http://www.atlp.org/pics/menage.jpg

Transportation

Part 1	Due before class on **10/21**
Part 2	Due before class on **10/28**
Part 3	Due before class on **11/4**
Part 4	Due before class on **11/6**
Final	Due before class on **11/11**

For this project, you will create a simulation of a transportation system. This project is intended to help you increase your skills in the following areas:

- Resourcefulness and Persistence
  - Searching for answers online, bookmarking commonly used webpages
  - Asking for help from Ira, the TA, or other students when you're stuck
- Logical Thinking
  - Carefully thinking through step-by-step logic for the program
- Using Unity
  - Creating new projects, files, and assets
  - Attaching scripts to objects
- Using GitHub
  - Storing and sharing your program as it changes over time
- Publishing a game to your own website
  - How to create a basic website and the tools required
  - FTP, basic text editor, Unity
- Organization
  - Well-organized and well-named files, folders, assets, and variables
  - Nicely formatted code with matching {} and indentation
  - Comments to help understand the code
- Types of variables
  - int, float, bool
  - How to create them and modify them
- Data structure: Arrays
  - What they are, how to create them and modify them
- Mathematical expressions
  - Comfortable with adding, subtracting, multiplying and dividing
- Conditional expressions
  - if(), else if(), else()
- Iterative statements
  - for() loops

Wow, that’s a lot! It’s amazing what you’ll be able to learn in just a few weeks of effort!

What To Hand In

For each part of this project, **deliver your work as follows**:

a) Within Unity, make a WebGL build (File -> Build Settings)
b) Upload the .html file, Build folder, and TemplateData folder to your website.
c) Upload your project files (especially .cs) to GitHub.
d) Find your tab on the spreadsheet below and add your website and GitHub links for the appropriate part of the project:
https://docs.google.com/spreadsheets/d/1Bjl9XKOAhpX2iPvoxRU8lQTS2qAHqVMEA6z8l74SmGk
e) Post in the #cs181_handin channel that you completed each part.

Part 1

By the deadline, create a game that functions as follows:

1) When the game starts, the player should see a row of 16 cubes.
    a. All cubes should be white.
    b. The cubes should be evenly spaced and stationary.
    c. The cubes should be created by the C# script. (Don’t prepopulate the 16 cubes by hand in Unity.)
2) When the player clicks on a cube, it should turn red.
    a. If the player clicks on a different cube, the new cube should turn red, and the old cube should turn white.

If you complete this part before the deadline, go on to the next part.

Part 2

By the deadline, create a game that functions as follows:

1) When the game starts, the player should see a grid of 16 x 9 cubes.
    a. The grid should be entirely clear sky, represented by white cubes, except for the upper left corner, which should be an airplane, represented by a red cube.
    b. The cubes should be evenly spaced and stationary.
    c. The cubes should be created by the C# script. (Don’t prepopulate the 16 cubes by hand in Unity.)
2) When the player clicks an airplane, it should highlight in some way to show that it’s the active airplane. (Change to a different color, glow, enlarge, etc.)
3) If the player clicks the active airplane, it should deactivate.
4) If the player clicks clear sky (a white cube) while there is an active airplane, the airplane should teleport to that location (i.e. the cube that was previously red should now be white, and the clicked cube should now be red).
5) If the player clicks a white cube while there is no active airplane, nothing happens.

If you complete this part before the deadline, go on to the next part.

Part 3

By the deadline, extend the game from Part 2 with these additional features:

1) The game takes place over many turns. Each turn takes 1.5 seconds.
2) The airplanes can carry cargo.
    a. Each airplane has a cargo capacity of 90 tons.
    b. At the beginning of the game, each airplane has no cargo.
3) Each turn a vehicle is in its starting location, it gains 10 tons of cargo (never above max capacity).
4) Show the current cargo value somewhere on screen.
5) There is a delivery depot in the lower right (black cube). If a vehicle reaches that location, it delivers all its cargo.
    a. The player gets 1 point for each ton of cargo delivered.

6) Show the player’s current score somewhere on screen.

If you complete this part before the deadline, go on to the next part.

Part 4

By the deadline, extend the game from Part 3 with this additional feature:

1) Instead of clicking to teleport the airplane that you implemented in part 2, make the airplane move based on keyboard input. If the player presses an arrow key while there is an active airplane, the airplane moves in that direction on the next turn. If the player touches many arrow keys between turns, only move one space total, in the most recently touched direction. The airplane is restricted to the grid.

Final

By the deadline, create a game that functions as follows:

1) The game starts with a 16x9 grid of white cubes, evenly spaced, created by the C# script.
2) There is one airplane that starts in the upper left, with a cargo capacity of 90.
3) There is a delivery depot in the lower right (black cube). If a vehicle reaches that location, it delivers all its cargo.
    a. The player gets 1 point for each ton of cargo delivered.
4) The game takes place over many turns. Each turn takes 1.5 seconds.
5) Each turn a vehicle is in its starting location, it gains 10 tons of cargo (never above max capacity).
6) When the player clicks a vehicle, it should highlight in some way to show that it’s the active vehicle. (Change to a different color, glow, etc.)
    a. If there is an active vehicle and the player clicks a different vehicle, the new vehicle should activate, and the old vehicle should deactivate.
7) If the player clicks the active vehicle, it should deactivate.
8) While there is an active vehicle, if the player clicks any location, the active vehicle should start to move to the clicked location. Movement should happen as follows:
    a. Movement happens one grid space at a time (diagonally is OK), once each turn.
    b. The vehicle remembers where it’s trying to go from turn to turn. Each turn it will move one space in the right direction, until it reaches its destination.
    c. The moving vehicle remains active, so it can be given a new destination partly through the movement.
9) If the player clicks a white cube while there is no active vehicle, nothing happens.

If you complete the final part before the deadline, try some Challenge by Choice items below.

Challenge by Choice: Do any or all of the following additional tasks. I've included categories below, but you can pick and choose however you want.

Game Design Complexity

1) There is one airplane, one train, and one boat, as follows:

Vehicle	**Cargo Cap.**	**Start Loc.**	**Color**	**Speed**
Airplane	90	Upper Left	Red	Fast
Train	200	Lower Left	Green	Medium
Boat	550	Upper Right	Blue	Slow

2) Movement:
    a. Fast vehicles move once each turn.
    b. Medium speed vehicles move every other turn.
    c. Slow vehicles move every third turn.
    d. Movement happens one grid space at a time (diagonally is OK), once each turn.
    e. The vehicle remembers where it’s trying to go from turn to turn. Each turn it will move one space in the right direction, until it reaches its destination.
    f. The moving vehicle remains active, so it can be given a new destination partly through the movement.
3) Grid locations may have more than 1 vehicle.
    a. If the player clicks on a location with multiple vehicles, do something reasonable to pick which vehicle becomes active.
    b. There should never be more than one active vehicle at a time, though.

Feedback to the Player

- Whenever players mouse-over any location (but before clicking on it), give a visual indication that it’s clickable (change the color a bit, enlarge it slightly, etc. Your choice of what to do, but do something.)
- Somewhere on the screen, display a count-up timer that shows how long the game has been going.
    - If showing the game timer, make it fancy: After 59 seconds, show 1:00 instead of 60. After 60 minutes, show 1:00:00 instead of 60:00.
- Somewhere on screen, display the player's current score.
    - Along with the player's current score, show the word "Score."
    - At the moment the player earns points, display the points earned somewhere (+10!).

Sound Effects

- Whenever the player clicks a vehicle, play a sound effect (SFX).
- Whenever the player clicks a vehicle, play a different SFX depending on which type of vehicle it is.
- Play music that loops, so there's constantly music playing during the game.

Visual Polish

- Make the vehicles and sky look prettier, instead of just being simple colored cubes. You can change the shape, texture, etc. The goal is making it pretty. Feel free to add a background to the game, too.
- Instead of having the vehicle be cubes, have the vehicles move on top of the cubes, from cube to cube.

Original project designed by Professor Ira Fay. Inspiration from Stone Librande, with help from Sam Goodspeed.