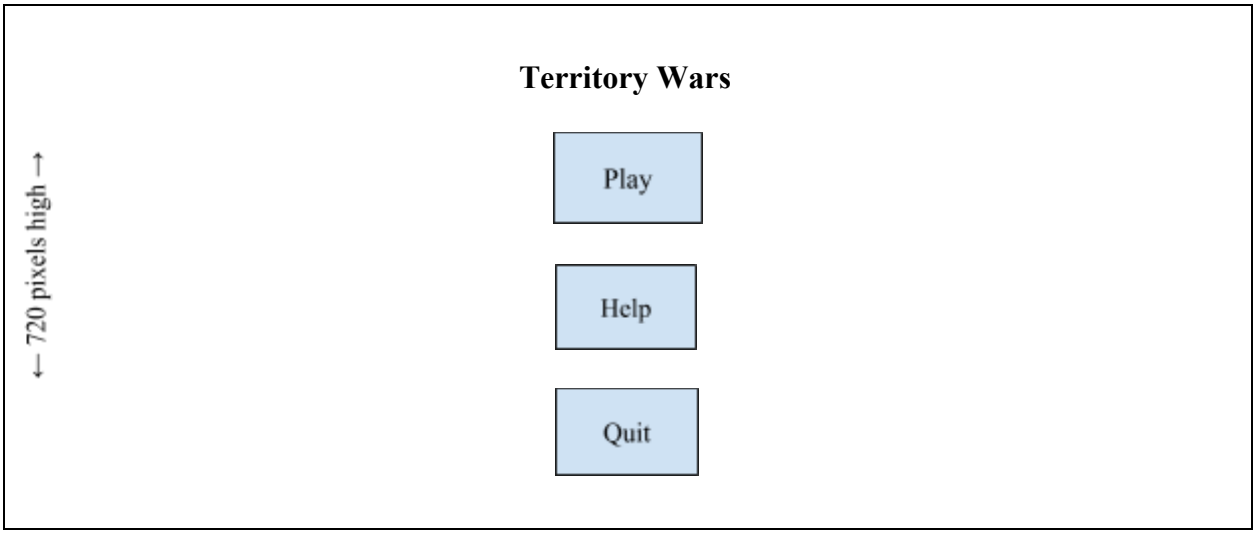
ICS4U1 Final CPT Requirements - Territory Wars

Use javaswing

- Import javax.swing library and use its components to create a graphical user interface (GUI)
- A variety of java swing components must be used
    - Java swing javadoc contains information for the components
    - Use menus, buttons, text fields, etc. (not all just animations)
    - Ex. JButton, JLabel, JTextArea, JFrame, JPanel, etc
- Game panel should be exactly 1280 x 720 pixels (shouldn't be able to resize the frame, turn off resizing features)



← 1280 pixels wide →

- Sample menu that be created using some java swing components
    - JButtons for play, help, quit buttons

Perform animations

- Animations must occur throughout the program
    - Animating shapes and/or images
    - Draw commands (fillRect, fillOval)
- This can be done using by creating a java file that extends JPanel and overrides its paint component method to draw in the default blank panel
    - Ex. public class TerritoryWarsPanel extends JPanel

- The players and bullet will be animated throughout the program and interacting with each other and the map

Use a combination of keyboard and mouse input

- The mouse AND keyboard should be actively used as input sources for the game
    - In other words, there should be more than one source of input
- This means the main program should implement MouseListener and MouseMotionListener (to listen to mouse events) and KeyListener (to listen to keyboard events)
    - Ex. public class TerritoryWars implements ActionListener, MouseListener, MouseMotionListener, KeyListener

- Main use of keyboard will be the arrow keys to obtain actions for player movement
- Keyboard can also be used to type in names, IP, chat

- Mouse input will be used mainly for clicking buttons
- Also can be used to shoot bullets

Use File IO

- There must be data being input and output into text files
- Must have one or more data files that contain the data for the game (ex. Objects for the game)
- Everyone can look the same
- One map using by using a comma separated value (CSV) file
    - Create a text file (map.txt) which is the map used for the whole game
    - In the text file, letters will be used to represent pictures
    - For example: g represents ground, s represents sky
    - If each tile is made 40 pixels by 40 pixels, there should be (1280/40) 32 tiles going across horizontally across, and (720/40) 18 tiles going vertically (total of 32*18 = 576 tiles on screen)

sniper.txt

```
10
10
20
5
```

- Sample text file for sniper.txt, containing data for the sniper gun
    - In order of bullet width, bullet height, bullet speed, and bullet damage

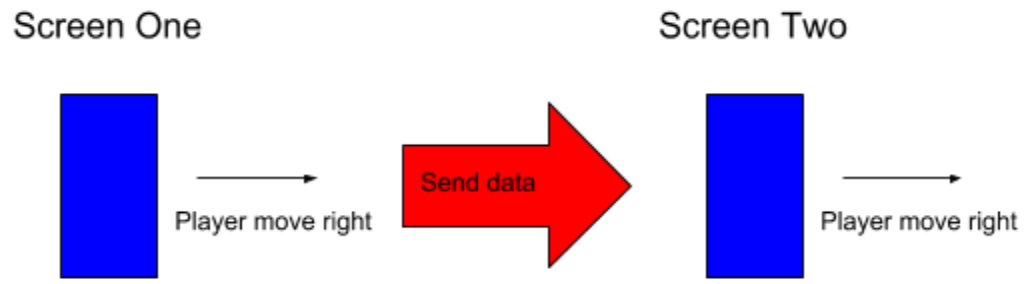
map.csv

s	s	s	s	s
s	g	g	g	s
s	s	s	s	s
g	s	s	s	g
g	g	s	g	g

- Small 5x5 sample of map.csv file
    - Letter s will be replaced by a sky image in code
    - Letter g will be replaced by a ground image in code

Use Socket IO

- Program must be able to talk over a network (interact with other devices)
    - Two-player game (involves a host server and client server)
    - This can be done through SuperSocketMaster, which can quickly open a Java network socket
- All programs should have some sort of chat functionality (ex. Chat area)
- Develop a networking protocol that will work for your game
    - Distinguish between gameplay and chat data (ex. Use comma split)
- Game must only be one window
    - The game shouldn't require multiple windows on the same device to be opened in order to function



- If player moves in one direction (ex. right) on one screen, he should also move on the other screen as well
    - This can be done by sending data through SuperSocketMaster

Be submitted and updated on Github by all members

- Final code must be uploaded on Github
- Code must also be regularly updated on Github throughout its development
    - This can serve as the daily log journal

- Create a repository on Github, add group members as contributors and upload code on a regular basis