The class project for 15-381 allows students to explore the application of AI techniques learned during the class to a specific problem domain. The goal of these projects is to let you think about a more open-ended problem than what is considered in the homeworks or exams, and one which may potentially lead to interesting research directions. However, note that for a successful project, you need to apply techniques from the class to solve the problem; very low scores will be given to any project that largely uses techniques not covered in the class to attack the problem, even if the resulting methodology is reasonable.

Please form a group of 3 students to work on these projects. Contact the instructors for another group size: no 1-student group will be accepted, 2 or 4-student groups will need to be justified; and no more-than-4 students in a group will be permitted.

**Project requirements:**

- By **Thursday, November 19**, please submit up a short (500 words) description of the project, with: (i) the names of the students in the project; (ii) the project selected; (iii) any details you know already about at the level of the concrete instantiation of the project, e.g., the puzzle you will be solving; and (iv) the techniques learned in the course that you plan on using in the project.

- On **Tuesday, December 8**, you will present, in class, a short presentation of your project (very few minutes (TBD) per project and student, which will be strictly enforced). The presentation should include at most three slides describing the project and key methods/results.

- A final report for the project of no more than **4 pages** (single spaced, 11 pt font, 1 inch margins) is due on **Thursday, December 17**. The project report should highlight the problem formulation, the main techniques that were used for the project, and a summary of the results.

**Project topics:**

Below is the list of the five topics for projects. **You must choose a project from one of the topics in this list.**

**Search:** For this problem you will apply search algorithms to solve a puzzle game. Select a puzzle (we specifically recommend the 2048 game puzzle or lunar lockout, but any other similar complex puzzle is sufficient with instructor approval) and design and implement a search algorithm to solve the puzzle.

**Planning:** You are to consider a space with obstacles and a robot (like a roomba) that needs to move in the environment. You can plan for the robot to cover the complete space or to be given the definition
of a goal (e.g., found an object) and stop when the goal is found. You will define the sensor(s) that the robot has and you will need to assume that there is uncertainty in the sensors and in the motion. You can assume just one bump sensor, or other sensors that will return the detection of some objects with uncertainty. You can also assume that the robot is given, learns, or does not have, the map of the environment, i.e., the position of the obstacles. You will need to describe your assumptions and build an algorithm that operates in this framework. Assume that the environment itself is static, i.e., that the obstacles do not move.

**Vision:** Build an image classifier from some “standard” dataset of images on the web. We recommend the Tiny ImageNet data set (http://cs231n.stanford.edu/tiny-imagenet-200.zip) the SUN database (http://groups.csail.mit.edu/vision/SUN/), or the CIFAR data set (http://www.cs.toronto.edu/~kriz/cifar.html) – you will almost certainly only want to consider some subset of the images, even for these relatively “small” data sets. You can use any of the vision techniques highlighted in class (e.g., using the OpenCV library, linear machine learning algorithms, or deep neural networks) to build your classifier. You can solve a problem of finding the images that have some feature, e.g., “outdoor images,” or you can classify images, or generate text captions for images.

**Machine Learning:** In this project you will design a machine learning problem and then apply algorithms learned in class to make predictions on the problem. In particular, you will need to look at some data sets released on public data from Pittsburgh, available at: https://data.wprdc.org/organization/city-of-pittsburgh We recommend the 311 data, the Police Incident Blotter, or the Building Energy Consumption data sets as potentially of interest here (many of the data sets on that page are simple geographic descriptions that would be hard to form any machine learning problem from). After looking at the data, you will need to construct a meaningful machine learning prediction task: this part is up to you, but you need to design a task that solves some reasonable problem. Build a training, validation, and test set from the data, and apply algorithms learned in class to the data set. Note that the problem setup, motivation, and experimental methodology will count toward your project grade more than any final absolute performance measure (accuracy, etc), on the problem you develop.

**Games:** Choose the game connect4, block battle, or warlightt (not any of the others) from the website http://theaigames.com/ and develop an algorithm for playing this game using the techniques highlighted in class. You can play the game against yourself or with or with opponents from the online platform. Your project should highlight the value (in terms of performance improvement) of incorporating different techniques into your code.

**Matching:** Consider any complex “matching” problem (e.g., dating, kidney exchange, buying a house, selecting a health insurance plan) with a large number of constraints. Formulate the problem as a CSP and apply different techniques to solve it. You can create a simulation of the problem or try to get real data.