The Road Ahead
Final Game Prototype
If your team didn’t make a submission, you missed your E3 demo

- I’m going to reopen the assignment, but the late penalty applies
- It’s better to turn in partial work than to turn in nothing
- It’s better to turn in an early version and then later turn in an updated version

I wanted to see each of you individually submit your game

- Part of the intent behind this is to encourage you to stay in communication with me
- However, I wasn’t here to emphasize this
  - So I’m going to make an exception to the late penalty
  - Submit a text file with a brief description of the parts of the game you worked on

Remember the code review assignment is due soon!
Final Game
Grading Criteria
The Batter

+5 post your game online (itch.io) and submit a link https://itch.io/jam/ucsc-cmpm-120-summer-2019

+5 game runs/executes without critical errors or crashes (Graders will use Chrome, so be sure you game works in that browser.)

+5 turn in a .zip of your final game file structure, including assets - the game MUST run via localhost with no code tweaking, folder rearranging, etc.

+5 game has appropriate title screen, end state, and ability to restart from within the game (based upon your game, genre, etc.)
The Batter

+5 the player can learn the controls from within the game, whether through a tutorial, instruction screens, or other diegetic means

+10 well-structured code, including comments, appropriate data structures, sensible ‘classes’ and states, etc.

+10 properly maintained and updated team GitHub (or equivalent) - submit this as a link in a .txt file
The Bake

**+10** game’s artistic cohesion *(does the art/sound/etc. reflect your stated aesthetic/experience goals? do your assets make sense together?)*

**+10** game’s mechanical cohesion *(do the mechanics reflect your stated aesthetic/experience goals? does the game feel good? are the mechanics well-implemented?)*
The Icing

+5 polish/creativity/originality/tilt (did we enjoy your game? does it stand out? does it go above and beyond the stated objectives?)
Future Topics
Top Requests

➔ Animation / Tweening
➔ Particles
➔ Mouse / Click & Drag
➔ Dialog / Branching Narrative
Also Requested

- Cutscenes
- UI
- Transition States
- Procedural Generation
- P2 Physics
Structuring Your Program
Model View Controller

Model
data and logic

View
representation of the model

Controller
converts input into commands

User
Observer

subject
list of functions
can add and remove from the list
calls functions on the list when thing happens

observer
when this function is called, react however we want

observer
when this function is called, react however we want

https://gameprogrammingpatterns.com/observer.html
Observer - you've seen this with collisions!

A collision has happened!

Notify the callback functions

Collision

(player, coin)
Increase the score
Remove the coin

(player, enemy)
decrease the player's hitpoints

https://gameprogrammingpatterns.com/observer.html
More Debugging Tips
Useful random debugging advice

1. When you find a problem, change something so that same problem can't happen again
   a. `assert()`
   b. Keep a debugging notebook

2. Make debug tools
   a. Quicker feedback is better
   b. Display values live if possible

3. Only make one change at a time and then test it

4. Just because you paused the game doesn't mean it's paused
   a. And stopping one update doesn't mean you stopped all of them

5. `console.log()` is slow
   a. Faster to print an array as a string than to individually print the contents
Useful random debugging advice

Walk through your code step by step, explaining to yourself what is supposed to happen
Useful random debugging advice

1. When you find a problem, change something so that same problem can't happen again
   a. assert()
   b. Keep a debugging notebook

2. Make debug tools
   a. Quicker feedback is better
   b. Display values live if possible

3. Only make one change at a time and then test it

4. Just because you paused the game doesn't mean it's paused
   a. And stopping one update doesn't mean you stopped all of them

5. console.log() is slow
   a. Faster to print an array as a string than to individually print the contents
Useful random debugging advice

1. When you find a problem, change something so that same problem can't happen again
   a. assert()
   b. Keep a debugging notebook

2. Make debug tools
   a. Quicker feedback is better
   b. Display values live if possible

3. Only make one change at a time and then test it

4. Just because you paused the game doesn't mean it's paused
   a. And stopping one update doesn't mean you stopped all of them

5. console.log() is slow
   a. Faster to print an array as a string than to individually print the contents
Useful random debugging advice

1. When you find a problem, change something so that same problem can't happen again
   a. assert()
   b. Keep a debugging notebook

2. Make debug tools
   a. Quicker feedback is better
   b. Display values live if possible

3. Only make one change at a time and then test it

4. Just because you paused the game doesn't mean it's paused
   a. And stopping one update doesn't mean you stopped all of them

5. console.log() is slow
   a. Faster to print an array as a string than to individually print the contents
Useful random debugging advice

1. When you find a problem, change something so that same problem can't happen again
   a. assert()
   b. Keep a debugging notebook

2. Make debug tools
   a. Quicker feedback is better
   b. Display values live if possible

3. Only make one change at a time and then test it

4. Just because you paused the game doesn't mean it's paused
   a. And stopping one update doesn't mean you stopped all of them

5. `console.log()` is slow
   a. Faster to print an array as a string than to individually print the contents
AABB characters and slopes

An example of a real-world physics-and-debugging problem in a game with 2D physics like yours

https://twitter.com/eevee/status/1133248372624613376