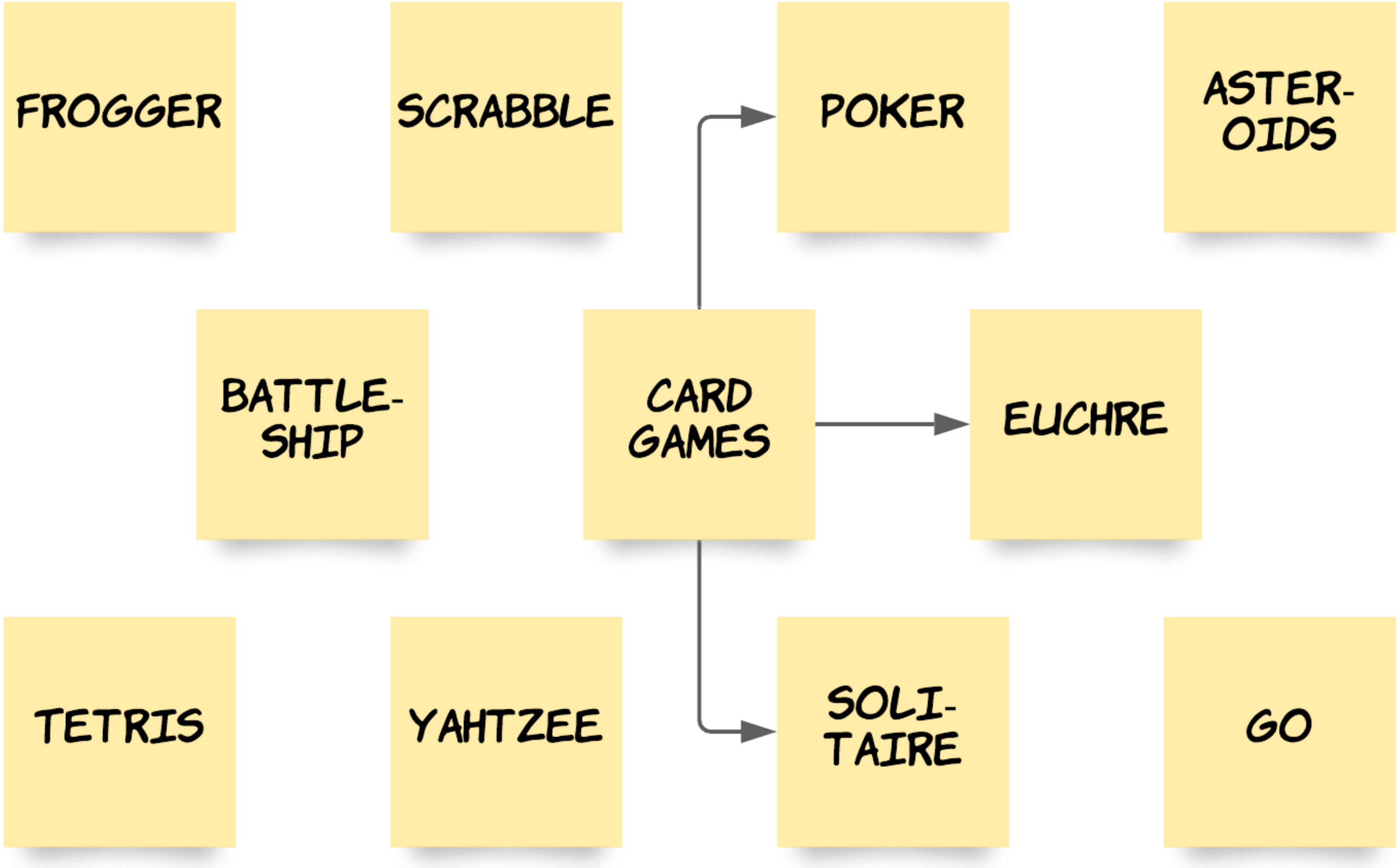
QUEEN'S THEMED BATTLESHIP

Team Cobol for
CISC/CMPE 320



FROGGER

SCRABBLE

POKER

ASTEROIDS

BATTLESHIP

CARD GAMES

EUCHRE

TETRIS

YAHTZEE

SOLITAIRE

GO

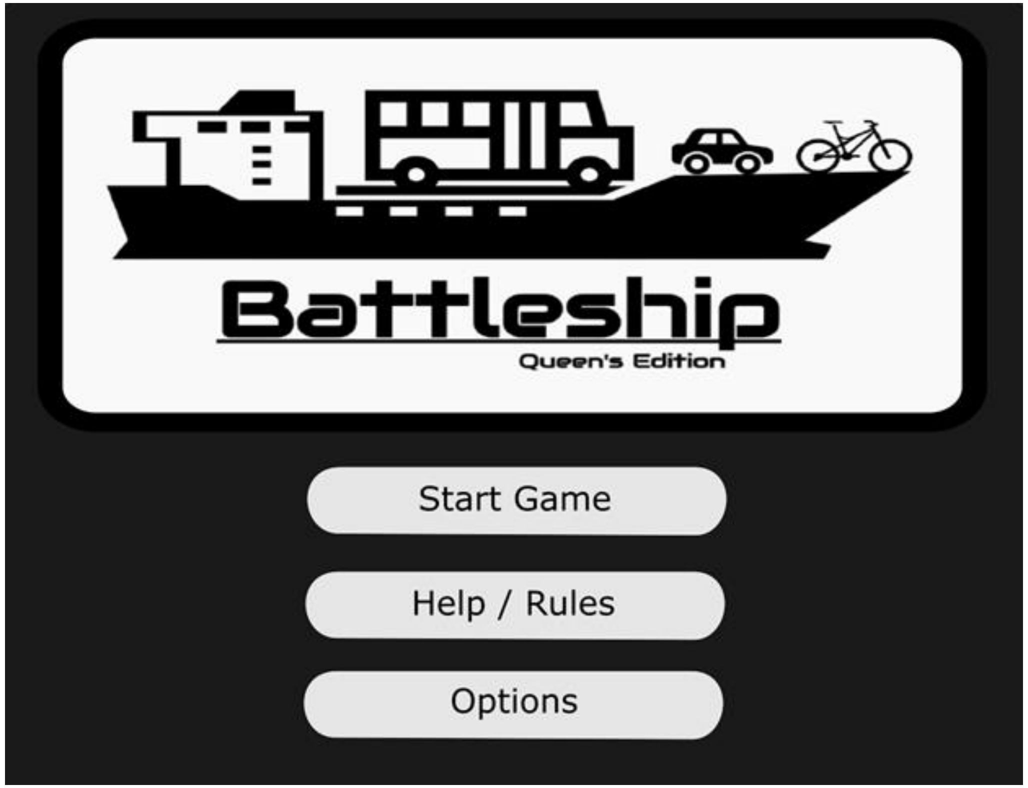
IDEA GENERATION/ EVALUATION

- Feasibility/complexity (5)
- Time commitment (4)
- Level of interest (1)
- Extra feature potential (3)

SPRINT 1 – RAD

Length: Oct. 5th to 9th (one week)

- List of roles assigned
- Sketches of UI
- Functional/non-functional requirements
- Features



TEAM ROLES

Team Lead:
Isabelle Quail

Primary Software Architect:
Danielle Mott

Tester:
Drew Anderson

Component Programmer:
Douglas Gowing

GUI Lead:
Eric Leuty

Illustrator:
Jeremy Browne

FUNCTIONAL REQUIREMENTS (BRIEFLY)

As a player, I want...

- To be able to play against another player on the same laptop so I have an opponent
- To be able to quit the application whenever I want
- The location of my ships to be hidden from my opponent, so they're forced to guess where they are located
- To be able to select a square on the grid to shoot so I can attempt to "sink" my opponent's battleships
- To know whether my last shot was a miss, a hit, or if I sunk a ship, so I can know if I should shoot in that area again
- To know whether I've sunk all my opponent's ships or vice versa, so I can know if I've won or lost

NON-FUNCTIONAL REQUIREMENTS (BRIEFLY)

As a player, I want…

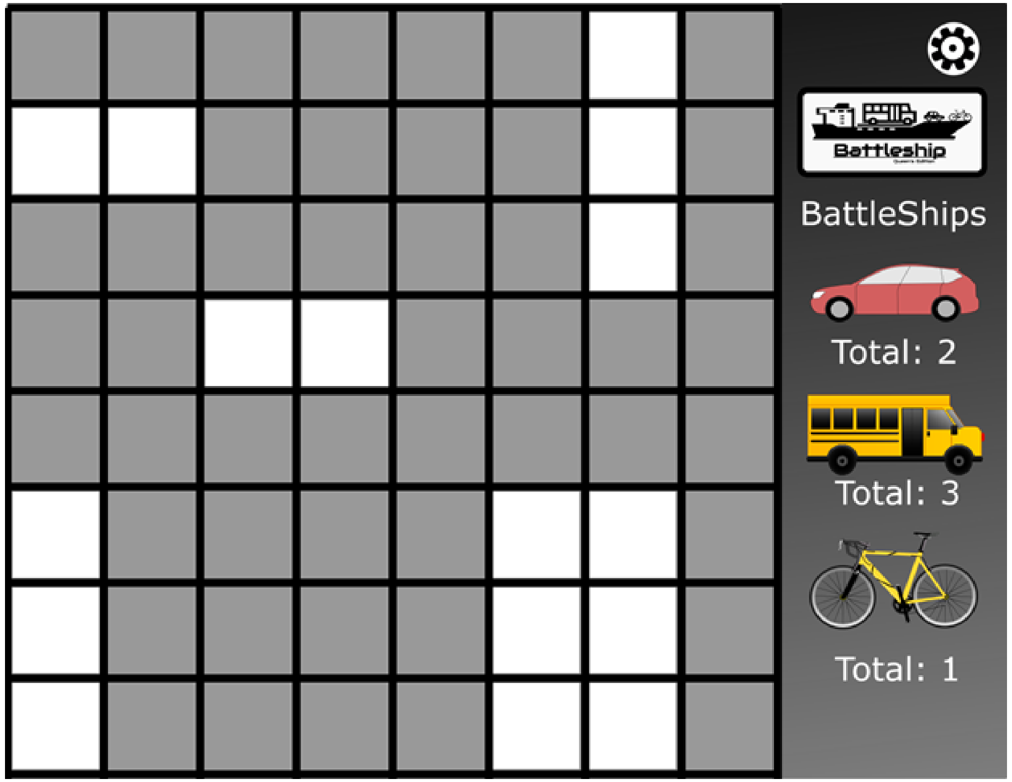
- To be able to enter my desired name so I can be identified while playing the game
- To be able to access the rules within one click from any screen
- My opponent's turn length to be reasonable (i.e. not very long)
- The game to have a fast loading time (< 100ms)
- The game not to crash when I play

Speaking: Douglas Gowing

SPRINT 2 – CONCEPT DESIGN

Length: Oct. 13th to 19th (one week)

- Clone repo
- Gameplay outline
- UI prototypes



SPRINT 3 – UI DESIGN AND BACKEND

Length: Oct. 19th to Nov. 3rd (two weeks)

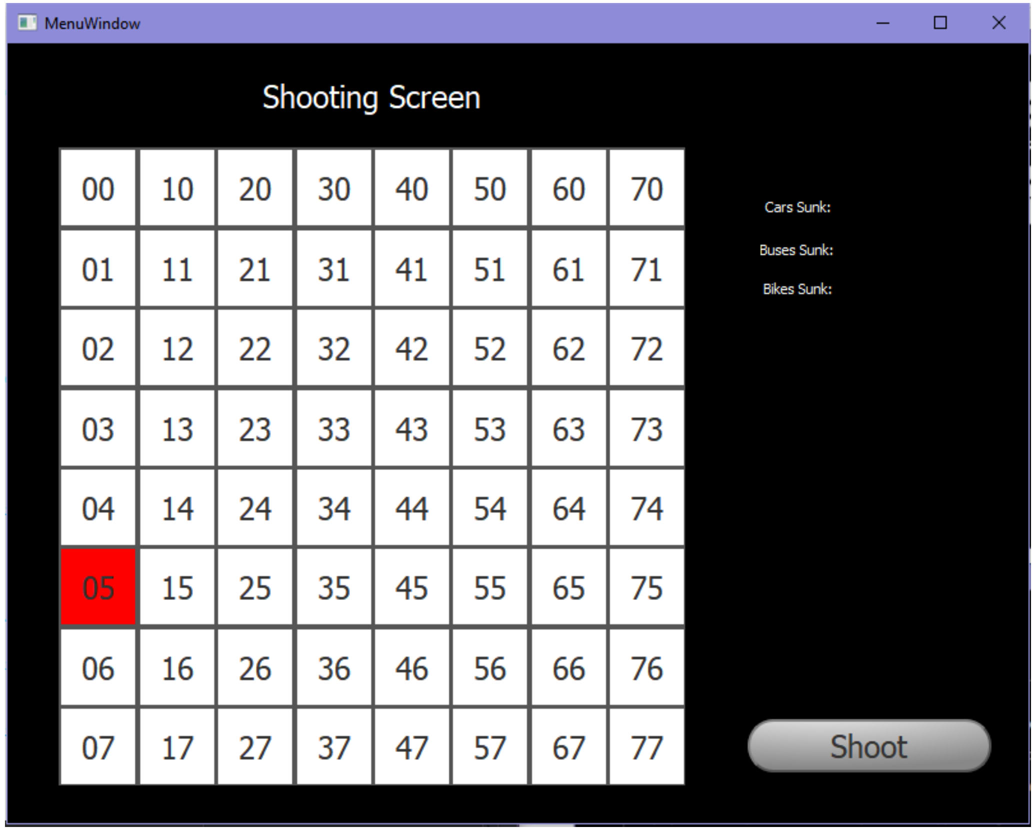
- Create main UI for Battleship
  - Main menu screen
- Pick backend
  - Jeremy prototype vs. Eric prototype

- Basic end-to-end game structure
- Rough class diagrams
- Flush out bugs

SPRINT 4 – WEEK 9

Length: Nov. 5th to 16th (one week)

- Write SDD
  - Compile class diagrams
  - Compile UI screenshots
- Development of base UI
  - All relevant game windows present (no functionality)
- Integration of UI with backend
  - Ship placement screen
  - Shot placement screen



Shooting Screen

00	10	20	30	40	50	60	70
01	11	21	31	41	51	61	71
02	12	22	32	42	52	62	72
03	13	23	33	43	53	63	73
04	14	24	34	44	54	64	74
05	15	25	35	45	55	65	75
06	16	26	36	46	56	66	76
07	17	27	37	47	57	67	77

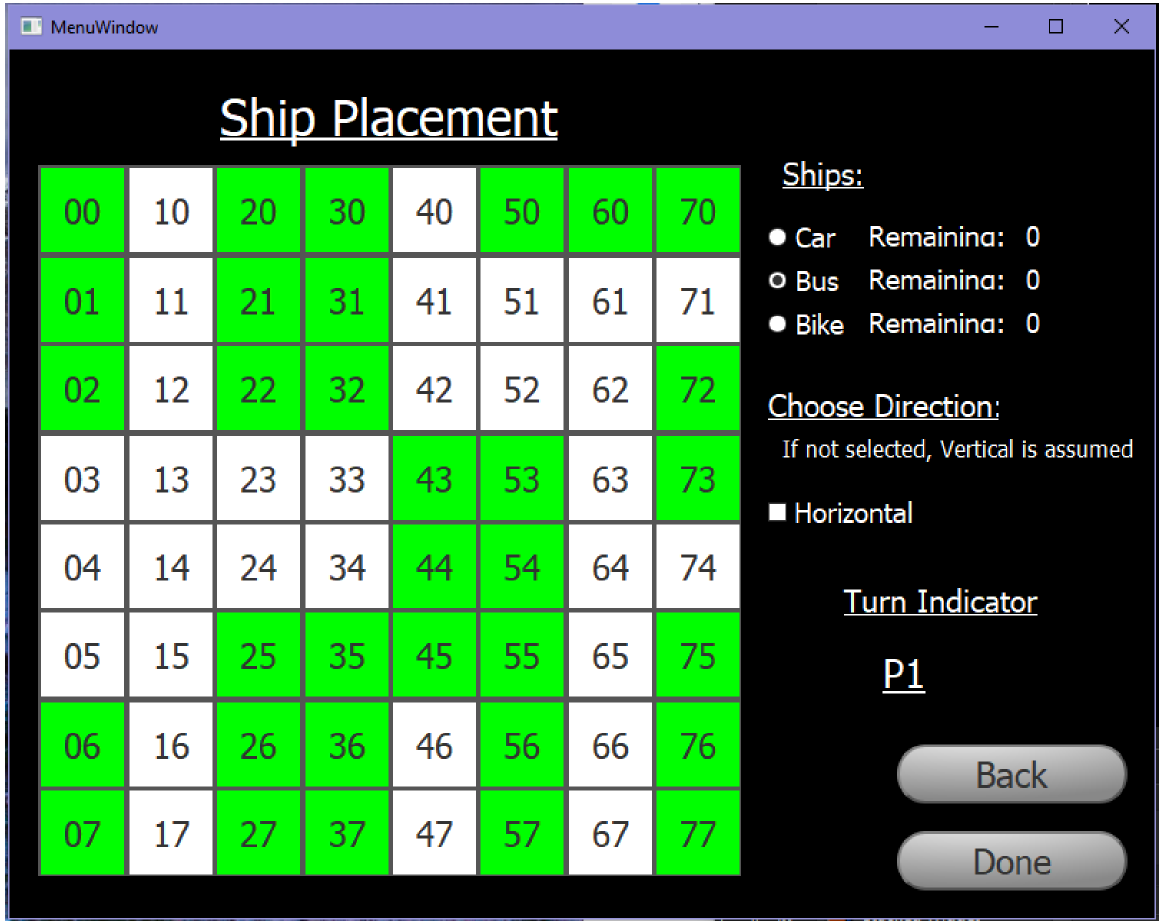
Cars Sunk:

Buses Sunk:

Bikes Sunk:

SYSTEM DESIGN DOCUMENT

- Required classes, attributes, methods
  - Ship/ShipException classes
  - Grid/GridException classes
  - Game class
  - (See next slides for full class diagram)
- Updated GUI screenshots
- Detailed timeline & specific coding assignments for coming weeks



Ship Placement

00	10	20	30	40	50	60	70
01	11	21	31	41	51	61	71
02	12	22	32	42	52	62	72
03	13	23	33	43	53	63	73
04	14	24	34	44	54	64	74
05	15	25	35	45	55	65	75
06	16	26	36	46	56	66	76
07	17	27	37	47	57	67	77

Ships:

● Car Remaining: 0
○ Bus Remaining: 0
● Bike Remaining: 0

Choose Direction:

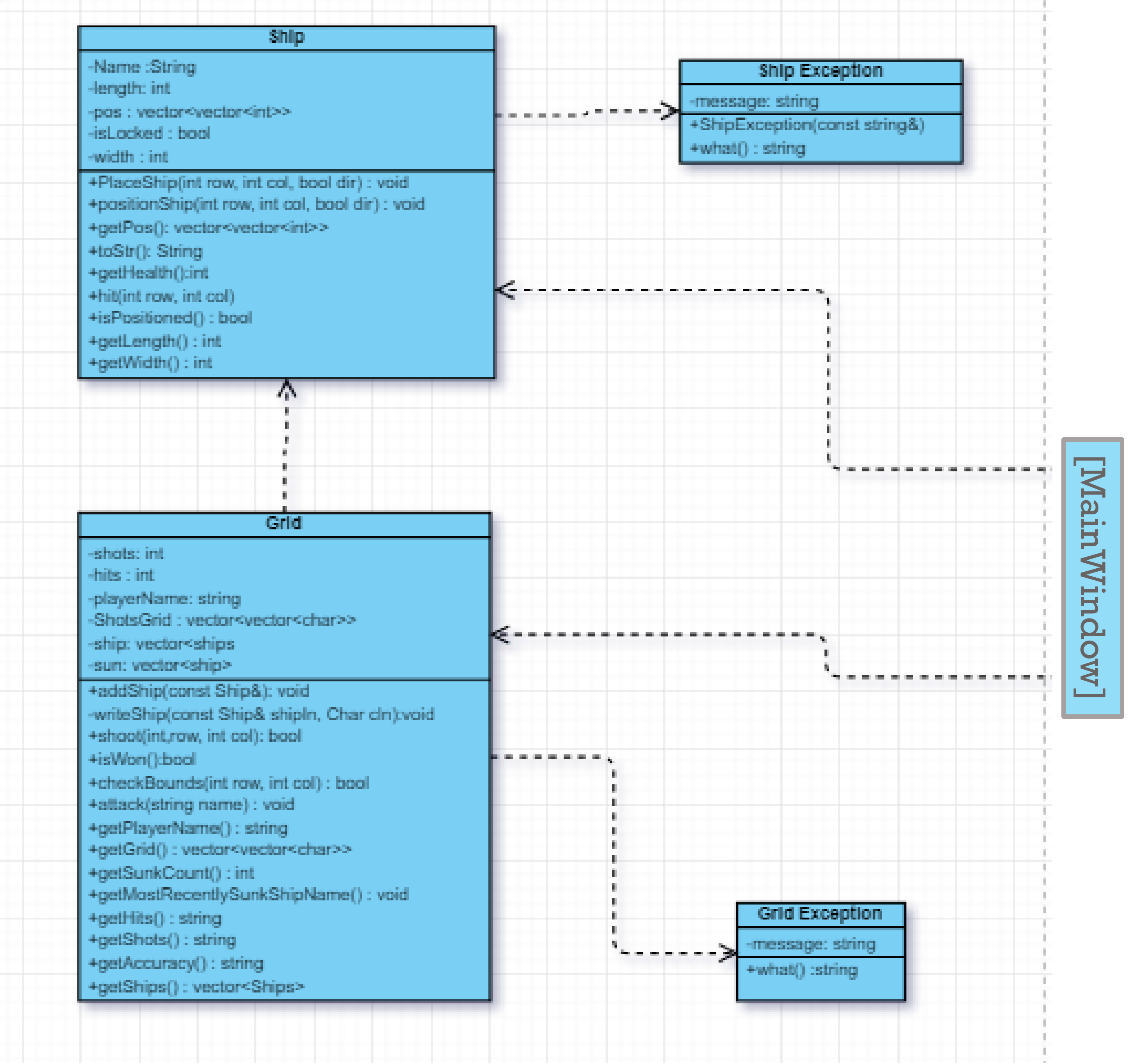
If not selected, Vertical is assumed

□ Horizontal

Turn Indicator

P1

CLASS DIAGRAM – FRONTEND



Ship

-Name :String
-length: int
-pos : vector<vector<int>>
-isLocked : bool
-width : int

+PlaceShip(int row, int col, bool dir) : void
+positionShip(int row, int col, bool dir) : void
+getPos(): vector<vector<int>>
+toStr(): String
+getHealth():int
+hit(int row, int col)
+isPositioned() : bool
+getLength() : int
+getWidth() : int

Ship Exception

-message: string

+ShipException(const string&)
+what() : string

Grid

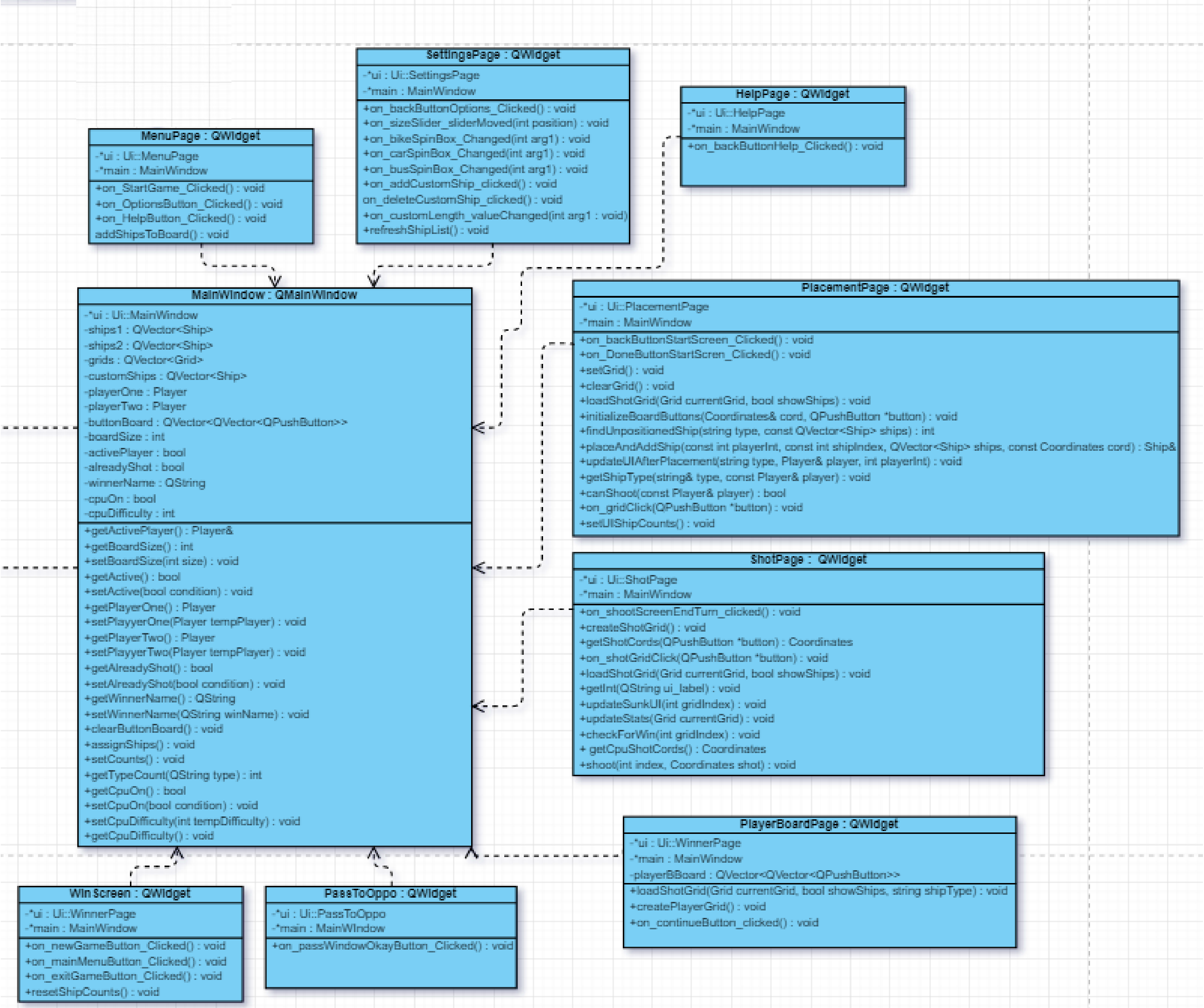
-shots: int
-hits : int
-playerName: string
-ShotsGrid : vector<vector<char>>
-ship: vector<ships
-sun: vector<ship>

+addShip(const Ship&): void
-writeShip(const Ship& shipIn, Char cIn):void
+shoot(int,row, int col): bool
+isWon():bool
+checkBounds(int row, int col) : bool
+attack(string name) : void
+getPlayerName() : string
+getGrid() : vector<vector<char>>
+getSunkCount() : int
+getMostRecentlySunkShipName() : void
+getHits() : string
+getShots() : string
+getAccuracy() : string
+getShips() : vector<Ships>

Grid Exception

-message: string

+what() :string



MenuPage : QWidget

-*ui : Ui::MenuPage
-*main : MainWindow

+on_StartGame_Clicked() : void
+on_OptionsButton_Clicked() : void
+on_HelpButton_Clicked() : void
addShipsToBoard() : void

SettingsPage : QWidget

-*ui : Ui::SettingsPage
-*main : MainWindow

+on_backButtonOptions_Clicked() : void
+on_sizeSlider_sliderMoved(int position) : void
+on_bikeSpinBox_Changed(int arg1) : void
+on_carSpinBox_Changed(int arg1) : void
+on_busSpinBox_Changed(int arg1) : void
+on_addCustomShip_clicked() : void
on_deleteCustomShip_clicked() : void
+on_customLength_valueChanged(int arg1 : void)
+refreshShipList() : void

HelpPage : QWidget

-*ui : Ui::HelpPage
-*main : MainWindow

+on_backButtonHelp_Clicked() : void

MainWindow : QMainWindow

-*ui : Ui::MainWindow
-ships1 : QVector<Ship>
-ships2 : QVector<Ship>
-grids : QVector<Grid>
-customShips : QVector<Ship>
-playerOne : Player
-playerTwo : Player
-buttonBoard : QVector<QVector<QPushButton>>
-boardSize : int
-activePlayer : bool
-alreadyShot : bool
-winnerName : QString
-cpuOn : bool
-cpuDifficulty : int

+getActivePlayer() : Player&
+getBoardSize() : int
+setBoardSize(int size) : void
+getActive() : bool
+setActive(bool condition) : void
+getPlayerOne() : Player
+setPlayyerOne(Player tempPlayer) : void
+getPlayerTwo() : Player
+setPlayyerTwo(Player tempPlayer) : void
+getAlreadyShot() : bool
+setAlreadyShot(bool condition) : void
+getWinnerName() : QString
+setWinnerName(QString winName) : void
+clearButtonBoard() : void
+assignShips() : void
+setCounts() : void
+getTypeCount(QString type) : int
+getCpuOn() : bool
+setCpuOn(bool condition) : void
+setCpuDifficulty(int tempDifficulty) : void
+getCpuDifficulty() : void

PlacementPage : QWidget

-*ui : Ui::PlacementPage
-*main : MainWindow

+on_backButtonStartScreen_Clicked() : void
+on_DoneButtonStartScren_Clicked() : void
+setGrid() : void
+clearGrid() : void
+loadShotGrid(Grid currentGrid, bool showShips) : void
+initializeBoardButtons(Coordinates& cord, QPushButton *button) : void
+findUnpositionedShip(string type, const QVector<Ship> ships) : int
+placeAndAddShip(const int playerInt, const int shipIndex, QVector<Ship> ships, const Coordinates cord) : Ship&
+updateUIAfterPlacement(string type, Player& player, int playerInt) : void
+getShipType(string& type, const Player& player) : void
+canShoot(const Player& player) : bool
+on_gridClick(QPushButton *button) : void
+setUIShipCounts() : void

ShotPage : QWidget

-*ui : Ui::ShotPage
-*main : MainWindow

+on_shootScreenEndTurn_clicked() : void
+createShotGrid() : void
+getShotCords(QPushButton *button) : Coordinates
+on_shotGridClick(QPushButton *button) : void
+loadShotGrid(Grid currentGrid, bool showShips) : void
+getInt(QString ui_label) : void
+updateSunkUI(int gridIndex) : void
+updateStats(Grid currentGrid) : void
+checkForWin(int gridIndex) : void
+ getCpuShotCords() : Coordinates
+shoot(int index, Coordinates shot) : void

PlayerBoardPage : QWidget

-*ui : Ui::WinnerPage
-*main : MainWindow
-playerBBoard : QVector<QVector<QPushButton>>

+loadShotGrid(Grid currentGrid, bool showShips, string shipType) : void
+createPlayerGrid() : void
+on_continueButton_clicked() : void

WinScreen : QWidget

-*ui : Ui::WinnerPage
-*main : MainWindow

+on_newGameButton_Clicked() : void
+on_mainMenuButton_Clicked() : void
+on_exitGameButton_Clicked() : void
+resetShipCounts() : void

PassToOppo : QWidget

-*ui : Ui::PassToOppo
-*main : MainWindow

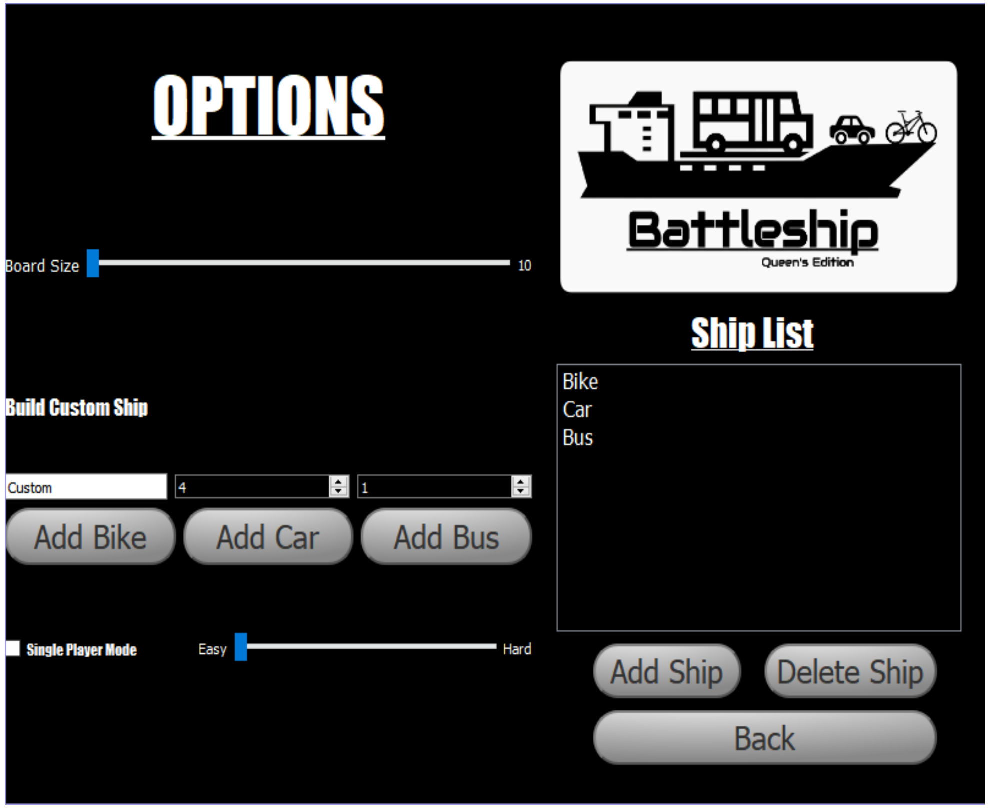
+on_passWindowOkayButton_Clicked() : void

CLASS DIAGRAM – BACKEND

SPRINT 5 – WEEK 10

Length: Nov. 16th to 23rd (one week)

- Integration of options screen
- Separation of classes in frontend
  - First iteration: one QStackedWidget class holding all game windows
  - Second iteration: each game window has an associated class and form file
- Page navigation
- Error checking for number of ships


Board Size 10

Build Custom Ship

Custom 4 1

□ Single Player Mode Easy Hard

Ship List

Bike
Car
Bus

SPRINT 6 – WEEK 11

Length: Nov. 23rd to 29th (one week)

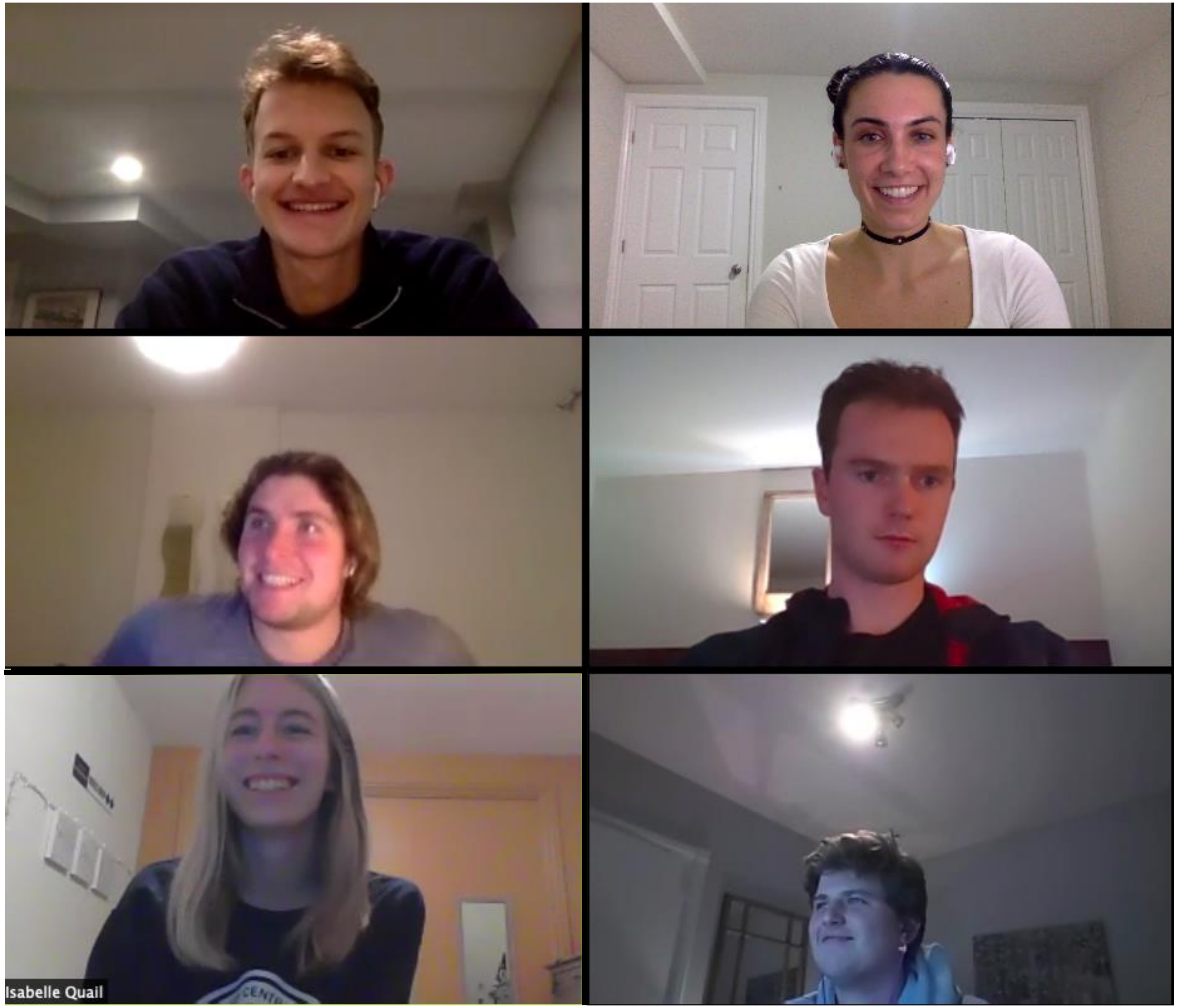
- Shot screen UI
- Debugging
  - Sunk stats not updating
  - Turn can be switched without taking shot first

- Integration of CPU
- Testing
  - Backend
  - Frontend
  - Integration

DEMO

REVISITING FUNCTIONAL & NON-FUNCTIONAL REQUIREMENTS

- **100%** of functional requirements were met
- **62.5%** of non-functional requirements were met



REFLECTION

As a team, we excelled at…

As a team, we struggled with…

If we were to do another project, we would…