team name	project name	project name	description	aproach	Risk Management
70/30	Lady Lazarus	Lady Lazarus	Lady Lazarus is a 2-D platformer inspired by Sylvia Plath's poem of the same name. Unlike the traditonal platformer where the player avoids death; in Lady Lazarus the player must die to complete the level. After death the player is revived to a spawn point with new powers dependent on how the player died. For example if killed by fire the player will come back with the ability to run faster. The player will have to use these abilites to solve puzzles to finish the level.	Languages ------	There's no significa
Autocomplete	Intro to tower defe	Intro to tower defe	We are going to design and build a tower defense style game. Our basic plan is to have a tower defense game where a single enemy will follow a set path from start to finish. We would have one tower that has a basic attack to defend the exit of the map. Once we have the basic game implemented we will move forward with more advanced features such as multiple enemies, multiple towers, free-open maps, etc. We hope to have simple graphics in the end that will add to the gameplay.	<p dir="ltr" style="margi	<span style="font-s
Ballmer Peak	Pixel	Pixel	Our project will consist of a blank window containing pixel emitters and pixel sinks. Pixels will eminate from the emitters and seek out sinks. Pixels that reach their sink will stick and become part of the sink. Pathfinding will be accomplished via evolution; pixels that reach a sink will influence how newly-created pixels behave. Given various difficulties that may arise from c++ and opengl, we only plan to implement the above functionalities. However, if we finish the project completely and ahead of schedule, we will continue implementing functionalities from the list below. 1. Create different species of pixels. Different species of pixel would be visually distinctive and possibly implement distinct pathfinding algorithms. Other species-specific behaviors such as killing non-same species have also been considered. 2. Implement a UI that allows for the placement and removal of pixel emitters. 3. Add complexity to make for a more visually interesting program. This may include implementing more intricate pathfinding algorithms, causing pixels to leave colored trails, adding various environmental obstacles/hazards, optimizing existing code to make higher populations of pixels a plausibility, or	The entire program will be	No significant risks.

Chromosominator	GPMA - Genetic Pr	GPMA - Genetic Pr	The project goal is to produce an application that models a given data set using a linear regression fitness function to produce a forecast of the data using n elements of a discrete time series set { t 1 ,t 2 ,...t n } where the training data for producing the population is a vector x of k elements with element k = t i+1 (the desired output) and the remaining elements are the set { t i , t i-1 ,...,t i-k-1 }. The test cases will be divided into training and validation sets (of equal size) to be combined into a best fitness individual. A percentage of the test cases will be reserved for an out-of-sample run to assess the generalization abilities of the solution. The logic of the modeling algorithm will use recursion to randomly build a population of expressions (individuals), get their output and calculate their fitness. The individuals will be selected via tournament and submitted to a process of mutation or crossover. Crossover will randomly select sub-tree structures from two parents, combine them into a new individual and add the new offspring to the population for subsequent runs (the worst fit individual will be removed from the population). Mutation will randomly select sub-trees of a chosen individual and then randomly alter either an operator or terminal of the sub-tree. The process will continue until an acceptably small error is discovered or a maximum number of runs have been conducted. The application will, at a minimum, run from the console and output the results to the screen, along with an output of the solutions of all test cases of the best individual to a delimited text file. The operator set for the individuals will be limited to: addition(+), subtraction(-), multiplication(*), protected division(/), sine(sin), and cosine(cos). The operands (terminal set) will simply be the input vector x , excluding the last element which contains the desired output. No random constants will be included. If time allows additional features will be included. The desired optional features include: 1. A GUI to allow selection of fitness functions, max runs, acceptable error, and hopefully some basic charting to display the training, validation, and out-of-sample data for graphical analysis. 2. A wavelet decomposition library to optionally decompose the inputs to wave coefficients to assess the effectiveness in increasing the applications ability to model the data. 3. A normalization function to explore how normalizing the data aids in the accuracy of the model.	Use c++ and maybe thw	No forseen risks
CIS330	ChatRoom	ChatRoom	In our project ,we are going to design a Sever-Client System base on the C language. We want to achieve that multiple user can connect on the Server at the same time to chatting with each other. Basically, we will build a Server first which will be able to accept multiple socket (client) to connect it. We are thinking to use multiple thre strategy to achieve our goal. To be more specific, we want to bulid this Chatting System be more like a "chatting Room" which will allow many people chat with each other at the same time, not just one person to one person. Also, we will have a Encryption to protect the private of our chatting, which means that only people who have the username and password can login to our Server, and chat with other. To make our System more funny, we are still discussing that we can add sticker, e moticon to our Client, and even video chat.	We are going to achieve c	No specific Risk mar
Dobis PR	Band	Band	We plan to develop an application that allows the user to choose a musical instrument, and then play that instrument using the keyboard as a controller. The program will include various menus for the user to select options such as instrument selection and stage background. While playing the instruments, some sort of graphical effect will happen to indicate what note (or drum) is being played. We will also include various backing tracks for aspiring music stars to practice soloing over.	We plan on implementing	We forsee no signifi

Flying Piglets	Oink	Oink	We'll be implementing a chat application in C++. After some online research, we are considering QT for creating our GUI and Boost for constructing our server/client. There will be different stages to developing our software, but we wish to include features such as an encrypted user database, a file transfer feature, a chat room, private chat, a friend's list, and some smilies. Depending on how secure we want our application to be, we might utilize a library for our security system. It's a similar case for our user database; we might implement a simple database through a text file or use something more complex like MySQL or FileMaker.	<p style="margin-bottom	<p class="MsoNorm
Jocknerds	Court Queue	Court Queue	Court Queue will facilitate the organization of disorganized pick up basketball games. It uses a tracking mechanism to count the number of wins each player get then stores it in a log file. As players show up to the court, Court Queue will take in a name for each player, create an object with a player profile, and store them chronologically in a queue or priority queue. The first five players that sign up will be assigned to a “red team” and the next five, the “blue team”. We will be implementing the GTK+ library for our GUI. After each game, the game length and win/loss will be recorded into the log file for each player. Whomever wins will stay in the queue by default and the next five players that signed up will be sorted into the blue to chronologically. This is a basic overview of how the program will work. The reason why we chose this program was to fulfill a need of organization. Pick up games at the UO rec center are often chaotic and can end in fights over who plays next or what not. Court Queue will hopefully be a solution to a few of these issues.	<p dir="ltr" style="margi	<p dir="ltr" style="r
Jon Zeller?	Robot Dragon 2 (v	Robot Dragon 2 (w	The project is a two-dimensional vertical shoot-'em-up game. The player will control a spaceship which must dodge large amounts of enemy bullets coming from the top of the screen. Much of the code will be recycled from an older project which was made with a team, but for which I wrote 90%+ of the code and all of the music; components I did not make will be erased. The game may involve wireframe vector-based 3D graphics similar in appearance to those used in old 3D arcade games; this will be entirely original to this version and used for visual effects only. To some extent, I will also refactor the code, because the old code was rushed. Vector graphics will be implemented through the use of matrices, which will be implemented in some way involving 2-dimensional arrays of floating points.	I intend to use C++, since	There should be no s
Peking Duck	Gomoku	Gomoku	An error occurred processing this row.		
Shootin for a C++	The Floor is Lava!	The Floor is Lava!	Our application will be a side-scrolling 2-D game. The main character will have to navigate a constantly size-variable tunnel whose walls are made of lava or some other dangerous substance. Crashing into the wall ends the game (or level, if we choose to implement levels one can complete- as of now we intend this to be a one-level game in which one achieves the highest distance possible before crashing). We may choose to include powerups or other bonuses/hazards as we see fit.	We plan to work in C++, p	We are reasonably c
std::cout << ”team nam	Chat Application	Chat Application	For our project we want to create a chat application with a clean, usable GUI. Users will be able to send messages to one another using an interface like that of many common chat room or instant messaging applications.	<!--<xml> <o:OfficeDoc	<!--<xml> <o:Offic
Team Jacob	Survival Game	Survival Game	We plan to use the SDL2.0 development library to develop a 2D game. The game is a survival based high score game. The player will need to avoid taking fatal damage from incoming obstacles and defeat enemies in pursuit of points. The game will consist of a menu a few different level layouts in survival mode.	We are planning to develo	No significant risks.
The Assemblers	Mr.Movl's Missing	Mr.Movl's Missing I	Mr.Movl's Missing Memory will be a 2D side-scroller/platformer game implemented using the Angel2D framework. The main protagonist, the honorable Mr. Movl, unfortunately has lost his memory and must set out on a quest to locate his missing memory blocks.	We will be implementing I	The biggest obstacle

The Bravest Warriors	Catbugs Last Stan	Catbugs Last Stan	We are building a 2d, side scrolling platforming/beat-em-up style game based on the Bravest Warriors web series (https://www.youtube.com/user/BravestWarriors). The game will be built in the Angel2d engine. The game will feature up to four player simultaneous local co-op multiplayer, with multiple playable characters that will feature various gameplay styles. The game will feature pixel art with characters and environments from the series.	The game will be written	One major risk is an
the Team with no name	Multiplayer Asterc	Multiplayer Asteroi	Project Description: We want to make a network/multiplayer version of the classic Asteroids game. We will start by getting the classic asteroids game to run over a network, and if we have extra time we will add additional features such as upgrades for the ship/weapons, more detailed physics (elastic collisions between asteroids), and hopefully the multiplayer aspect.	<p dir="ltr" style="margi	<p dir="ltr" style="'
Undecided	<Unanswered>	<Unanswered>	<Unanswered>	<Unanswered>	<Unanswered>
UO-OCR	Optical Character	Optical Character I	We wish to create a basic Optical Character Recognition (OCR) program in C++. First we will need to do some pre-processing to isolate letters. Other pre-processing steps may be applied as well. These include de-skewing, decolorizing, normalizing the contrast, normalizing the aspect ratio, and line detection. Once pre-processing is complete we will pass the pre-processed characters into our primary processor. The identity of each character will then be determined using a machine learning algorithm. We will most likely use the k-nearest neighbors to do OCR; however, if we find an algorithm which is better suited to our project, we are open to changing this component. Once characters have been identified, we will optionally begin post-processing. This can consist of using word and character frequency to correct words, or going further and analyzing n-grams. We could also format the output in a more readable fashion. The project can be easily scaled to include more features. The most basic version would simply take in human processed characters and output each character as text. Additional features will be added as we go to make the program more usable and robust.	Languages All m	The project runs the
What Would GCCesus Do	Chats Diddy	Chats Diddy	We are going to create a chat application that is run through the Unix command line. There will be commands for logging in, seeing which of your friends are already online, adding and deleting friends, logging out, etc. It will be mostly command-line based, except when you use the command to start a chat with an online friend, it will open a small GUI chatbox. Closing this chatbox will end that chat. We may also look into the ability to send files between users.	We are going to write this	No significant risks,