

Spring 2017

# CS349 User Interfaces

- [Description](#)
- [Staff](#)
- [Office \(Lab\) Hours](#)
- [Grading](#)
- [Assignments](#)
- [Git Repository Setup](#)
- [Policies](#)

## Description

This course teaches the principles of *constructing* user interfaces. In this course you will learn how to actually *implement* user interfaces. Some attention is paid to issues of design and usability, but CS 449 provides more complete treatment of these topics.

## Objectives

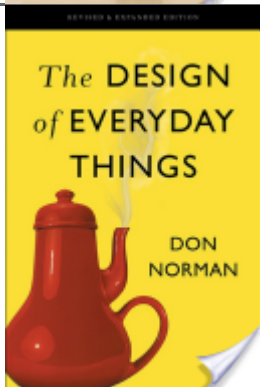
This course provides an introduction to contemporary user interfaces, including the basics of human-computer interaction, the user interface design/evaluation process, and the architectures within which user interfaces are developed. Students implement and evaluate portions of typical user interfaces in a series of programming assignments.

## Recommended Texts



*Building Interactive Systems*, 2010, by Dan R. Olsen Jr.

A copy is available on 3-hour loan at the DC library under call number UWD1512.



*The Design of Everyday Things*, 2013, by Don Norman.

A copy is available on 3-hour loan at the DC library under call number TS171.4 .N67 2013.

*Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Guidelines*, 2013, by Jeff Johnson

A copy is available on a 3-hour loan at the DC library under call number UWD1568.



---

## Announcements

Lectures begin Monday, May 1<sup>st</sup>. Please note that during the first two weeks of the term, office hours will happen at DC 3540 (Mon-Thu) or MC 4065 (Fri). After May 15, all office hours will be held at MC 3005 as informed.

We'll be using [Piazza](#) for all announcements, as well as questions and answers about the course material and assignments. Please create an account at the start of the term, using your real name and your Waterloo username; this will make it easier for course staff to identify you if required.

## Lectures

MWF 08:30 - 09:20 - MC 2034 (Section 002) - Gustavo Fortes Tondello

MWF 11:30 - 12:20 - MC 2038 (Section 001) - Gustavo Fortes Tondello

## Staff

(all email addresses are *userid@uwaterloo.ca*)

### Instructor

The instructor is responsible for the course as a whole: lectures, setting assignments and exams, developing marking schemes and resolving issues related to the course.

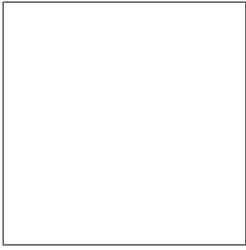


**Gustavo Fortes Tondello** (gfortestondello@)  
Office Hours: Fri 9:40-11:00, MC 3005, or by appointment

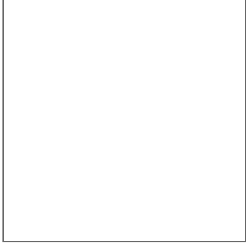
---

## Instructional Apprentices

Instructional apprentices are responsible for helping the instructor develop assignments and marking schemes, helping maintain marking consistency, resolving mark appeals, etc.



**Jingjie Zheng** (j49zheng@)  
Wed 10:00-11:00, MC 3005



**Terence Dickson** (tpdickso@)  
Wed 10:00-11:00, MC 3005

---

## Teaching Assistants

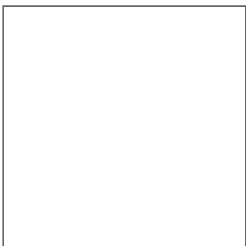
Teaching assistants are responsible for marking assignments, marking exams, and holding office hours.



**Dallas Fraser** (d6fraser@)  
Mon 10:00-11:00, MC 3005



**Jeremy Hartmann** (jeremy.hartmann@)  
Mon 10:00-11:00, MC 3005



**Qi Feng (Edmund) Liu** (qfliu@)  
Thu 2:00-3:00 pm, MC 3005

---

**Shaishav Siddhpuria** (spsiddhp@)  
Tue 10:00-11:00, MC 3005



---

**Will Callaghan** (wrcallag@)  
Tue 10:00-11:00, MC 3005



---

## Instructional Support Coordinator

The ISC is responsible for long-term support for the course, maintaining the marks spreadsheet, coordinating accommodations (illness, alternate exams, etc), screening for cheating and dealing with the results.



**Caroline Kierstead** (ctkierst@)  
MC 4009

---

## Office (Lab) Hours

Office hours will be held at the computing lab at MC 3005 whenever possible. If you wish, you can work on your assignments in the lab during these lab hours and at least one course staff will be there to aid you in case you have any question. Regular office/lab hours will occur weekly at MC 3005 at the following times (see also [Course Staff](#) details):

- Mondays 10:00-11:00 - Jeremy and Dallas
- Tuesdays 10:00-11:00 - Shaishav and Will
- Wednesdays 10:00-11:00 - Terence and Jingjie
- Thursdays 2:00-3:00 - Edmund
- Fridays 9:40-11:00 - Gustavo

Changes on these regular times might occur throughout the term according to the availability of the staff and the lab, and will be announced through [Piazza](#). Therefore, be sure to check Piazza regularly for announcements.

# Grading

There are three [assignments](#) (submitted in four dates) which are together worth 40%, a midterm worth 20% (see [schedule](#) for date and location), and a final exam worth 40%.

Individual assignments will not receive a numeric grade. Instead, students are required to submit a self-evaluation of their skills together with each assignment submission, which will be assessed by course staff together with the assignment. The final assignments grade will be an average of the proficiency acquired by the student on all the required skills. Please check the [Skill Tree](#) for additional information.

Final course grades will be calculated as follows:

```
exams = (.20 * midterm + .40 * final) / .60
normal = .60 * exams + .40 * assignments
if assignments < .50 or exams < .50
    grade = min(.47, exams, normal) * 100
else
    grade = normal * 100
```

**Note:** to pass the course you must pass the weighted average of the assignments and the weighted average of the exams.

# Assignments

Assignments are meant to provide meaningful, engaging experiences in constructing interfaces while giving you the opportunity to create applications you will want to share with others. There's lots of room for creativity in assignments and each will have a component for going above and beyond the basic assignment specification.

The assignments in this course require a significant amount of time. Do not underestimate the time it takes to code interactive applications.

All assignments are to be completed individually unless otherwise stated. You are allowed to discuss strategies and solutions with the course staff and other students; however, you must write your code independently. Therefore, even if you implement a similar solution than other student, we expect your code not to be identical.

**Plagiarism detection software is used to screen assignments in this course to verify that students aren't sharing code in their assignments.** We will report suspicious activity, and penalties for plagiarism/cheating are severe. Please read the [available information](#) about academic integrity very carefully.

Detailed assignment descriptions and due dates are posted in the [course schedule](#) when available.

There are three assignments in the course, plus an initial assignment (A0) that is meant to get you started with Git and Java.

- [A0](#): Using Git and Java; Course set-up (Kickstart Assignment)
- A1: Interactive game with level editor (Java)
  - A1.1: UI design
  - A1.2: Basic UI and gameplay (MVC, layout, events, drawing)
  - A1.3: Level editor (MVC, layouts, direct manipulation, graphics)
- A2: Event queue and drawing (Java)
- A3: Web services (HTML + JavaScript)

# Assignment Policies

1. Assignments consist of writing complete, working programs that satisfy a specification. You are expected to submit working source code. For Java code, you are also expected to submit an Apache Ant `build.xml` file that can build it, unless otherwise directed.
2. Assignments will be done in Java and HTML+JavaScript, as specified on the assignment. Code should be "platform-neutral", and can be developed on the desktop platform of your choice (Windows, OS X, Linux). Exact details will be provided in the assignment specifications.
3. A0 will be due on Wednesday, May 17 at 11:59 PM. You must complete it before we will assess any other assignment.
4. The remaining assignments shall be submitted at one of the specified due dates (see the [course schedule](#) for details). You are encouraged to submit each assignment on the due date suggested in the assignment details. However, you are allowed to submit different portions of the assignments at each due date if you wish (for example, at the suggested due date for A1.2, you can submit portions A1.1 or A1.3 if you wish). But you must know that course staff will be assessing your assignment at each due date. Therefore, we encourage you to submit something at each due date, even if it's not complete, to receive assessment and feedback.
5. You are also allowed to resubmit previous portions of the assignments you have previously submitted. Thus, we encourage you to improve your implementation following advice from the TA's feedback and submit the improved version. The goal of this practice is that you learn from the TA's feedback and improve your UI implementation skills. However, resubmissions will only be reassessed at each assignment due date.
6. You are required to submit a self-evaluation of the skills you acquired after working in each assignment, together with your code submission. Please check the [Skill Tree](#) for information on how the assignments will be assessed and graded.
7. Submission must be via your [Git repository](#).
8. Assignments are your individual work:
  - You must design and implement the assignments by yourself. There are no group assignments.
  - You can use code examples provided in-class. You are allowed to directly use or include portions of in-class samples in your assignments. Some assignments may also, at the instructor's discretion, include starter code that you can use.
  - You cannot use anyone else's code in your assignments (this includes current or past students of this course), and you are prohibited from sharing your code with anyone else in the course. You can discuss how you accomplished something in general terms with other students, but sharing code in any way is prohibited.
  - No other third-party code or libraries are allowed in your assignments. You should not be searching for, and using, code that you find on the Internet (this includes GitHub, Stack Overflow and similar sites). It is perfectly acceptable to search for a general technique (e.g. how does this class work?) but you cannot use any third-party code in your assignments.
  - You should not make your code publicly available on any hosting sites (e.g. GitHub), even after the end of the term. If you wish to show your code to potential employers, you should post in a private/restricted repository instead.

## Java Development

For assignments A1 and A2, we use [Java SE 8 SDK \(8u101 or later\)](#). We also use [Apache Ant \(1.9 or later\)](#) to manage builds. A Java IDE is also recommended; either [IntelliJ](#) (you can get a free student licence) or [Eclipse](#).

## Web Development

For assignment A3, we use HTML 5, CSS 3, and JavaScript. A IDE is also recommended; either [IntelliJ](#) (you can get a free student licence) or [Visual Studio Code](#). Your code must run independently of web browser. We recommend Google Chrome or Mozilla Firefox for testing.

# Git Setup

We will be using [Git](#), a distributed version control system, for assignment submissions. You are expected to setup your account on the [UW Git installation](#), and "push" your assignments to a private repository on that server to submit them. As per assignment policies described above, you are not allowed to publically post your assignments online in any other location.

To access your private repository for this course, you need to git-clone the repo to your local machine.

The command looks like this (replace username with your Quest login name, e.g. jdoe.git):

```
git clone https://username@git.uwaterloo.ca/cs349-spring2017/username.git
```

To submit assignments, you need to 'git add' new files, 'git commit' changes and 'git push' to the server (push sends it to the server to be graded!). See [Git slides](#) for more information.

## Policies

### Intellectual Property

Students should be aware that this course contains the intellectual property of their instructor, TA, and/or the University of Waterloo. Intellectual property includes items such as:

- Lecture content, spoken and written (and any audio/video recording thereof);
- Lecture handouts, presentations, and other materials prepared for the course (e.g., PowerPoint slides);
- Questions or solution sets from various types of assessments (e.g., assignments, quizzes, tests, final exams); and
- Work protected by copyright (e.g., any work authored by the instructor or TA or used by the instructor or TA with permission of the copyright owner).

Course materials and the intellectual property contained therein, are used to enhance a student's educational experience. However, sharing this intellectual property without the intellectual property owner's permission is a violation of intellectual property rights. For this reason, it is necessary to ask the instructor, TA and/or the University of Waterloo for permission before uploading and sharing the intellectual property of others online (e.g., to an online repository). Permission from an instructor, TA or the University is also necessary before sharing the intellectual property of others from completed courses with students taking the same/similar courses in subsequent terms/years. In many cases, instructors might be happy to allow distribution of certain materials. However, doing so without expressed permission is considered a violation of intellectual property rights.

### Academic Integrity

In order to maintain a culture of academic integrity, members of the University of Waterloo community are expected to promote honesty, trust, fairness, respect and responsibility. [Check [www.uwaterloo.ca/academicintegrity/](http://www.uwaterloo.ca/academicintegrity/) for more information.]

### Grievance

A student who believes that a decision affecting some aspect of his/her university life has been unfair or unreasonable may have grounds for initiating a grievance. Read Policy 70, Student Petitions and Grievances, Section 4, [www.adm.uwaterloo.ca/infosec/Policies/policy70.htm](http://www.adm.uwaterloo.ca/infosec/Policies/policy70.htm). When in doubt please be certain to contact the department's administrative assistant who will provide further assistance.

## Discipline

A student is expected to know what constitutes academic integrity [check [www.uwaterloo.ca/academicintegrity/](http://www.uwaterloo.ca/academicintegrity/)] to avoid committing an academic offence, and to take responsibility for his/her actions. A student who is unsure whether an action constitutes an offence, or who needs help in learning how to avoid offences (e.g., plagiarism, cheating) or about 'rules' for group work/collaboration should seek guidance from the course instructor, academic advisor, or the undergraduate Associate Dean. For information on categories of offences and types of penalties, students should refer to Policy 71, Student Discipline, [www.adm.uwaterloo.ca/infosec/Policies/policy71.htm](http://www.adm.uwaterloo.ca/infosec/Policies/policy71.htm). For typical penalties check Guidelines for the Assessment of Penalties, [www.adm.uwaterloo.ca/infosec/guidelines/penaltyguidelines.htm](http://www.adm.uwaterloo.ca/infosec/guidelines/penaltyguidelines.htm).

## Appeals

A decision made or penalty imposed under Policy 70 (Student Petitions and Grievances) (other than a petition) or Policy 71 (Student Discipline) may be appealed if there is a ground. A student who believes he/she has a ground for an appeal should refer to Policy 72 (Student Appeals) [www.adm.uwaterloo.ca/infosec/Policies/policy72.htm](http://www.adm.uwaterloo.ca/infosec/Policies/policy72.htm).

## Students with Disabilities

AccessAbility Services collaborates with all academic departments to arrange appropriate accommodations for students with temporary or permanent disabilities without compromising the academic integrity of the curriculum. If you require academic accommodations, please register with the [AccessAbility Services](#) at the beginning of each academic term.

---

[University of Waterloo](#)

[Cheriton School of Computer Science](#)

[Gustavo Fortes Tondello](#)



Spring 2017

# CS349 User Interfaces

[\(show/hide past dates\)](#)

This schedule is subject to change.

1	Mon, May 01	Introduction	1-Introduction	
2	Wed, May 03	History	2-History	
3	Fri, May 05	Tools tutorial	3.1-Git tutorial	3.2-Java tutorial
4	Mon, May 08	Design principles		
5	Wed, May 10	Design principles		
6	Fri, May 12	Visual design		
7	Mon, May 15	Visual design		
8	Wed, May 17	Design Process		
		A0: Java and Git Kickstart	A0	
9	Fri, May 19	Design Process		
	Mon, May 22	Victoria Day (no lecture)		
10	Tue, May 23	Windowing Systems		
11	Wed, May 24	Model-View-Controller (MVC)		
12	Fri, May 26	Model-View-Controller (MVC)		
13	Mon, May 29	Drawing and Events		
14	Wed, May 31	GUI implementation in Java		
		Assignment due date. Suggested submission: A1.1: Game (UI design)		
15	Fri, Jun 02	Widgets		
16	Mon, Jun 05	Event Dispatch and Handling		
17	Wed, Jun 07	Layouts		
	Thu, Jun 08	Midterm (7:00-8:50pm at M3 1006)		
18	Fri, Jun 09	Input Devices		
19	Mon, Jun 12	Input Performance		
20	Wed, Jun 14	2D Graphics		
21	Fri, Jun 16	2D Graphics		
22	Mon, Jun 19	Direct Manipulation		
		Assignment due date. Suggested submissions: A1.2: Game (basic UI and gameplay) and A2: Event queue		
23	Wed, Jun 21	Features: Undo		
24	Fri, Jun 23	Features: Cut 'n Paste; Drag 'n Drop		
25	Mon, Jun 26	Mobile Development		
26	Wed, Jun 28	Web Development		
27	Fri, Jun 30	Responsiveness (layout)		
	Mon, Jul 03	Canada Day (no lecture)		
28	Wed, Jul 05	Responsiveness (time)		
		Assignment due date. Suggested submission: A3: Game (Level editor)		
29	Fri, Jul 07	Touch interfaces		

---

30	Mon, Jul 10	Touchless interfaces
31	Wed, Jul 12	Wearables
32	Fri, Jul 14	Gamification
33	Mon, Jul 17	Gamification
34	Wed, Jul 19	Visual perception and Cognition
35	Fri, Jul 21	Accessibility
36	Mon, Jul 24	UI Evaluation

---

Assignment due date.  
Suggested submission: A3: Web services

---

---

[University of Waterloo](#)

[Cheriton School of Computer Science](#)

[Gustavo Fortes Tondello](#)

Spring 2017

# CS349 User Interfaces

The *skill tree* represents the skills that students will acquire by the end of the course and will replace assignment marks for this course.

Each student is required to do a self-evaluation of skill proficiency by each assignment submission date. You will rate your proficiency in each one of the skills in a scale of 0 to 4 (see below). In addition, you will provide evidence of the achievements that demonstrate that you have learned each skill, meaning that you will explain what you did in each assignment that demonstrate that you have learned each skill.

The TAs will then assess your assignment and your self-evaluation and provide feedback for you to improve your skills.

The skill levels are:

0. **No proficiency:** you don't know how to use the skill yet.
1. **Beginner:** you have an initial understanding of the skill, but you can't use it with proficiency yet.
2. **Learning:** you are able to use the skill for basic tasks, but you still struggle to use it in complex tasks.
3. **Proficient:** you are able to use the skill for moderate and complex tasks, but sometimes you still need some time to figure how to use it for complex tasks.
4. **Mastery:** you have used the skill a few times for moderate and complex tasks and you are confident that you can use it for any task.

Assignment will not have a numeric grade, only an evaluation of the skills you acquired after working on them.

Therefore, your final assignment grade for the course will be derived from the percentage of skills that you have acquired. For example:

- The passing grade for the course is 50%, meaning that in average you have achieved level 2 (Learning) on all skills.
- If you complete all assignments without error, but also without enhancements, you will achieve level 3 (Proficient) on all skills, which correspond to a final grade of 75%.
- If you add enhancements to your assignments, you may achieve level 4 (Mastery) on some of the skills.
- A final grade of 100% on the assignments would be granted if you achieve level 4 (Mastery) on all the skills.

Additional information on how to submit your self-evaluation of skills will be provided as the assignments are revealed.

This is the complete skill tree for CS 349 (please scroll horizontally to see all the skills):

