allocation must be done for this application as the amount of data being collected will be of a fixed size and will be updated frequently. The same block of memory will be used to store the topographical data on repeat executions. This will keep the program from wasting the computers resources.
17.13 Print statements	Print statements are only allowed for debugging. They will not be allowed to be included in the final production version of the software as they will negatively affect the programs execution time. If output is required during normal operation, use an output file.
17.14 Output files	Output files have two basic purposes: data storage and debugging. Data storage is a requirement of the software and must be done to the specified format. All output files must be closed after the information has been written.
17.15 Condensed Code	Clever code is not syntactically incorrect, but as it does not aid in any development across multiple users, it is not allowed. Verbose, easy to follow code is required which will allow all programmers to easily understand an modify any sections of code.
17.16 Program Title Block	Regardless of C or Python, the program header block will take the following form and will be in a block comment: Program Name: Program Author: Date of Creation: Revision: Revision Updates: Program Description:
17.17 Program Function Block	Regardless of C or Python, the program function block will take the following form and will be in a comment block: Function Name: Function Author: Date of Creation: Function Parameters:

	Return Parameters: Function Description:
17.18 Program Header Block	Headers will be used for functions that are frequently used and therefore will contain some repeated information as the function block. The following header block is for C only and will take the following form and will be in a comment block: Header Name: Header Author: Date of Creation: Function Name: Function Author: Date of Creation: Function Parameters: Return Parameters: Function Description:
17.19 Comment Block	Comment blocks are defined as comments in the program that extend more than one line. This is done in the following way: C: /************************* * This is a multiple line comment **************************/ Python: ‘’’This is how to do multiple line commenting in Python’’’
17.20 Single Line Comment	Single line comments will be done in the following way: C: //This is a single line comment in C Python: #This is a single line comment in Python
17.21 White space	White space between lines of code is not a requirement. Excessive space is not allowed, however double spacing your code for easier reading is allowed and up to the programmer’s discretion.
17.22 Visible function/program separation	Frequently it helps to break up the program into visible sections. This is allowed in the following ways: C: //--- Python: #---

17.23 Brackets	Brackets are used and required explicitly for all functions, loops, if statements, etc. They will be done in the following ways:
	void exampleFunction(void)
	{
	        //Function contents
	}

<table>
<tr><td></td><td>

```
If (variable == 1)
{
	//Do this
}
elseif (variable == 2)
{
	//Or this
}
else
{
	//Then this
}
```

The following is an example of what is not allowed:

```
void exampleFunction(void) {
	//Function contents
}
```

</td></tr>
<tr><td>17.24 Revisioning</td><td>The program title block has a section for revision. This is to be used when developing the program. When the program is in its prototype phase and is not functioning as required, the revision used will be alphabetic only. Ex. Revision: A. Once the program reaches the point when the program is functioning as required and changes made are to improve functionality, speed, interfacing, etc., then the revision will be a two-digit numeric only. Ex. Revision: 00. When saving a new revision, update the section that describes what changes were made since the previous revision. Use revisions liberally, as they are a great way to keep a program history of what changes have been done and allows for easy roll back if necessary.</td></tr>
<tr><td>17.25 Multiple Source Files</td><td>Multiple source are allowed but must be saved in the same directory as the main file. When using multiple sources, they must be verbosely explained and contain the same title block and description of operation as any other files. In general, unless the program becomes exceedingly long, it is best to make use of header files before using multiple source files.</td></tr>
</table>

Augmented Reality Sandbox

Dec 2, 2017

Senior Design Group

Project manager	Nathan Schram
Project dates	Sep 18, 2017 - Jan 16, 2018
Completion	33%
Tasks	30
Resources	3

Augmented Reality Sandbox
Project Progress Tracking
Group Members: N. Schram, R. Heifferon, R. Butler

Tasks

Name	Begin date	End date
Project Kickoff	9/18/17	10/6/17
Project Directives	9/18/17	9/18/17

SENIOR DESIGN I
--
Group Members:
1. Nathan Schram
2. Ryan Heifferon
3. Randy Bulter

Welcome Gentlemen,
This is going to be the working Gantt software that we use to track every step of our progress through the entire semester. I have created the general structure for each thing that is required this semester (with the exception of the individual assignments).

To differentiate our tasks and progress through individual tasks, I have assigned a color to each of us:
1. N. Schram - Purple
2. R. Heifferon - Orange
3. R. Butler - Green

Individual Task Notes:
It is required that you write a brief summary of the task as you work on or complete you assigned tasks. This does not have to be exteremly verbose, but something so that the group can see what you are working on and how it is going. This is done in the "Edit Notes" section to the right.

Name	Begin date	End date
Define Problem	9/25/17	9/27/17

2017-NOV-30:
Redefinition of our original problem is completed and located in the project requirements document

Name	Begin date	End date
Develop Requirements	9/28/17	10/6/17

2017-NOV-27:
First pass of the software requirements document has been completed. Posted to google drive but have not heard back any feedback from the group. I will try and wrap that up this week.

2017-DEC-02:
The final requirements document has been completed. The last section of which contains best software practices that we will use to write code for the project. This document also contains the problem statement, assumptions and constraints all in one place.

Name	Begin date	End date
Develop Constraints	9/28/17	10/2/17

2017-NOV-30:
Project constraints are rolled into the requirements, as they are for the most part the same thing. Any unique constraints that develop separetely from the requirements will be mentioned verbosely.

Name	Begin date	End date
Develop Concepts	10/17/17	10/24/17
Brainstorming	10/17/17	10/17/17

Tasks

Name	Begin date	End date
Mind-Map	10/18/17	10/18/17
Functional Analysis	10/19/17	10/19/17
TRIZ	10/23/17	10/23/17
SCAMPER	10/24/17	10/24/17
Select Concept	10/30/17	11/1/17
Six Hats	10/30/17	10/30/17
Weighted Analysis	10/31/17	10/31/17
ROM Cost	11/1/17	11/1/17
Patent Search	11/2/17	11/2/17
Analyze Risks	11/3/17	11/3/17
Risk Radar	11/3/17	11/3/17
Preliminary Design	11/6/17	11/13/17
Layout Drawing *2017-NOV-29: Completed the block diagram. I assume that was what this was meant for. A more detailed schematic style diagram will be created in Altium but will be documented under the "Detailed Design" section/OrCad.*	11/6/17	11/6/17
Master Equipment List *2017-NOV-30: Completed the bill of materials for the project. I broke the document down into two sections: prototype and final production. The final costs of the production unit is roughed in for materials but not for costs. Once the final mechanical design pieces are done we can estmiate the final costs.*	11/8/17	11/8/17
Power Profile *2017-NOV-30: Power profile of primary equipment has been complete. As this project is primarily a software related project, only the main components have been calculated for power consumption: 1. Laptop computer 2. RGB+D Camera 3. Digital Projector 4. 3D Printer*	11/13/17	11/13/17

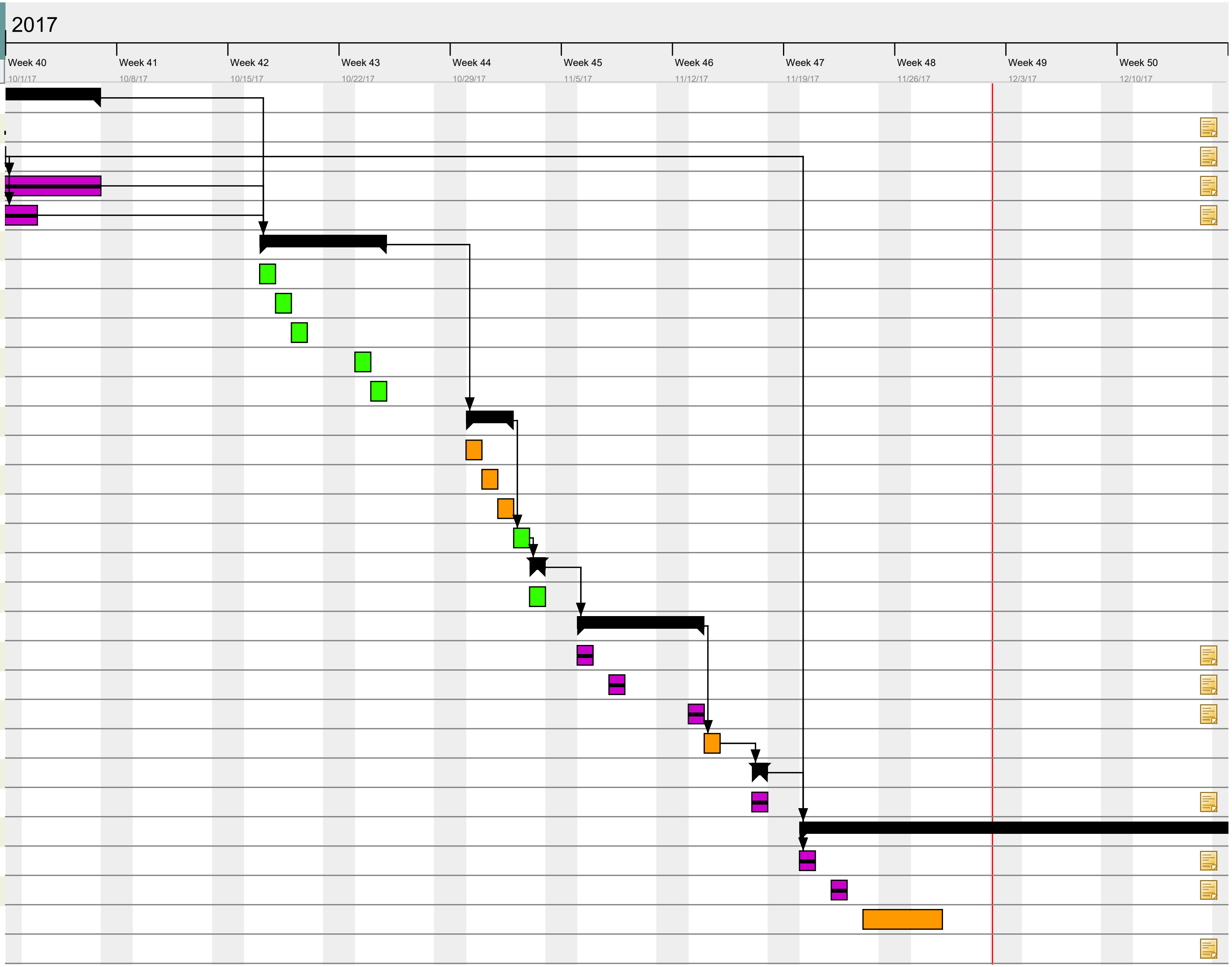
Tasks

Name	Begin date	End date
Analysis	11/14/17	11/14/17
Intermediate Design	11/17/17	11/17/17
Architecture	11/17/17	11/17/17
2017-NOV-29: Software architecture document completed. Will go over the document with the group at our next meeting.		
Detailed Design	11/20/17	1/15/18
Altium Schematic	11/20/17	11/20/17
2017-NOV-30: Completed the electrical schematic in Altium designer.		
Matlab	11/22/17	11/22/17
2017-NOV-30: The Matlab section of code is being marked as complete as the only use for this software is going to be in post processing of data for the 3D plot. It is likely that this will be done entirely in Python and will not be implemented at all.		
Final Presentation	11/24/17	11/28/17
Software	12/18/17	1/15/18
2017-NOV-29: This is the actual start of developing the software. This is not scheduled for the fall 2017 term but is something I will start working on over the winter break.		

Resources

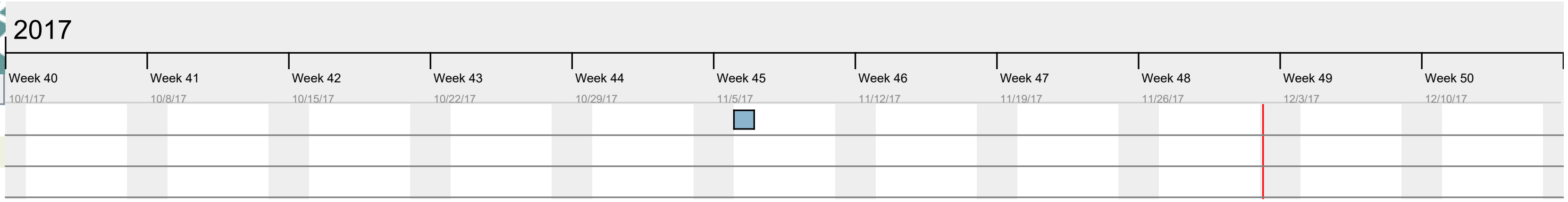
Name	Default role
Nathan Schram	project manager
Ryan Heifferon	undefined
Randy Butler	undefined

Name	Begin date	End date
Project Kickoff	9/18/17	10/6/17
Project Directives	9/18/17	9/18/17
Define Problem	9/25/17	9/27/17
Develop Requirem...	9/28/17	10/6/17
Develop Constraints	9/28/17	10/2/17
Develop Concepts	10/17/17	10/24/17
Brainstorming	10/17/17	10/17/17
Mind-Map	10/18/17	10/18/17
Functional Analysis	10/19/17	10/19/17
TRIZ	10/23/17	10/23/17
SCAMPER	10/24/17	10/24/17
Select Concept	10/30/17	11/1/17
Six Hats	10/30/17	10/30/17
Weighted Analysis	10/31/17	10/31/17
ROM Cost	11/1/17	11/1/17
Patent Search	11/2/17	11/2/17
Analyze Risks	11/3/17	11/3/17
Risk Radar	11/3/17	11/3/17
Preliminary Design	11/6/17	11/13/17
Layout Drawing	11/6/17	11/6/17
Master Equipment ...	11/8/17	11/8/17
Power Profile	11/13/17	11/13/17
Analysis	11/14/17	11/14/17
Intermediate Design	11/17/17	11/17/17
Architecture	11/17/17	11/17/17
Detailed Design	11/20/17	1/15/18
Altium Schematic	11/20/17	11/20/17
Matlab	11/22/17	11/22/17
Final Presentation	11/24/17	11/28/17
Software	12/18/17	1/15/18



Resources Chart

Name	Default role
Nathan Schram	project mana...
Ryan Heifferon	undefined
Randy Butler	undefined



Senior Design I
Group Meeting Notes/Milestones
2017-NOV-26

Group Update:

After much deliberation and research we decided to simplify our overall design for our senior project. We worked with several professors and determined that our goals to implement antennas for object detection was at a level higher than the undergraduate. Due to time constraints, budget and support we decided to drop the EM aspect entirely and focus all our efforts on an embedded application for augmented reality. The general idea now is to create an augmented reality sandbox consisting of two major pieces of hardware: Xbox Kinect V2 (for our RGB+D camera) and a USB/HDMI projector. This project is now primarily an embedded software project (with the exception of not selecting a host to run the program on). So for now, this is a pure software project. We will prove the concept with a modern laptop and the current goal is to write the entire program in Python. The requirements document is currently being updated to reflect the changes we decided as well. More document updates will follow this general updated summary of progress.

Ryan:

- Getting up to speed on Python
- Currently working some simply Python examples and doing some self-education with this language to be semi up to speed with the rest of the group with respect to the language.
- Completed ethics homework assignment

Nathan:

- Researched our projector options and became familiar with the specifications
- Searched for similar products that have the approximate specifications and fit our budget
- Ultimately failed in finding something, mainly due to constraints of cost
- Convinced our department chair to allow us to spend both semesters budget at one time on one piece of hardware and found the exact project online for a deeply discounted price ($699).
- Karla ordered our component and it should be arriving this next week (Week ending December 1st)
- Currently working on ethics homework assignment

Randy:

- Researched and found our ideal candidate for our projector
- Found the model and cost (~$1000), which is beyond the budget
- Turned over that information to Nathan to try and source an alternative equivalent component

Next Milestones: 2017-DEC-03

Ryan:

- Generate a "fake" point map for Randy to read in as a precursor to getting the actual values from the Kinect
- Create this map based on our current prototyped size of 12" x 12" so that we can actually mount and project the image onto our small sized sand box
- Create a mechanical drawing of the final sized projector, take into account the aspect ratio of the device so that we know how high we need to mount both devices to accurately map back down onto the sand
- The mechanical aspect should be created in some relevant software (I.E. Solidworks or Autodesk)

Nathan:

- Continue working with the Xbox Kinect V2 and get data into Python

- This requires many external dependency downloads with zero guarantee of success
- Look for a walk through on how to pull the Z-axis data from the camera
- Get the actual data out of this device and store into very large matrices
- Complete the updates to the requirements document
- Complete the ethics homework

Randy:

- Take in the “fake” point map created by Ryan and create a connect the dots type map in Python using any dependency you like
- Check out matplotlib, seaborns & scipy
- Display “fake” map with all connected contours out through the HDMI port to the projector
- Leverage Ryan or Nathan for assistance as needed
- Complete the ethics homework

Senior Design
Group Meeting Notes/Milestones
2017-OCT-21

<u>Group Update:</u>

After shaking off a particular professor's opinion of our knowledge on certain subject matter, we decided to take a step back and re-access our project goals and objectives. We split our group effort into two parts: embedded systems and electromagnetics. The following was done independently:

<u>Ryan:</u>

Met with Dr. Golcowski and Dr. Harid discussing options for how to proceed with "hot spot" identification. May be able to do far field/near field but need to verbosely do the math to prove the concept and required power. Will be moving over to looking at a simplified general sensing region to prove the concept and determine if specific object shapes (I.E. square, circle & triangle) could be detected if machined from highly conductive material.

<u>Nathan/Randy:</u>

Met with Dr. Liu and Dr. Connors to discuss applications and use of sensors that are within our budget and also have the potential to work with depth. Was given a structure.io sensor from Dr. Connors and was evaluated but found to have a minimum sensing depth issue (>35 cm from source). This pushes the requirements to be much further away from our sandbox than we originally thought. Also, this sensor only has hardware capability with the Apple IPad and the company does not like to give away hardware without the supporting software (downloaded from the Apple Store). Dr. Liu also gave us an Xbox Kinect which has not yet been evaluated but will have the same problem with overhead mounting and distance away from the sandbox. This may not work well for the prototype but may work quite well for the final design (which will be much larger in size). Additionally due to incurred cost of use Plexiglas we will likely move the physical construction over to a cheaper material (I.E. wood). Began working on generic python code for showing/reading topographical data using JetBrains PyCharm IDE. Was able to generate a simple 2D plot of hand drawn single topographical layer. This can be easily extrapolated to multi-layers given a resolution of the sensor. For now we will continue down the line of the Xbox Kinect due to already having a sensor.

<u>Next Milestones: 2017-NOV-04</u>

<u>Ryan:</u>

1. Investigate the solenoid receiver option for simple quadrant object detection
    a. Use software to fill the gap if required
2. Investigate use of "off the shelf" pre-built antenna options
3. Evaluate pricing for "off the shelf" options against our budget
4. Power calculations on power for a given frequency and FCC regulation for legality
5. Engineering notebook
6. Add GanttProject and start tracking project task completion

<u>Nathan:</u>

1. Email/Call Zivid for information on RGB+D sensor
2. Discuss Sterolabs options with Dr. Connors and pull datasheets for info
3. Start playing with the Kinect mounted above our prototype
4. Complete requirements document
5. Software/Hardware flowchart
6. Add GanttProject and start tracking project task completion
7. How to pull of Kinect data and read into python
8. Parse vectors and matrices for z-height
9. HWSW Acceleration (at the back end)
10. See python code (Augmented Reality Sandbox.py) for detailed next steps

<u>Randy:</u>

1. Investigate GPR with applications and examples
2. URL for previous work (benchmarking for component selection on a budget)
3. Patent search, create document with results, be verbose and include all results that are close enough to our global project goals (Augmented Reality Sandbox)
4. Add GanttProject and start tracking project task completion
5. How to take Nathan’s parsed data from python (2D representation of height) and project that out through HDMI using python only

Wunderlist

- -App, fancy grocery list to share with other people
- -phone app that updates in real time
- -easy way to work through simple tasks from a multi view

Hardware

- -Lidar sensors & costs
  - -Garmin Lite V3
- -Antenna
  - -dipole for tx
  - -loop for rx
- -Controllers
  - -Beagle Bone Blue
  - -RPi V3
  - -BBB
  - -Snapdragon
  - -Nvidia Tegra K1
  - -other options
- -Mapping topo lines
  - -micro projector
  - -alternative projecting methods
- -1 foot cube made out of clear plexi glass

Project Code Names:

- -Topographical arial mapping and finding (T.A.M.F.)
- -Topographical arial mapping utility (T.A.M.U.)
- -Real Time Topographical Mapping and finding (R.T.T.M.F.)
- -continue to spitball names

<u>Goals by person</u>

1. Ryan
    - 1.1. Research the impacts of sand
    - 1.2. Types of Antennas used for Tx and Rx
    - 1.3. Skin Depths
    - 1.4. Sand that can be penetrated easily
    - 1.5. Salt?
    - 1.6. Costs
    - 1.7. 3 PowerPoint slides for presentation
    - 1.8. Brochure section
    - 1.9. EM & Lidar interference
    - 1.10. Signal processing
    - 1.11. Budget.
    - 1.12. Crowd source funding
2. Randy
    - 2.1. Lidar sensor research
    - 2.2. Point cloud research
    - 2.3. Costs
    - 2.4. Alternative sensors
    - 2.5. Projectors
    - 2.6. How to project back down onto the sand
    - 2.7. Kinect Sensor investigation
    - 2.8. 3 PowerPoint slides for presentation
    - 2.9. Brochure section
    - 2.10. Budget
    - 2.11. Crowd source funding
3. Nathan
    - 3.1. Controllers
    - 3.2. MCU/MPU options
    - 3.3. FPGA?
    - 3.4. Interfacing
    - 3.5. Costs
    - 3.6. HW/SW
    - 3.7. Debian?
    - 3.8. Latency
    - 3.9. 3 PowerPoint slides for presentation
    - 3.10. Brochure section
    - 3.11. Lidar development via laser
    - 3.12. MATLAB point cloud implementation
    - 3.13. Budget
    - 3.14. Crowd source funding

Deliverables for 2017-OCT-2:

1. Each person to have completed their slides and brochure for the presentation

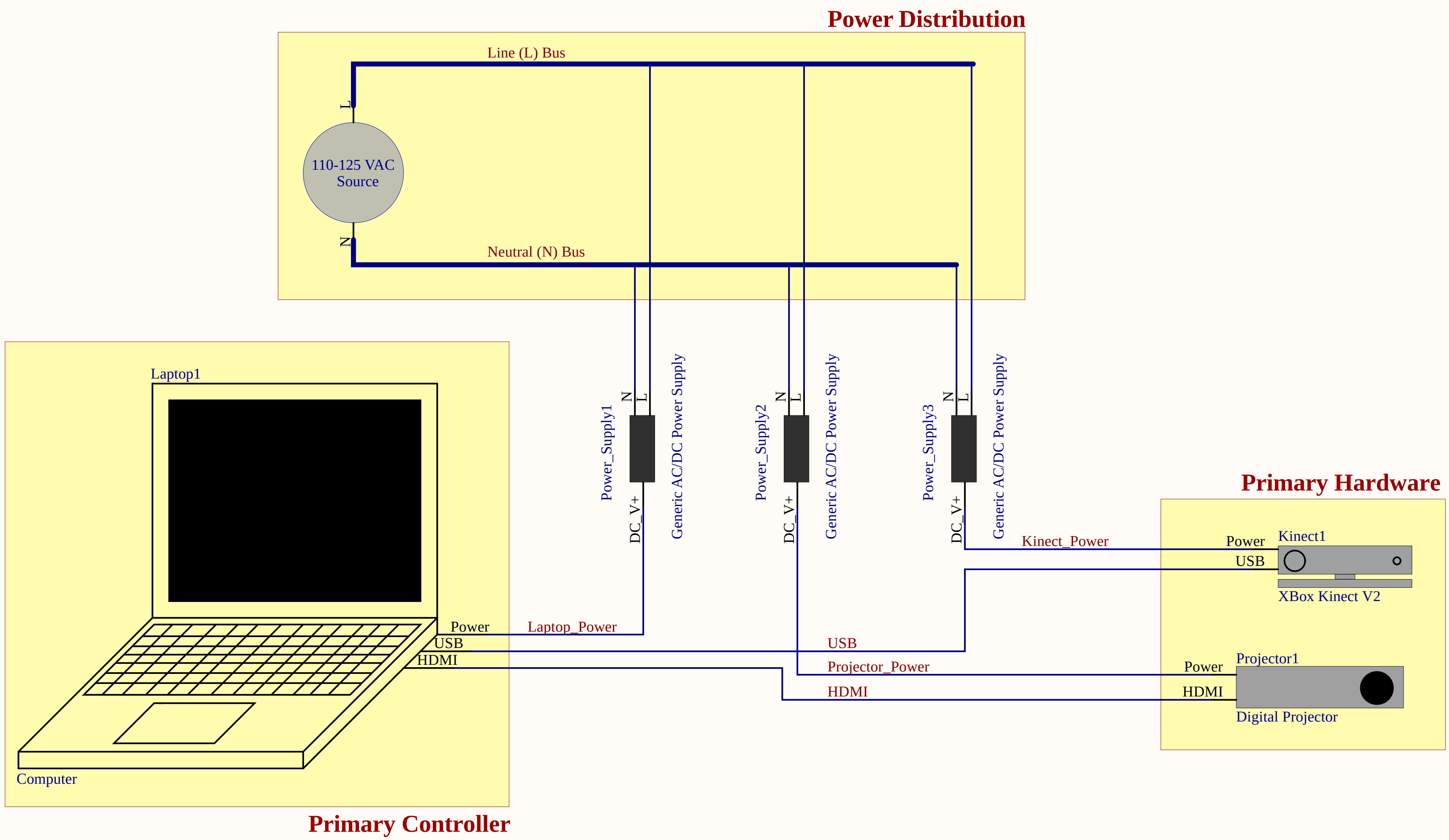
Deliverables for 2017-OCT-7:

1. Complete the research for the listed topics per person to be able to discuss as a group
2. Gantt Installed
3. Wunderlist installed

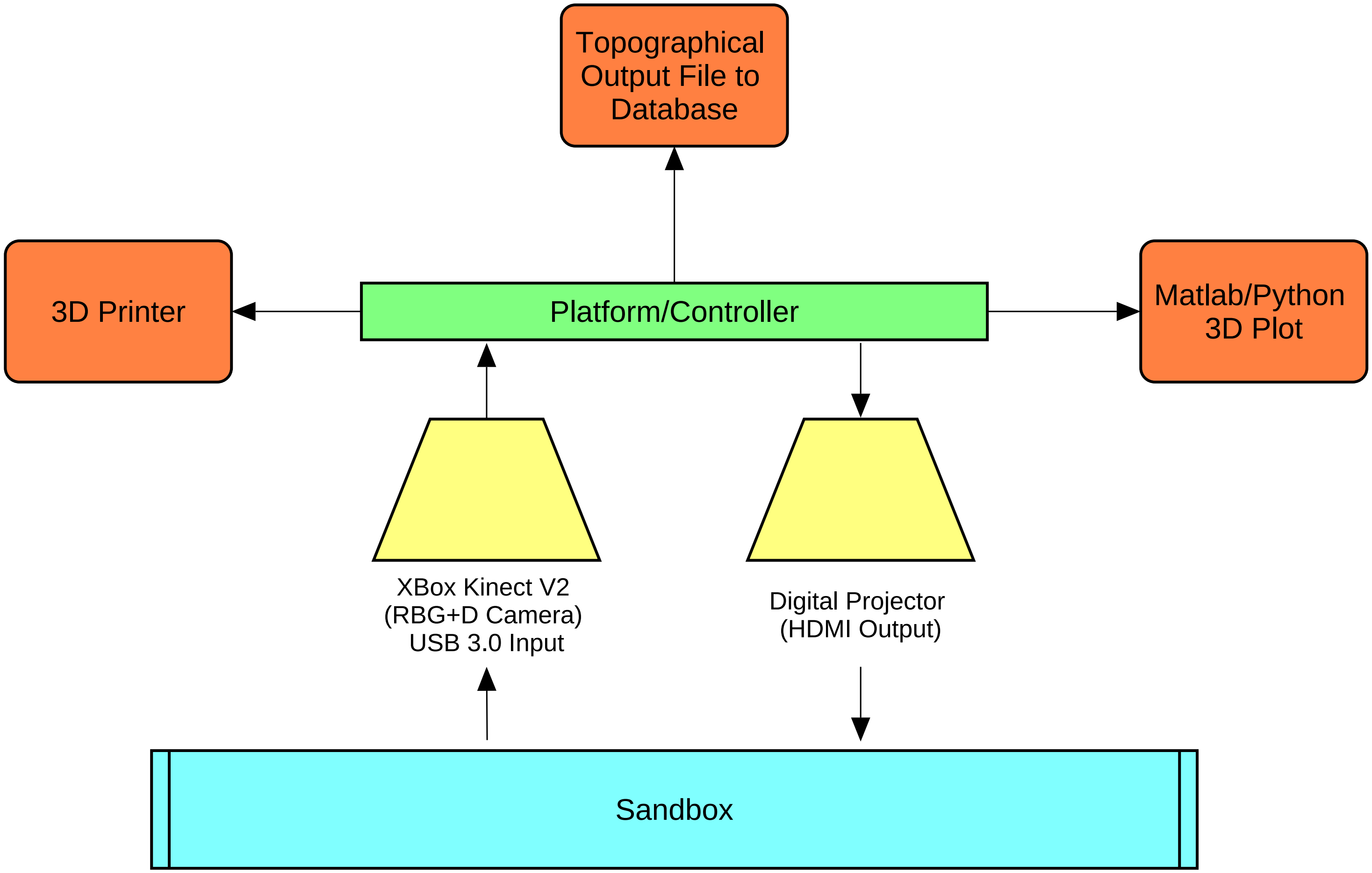
Augmented Reality Sandbox
Power Profile
Group Members: N. Schram, R. Heifferon, R. Butler

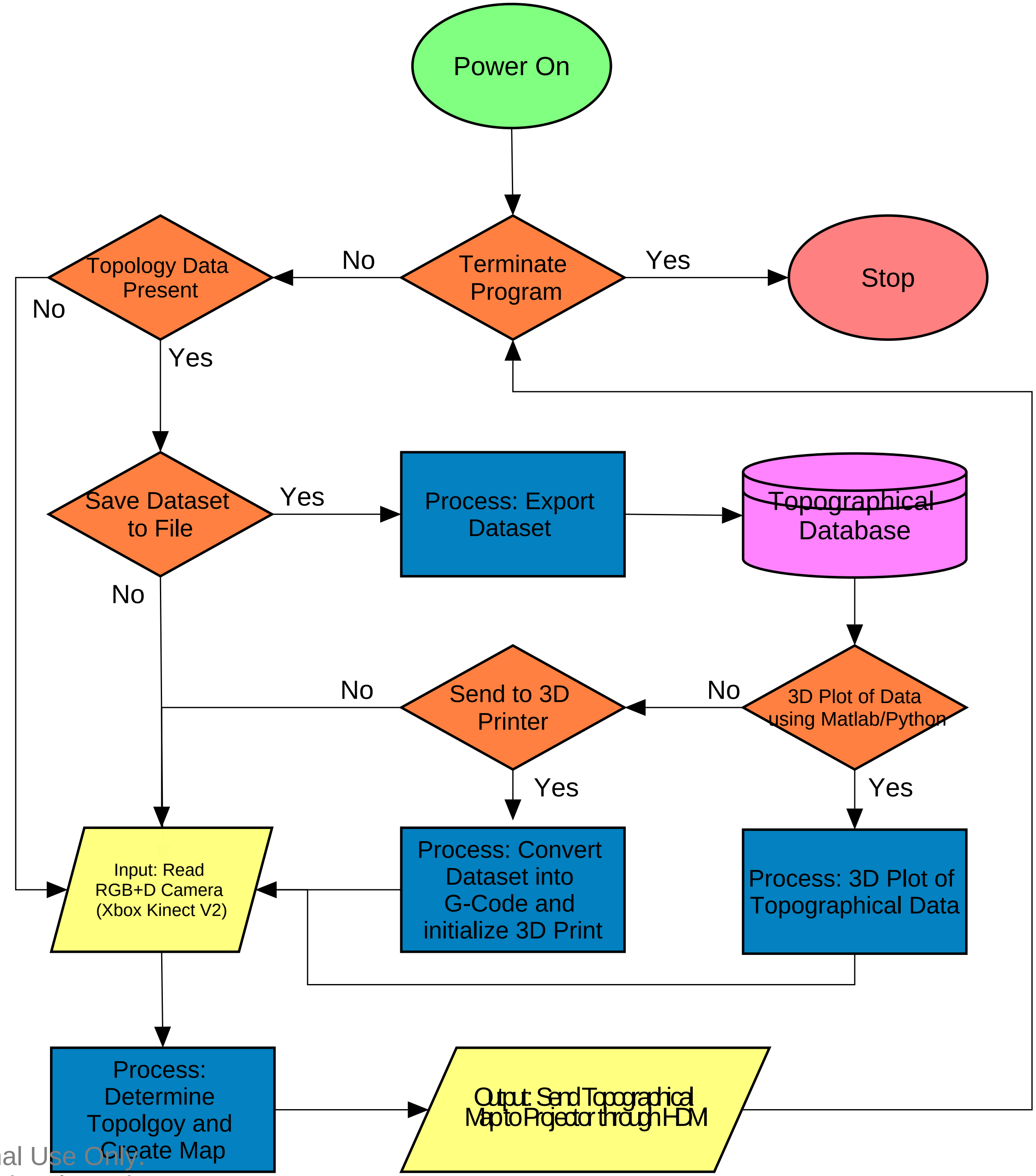
Item Number	Part Number	Description	Power Requirements (Maximum Draws) (Assuming 120 VAC)	
1	N/A	AC Mains (110-125 VAC), 3 Prong Grounded Outlet	N/A	**Source**
2	ARS-001	Laptop Computer	804W, 6.7 A	**Power Draws**
3	ARS-002	Digital Projector	305 W, 2.54 A	
4	ARS-003	RGB+D Camera	320W, 2.67 A	
5	ARS-018	3D Printer	636W, 5.3 A	
			Global Power Consumption: 2065W, 17.21	

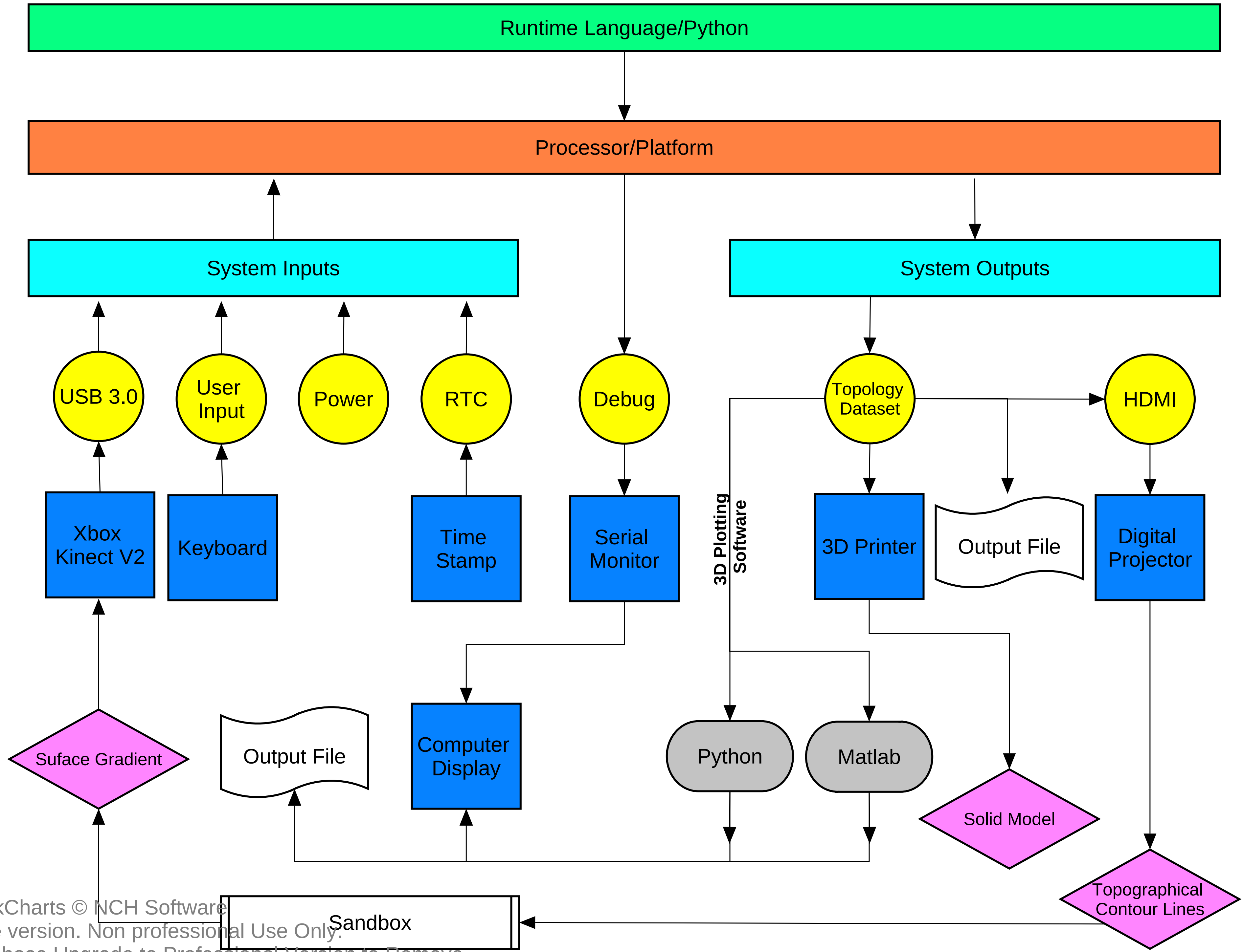
Disclaimer: This power calculation was completed assuming peak performance, as if everything is under heavy load simultaneously. Equipment consumes significantly less power when operating normally.



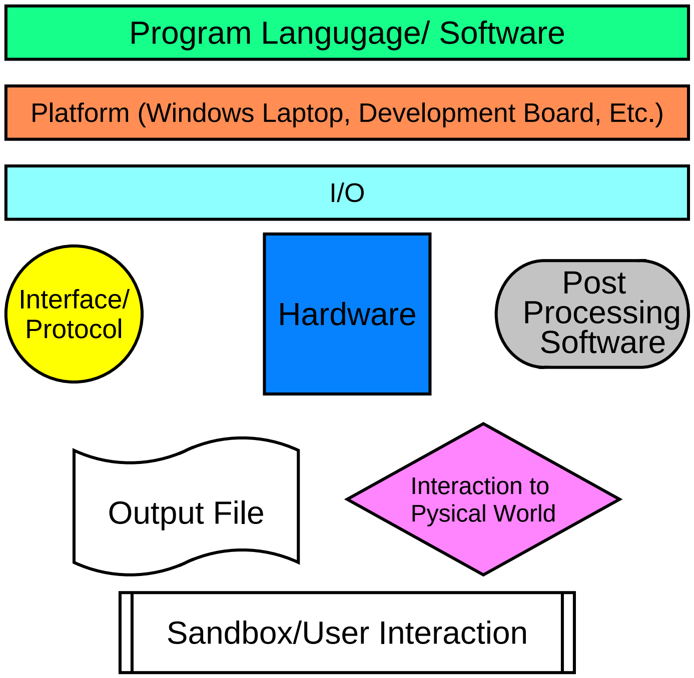
Title	Augmented Reality Sandbox Schematic		
Size A	Number N/A		Revision 00
Date:	11/30/2017	Sheet 1 of 1	
File:	F:\UCD\..\ARS_Schematic.SchDoc	Drawn By: N. Schram	


Topographical Output File to Database
3D Printer
Platform/Controller
Matlab/Python 3D Plot
XBox Kinect V2 (RBG+D Camera) USB 3.0 Input
Digital Projector (HDMI Output)
Sandbox


Power On
Terminate Program
Yes
Stop
No
Topology Data Present
No
Yes
Save Dataset to File
Yes
Process: Export Dataset
Topographical Database
No
3D Plot of Data using Matlab/Python
No
Send to 3D Printer
No
Yes
Yes
Input: Read RGB+D Camera (Xbox Kinect V2)
Process: Convert Dataset into G-Code and initialize 3D Print
Process: 3D Plot of Topographical Data
Process: Determine Topolgoy and Create Map
Output: Send Topographical Map to Projector through HDMI


Runtime Language/Python
Processor/Platform
System Inputs
System Outputs
USB 3.0
User Input
Power
RTC
Debug
Topology Dataset
HDMI
Xbox Kinect V2
Keyboard
Time Stamp
Serial Monitor
3D Plotting Software
3D Printer
Output File
Digital Projector
Suface Gradient
Output File
Computer Display
Python
Matlab
Solid Model
Topographical Contour Lines
Sandbox

LEGEND:


Program Langugage/ Software
Platform (Windows Laptop, Development Board, Etc.)
I/O
Interface/ Protocol
Hardware
Post Processing Software
Output File
Interaction to Pysical World
Sandbox/User Interaction

US009824490B1

(12) United States Patent
Côté et al.

(10) **Patent No.: US 9,824,490 B1**
(45) **Date of Patent: Nov. 21, 2017**

(54) **AUGMENTATION OF A DYNAMIC TERRAIN SURFACE**

(71) Applicant: **Bentley Systems, Incorporated**, Exton, PA (US)

(72) Inventors: **Stéphane Côté**, Lac Beauport (CA); **Ian Létourneau**, Quebec (CA); **Jade Marcoux-Ouellet**, Quebec (CA)

(73) Assignee: **Bentley Systems, Incorporated**, Exton, PA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 85 days.

(21) Appl. No.: **14/733,660**

(22) Filed: **Jun. 8, 2015**

(51) **Int. Cl.**
G06T 17/05 (2011.01)
E02F 9/24 (2006.01)
G06T 19/00 (2011.01)
G06T 17/10 (2006.01)

(52) **U.S. Cl.**
CPC ***G06T 17/05*** (2013.01); ***E02F 9/245*** (2013.01); ***G06T 17/10*** (2013.01); ***G06T 19/006*** (2013.01)

(58) **Field of Classification Search**
CPC G06T 17/05; G06T 19/00; G06T 19/006; G06T 2219/004; E02F 9/264; E02F 9/261
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

8,249,306 B2 8/2012 Nielsen et al.
8,384,742 B2 2/2013 Nielsen et al.
8,914,199 B2 12/2014 Nomura et al.
9,465,129 B1 * 10/2016 Olsson G01V 3/15
2014/0184643 A1 * 7/2014 Friend G09G 3/003 345/633
2014/0200863 A1 * 7/2014 Kamat G01C 15/00 703/1

(Continued)

OTHER PUBLICATIONS

Kreylos, "Augmented Reality Sandbox", URL:<http://www.iday.ucdavis.ed/~okreylos/ResDev/SARndbox/MainPage.html>, Jun. 4, 2015, 3 pages.*

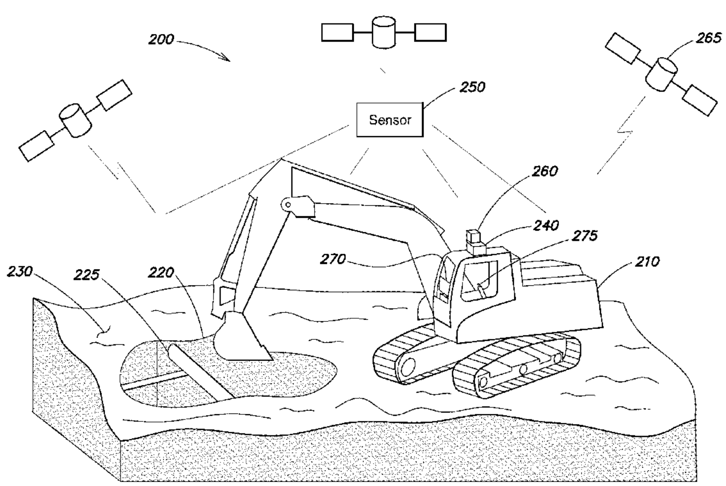
(Continued)

Primary Examiner — Gregory J Tryder
Assistant Examiner — Diane Wills
(74) *Attorney, Agent, or Firm* — Cesari and McKenna, LLP; James A. Blanchette

(57) **ABSTRACT**

In one embodiment, an augmented view is generated that accounts for dynamically changing terrain surface at a site. A sensor captures live georeferenced terrain surface topography for the site. A camera captures an image of the site. Further, a tracking system determines a georeferenced camera pose of the camera. An augmented reality application aligns a georeferenced three-dimensional (3-D) model for the site with the live georeferenced terrain surface topography. Then, using at least the captured image, the georeferenced camera pose, the georeferenced 3-D model and live georeferenced terrain surface topography, the augmented reality application creates an augmented view of the site that shows graphical representations of subsurface features. At least a portion of the graphical representations are dynamically conformed to the contours of the terrain surface in the image based on the live georeferenced terrain surface topography. The graphical representations may include virtual excavation and/or virtual paint markings.

20 Claims, 9 Drawing Sheets



Our project is much more centralized around Hardware and Software. Because we did not secure any exterior funding, we are very constrained by our hardware choices. Our RGB-D camera was donated by one of our professors and our entire capstone budget was spent on the projector device which is absolutely required. We did not trust purchasing a refurbished or pre-owned projector. If any of our hardware is to fail we will likely have to provide our own money to replace the parts.

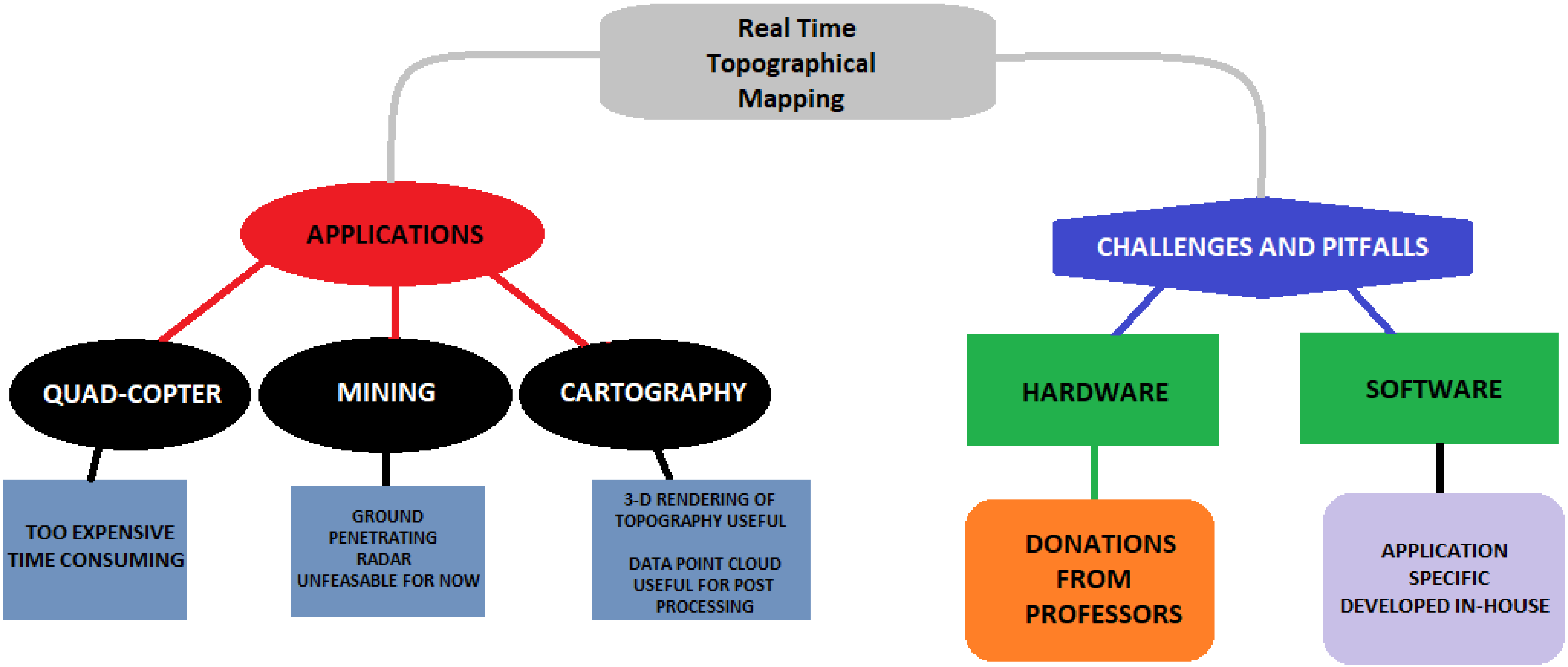
Because we're generating the software ourselves, there is a larger constraint of completion based on our ability to troubleshoot and correct software issues. But, because we're writing the code ourselves it should, theoretically, be easier to debug as we'll be fully immersed in the project.

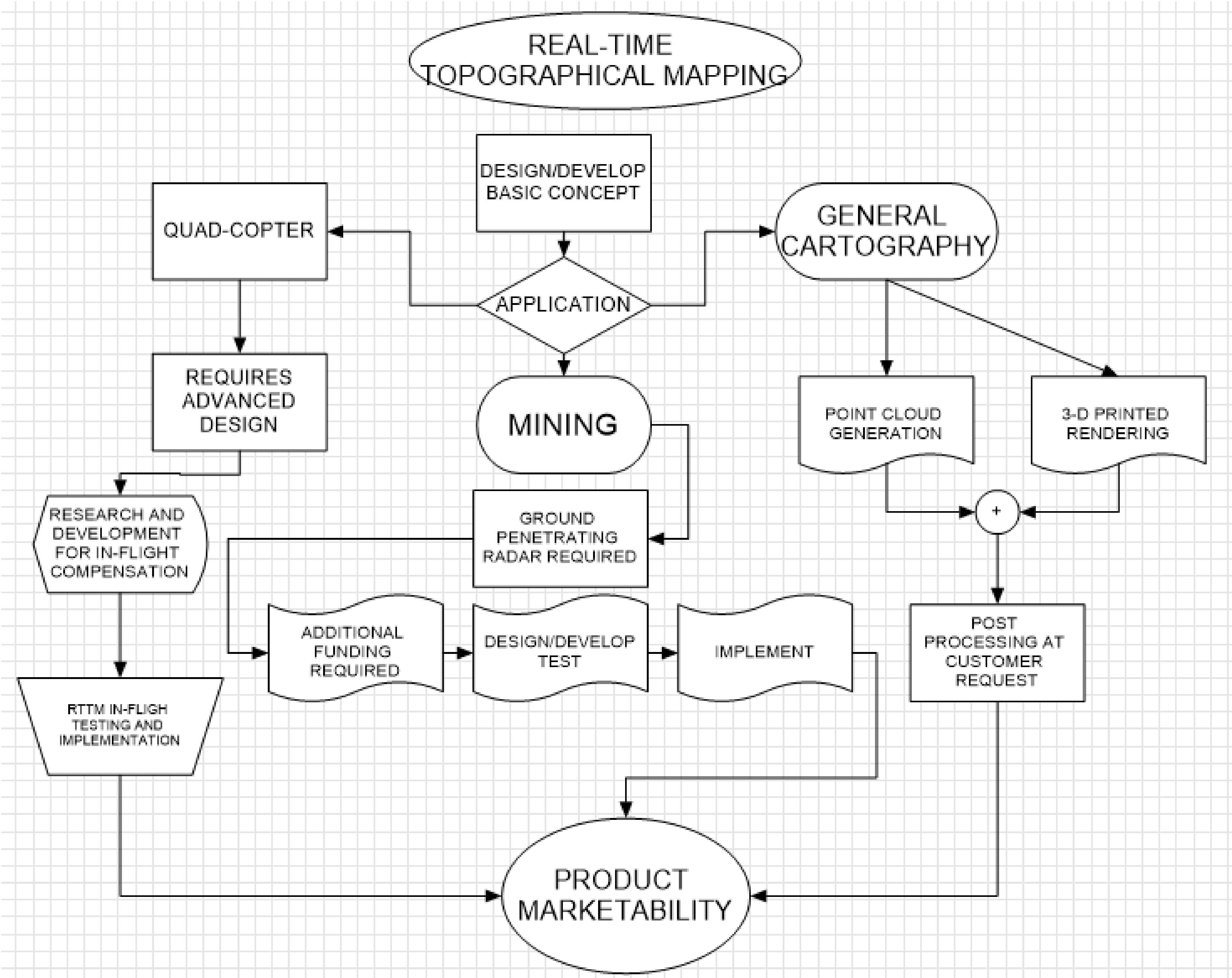
Time management is our last big concern. As long as we affectively layout and achieve milestones within a reasonable amount of time we should have "00.00" problems completing our project long term.

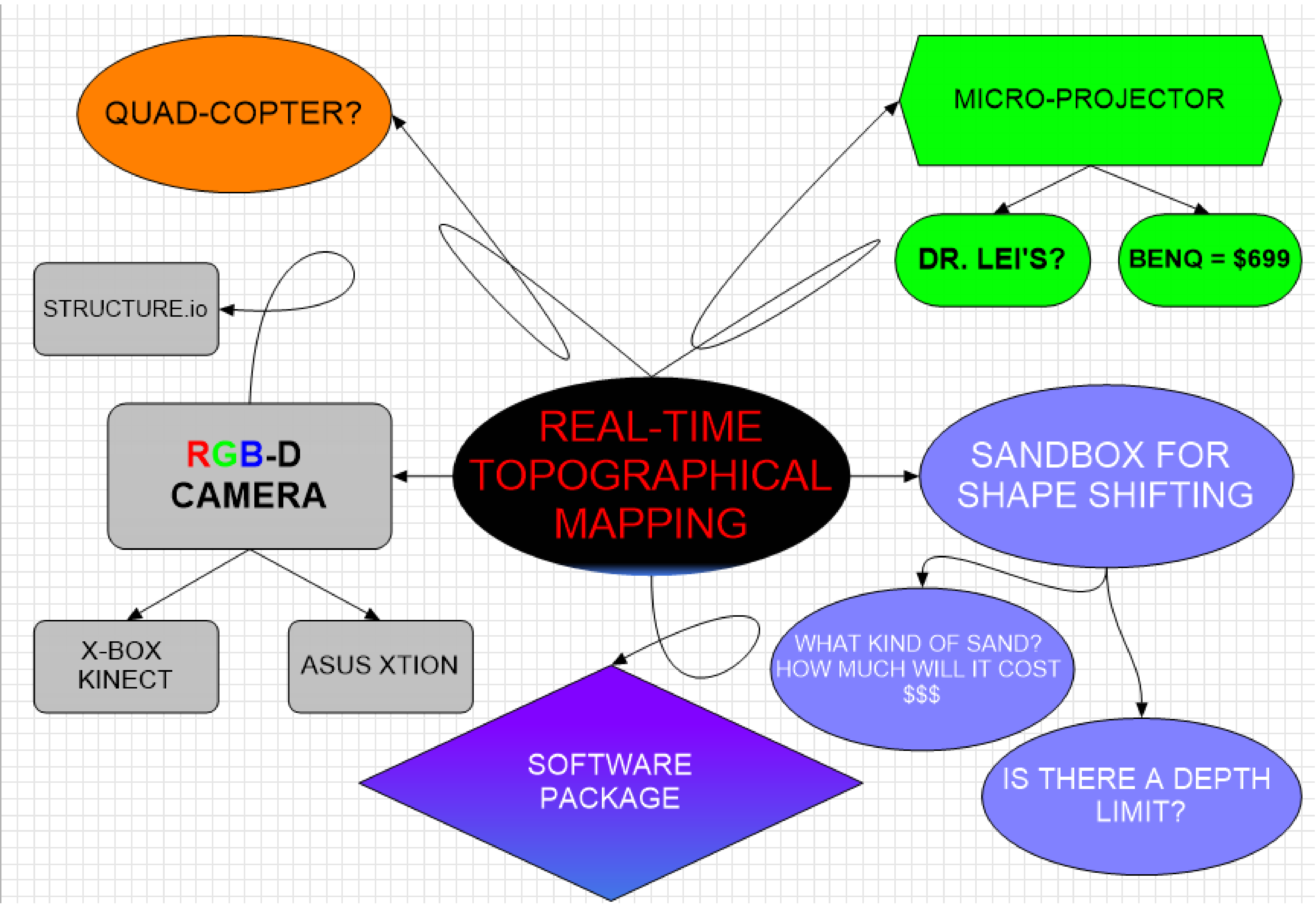
	TIME MANAGEMENT		SOFTWARE FAILURES	DESTROYING HARDWARE
EXTREME				GG ☹
MODERATE				
LOW				
VERY LOW				

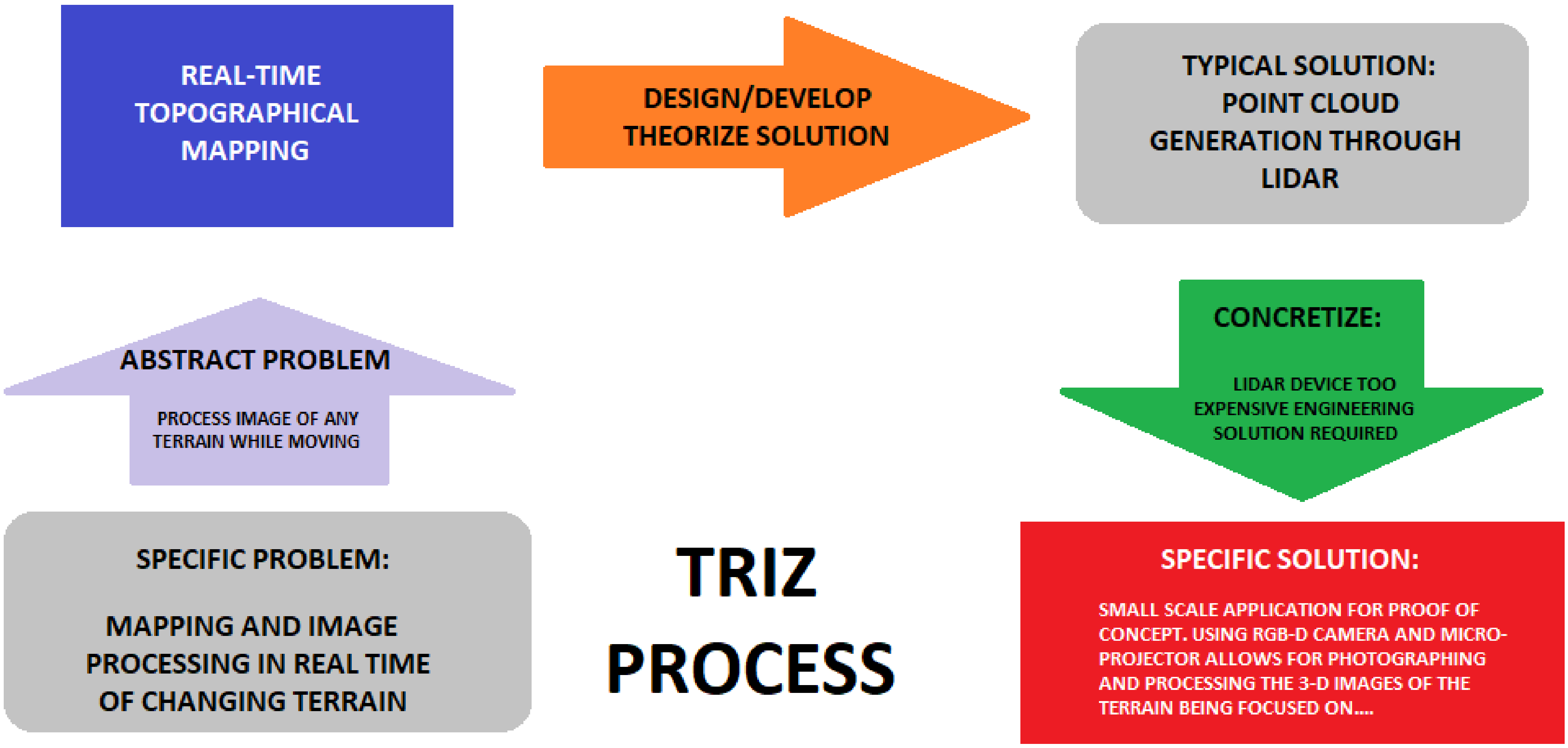
AS TIME APPROACHES THE COMPLETION DEADLINE

Brainstorming....




REAL-TIME TOPOGRAPHICAL MAPPING
DESIGN/DEVELOP BASIC CONCEPT
QUAD-COPTER
APPLICATION
GENERAL CARTOGRAPHY
REQUIRES ADVANCED DESIGN
MINING
POINT CLOUD GENERATION
3-D PRINTED RENDERING
RESEARCH AND DEVELOPMENT FOR IN-FLIGHT COMPENSATION
GROUND PENETRATING RADAR REQUIRED
+
ADDITIONAL FUNDING REQUIRED
DESIGN/DEVELOP TEST
IMPLEMENT
POST PROCESSING AT CUSTOMER REQUEST
RTTM IN-FLIGH TESTING AND IMPLEMENTATION
PRODUCT MARKETABILITY


QUAD-COPTER?
MICRO-PROJECTOR
DR. LEI'S?
BENQ = $699
STRUCTURE.io
RGB-D CAMERA
REAL-TIME TOPOGRAPHICAL MAPPING
SANDBOX FOR SHAPE SHIFTING
X-BOX KINECT
ASUS XTION
WHAT KIND OF SAND? HOW MUCH WILL IT COST $$$
SOFTWARE PACKAGE
IS THERE A DEPTH LIMIT?


REAL-TIME TOPOGRAPHICAL MAPPING
DESIGN/DEVELOP THEORIZE SOLUTION
TYPICAL SOLUTION: POINT CLOUD GENERATION THROUGH LIDAR
CONCRETIZE:
LIDAR DEVICE TOO EXPENSIVE ENGINEERING SOLUTION REQUIRED
SPECIFIC SOLUTION:
SMALL SCALE APPLICATION FOR PROOF OF CONCEPT. USING RGB-D CAMERA AND MICRO-PROJECTOR ALLOWS FOR PHOTOGRAPHING AND PROCESSING THE 3-D IMAGES OF THE TERRAIN BEING FOCUSED ON....
TRIZ PROCESS
SPECIFIC PROBLEM:
MAPPING AND IMAGE PROCESSING IN REAL TIME OF CHANGING TERRAIN
ABSTRACT PROBLEM
PROCESS IMAGE OF ANY TERRAIN WHILE MOVING