1-Page PDF Proposals Due: Thursday, November 21st, 11:59pm
Individual Design Checks: Sunday, November 24th - Monday, November 25th

**On-Time Handin:** Saturday, December 14th, 11:59pm*
*Please note that although the due date is 11:59pm, the SunLab will close at 10:00pm
**Late Handin:** Monday, December 16th, 5:00pm
You cannot use a late pass on this assignment.

Demos¹: cs0150_runSnazzyDemo  Indy
*Demos may not work well over ssh! Try FastX or the Sunlab.

To install: cs0150_install Indy
To hand in: cs0150_handinIndy <nameOfFolderWithYourCode>

**Table of Contents**

- Table of Contents
- **Staff**
- **Silly Premise & Intro**
- **What qualifies as an Independent Project?**
- **Who can do an Independent Project?**
- **Timeline**
- **What We Are Expecting**
- **TA Hours**
- **User Guide**
- **Final Project Handin**

---

¹ Note that some of these demos were done in an older Java graphics package (Swing), so elements may show up slightly differently if you try to replicate specific parts.
Silly Premise & Intro

“Stanley, I have a job for you. I need you to make us a Pacman machine for the office!”

Stanley looked into the void. In his heart there was a built up rage after all the years of being ordered around by Michael. He felt like he didn’t even have anything to work for — pretzel day was still 358 days away.

“No, Michael” he said, as always. But this time he meant it.

Stanley wanted to do something that was unique, and stretched beyond what was expected in the often mundane world of office paper: a project that demonstrates strong independence, and illustrates an exceptional command of object-oriented programming. Help Stanley quit his job at Dunder-Mifflin to make the project of his dreams!

What qualifies as an Independent Project?

To be approved, your potential project must meet the following criteria:

1. **Length**: LiteBrite does not qualify as a final project! Independents must be at least as long as an average final project, which, by a rough estimate, is about 2000 lines of code.

2. **Data Structures**: The program must contain advanced data structures, e.g. multidimensional matrix (3D or higher), linked lists, stacks, queues, etc. Another possibility is to explore and employ a data structure which has not been used in previous project, e.g. HashMaps or Trees.
3. **Algorithms:** You may not do a project which performs a very simple task even if it takes a lot of code. For example, a program which draws 63 different columns (even if each were a different color!) would not meet the algorithm standards. Thus, your project should utilize a reasonably complex algorithm (at least as difficult as that in Tetris). When designing your project, be sure to take full advantage of polymorphism. It is also important to remember that there are algorithms that are short, but are hard to implement. Make sure you choose your algorithm wisely!

4. **Originality:** Finally, the program must be different enough from the standard CS15 projects to warrant its independent status. If you insist on expanding one of the regular class final projects, be prepared to explain and defend the reasons why your variation cannot just be implemented as extra credit for the regular project.

5. **Language:** Your whole project must be written in Java!

We want you to realize exactly what you’re getting yourself into. You will have a limited amount of time to do a project that hasn’t been done before, so be prepared to work hard. However, you will have a dedicated staff of 4 passionate, creative TAs who will do their best to make sure your project is a success.

**Who can do an Independent Project?**

You must have a firm grasp of the material covered in the course to do an Independent Project. In any case, you must be able to work on your own with relatively little supervision. Before you begin to even write pseudocode for any program, you obviously must first know what it is that you are trying to accomplish. The other students will have this spelled out for them in their handouts; you must devise this yourself. Remember, you will not be getting explicit program specs, design hints, or demos. You must create all this for yourself. Actually writing the program should take about the same amount of time as writing one of the standard projects. The difference is that you will be spending MUCH more time on the design and development end of things. Since an independent project provides you with greater freedom, it is necessary that you be someone who works well with that freedom and lack of structure.

We have a system for screening Indy applicants. In order to qualify for Indy, you have to have shown a strong track record in the previous projects. This means we’re looking for students who have consistently done bells and whistles and performed pretty well on the other projects. This is not to discourage people from taking on the Indy challenge, but rather to make sure that Indy is the right choice for you. If you have any questions about this policy, send an email to the Indy TAs.
Timeline

Here is a roadmap of what the project looks like:

- **Thursday, November 21st**: Attend a help session for a backup final project.

- **Thursday, November 21st by 11:59pm**: By this time, all students interested in doing Indy must send a 1 page PDF proposal to the indy staff (jwang73@cs.brown.edu, wbuerger@cs.brown.edu, zbeckman@cs.brown.edu, and zmothner@cs.brown.edu). This proposal should give a short description of your project, clearly specifying what algorithms and data structures you think you might use. No need to go too in-depth; this is meant for us to get a general sense of what you want to do.

- **Friday, November 22nd**: An Indy TA will reach out to you EOD to schedule a design check.

- **Saturday, November 23rd by 5 pm**: Indy mini assignment due as a PDF, sent to the email of the TA who is running your design check.

- **Sunday, November 24th - Monday, 25th**: We will be holding individual Design Checks, where you will have the chance to defend your answers to the design questions. We will let you know if your project has been accepted or denied by end of day on the day of your design check. If your project is accepted, you will receive a personal Indy TA and you must set up a progress check with them at this time.

- **Friday, December 6th - Saturday, December 7th**: Progress Check with your Indy TA. Note that the progress check is a mandatory meeting with a TA to demo your progress so far. This is meant to make sure that you're on track, since there's so much freedom with the assignment. In your first design check, you will outline the requirements for your progress check. This will be part of your final grade.

- **Tuesday, December 10th**: By this time, you will be assigned a different Indy TA who will grade your project. They will reach out to you for scheduling a final demo of your project.

- **Saturday, December 14th at 11:59pm**: On time hand in. Your code **must** be able to compile in the command line on the CS department machines.

- **Monday, December 16th at 5pm**: Late handin. You must have completed a Final Demo with your Indy TA grader by this time.
Once again, although you shouldn’t be pessimistic, your design check could be denied, so be realistic enough to go to the help session of one of the other final projects. After the initial extra work for your project design check, your life will be pretty much the same as any other CS15 student except with more work and more fun (in other words, you are very similar to everyone else, but different 🌟).

What We Are Expecting

Just to be clear, we want you to be successful. If you prove that you have thought hard about your design, then we will accept your project. We do not have a limit on the amount of projects we can accept.

To that end, here is a list of things you can do to show us you have thought your design through:

- Make sure your code can compile in the command line. You are welcome to and encouraged to use eclipse, but in order for us to grade your project you must make sure you test command line compiling. If your code cannot compile in the command line, you will receive a significant deduction.
- If you wish to use images in your project, you must follow the directions of this guide or you will also receive a deduction.
- Bring thorough pseudocode with you. This does not mean you need to have pseudocode for every single method, but you should have the core functionality planned out.
- Set aside a reasonable amount of time to plan out your project. It will be obvious if you attempt to complete your design questions in the hour before your check in.
- If necessary, bring other materials that could help supplement your thoughts. For example, if you plan to make Scrabble, then perhaps bringing in a real Scrabble board would be helpful!
- Be open to change. You will probably change your design a few times before you start coding. And even when you start coding, your design can still change.

As always, please reach out to us if you need any clarification on what we are expecting.

TA Hours

TA Hours will be by appointment only. Please do not sign up for normal TA hours. They are reserved for students doing other final projects. Prior to being assigned an Indy TA, you may reach out to the entire Indy Staff. Once you get your Indy TA, please email them individually to
set up an appointment. Our emails are jwang73@cs.brown.edu, wbuerger@cs.brown.edu, zbeckman@cs.brown.edu, and zmothner@cs.brown.edu

User Guide

In addition to your final project, you must hand in a final User Guide. This document should describe your user interface completely, as well as provide sample input/output and example screens. It should highlight your program's capabilities (and limitations). The User Guide is due at the same time as the rest of your project. This should be emailed to the Indy TAs and included in the folder you hand in. This is different from the README, which you must also have. The README pertains only to your code. This includes code design, architecture choices, known bugs, broad design choices for your algorithms, etc.

Final Project Handin

You must hand in your program using cs0150_handinIndy <nameOfFolderWithYourCode>.