Product Design

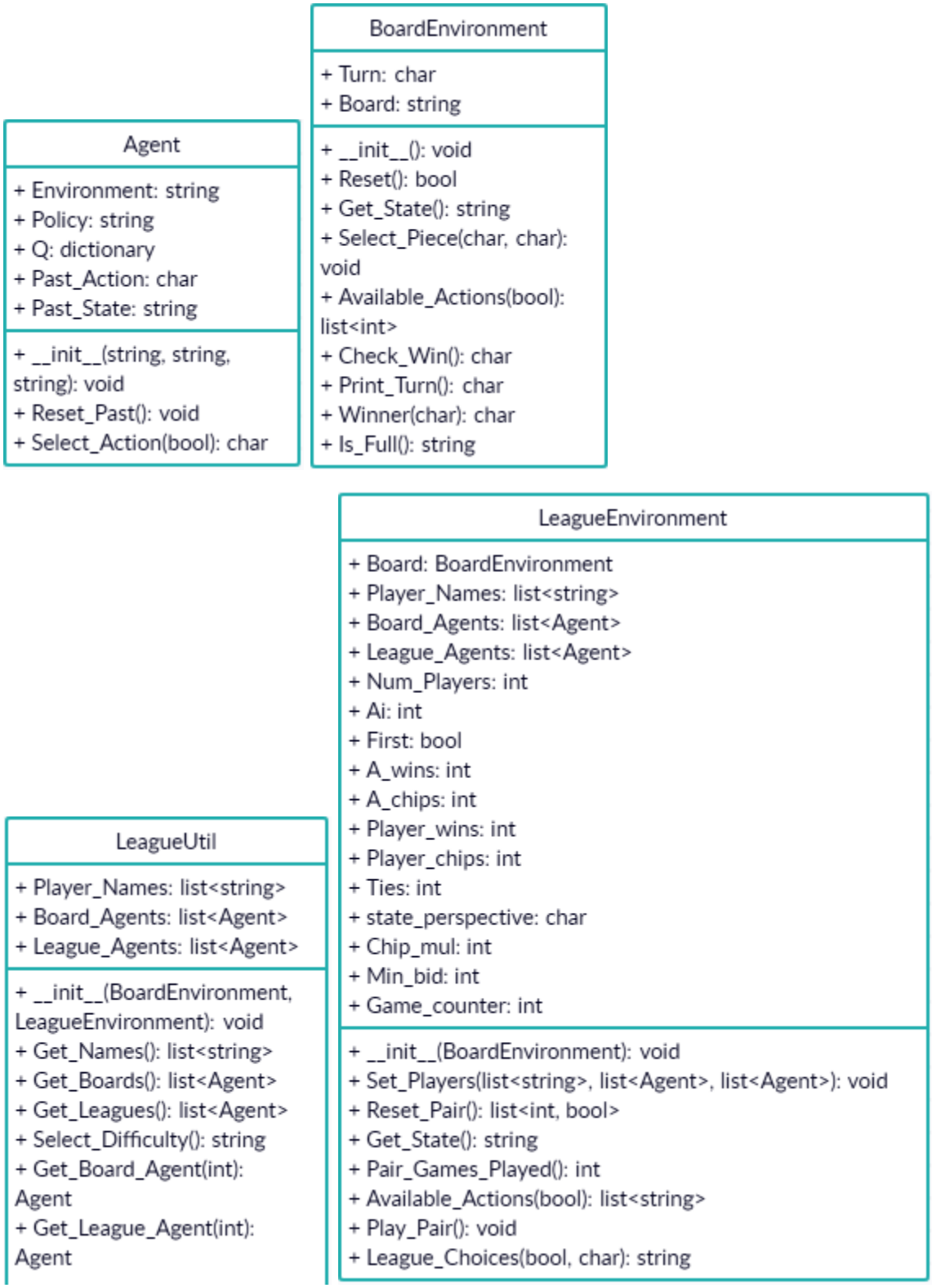
Team Name: Team WJNKCW

Brian Little, James Peters, Nikki Meyer, Kyle Petrie, Caleb Halter, Wafi Hussain

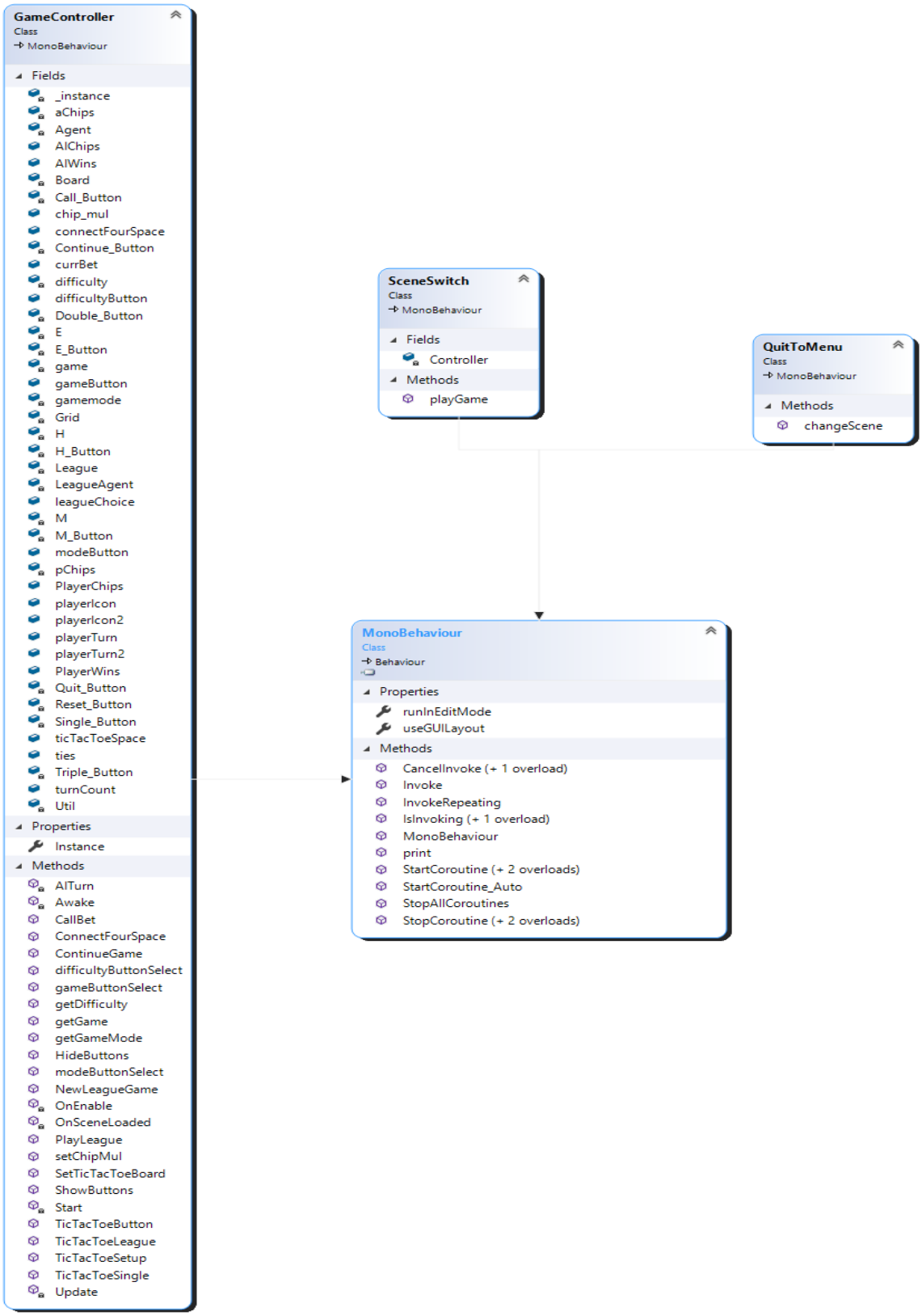
Note: Much of the design our team is conducting will be operating off of and building on top of existing designs made by a previous group and the master's students we are working with.

Revision Number	Revision Date	Summary of Changes	Authors
0.1	2/24/2021	Initial Creation of Document	Brian Little

Class Diagrams

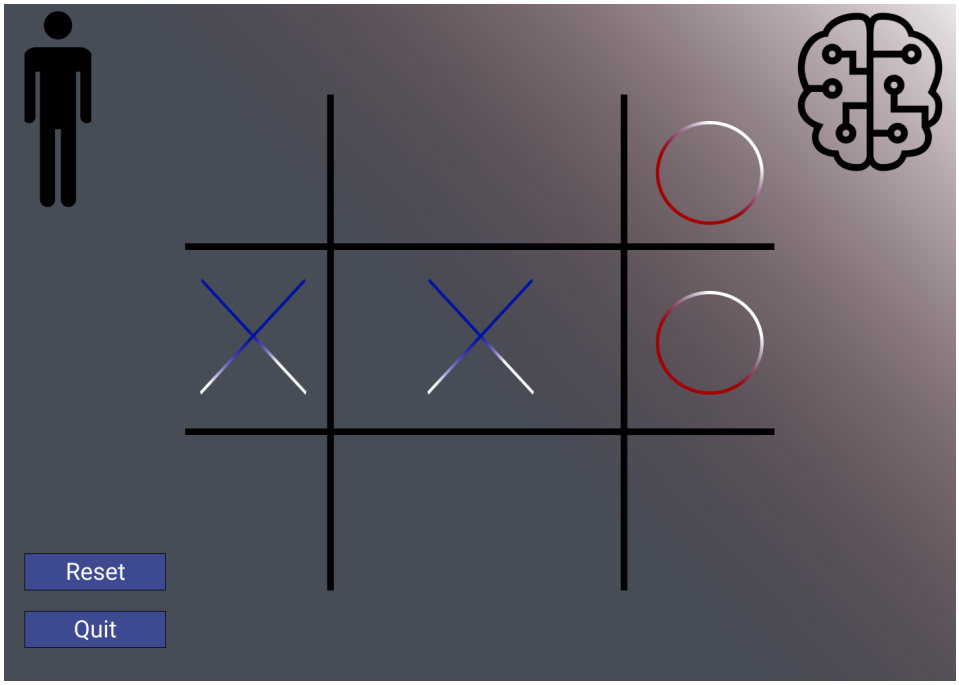


ER Diagrams


GameController
Class
MonoBehaviour
Fields
_instance
aChips
Agent
AIChips
AIWins
Board
Call_Button
chip_mul
connectFourSpace
Continue_Button
currBet
difficulty
difficultyButton
Double_Button
E
E_Button
game
gameButton
gamemode
Grid
H
H_Button
League
LeagueAgent
leagueChoice
M
M_Button
modeButton
pChips
PlayerChips
playerIcon
playerIcon2
playerTurn
playerTurn2
PlayerWins
Quit_Button
Reset_Button
Single_Button
ticTacToeSpace
ties
Triple_Button
turnCount
Util
Properties
Instance
Methods
AITurn
Awake
CallBet
ConnectFourSpace
ContinueGame
difficultyButtonSelect
gameButtonSelect
getDifficulty
getGame
getGameMode
HideButtons
modeButtonSelect
NewLeagueGame
OnEnable
OnSceneLoaded
PlayLeague
setChipMul
SetTicTacToeBoard
ShowButtons
Start
TicTacToeButton
TicTacToeLeague
TicTacToeSetup
TicTacToeSingle
Update
SceneSwitch
Class
MonoBehaviour
Fields
Controller
Methods
playGame
QuitToMenu
Class
MonoBehaviour
Methods
changeScene
MonoBehaviour
Class
Behaviour
Properties
runInEditMode
useGUILayout
Methods
CancelInvoke (+ 1 overload)
Invoke
InvokeRepeating
IsInvoking (+ 1 overload)
MonoBehaviour
print
StartCoroutine (+ 2 overloads)
StartCoroutine_Auto
StopAllCoroutines
StopCoroutine (+ 2 overloads)

User Interface Wireframes/Screenshots

Title Screen User Interface mock-up with options for user to select game type



Tic Tac Toe game board mock-up

These UI designs should be easy to adapt to a mobile screen size without losing any functionality.

Current UI for title screen from previous group that we will be improving upon.

Design Summary

The project is already in progress and a lot of work has already been done. The focus of our design and implementation will be expanding on the work that already exists.

We will be implementing Python scripts into the Unity Game engine. We will also be making changes to file pathing with MacOSX, iOS, and Android versions to deploy the project on operating systems other than Windows. Finally, we will be fixing bugs and adding onto the existing features of the game (improving visuals, improving q-tables, testing, etc.)

Revision 0.1

_____Currently, we are using the class and ER diagrams from the previous semester's group. These are the final diagrams they presented at the end of the semester and represent the current state of the project. At the moment, our plans for development are to improve and implement these existing designs. We are not currently intending to add any classes or entities.

However, after our current tasks are done, we will reassess how to improve the product. Based on the outcomes from that assessment, we may add classes/entities, and then make edits to the relevant diagrams.

Design Rationale

Visual Designs

Our visual designs are intended to be sleek, visually engaging, and easy to distinguish, with a focus on accessibility. This is because the product's intended users are students of various education levels. Our designs have to be visually engaging to a wide variety of age groups, and they also have to be easily used by those same age groups.

We went for a background gradient with a bright pop and a subdued blue-grey. The color feels metallic, which looks sleek and feels as if it visually fits a program focusing on artificial intelligence.

The current designs for the project from previous groups feel a bit hard to distinguish; the text boxes and game pieces blend in with the wood-patterned background. Additionally, the wood background fits the idea of a board game but feels as if it clashes with the technological idea of an artificial intelligence.