Course Project Overview

Overview:

Each team will choose one of 21 possible application concepts. No two teams can be assigned the same project. A *general* project description is provided below and each project has an associated client/end user. The team will be responsible for scheduling meetings with the client to determine exact and specific requirements for each application concept.

Team Formation

There will be six teams of 4 members each. You are left to choose and form your own teams. Teams should agree on the application concept to work on: if you do not choose an application, one will be chosen for you.

Project Assignment / Memo of Understanding:

Once formed, each team should come up with a name for their team and then create a memo of understanding between the team and Dr. Benton requesting to develop the software for the associated application concept. The memo should state that you will undertake the project and complete it by the deadlines given and provide all required deliverables on time. I would suggest having a second and third choice ready in case you do not get your first choice. You should conduct a skills inventory of each team member that includes, for example, relative programming/coding skill, Java experience, Android experience, graphic design experience, database design experience, management experience, skill with using certain tools, etc.

Create your MOU in Microsoft Word or other editor, but create a PDF file for turn in (this function is available automatically in Microsoft Office products through Save As… PDF).

At a minimum, each team should pick a Project Manager**.** This person does not necessarily have to be the strongest or most experienced programmer, but rather someone who may have experience with management. The Project Manager will be the primary POC between the team and the client and be responsible for project management activities. Other team members should be assigned **tentative** initial roles for the project kick-off. Members may have more than one role, including the Project Manager.

Example roles include, but are not limited to:

- Programmer / Implementer
- Tester / Quality Assurance
- Requirements Analyst / System Architect
- User Interface Designer
- Team Scribe / Document Editor
- Configuration/Deployment Manager
- Database Designer/Manager

Each team must determine which deployment platform they are targeting (Android or Java Application) based on relative strengths or learning goals of the team.

The MOU must have the following content:

- **Team Name**
- **Project Manager**
- **Team Members/Roles**, with contact information (email)
- **Proposed Project**
- **Deployment Platform Chosen (Java or Android)**
- **Milestones:** based on the constraints listed below and the dates provided in the course lab schedule (posted with the syllabus on the Sakai course website), list the milestones for deliverables and reviews required for the project. The major milestones are 2 “sprints” and focused subject reviews in between.
- **Rationale**: provide a 3-4 sentence rationale, based on the skills inventory of your team, as to why your team can succeed in developing this application (based on the expected technical skills required for the application). For example, if team members have prior Android development experience or knowledge, database design experience (if the application will more than likely require a database), prior Java graphics experience, knowledge of appropriate algorithms, etc.
- **Development Strategy and Approach:** based on the development constraints listed in the below sections, detail the software lifecycle methodology and software development paradigm you will follow.
- **Communication Plan:** detail the methods that your team will use to communicate and meet outside of lab times. Determine and provide (common) available times for all team members for in-person meetings and/or on-line meetings.

If the memo does not meet these requirements, your choice will not be considered.

Projects are assigned on a first come, first serve basis. Email your MOU to Dr. Benton once it is complete. If your MOU and choice is approved, your team will be notified via email and a class-wide email will announce the formation of your team and project selection. Dr. Benton will coordinate introduction of your team with the client.

All teams must be formed and have an approved project NLT Wednesday, 18:00, 24 August. Project kickoff begins 25 August.

Technical Requirements:

The major technical requirement for the application involves the deployment environment. You must choose one of two deployment platforms: an app in the **Android application environment (Android Version 4.0.3 or below)** or a **Java/Swing application (Java 7.0 or 8.0)**. For Android applications, a Toshiba AT200 tablet will be provided for use by the team for prototyping and demonstration purposes.

Development Constraints:

Teams will be constrained in the use of certain tools and techniques for software lifecycle methodology, source code development, configuration management, and UML diagramming.

- All teams will follow an iterative/agile software lifecycle approach with two iterations (two sprints). The first sprint must produce a working prototype (no business rules or logic), in the
- Each team must follow an object-oriented development paradigm
- Teams must use Eclipse as their IDE.
- Teams will produce either a Java application user interface (Swing) or Android app user interface.
- Teams will use Astah Community Professional for their UML tool. Astah provides a free professional license for students. See http://astah.net/student-license-request/
- Teams must document the API of their source code with Javadoc. The HTML API of the source code will be published on a
- Teams must use JUnit for unit testing.
- Teams must follow a minimal Java style guide, to be provided by Dr. Benton.
- Teams will use Subversion for configuration management and source code control, using School of Computing accounts and repositories. Mr. Keith Lynn will provide account information and instructions will be provided on how to integrate Subversion into the Eclipse environment during lab times.
- Teams will use Microsoft Project for their project management scheduling and GANTT charts. Microsoft Project is available as part of the USA Dreamspark available software.
- For applications that require a database interaction, teams will use a MySQL Workbench local client (for Java Applications) or a SQLite local client (for Android applications).
- Each team will use a Sakai Project site to hold deliverables, team meeting journals, and all associated project files. The project site will be created for you and will follow a common structure.

Team Meetings, Communications, and Project Status:

Teams will meet at a minimum once weekly as a group (all members). Other meetings with less than full team members are of course expected given the diverse schedules of your group. Non lab times on Tuesdays should be set aside for team meetings (everyone in the class should have 12:30 – 13:30 free). Some lab times on Thursday may be available for team meetings, but most will be used for reviews and targeted instruction related to the project. Each team should come up with a plan on how/where/when to have meetings outside of lab time: the Sakai Big Blue Button meeting app is provided as an option as well as Sakai's online chat option. Other options might include Skype, social media options, etc.

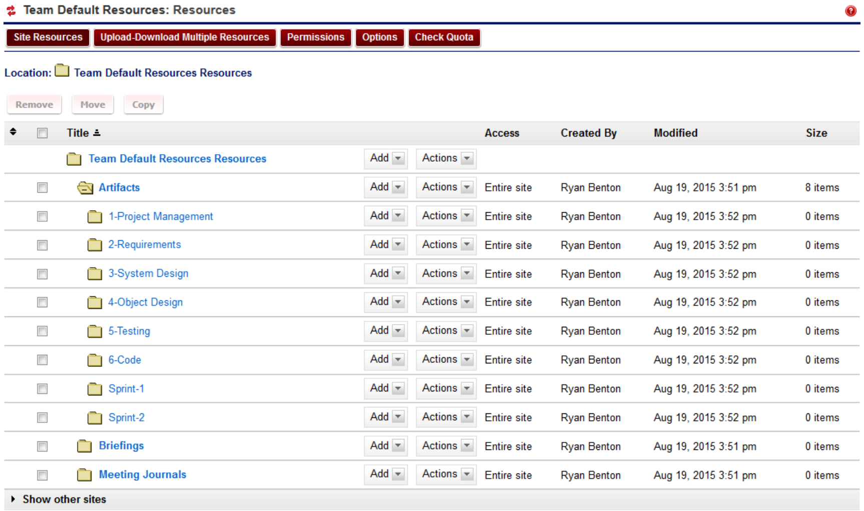
- Each team will be responsible for journaling their development activities and providing rationale management. A template Sakai site will be provided for each team that will be used to hold artifacts, keep project related documents/briefings, and keep meeting journals. Each team will keep a meeting journal for every week of the project.
- Apart from customer meetings, each team Project Manager will be responsible for providing Dr. Benton a bi-weekly status update email on their project (regardless of client): the email should summarize the current work on the project for that time period and a 2-3 sentence summary of activities. The email serves as a communication to the instructor on the current health of the project, status of identified risks, progress toward meeting milestones and reviews, concerns, and team member issues. The subject line of the email should be: CSC-331 Team <Name> Status. Using the Sakai project web site for email communications will automatically archive any sent messages.

  Status updates are due on the following dates:
  8 Sept
  22 Sep
  6 Oct
  20 Oct
  3 Nov
  17 Nov
  1 Dec

- Each team is responsible for scheduling meetings and demonstrations with their client. Client meetings will commence after Project Kickoff. Teams are responsible for maintaining communication with their client on progress or milestones as they see fit (if the client is Dr. Benton, the bi-weekly status update will meet this requirement). ***Clients may require additional communication requirements!***
- Each team is responsible for scheduling a client demonstration test, prior to final project presentations on the last day of class.
- Each team will, at the end of project, must have all final documents and artifacts checked into their project site on Sakai. This must be done by 1 PM, December 6, 2016.

Deliverables and Reviews:

Each team will be responsible for providing rationale management in the form of an artifacts slide set. Each team will give a 10 minute review on major development aspects (requirements, analysis, design, testing) during the semester and also provide a review for each of the two sprints that are planned. Artifacts are due at the beginning of the lab time that is recorded in the lab schedule. All required artifacts and slides will be directly downloaded from the Sakai project web site. Specific instructions will be provided for deliverables and reviews. A final report that incorporates incremental reviews and artifacts will be produced as a final deliverable.



Location: Team Default Resources Resources

□	Title			Access	Created By	Modified	Size
	Team Default Resources Resources	Add	Actions				
□	Artifacts	Add	Actions	Entire site	Ryan Benton	Aug 19, 2015 3:51 pm	8 items
□	1-Project Management	Add	Actions	Entire site	Ryan Benton	Aug 19, 2015 3:52 pm	0 items
□	2-Requirements	Add	Actions	Entire site	Ryan Benton	Aug 19, 2015 3:52 pm	0 items
□	3-System Design	Add	Actions	Entire site	Ryan Benton	Aug 19, 2015 3:52 pm	0 items
□	4-Object Design	Add	Actions	Entire site	Ryan Benton	Aug 19, 2015 3:52 pm	0 items
□	5-Testing	Add	Actions	Entire site	Ryan Benton	Aug 19, 2015 3:52 pm	0 items
□	6-Code	Add	Actions	Entire site	Ryan Benton	Aug 19, 2015 3:52 pm	0 items
□	Sprint-1	Add	Actions	Entire site	Ryan Benton	Aug 19, 2015 3:52 pm	0 items
□	Sprint-2	Add	Actions	Entire site	Ryan Benton	Aug 19, 2015 3:52 pm	0 items
□	Briefings	Add	Actions	Entire site	Ryan Benton	Aug 19, 2015 3:51 pm	0 items
□	Meeting Journals	Add	Actions	Entire site	Ryan Benton	Aug 19, 2015 3:51 pm	0 items

Grading:

Each team will receive a team score based on reviews and artifacts that are due at various milestones during the semester. Your individual score will be modified based on peer feedback that will be collected during each of the key review phases. Each team will have an opportunity to provide direct feedback on member participation and support to Dr. Benton. Typically, all team members will be assigned the same grade, but non-participation and lack of effort may affect how many points you receive individually for each graded deliverable.

Team Grade	**Kickoff/MOU/Startup**	**5%**
	SPMP/SCMP Review	**5%**
	RAD Review	**15%**
	SDD Review	**10%**
	Sprint 1 Review / Prototype	**15%**
	ODD Review	**10%**
	Test Plan Review	**10%**
	Sprint 2 Review / Final Report	**15%**
	Customer Demonstration	**5%**
	Code / Style Guide / Javadoc	**5%**
	Management Artifacts / Journal	**5%**
		100%

Application #1: Fantasy Sports League

This application will manage participants in a Fantasy Sports League: the app will be single-themed sport and could be Football, Baseball, Soccer, etc., chosen by the development team.

For example: Fantasy football is an interactive competition in which users compete against each other as general managers of virtual teams built from real players. The players that an individual is able to manage are professional American football players in the National Football League. The different actions people are able to make are drafting, trading, adding or dropping players, and changing rosters. Due to the growth of the Internet, fantasy football has vastly increased in popularity, particularly because fantasy football providers such as ESPN, Yahoo, CBS, and the NFL itself are able to keep track of statistics entirely online, eliminating the need to check box scores in newspapers regularly to keep track of players. Most leagues have a single week championship in week 16 of the NFL season. [http://en.wikipedia.org/wiki/Fantasy_football_%28American%29]

Application #2: Sports Seer

Sport Seer is a predictive sports application that allows a user to estimate the performance of teams in a particular sport given a specific matchup, player performance statistic, or win/loss record for a season. The app will be a single-themed sport and could be Football, Baseball, Soccer, etc., chosen by the development team. The prediction engine is based upon past performance of players and teams combined. A user should be able to choose team matchups and player statistic records.

This app would be an ideal companion for professional sports analysts. For example, Vin Scully is one of the most respected sport's analysts of all time. At the age of 25, Scully became the youngest person ever to broadcast a World Series, which is a record in and of itself. Some of Scully's most memorable predictions were:
* The 1955 World Series, when the Brooklyn Dodgers won their first World Championship.
* Hank Aaron's 714th home run against the Dodgers in 1974.
* Sandy Koufax's perfect game in 1965.
* The Dodgers' four consecutive home runs against the San Diego Padres on Sept. 18, 2006.

As another example, what would be the result if the 1993 Dallas Cowboys were matched up against the 1992 Dallas Cowboys (Super Bowl champs both years)? What would be the point spread? What would happen if the Auburn Tigers played the Air Force Academy Falcons? What would be the outcome of the USA Jaguars and the Alabama Crimson Tide?

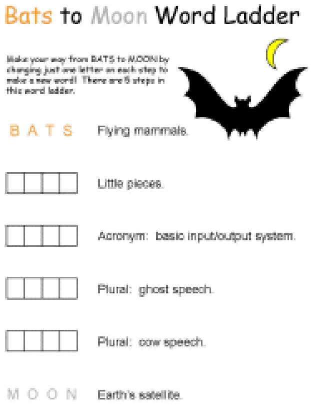
These are just examples of the type of analysis that Sports Seer might produce.

Application #3: Word Games

This application is a collection of two single-player word games. One word game must be "Word Ladders" the other word game will be "Jumble".

Word Ladders are games that have a starting word and an ending word. There is some number of intermediate words in between the start word and end word, with each intermediate word differing by exactly one character from the previous word. Some Word Ladders provide hints or clues or partial letters to help the player.

For example: [http://www.dltk-kids.com/type/word_ladders.htm]



Bats to Moon Word Ladder

Make your way from BATS to MOON by changing just one letter on each step to make a new word! There are 5 steps in this word ladder.

BATS Flying mammals.

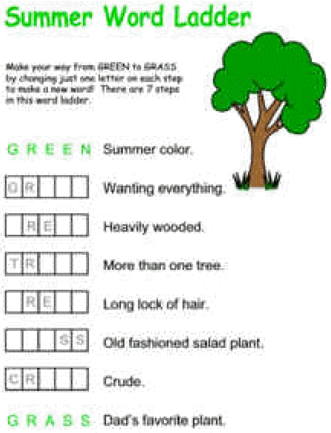
□□□□ Little pieces.

□□□□ Acronym: basic input/output system.

□□□□ Plural: ghost speech.

□□□□ Plural: cow speech.

MOON Earth's satellite.



Summer Word Ladder

Make your way from GREEN to GRASS by changing just one letter on each step to make a new word! There are 7 steps in this word ladder.

GREEN Summer color.

GR□□□ Wanting everything.

□RE□□ Heavily wooded.

TR□□□ More than one tree.

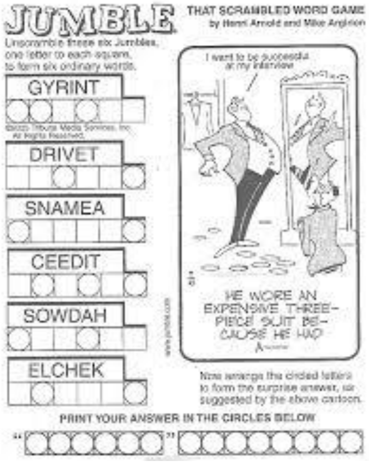
□RE□□ Long lock of hair.

□□□SS Old fashioned salad plant.

CR□□□ Crude.

GRASS Dad's favorite plant.

Jumbles are word games that are very common in newspapers. They are simply a set of rearranged words, with some letters in each word that form a final jumbled word. The classic Jumble games look like this =====>



JUMBLE THAT SCRAMBLED WORD GAME by Henri Arnold and Mike Argirion

Unscramble these six Jumbles, one letter to each square, to form six ordinary words.

GYRINT

©2020 Tribune Media Services, Inc. All Rights Reserved.

DRIVET

SNAMEA

CEEDIT

SOWDAH

ELCHEK

I want to be successful at my interview
HE WORE AN EXPENSIVE THREE-PIECE SUIT BE-CAUSE HE HAD

Now arrange the circled letters to form the surprise answer, as suggested by the above cartoon.

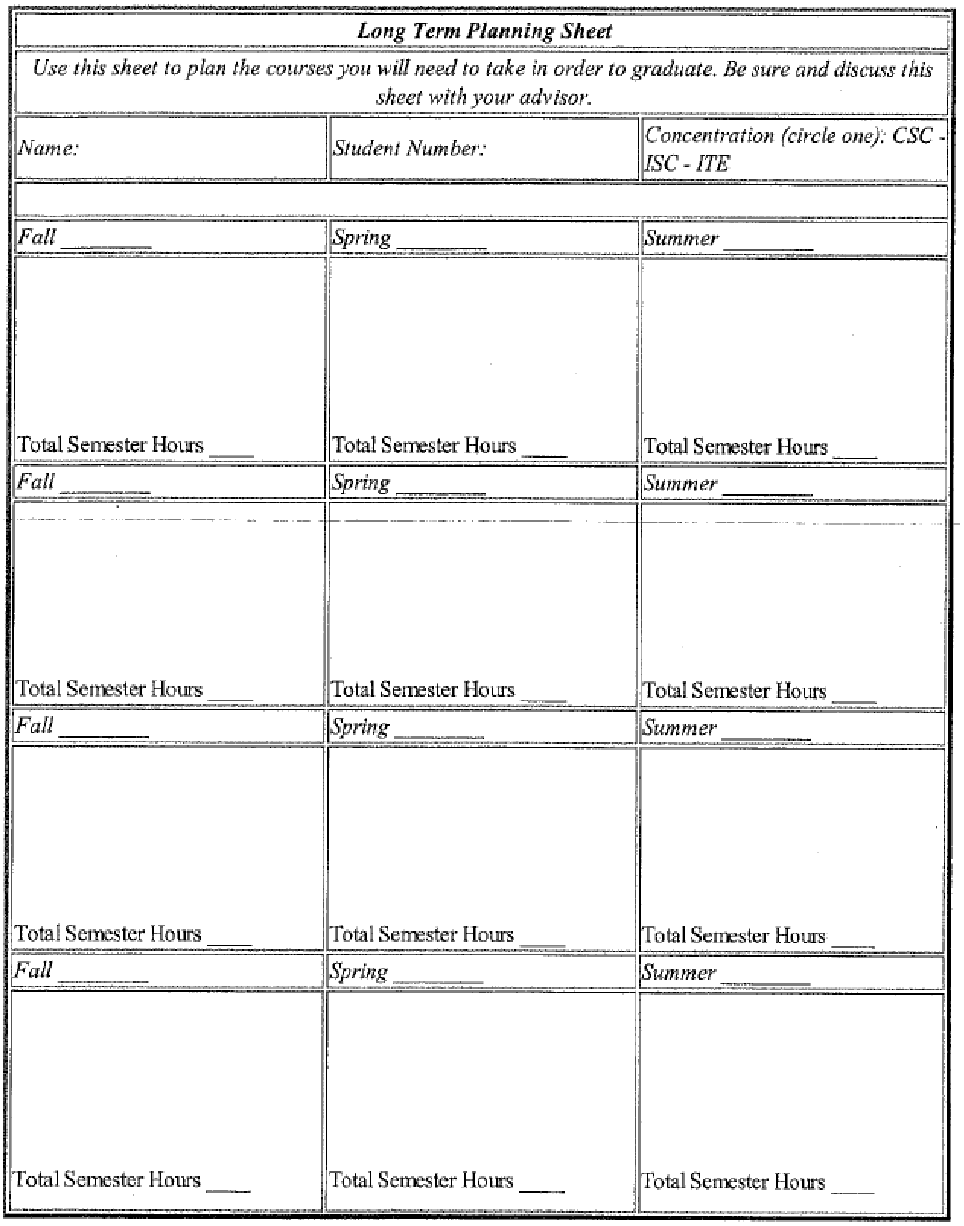
PRINT YOUR ANSWER IN THE CIRCLES BELOW

"○○○○○○" ○○○○○○○○

Application #4: USA SoC Co-Advisor

Students and their advisors often want to know what their academic program will look like. This application will assist a student in mapping out their Computer Science course curriculum, starting at any point in the program. The program will operate on a bulletin which assigns mandatory and elective courses for students to take and any courses that have already been taken. Courses also have pre-requisite and co-requisite constraints that must be met. The app will help the student create a long term plan based on their course choices.

The app will mimic the function of the "Long Term Planning Sheet" which is currently used for advising:



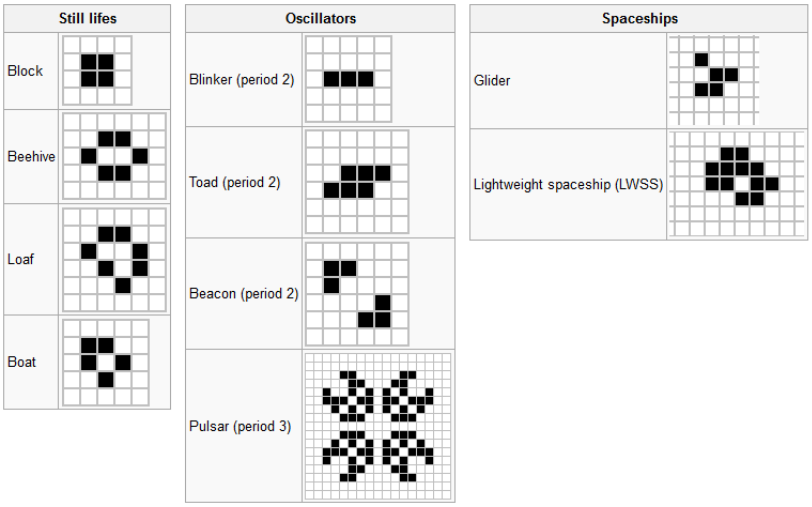
Long Term Planning Sheet		
Use this sheet to plan the courses you will need to take in order to graduate. Be sure and discuss this sheet with your advisor.		
Name:	*Student Number:*	*Concentration (circle one): CSC - ISC - ITE*
Fall ________	*Spring* ________	*Summer* ________
Total Semester Hours ____	Total Semester Hours ____	Total Semester Hours ____
Fall ________	*Spring* ________	*Summer* ________
Total Semester Hours ____	Total Semester Hours ____	Total Semester Hours ____
Fall ________	*Spring* ________	*Summer* ________
Total Semester Hours ____	Total Semester Hours ____	Total Semester Hours ____
Fall ________	*Spring* ________	*Summer* ________
Total Semester Hours ____	Total Semester Hours ____	Total Semester Hours ____

Application #5: Game of Life

This app will provide a simulation of Conway’s classic game of “Life”, which is a cellular automaton. As such it is a “zero-player” game that evolves based on its initial state. The simulation will provide options for user configuration and will provide a variety of statistics based on the patterns that are generated. The basic games consists of a grid of cells, each cell with a state of “live” or “dead”. Every cell interacts with its eight neighbours, which are the cells that are horizontally, vertically, or diagonally adjacent. At each step in time, the following transitions occur:

Any live cell with fewer than two live neighbours dies, as if caused by under-population.
Any live cell with two or three live neighbours lives on to the next generation.
Any live cell with more than three live neighbours dies, as if by overcrowding.
Any dead cell with exactly three live neighbours becomes a live cell, as if by reproduction.

During simulation, many different shapes are created and the application will provide statistics for these. For example, blocks, beehives, loafs, boats, blinkers, gliders, etc. are all terms associated with shapes that appear on the grid at any given time. The app will track numbers of such shapes and present a summary report to the user.

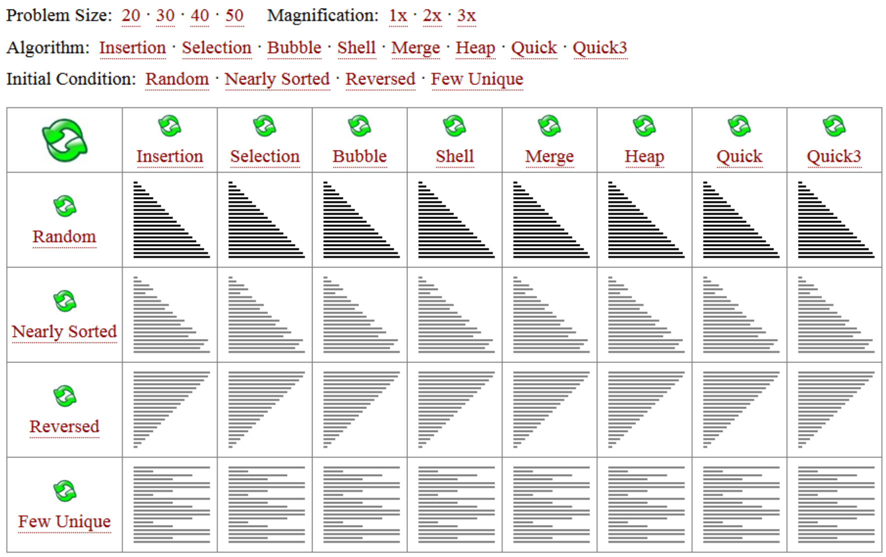


[http://en.wikipedia.org/wiki/Conway%27s_Game_of_Life]

Application #6: Algorithm Visualizer

This application will provide a visual representation of both data structures and sorting algorithms. Data structures may include abstractions of linear and non-linear structures such as linked lists, arrays, stacks, queues, hash tables, graphs, sets, and trees. Sorting algorithms may include procedures such as insertion, selection, bubble, shell, merge, heap, and quick sort.

As a rudimentary example of how sorting algorithms MIGHT be visualized, see:
http://www.sorting-algorithms.com/



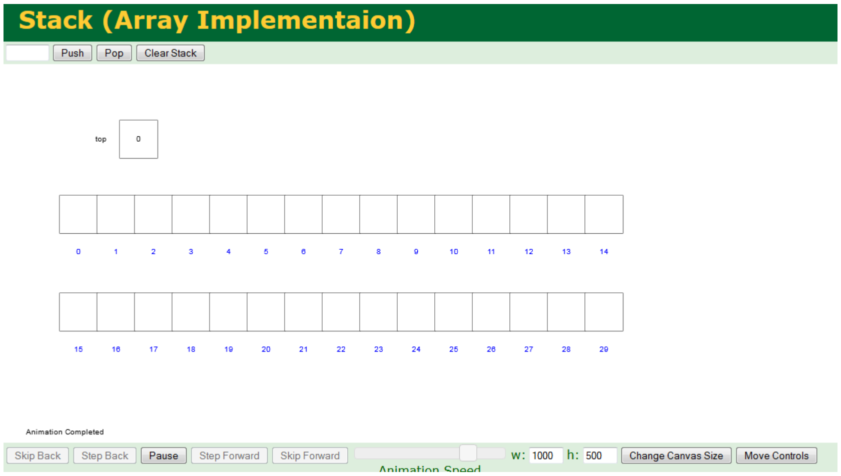
Problem Size: 20 · 30 · 40 · 50 Magnification: 1x · 2x · 3x

Algorithm: Insertion · Selection · Bubble · Shell · Merge · Heap · Quick · Quick3

Initial Condition: Random · Nearly Sorted · Reversed · Few Unique

	Insertion	Selection	Bubble	Shell	Merge	Heap	Quick	Quick3
Random								
Nearly Sorted								
Reversed								
Few Unique								

As a rudimentary example of how data structures may be visualized, see:
http://www.cs.usfca.edu/~galles/visualization/Algorithms.html



Stack (Array Implementaion)

Animation Completed

w: 1000 h: 500

Application #7: Bluetooth Attendee

This application would allow registered users (classroom attendees) to automatically register their attendance in class and is specifically envisioned to be an Android application. Since most all cellphones have a Bluetooth capability, the application would recognize when registered devices are in range and record their presence. Client devices would run the application which would broadcast and connect via Bluetooth to the master attendance application. For students or attendees without Bluetooth devices, the application would need to provide an alternate means to record their attendance.

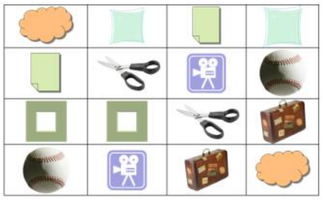
The application will provide record keeping of attendance, statistics, and other customer-desired information.

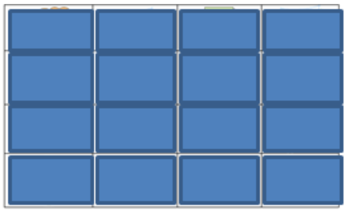
Application #8: Brain Games

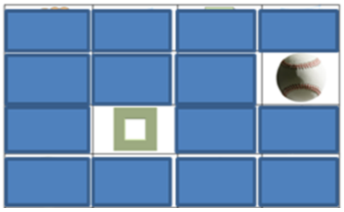
This application is a collection of two single-player "brain" games. One game must be "Memory Match" the other memory game will be "Master Mind".

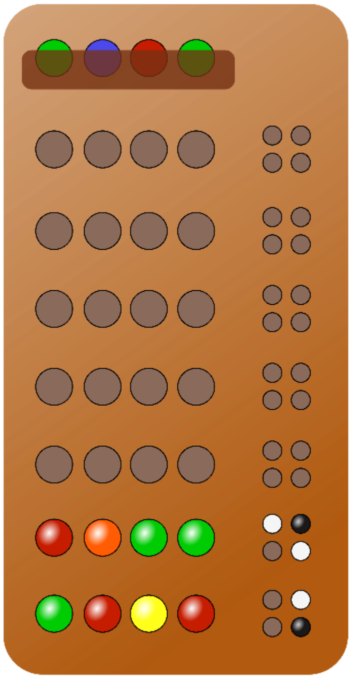
Memory Match defines a simple grid-based game where each cell has a picture and there are sets of 2, 3, or 4 matching pictures randomly distributed on the grid. Grid sizes may vary and be configured by the player, but a standard grid is 6 x 6, and grid sizes are even numbers.

For example, in a 2-set game there are **pairs** of matching pictures. When the game begins, all cells are covered. As the user clicks on a cell, the picture is revealed. For pairs, the player can click up to cells and reveal their contents, then the cells are covered after a certain time period. As pairs are identified, they remain uncovered. Games are typically timed.









Mastermind is a game that has a board, with a number of rows, each with large four holes and four small holes, coding pegs (each code peg) and "key" pegs. The goal is to determine the "code" that the computer has assigned; the code is composed of a code peg. A code peg represents a color; each peg can be assigned one of six colors, with duplicates allowed. The codebreaker is attempting determine both the colors of the code as well as the order within X turns; the number of turns is the same as the number of rows. After each test, feedback is given: For each peg that is correct in color and placement, a black key peg is placed in small hole; for each guess that is correct in color but no placement, a white is given. The order of the kep pegs is not tied to the order of the code. The game ends with a correct decoding or if the number of terms are exceed. For a more detailed explanation, please see: https://en.wikipedia.org/wiki/Mastermind_(board_game)

Application #9: Math Coach

The intent of this game is to present high-speed, high density practice in adding two single digits together to get the sum without counting on one’s fingers; with the answer becoming automatic through rote practice with the sight and sound of the basic addition table. It may be the case that when this task is mastered and addition with carrying is mastered, we may have a breakthrough in arithmetic ability for someone who otherwise would never be able to perform an arithmetic computation, and therefore not have confidence to make independent purchases in, for example, a grocery store.

The game should be exciting to a seven year old, along the lines of the Super Mario Brothers Nintendo game of jumping from platform to platform, but with fewer complicating factors. This oddly compelling game form seems to be a better learning platform than a “first person shooter” for a population whose response may be hard to predict, despite the popularity of war games.

Some possible design considerations include:

1. Selectable music
2. Adjustable speed
3. The addition problem should be presented in standard scholastic form (i.e., 2 + 3 = ? and

$$\begin{array}{r} 2 \\ \underline{+3} \\ ? \end{array}$$

4. The order of digits should be random (e.g., 2 + 3 = ? and 3 + 2 = ?)
5. Selection of digits should be random (e.g., 1 + 5, 7 + 2, etc.)
6. The range of the digits presented should be selectable (e.g., 0 to 4, 5 to 7, 0 to 9, etc.)
7. An avatar for the user should be available from a choice of approximately 10
8. The user’s response (correct/incorrect and latency to answer) to the addition problem presented should be the controlling factor in making progress in the game
9. Correct / incorrect scores, number of trials, and speed of play should be recorded along with a date / time stamp
10. The most effective end result / objective and the number of levels of the game remain to be determined
11. Play on a computer is probably the best first application, since many schools have computers; not all schools have tablets.

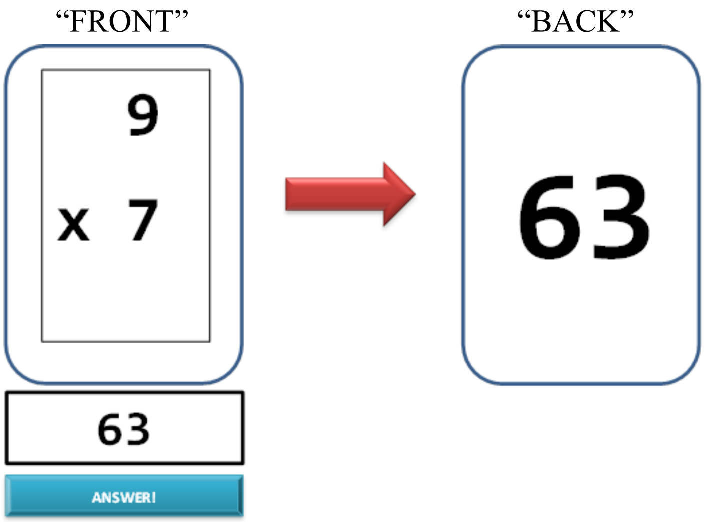
Application #10: Flash Card Mania

Students and teachers at all grade levels often have the desire to provide memory reinforcement exercises for various subjects. This application will provide the ability for a user to create a set of flash cards as a flash card "group" and then allow users to test themselves using the set. Card groups should be categorized into subjects which are called "libraries". Each "card" has a front side (which is an image or text that represents a problem to work out, a picture to recognize, etc.) and a back side (which is the answer, always in text form). The application will provide features such as timers to track study time, ability to associate hardness level with each card, ability to provide hints for cards, and statistic calculations for tests taken.

The application should provide the ability to create new libraries, import graphics, edit existing libraries, and view existing libraries. Ideally, libraries should be sharable by all users of the application.
At a minimum, 3 built in libraries should be provided with the application with at least 3 card groups per library:

1) **Chemistry Tutor Library:** these are card groups that aid in practice for learning names of chemical compounds. See http://misterguch.brinkster.net/pra_namingwkshts.html for a sample list of exercises for naming ionic and organic compounds
2) **Periodic Table Library:** these are card groups that aid in recognition of symbols in the periodic table and other information associated with an element (# of protons, # of neutrons, etc.)
3) **Times Table:** these are card groups that aid in memorizing the multiplication times table

A conceptual notion of what a flash card would like for a times table problem would be:



Application #11: StockSeer – a Stock Market Investment Fantasy League

StockSeer is a fantasy game that allows aspiring investors to learn how to trade financial stocks. The game supports trading activities and simulates a broker who pretends to execute trading orders. This tool is for an investor who does not have too much time to do a thorough research on all aspects of a particular company, neither do they have understanding and mastery over financial number crunching. The application should leverage freely available financial information and databases, in real time, to help make the game realistic for the would-be investor.

Application #12: E-Waiter

The goal for this project is to introduce automation in privately-owned restaurants, that is, small- to medium-sized establishments. Typical problems restaurant personnel are facing include:

- Coordination of their work activities
- Anticipating and handling periods of low/high patron traffic
- Recognizing trends early enough to take advantage of bestsellers or abandon the flops
- Lowering operating costs, and increasing efficiency/productivity and profits

Many restaurants are still operated using pen and paper methods, with little or no automation. Patrons enter the facility to be greeted by a host, who often times has a “dry erase” diagram of the tables, maintained on a blackboard. The host can see the status of the tables based on whether or not they or someone else physically updates the diagram. Once seated a waiter tends to the costumers by jotting down the orders onto a piece of carbon paper and delivers it to the kitchen for proper food preparation. The waiter then has to periodically check back to find out when the meal is ready. When the food is done, the piece of carbon paper is saved for proper record keeping by the management. This “old fashion” system works but yields a large amount of tab receipts, wastes a lot of time and is simply out-of-date. In old fashion systems, waiters have to carry pads around to take orders, always have a working pen and be sure to keep each bill organized and “synchronized” with the proper table.

The application is specifically Android focused and would involve target functionality for hosts in assigning patrons to tables and waiters to tables.

Application #13: Video Poker/Jacks or Better

Implement a simulation of a popular casino game called video poker. The poker game will follow the "Jacks or Better" scoring approach. The card deck contains 52 cards, 13 of each suit. At the beginning of the game, the deck is shuffled. You need to devise a fair method for shuffling. (It does not have to be efficient.)

A player begins play by entering their name and a beginning amount of credits. A player can continue to play until they get tired and want to quit or until they run out of credits. The player places a bet of 1-5 credits and then tells the system to "deal" a hand. Then the top five cards of the deck are presented to the player as their "hand". The player can reject none, some or all of the cards from their hand. The rejected cards are replaced from the top of the deck. Now the hand is scored.

Your program should pronounce it to be one of the following:

- Pairs of Jacks or Better - All high pairs (Jack-Jack, Queen-Queen, King-King, Ace-Ace).
- Two Pairs - Two cards of one kind and another two cards of one kind (for example, 3-3 and 6-6).
- Three of a Kind - three cards of one kind (same rank).
- Straight - Five cards of sequential rank (7-8-9-10-Jack).
- Flush - Five cards of the same suit.
- Full House - A combination of Three-of-a-Kind and a Pair.
- Four-of-a-Kind - Four cards of one kind (same rank).
- Straight Flush - Straight of one suit.
- Royal Flush - Ace-High straight of one suit.

The player can wager from 1 to 5 credits per game. Once a wager is made, the credits are deducted from the player's pot. If the player wins, their payout will be based on the following payout chart.

Hand	Odds	1 Credit	2 Credits	3 Credits	4 Credits	5 Credits
Pair of Jacks or Better	1.36 : 1	1	2	3	4	5
Two pairs	20.0 : 1	2	4	6	8	10
Three of a Kind	46.3 : 1	3	6	9	12	15
Straight	254 : 1	4	8	12	16	20
Flush	508 : 1	6	12	18	24	30
Full House	693 : 1	9	18	27	36	45
Four of a Kind	4,164 : 1	25	50	75	100	125
Straight Flush	72,192 : 1	50	100	150	200	250
Royal Flush	649,739 : 1	250	500	750	1000	4000

Application #14: RAILSYS

The purpose of a railroad is to carry car loads of products from one industry to another in freight cars and people from station to station in passenger cars. Railroad cars are assembled into trains. Each train follows a designated *route* and stops at each location on the route for some amount of time, either to allow passengers to get on or off the train, or to add (pick up) some freight cars or leave (drop) some freight cars.

You have been approached by the operations department of a major railroad to develop a train scheduling system to assist railroad personnel in specifying the arrival and departure times for each location along a train's route. The scheduling system will consist of a set of trains, a set of routes, a set of locations, and a set of schedules. The user will select a specific train and specify its starting time. The system will prompt the user for the length of time (possibly zero) that the train should remain stopped at each location on the route. Based on the train's average speed, the system will then calculate the time of arrival and the time of departure for each location on the route, and save that information in a schedule for the train.

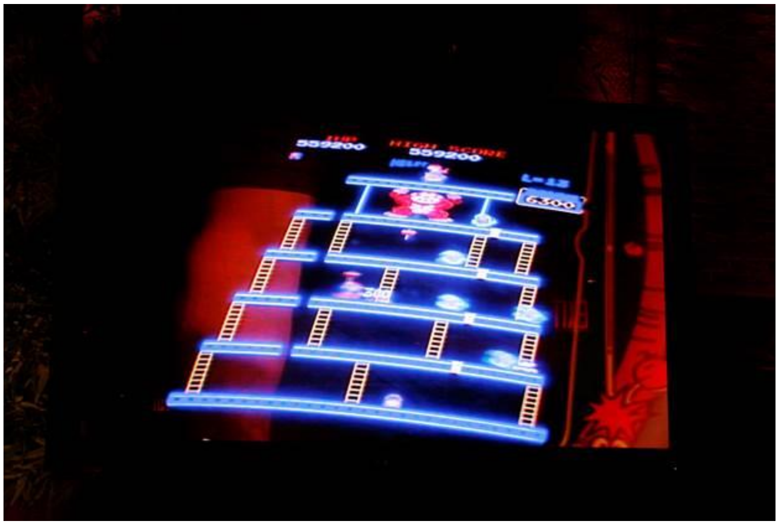
Application #15: LIBSYS

A system is needed to administer a lending library. Membership is required for borrowing books, but not for reading them in the library premises. Books may be ordered and can be obtained from associated libraries. Overdue books incur fines. Various management reports are required.

Librarians will interact with this system for library patrons. Not all library books are available for borrowing—some can only be read in the library. Loans can be renewed only once. Members can reserve books—but the reserved book is only held for 2 weeks when it is returned. When the reserved book is returned the member is notified by email (preferred) or post. Non-members may not reserve books. Membership expires after 3-years; however, there is no membership fee—renewal serves to update a patron’s membership information. The system should support basic functions such as borrowing a book, returning a book, renewing a loan, reserving a book, searching the catalog, and managing memberships.

Application #16: Angry Donkey Kong

10 years ago Mario (a crack commando) was sent to an underground prison for a crime he did not commit. Can you create a game that takes an old version of Donkey Kong and helps Mario find a way to battle his way to freedom and save the girl using guns, cannons and volatile barrels? It is time for explosive barrel roles!!! If you have graphical skills, you are Android savvy and you like things that blow-up, then this project is for you. I love it when a plan comes together.



Application #17: Burn It Up!

Create a calorie counter and exercise tracking application that aims to help you lose weight the old fashioned way: by expending more calories than you take in. The application will require a file of foods with associated calories that will allow a user to quickly log the calories they take in. For workout routines, associated caloric outtake algorithms and data will be required to provide the “minus”.

Application #18: Teacher Assistant

Teacher Assistant will help grade school teachers manage classroom activities and work. The application should allow an instructor to create seating charts, record attendance, log behavior, and possibly track grades for a classroom of students.

Application #19: Numble

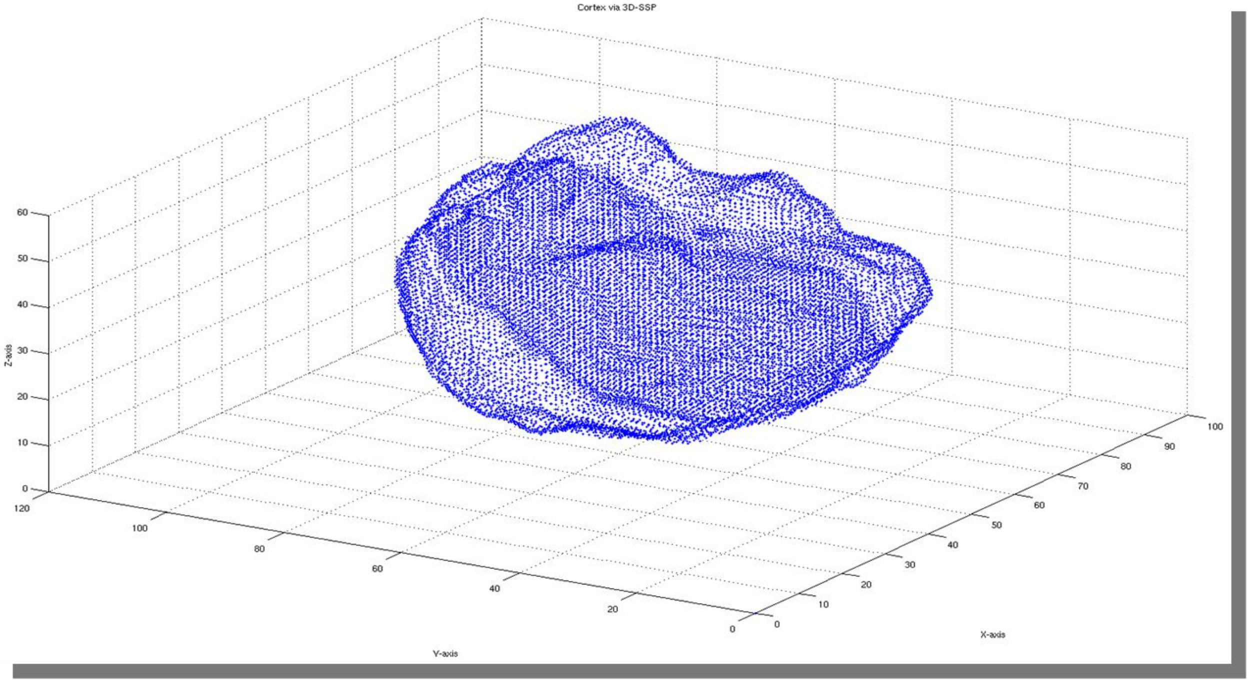
Numble is a mathematics based game that is similar to Scrabble. Instead of creating words from letters, the game will create equations from numbers and math operations on a game board.

Application #20: Neuro Assist

This application will display a large set of uncharacterized EEG data for help in determining neurological disorders such as epileptic seizures. Electroencephalography (EEG) is the recording of electrical activity along the scalp. EEG measures voltage fluctuations resulting from ionic current flows within the neurons of the brain. In some cases, EEG is recorded during analysis periods for epilepsy patients, but has not been characterized by a neurologist to determine whether a seizure has occurred based on the data. The application would display a time window of EEG from a recorded session and allow a neurologist to annotate the session for seizure onset and time.

Application #21: Brain Visualizer

This application will display a 3D display composed of point cloud data that has been extracted from PET images of the brain and predict if the brain is representative a normal brain, a brain with dementia, or uncertain. In addition, two different PET scans (potentially of different brains) will be display side-by-side, allowing for visual comparisons.


Cortex via 3D-SSP
Z-axis
60
50
40
30
20
10
0
120
100
80
60
40
20
0
Y-axis
0
10
20
30
40
50
60
70
80
90
100
X-axis