Contents

Syllabus **1**
- Course Content . . . 1
- Course Structure . . . 1
- Course Grading . . . 1
  - EMRN Scoring . . . 2
  - Solo Project . . . 2
  - Group Project . . . 2
  - Letter Grades . . . 2
- Academic Integrity . . . 3

Syllabus

Course Content

We will cover the following topics this semester:

1. Platform Overview (1 week)
2. Application Lifecycle and Visual Elements (2 weeks)
3. Application State, Graphics, and Animation (2 weeks)
4. Physics Properties, Permissions, and Location (2 weeks)
5. Persistence, Threading, and Network Services (2 weeks)
6. Styling, Sensors, and Profiling (2 weeks)
7. Events and Notifications (2 weeks)
8. Special Topics (1 week)
9. Group Project Demonstrations (1 week)

Course Structure

Class times will be split between lectures and hands-on work, roughly alternating weeks. For the hands-on work, we will focus on developing an application that illustrates the majority of the course concepts.

The graded work in the course will consist of two projects, a multiple-phase solo project and a group project. You will have considerable flexibility in how rapidly you complete these, but we will send you reminders if you are falling behind.

Course Grading

This course employs *Standards-Based Grading*, rather than traditional points-based grading. Rather than accumulating points that are then used to determine the cut-offs for various letter grades, there are specific *learning objectives* for students to demonstrate mastery of. A final grade in the course is determined by how well these objectives were met, and you will be able to choose the grade you are trying to obtain in accordance with the amount of time you have to devote to the course.

There are two sequences of objectives to meet:

1. The seven phases of the solo project;
2. The five stages of the group project.

Some of these objectives are scored as Completed/Not Completed, and serve as gateways to later objectives. Other objects are EMRN-scored (see below), and determine your final grade. In all cases, each step of a sequence must be completed satisfactorily before the next step may be accessed.

EMRN Scoring

The following is the EMRN scale. The points are only there to work with the Canvas gradebook.

- **E:** (3 points) Excellent; demonstrates mastery of the material
- **M:** (2 points) Meets Expectations; approaching mastery, with minor updates or learning required
- **R:** (1 point) Requires Resubmission; demonstrates progress towards mastery, but additional learning and major updates
- **N:** (0 points) Not Assessable; does not indicate that progress has yet been made

The specifics of these assessments will be specified in project phase descriptions.

Solo Project

This will be an app, called Terpiez, that everyone will implement. It will ultimately need to interoperate with a single course-wide database, which we will set up, so it is important to set things up exactly as specified.

Phase 0 of the project is a gateway, the six remaining phases are EMRN-scored. In order to progress to a later phase, you must receive at least an M on the prior one. You will be able to resubmit project phases until you reach the assessment level you need for the grade you want. When resubmitting a project, you must also include a brief discussion of what enabled you to make progress. Because the resubmission attempts are unlimited, there will be no extensions granted. No excuse needs to be given, merely an acknowledgement that more work was needed.

*You may **not** use Generative AI, such as ChatGPT, to produce your code, and you may not upload the code we provide to any such system (this would be a copyright violation). If you are using code you find elsewhere, you must cite it in the comments.*

Group Project

There is also a team project, where each team of up to 5 students will propose an app and then implement it. Proposals will include a set of minimal goals (required functionality) and stretch goals (extra functionality). The initial proposal will be evaluated and approved by Dr. Marsh, after which a TA will be assigned to each team, who will monitor the progress and assign final project scores.

Projects will be graded as follows:

- E: All minimal goals met, and at least one stretch goal met.
- M: All minimal goals met.
- R: App implements substantial functionality, but does not yet meet all minimal goals.
- N: App is lacking substantial functionality.

Teams and TAs will agree to a schedule of three milestones, where the TA will assess the state of the project, and all parties will discuss any necessary changes to the design or schedule. No milestones may be scheduled until the project proposal is approved.

*You may **not** use Generative AI, such as ChatGPT, to produce your code, and you may not upload the code we provide to any such system (this would be a copyright violation). If you are using code you find elsewhere, you must cite it in the comments.*

Letter Grades

Your final grade in the course will be determined by the number of EMRN-scored assessments you have at various levels. There are a total of 7 such assessments.

Grade	# E	# M	# R
A	7	0	0
A-	6	1	0

Grade	# E	# M	# R
B+	5	2	0
B	4	3	0
B-	3	4	0
C+	2	5	0
C	1	6	0
C-	0	7	0
D	6		1

Academic Integrity

You are expected to adhere to the University's academic integrity policy. All work that is not explicitly stated to be a team effort must be done by you alone. *You have 24 hours after the deadline of any graded work (including tests) to self-report lapses in academic integrity*, which we can then hopefully resolve without a reference to the Office of Student Conduct.

Posting of any project/exercise implementation (even after the course is over) in a publicly available online location (e.g., github) is prohibited under the Code of Academic Integrity (facilitation of academic dishonesty). Any student responsible for this will be reported to the Office of Student Conduct and risks the sanction of an "XF" in the course. The CS Department gitlab server (https://gitlab.cs.umd.edu (Links to an external site.)) is an acceptable place for your code, as long as access is appropriately restricted.