"Thinking" About the Next Generation of Gaming

Chris Howard[1], Jason Smith[1] and Jim Durante[1]

[1] *University of New Haven, Computer Science Students, CT, USA*

Abstract— The goal of a virtual environment is to facilitate an immersive user experience. In 2021, consumer grade virtual reality (VR) headsets have increased in functionality and demand. Our team intends to develop an innovative immersive experience that complements a VR environment and increases the overall accessibility of the virtual landscape. We intend to meet these goals by merging the immersion of a VR headset with the intuition of a brain computer interface (BCI). To validate our solution we propose an experiment to determine the feasibility of our solution in production and provide useful metrics for developers to validate their design.

Keywords— Virtual Reality, Extended Reality, Mixed Reality, Brain Computer Interface

1 Context

The gaming industry is relatively new and is still rapidly expanding in terms of technological advancements around the world. One of these innovative ideas has been the creation of an immersive virtual space through virtual and augmented reality implementations. Currently the most difficult problem in VR is implementing the most intuitive user control. Companies like Valve have been leading the way in innovative controller designs. However, our team has a vision for the future of gaming beyond traditional user controls. Using a Brain-computer interface (BCI) and VR technology there could be a way to enjoy your favorite game without having to move a muscle.

2 Problem Formulation

The research questions we are trying to answer with our prototype are, "Are pure BCI controls, traditional controls, or a mix of the two paradigms the most engaging?", “Can we give real-time insight on user engagement?”, and “Will BCI controls increase user accessibility for users who may not have access to traditional controls?”

To answer these questions, we must compare the quantitative data observed during the testing of our prototype. This includes looking at brain waves like alpha and beta waves in order to see when the brain is being engaged. Along with looking at brain waves we will also note the scores of the user and see if they correlate with the engagement the user is having with the game using the BCI model or the traditional model.

3 Relevance and potential outcomes

The minimum viable product for our project was to combine VR and an intuitive BCI enabled user interface to create a novel experience for gaming and for the future of technology. We are currently validating our hypothesis through a volunteer experiment on campus. By expanding our core concept, our product could be monetized by demonstrating that our system could be used to categorize user engagement beyond just our project. Game developers could gain quantifiable and actionable data on the level of engagement of a new product in one of many applications for our project.

4 Related Work

Quarter four of 2021 has proven to be a pivotal time for Virtual Reality (VR) technology. At least five new headsets have been released from companies such as Pimax with the 8k X (Pimax, 2021), Varjo with the XR-3 (Varjo, 2021), HTC Vive with the Vive Flow (Vive, 2021) and Sony with PS VR (Playstation VR 2021) to name a few. In addition to the headsets which have already been released there are several more plans to include Project Galea from the company Valve. “Galea is a hardware and software platform that merges next-generation biometrics with mixed reality” (Introducing: Galea, 2021). The Galea headset highlights the importance of our project because they share our same vision of a headset which combines BCI controls in a VR environment. Lastly one of the biggest names in technology recently released their vision for the future of VR infrastructure under the name Metaverse. Rebranding themselves from Facebook to Meta. “From now on, we’re going to be metaverse first, not Facebook first," CEO Mark Zuckerberg said Thursday” (D.W., 2021). The combination of these varied market shifts lets our team know that we are thinking in the right direction.

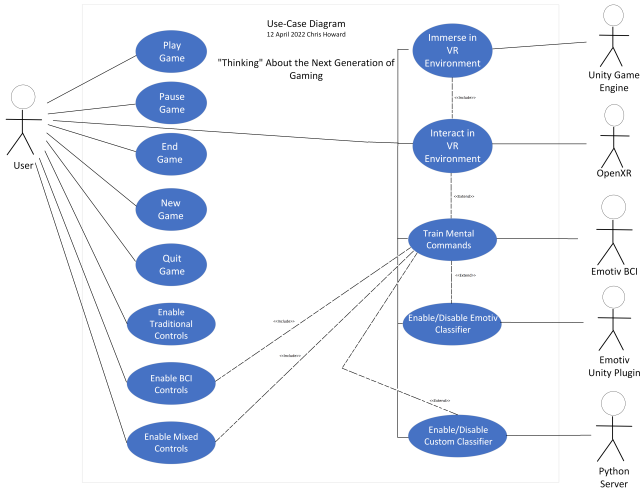
5 Software Requirement Engineering

5.1 User Needs

Our application will provide a useful tool for analyzing user immersion in VR games with BCI enabled user control. The application will have three primary types of users

to include: consumers, analysts, and executives. The consumer will need an easy-to-use interface that increases user immersion in the game. The analyst will require data to analyze their company or software objectives. The executive will need a way to visualize the data in a way that allows them to efficiently make design decisions. We intend to make a software application that will meet or exceed the needs of our three primary users.

5.2 Use Case Diagram (UML)



5.3 Process Descriptions

5.3.1 Use-case: Connect EEG Headset

Preconditions: User must have access to an Emotiv EEG headset. The headset must have been preconfigured to the bluetooth connection on the controlling computer.

Main Sequence: 1. Inspect the individual sensor felt pads and replace them if required. 2. Place saline solution or gel on the sensors of the EEG. 3. Place the EEG headset on the user's head. 4. Power on the EEG headset. 5. Verify all sensors have a good connection and adjust as necessary.

Outcome: The EEG headset is connected and operational.

5.3.2 Use-case: Train Mental Commands (Emotiv)

Preconditions: EmotivBCI must be running, and the headset must be connected.

Main Sequence: 1. Start EmotivBCI. 2. The user must verify that all the sensors are properly connected. 3. Then the user must train the baseline neutral for the day. 4. The user can then choose which mental command they intend to train. 5. The user must train a minimum of two iterations per mental command.

Outcome: A profile is either created or updated with associated training saved or discarded.

5.3.3 Use-case: Train Mental Commands (Custom)

Preconditions: The host software must be running, and the headset must be connected.

Main Sequence: 1. Start Python Server. 2. Configure data store. 3. Select "Train Mental Commands." 4. Select the mental command that the user wants to train. 5. Select the model to train.

Outcome: The trained model is prepared to be streamed to the game application.

5.3.4 Use-case: Save profile

Preconditions: A profile must have been created.

Main Sequence: 1. The user will first access the main menu. 2. The next menu they must access is the mental training menu by pressing the "Train" button. 3. The user will have two options to include: Load Profile and Train Mental Commands. If a profile has been created a third option will appear named "Save Profile." 4. The user will press the "Save Profile" button. 5. A menu will appear asking them which profile to save to. The user will select the profile they wish to write to or create a new profile.

Outcome: A profile will be created with associated training profiles

5.3.5 Use-case: Play Game

Preconditions: None.

Main Sequence: 1. The user will first access the main menu. 2. The user will then select the "Play" button.

Outcome: The game will start.

5.3.6 Use-case: Pause Game

Preconditions: Game must have been started.

Main Sequence: 1. The user must press the appropriate pause button (Secondary OpenXR Controller Button).

Outcome: The game will pause resulting in the animation stopping

5.3.7 Use-case: End Game

Preconditions: End game objective must be reached, player must be incapacitated, or the user must have quit the game.

Main Sequence: 1. An end-game function will be encountered. 2. The end game screen will appear. 3. The user will be given the option to end the game or start a new game.

Outcome: Record the user score.

5.3.8 Use-case: New Game

Preconditions: An end game event must be encountered. Such as winning the game or losing the game.

Main Sequence: 1. The end game menu will appear. 2. The user must select the "New Game" menu button.

Outcome: The player's position will be reset to the starting position. The main menu will be displayed.

5.3.9 Use-case: Quit Game

Preconditions: Pause button (Secondary OpenXR Controller Button) must be pressed.

Main Sequence: 1. The main menu will be displayed. 2. User must select the "Quit" menu button.

Outcome: The application will quit. Returning control to calling application (SteamVR or Operating System).

5.3.10 Use-case: Enable Traditional Controls

Preconditions: The options menu must be selected from the main menu.

Main Sequence: 1. Select "Enable Traditional Controls"

Outcome: The locomotion system for the OpenXR controllers will be enabled.

5.3.11 Use-case: Enable BCI Controls

Preconditions: 1. The custom or Emotiv classifier must be enabled. 2. If custom classifier is enabled. The python server must be on. 3. The options menu must be selected from the main menu.

Main Sequence: 1. Select "Enable BCI Controls"

Outcome: The locomotion system for the OpenXR controllers will be disabled. BCI control will be enabled.

5.3.12 Use-case: Enable Mixed Controls

Preconditions: 1. The custom or Emotiv classifier must be enabled. 2. If custom classifier is enabled. The python server must be on. 3. The options menu must be selected from the main menu.

Main Sequence: 1. Select "Enable Mixed Controls."

Outcome: Locomotion will be enabled via the OpenXR controllers. The jump button will be enabled via the lift mental command.

5.4 Software Architecture Style

5.4.1 Presentation-Abstraction-Control (PAC) Architecture

The Presentation-Abstraction-Control (PAC) architecture reimagines the Model-View-Controller (MVC) paradigm. This model fits the requirements for the scene control and home screen of our game. We chose to use PAC to model these parts of the project because we want an architecture that can manage human computer interaction. MVC is suitable for the user interaction of a web page. However, our system is managing multiple independent agents in a real-time interactive environment. Using this architecture will provide increased granularity and the possibility to scale in the future.

5.4.2 Component Based Architecture

The primary benefit of a Component Based Architecture is that each component is self-contained and modular. Our software system is composed of multiple components that are independent of each other. In fact, several of the components are third-party solutions. However, we still want our system to operate if the third-party system is no longer available. Therefore, we intend to build a software system with modular components in lieu of a closely coupled monolithic solution.

5.5 Architectural Goals and Constraints

5.5.1 Goals

1. Flexibility through a Modular Design and Reusable Components
2. Maintainability through Reliability and Configuration Management
3. Scalable

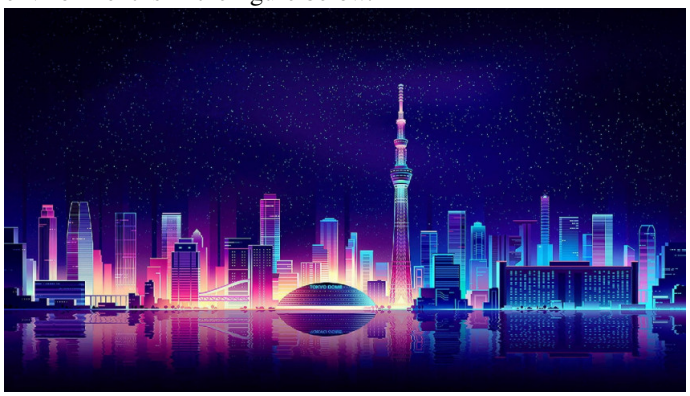
5.5.2 Constraints

1. Limited in certain situations by the Emotiv Cortex API.
2. Components (Limited to Emotiv Headsets)
3. Complexity

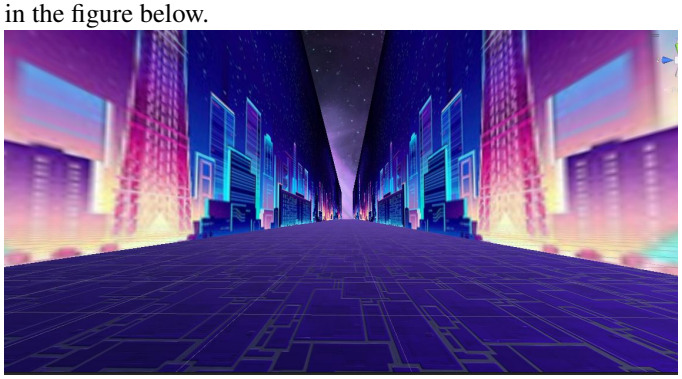
6 Data Dash VR Game Prototype

6.0.1 Game Prototype

We designed a VR game in the continuous runner genre. We drew our inspiration from a Tron like dystopian future. The game was implemented using the latest version of the Unity Engine at the time of development 2021. The interaction with the VR world was implemented using the latest version of OpenXR to facilitate a working game on multiple different VR devices. Our inspiration for the game environment is in the figure below.



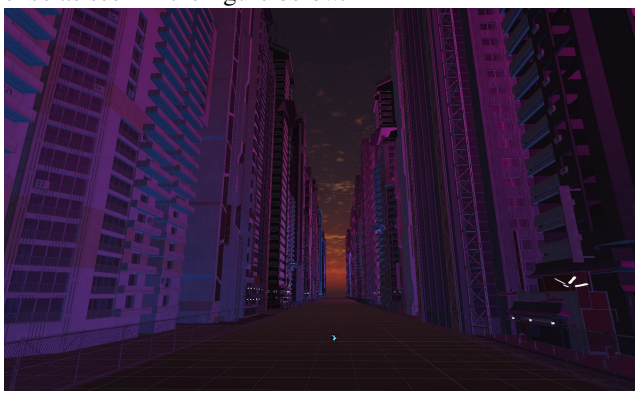
We then created a working prototype to implement the core functionality based on our original concept as depicted in the figure below.



Once the core game functionality was implemented we proceeded to work on the aesthetics of the game by implementing several different 3D models until we found some that were freely available and aesthetically appealing

as depicted in the figure below.

We then implemented lighting and audio into the game to create a more immersive environment for the intended audience as seen in the figure below.



In addition to the VR environment we also implemented a user interface (UI) via the menu design pattern. Consisting of a Main Menu, BCI menu, Options Menu, and Controller Menu. The menu was created using free sprite assets from the Unity asset store and adding animations when the user interacts with the UI. The UI was designed to compliment our art theme and is easily opened in closed via the secondary button on any controller that interfaces with OpenXR as depicted in the figures below.

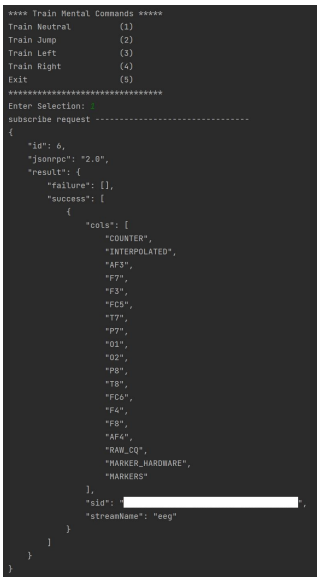
Upon game completion either through a collision with an obstacle or crossing the finish line. An end game event is triggered and the end game menu is displayed. This menu shows the final score and allows the user to start a new game or quit the application as referenced in the figure below.

6.0.2 Custom ML Solution

The purpose of our game is to test our hypothesis. Our hypothesis was that a mixture of traditional and BCI controls were going to increase user immersion in contrast to strictly BCI controls or strictly traditional hand held controls. Therefore, we implemented a brain computer interface in addition to traditional handheld controllers. However, in all paradigms the UI menu was strictly opened using the secondary button on the OpenXR controller.

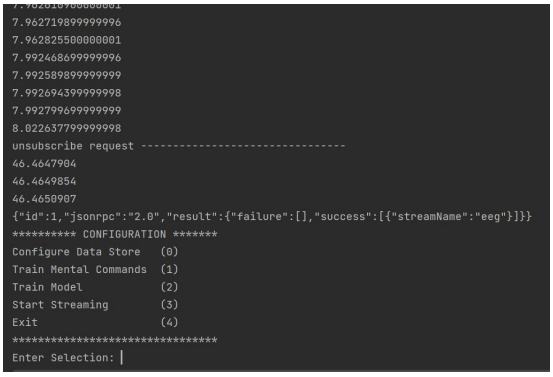
Initially there was an error with the Emotiv Unity Plugin. While the troubleshooting process was in progress on this error we implemented a custom machine learning classifier in Python. The server connects to the Emotiv Cortex API.

Once the server has connected we subscribed to the EEG or mental command stream.

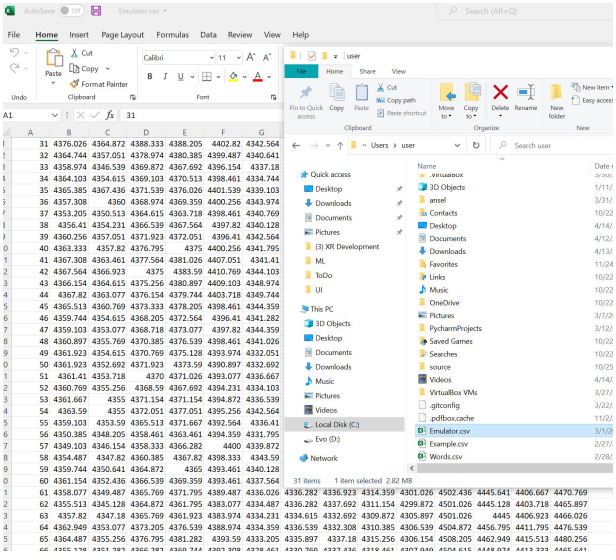


This way we could implement the Emotiv Classifier into our game or run the data through our own model.

Our custom machine learning solution can then collect the user's EEG data.

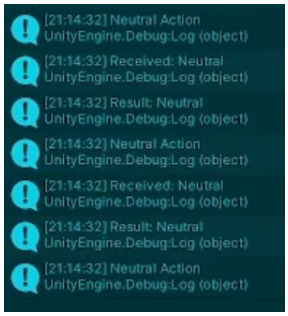


Then stores the data in a comma separated value (CSV) file. This CSV file acts as the user's primary and persistent data store (profile).



We then ingest the CSV file into one of five different models that may be selected by the user. The data is then used to train/test the selected model.

Once the model is trained we then are able to classify and stream the raw EEG data in real-time to our VR game.



7 Results

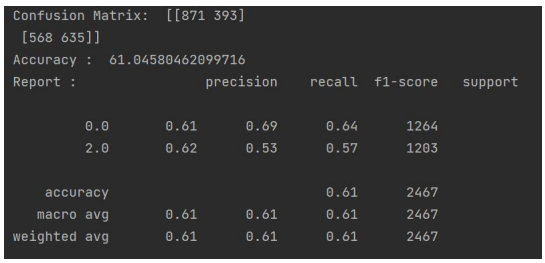
The result of our teams work is a basic prototype as demonstrative proof our concept can be accomplished. Our proof-of-concept prototype encompassed a real-time connection to an emulated Emotiv Headset and a functional Unity Project. Our prototype was implemented using the latest Emotiv Epoc X headset and EmotivBCI software.

7.1 Machine Learning Model Metrics

7.2 Neural Net Models

7.2.1 Neural Net Model 1

7.2.2 Neural Net Model 2



7.2.3 Neural Net Model 3

```
Confusion Matrix:  [[999 238]
 [268 945]]
Accuracy :  79.3469387755102
Report :
              precision    recall  f1-score   support

         0.0       0.79      0.81      0.80      1237
         2.0       0.80      0.78      0.79      1213

    accuracy                           0.79      2450
   macro avg       0.79      0.79      0.79      2450
weighted avg       0.79      0.79      0.79      2450
```

7.2.4 Decision Tree

```
self.model = DecisionTreeClassifier(criterion="gini", random_state=100, max_depth=3, min_samples_leaf=5)
```

```
Confusion Matrix:  [[835 432]
 [545 655]]
Accuracy :  60.397243615727604
Report :               precision    recall  f1-score   support

         0.0       0.61      0.66      0.63      1267
         2.0       0.60      0.55      0.57      1200

    accuracy                           0.60      2467
   macro avg       0.60      0.60      0.60      2467
weighted avg       0.60      0.60      0.60      2467
```

7.2.5 Support Vector Machine (SVM)

```
self.model = svm.SVC(gamma=0.001, C=100)  # Gamma is learning rate.
```

```
Confusion Matrix:  [[835 432]
 [545 655]]
Accuracy :  60.397243615727604
Report :               precision    recall  f1-score   support

         0.0       0.61      0.66      0.63      1267
         2.0       0.60      0.55      0.57      1200

    accuracy                           0.60      2467
   macro avg       0.60      0.60      0.60      2467
weighted avg       0.60      0.60      0.60      2467
```

8 Discussion

We implemented five different machine learning models with varying success. According to the metrics the best model is neural net model 3. However, in practice we did not get the best results. Therefore, we believe further calibration of our models or implementation is required. This could be an issue with our implementation or a lack of available data.

The decision tree, support vector machine, and neural net model 2 are tied for second place. Then the worst model was neural net model 1. Although we did not have the greatest success with our current implementation the metrics are congruent with our expectations. Therefore we believe future work could be done in this area.

For our experiment we intend to rely heavily on the Emotiv classifier. The Emotiv classifier is backed by the current leader in consumer EEG devices. In addition to their leadership they have collected a massive amount of user data that is used to fine tune their model. The lack of available data is an apparent flaw in our current iteration.

9 Conclusion and Future Work

Brain computer interfaces, machine learning, and virtual reality are all bleeding edge technologies that we have learned a significant amount about and implemented in our own project. Along the way we have ran into several obstacles with both hardware and software. To include several bugs that we played a key role in resolving with the Emotiv developers. Despite the roadblocks that we encountered we left the state of the art in brain computer interfaces at least a couple steps further than it was when we started the project.

However, there is still work that can and should be done. If awarded the opportunity or given more time we believe we could still continue working on several different aspects of our game, machine learning classifier, and hardware. The VR game could benefit from custom art work, music, and sound effects. If we had more time or a team of artists we could make significant improvements. In addition to aesthetics we could also use additional guidance from experts in the machine learning field to calibrate our models to be more effective at classifying the intent of the user. An additional aspect that could be improved would be the hardware. A custom hardware solution that integrates an EEG and VR head mounted display would be an ideal upgrade to our project.

9.1 Virtual Reality Game controlled by Motor Imagery

We created a Virtual Reality Game that is controlled directly by the brain. We used a non-intrusive electroencephalogram as the interface between the brain and the virtual reality game. The game genre that we implemented was a continuous runner action game in which the player dodges obstacles by moving left, right, and jump. They utilize the power of their brain to attempt to make it safely to the finish line.

9.2 Custom Machine Learning Solution

In tandem with the game creation we implemented a custom machine learning solution. This solution includes five machine learning models to include a Support Vector Machine (SVM), a Decision Tree, and three different Neural Nets. We trained these models using TensorFlow and integrating the solution with our game using a Python server sending data directly to Unity. We were able to test different machine learning models that was being used as a back up to the Emotiv machine learning classifier do to issues encountered early on in the development process.

9.3 Anticipated Experiment Results

Our original inquiry was whether BCI controls, traditional controls, or a mixture of the two paradigms would be more engaging. We are currently testing our hypothesis through an experiment on campus. Our team's hypothesis is that a mixture of the two control paradigms will be the favorite choice amongst the volunteer test subjects. BCI literacy is still in its infancy amongst the general population. Therefore, we expect to see greater functionality in BCI controls when "thinking" about the next generation of gaming.

10 Documentation

10.1 Surveys

"Thinking" about the next gen of gaming volunteer survey

A short survey to see if you are eligible to participate in our experiment regarding BCI devices and virtual reality.

Name *

Short answer text

email *

Short answer text

Do you have any medical conditions related to flashing lights or bright screens? (Epilepsy, seizures, etc) *

- yes
- no

By clicking this button you agree that Group 4 is not responsible for any damage or injuries that occur during this experiment. *

- yes

"Thinking" About the next gen of Gaming results

A short survey to get feedback on the gaming experience

Please give a rating on how much you enjoyed this experiment. (1 being the least enjoyment and 5 being the most enjoyment) *

1	2	3	4	5
○	○	○	○	○

Please Give a rating on How much you enjoyed the Traditional Controls (1 being the least enjoyment and 5 being the most enjoyment) *

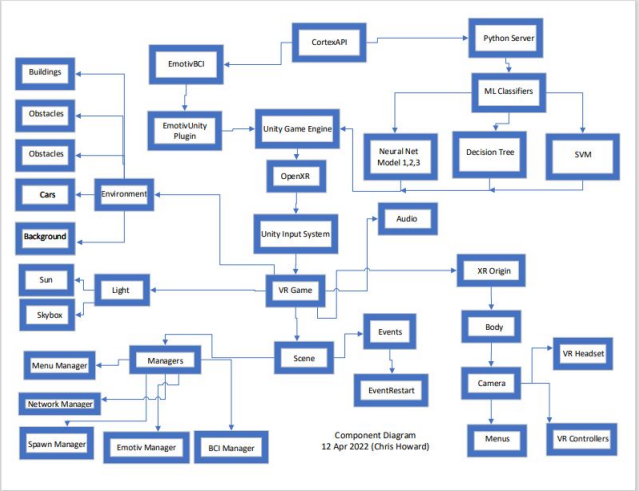
1	2	3	4	5
○	○	○	○	○

Please Give a rating on how much you enjoyed the Mixed Controls (1 being the least enjoyment and 5 being the most enjoyment) *

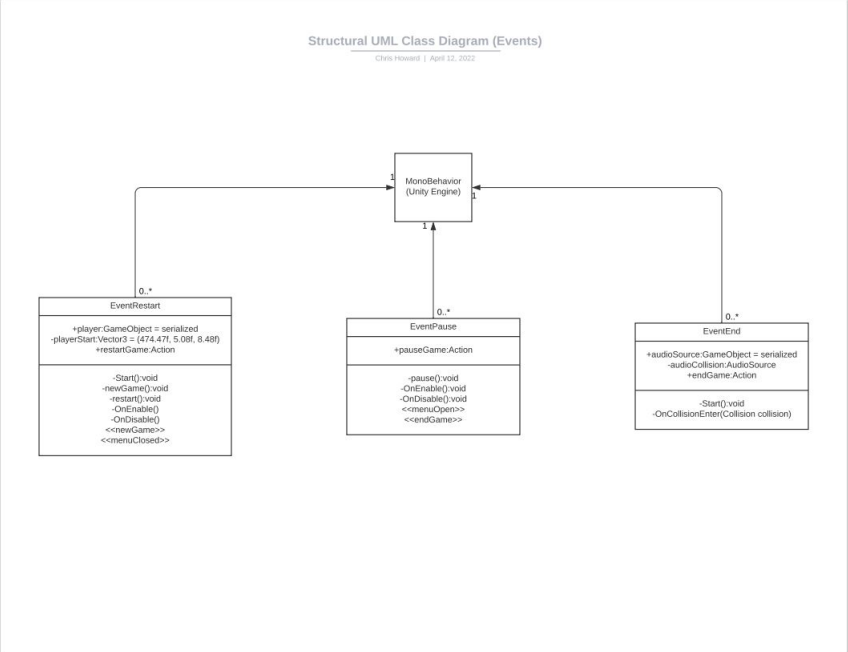
1	2	3	4	5
○	○	○	○	○

10.2 Diagrams

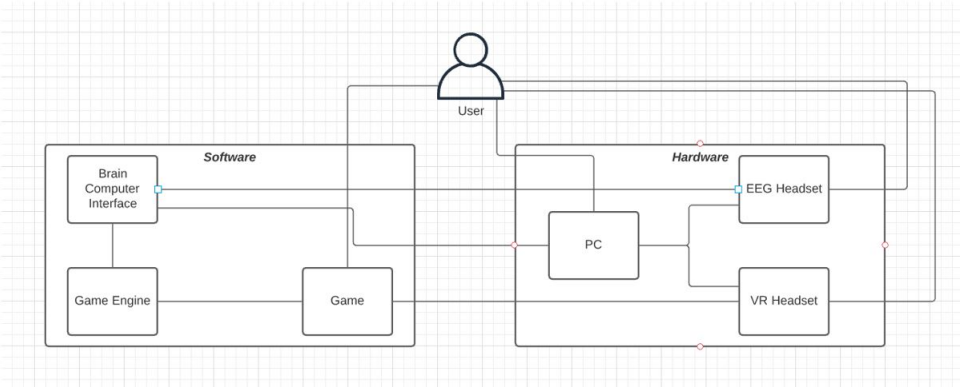
10.3 Component Diagram



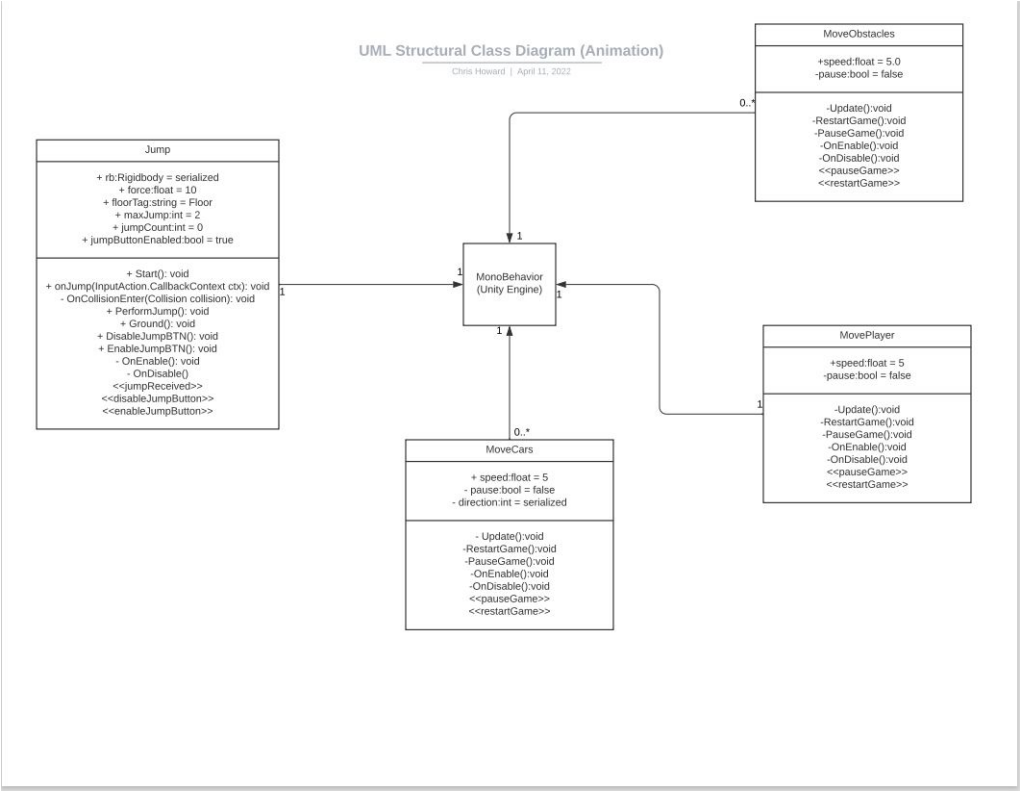
10.4 Structural UML Class Diagram (Events)



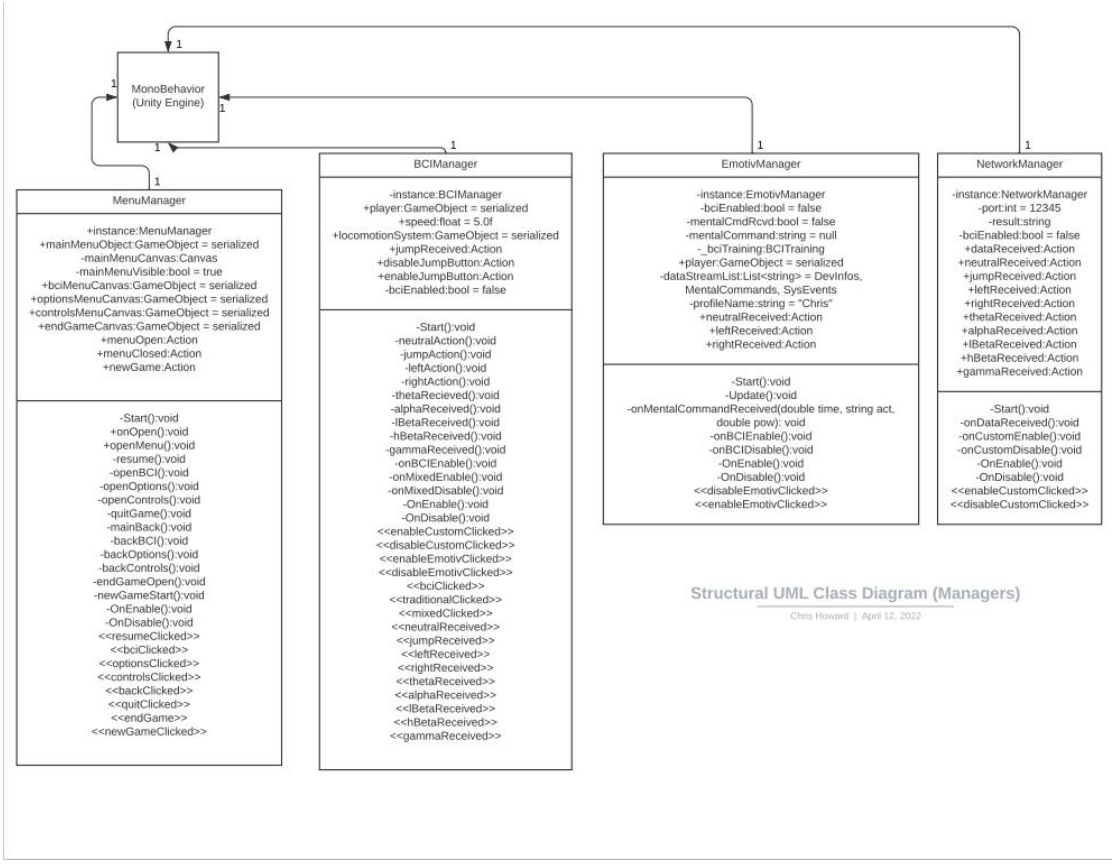
10.5 Deployment Diagram



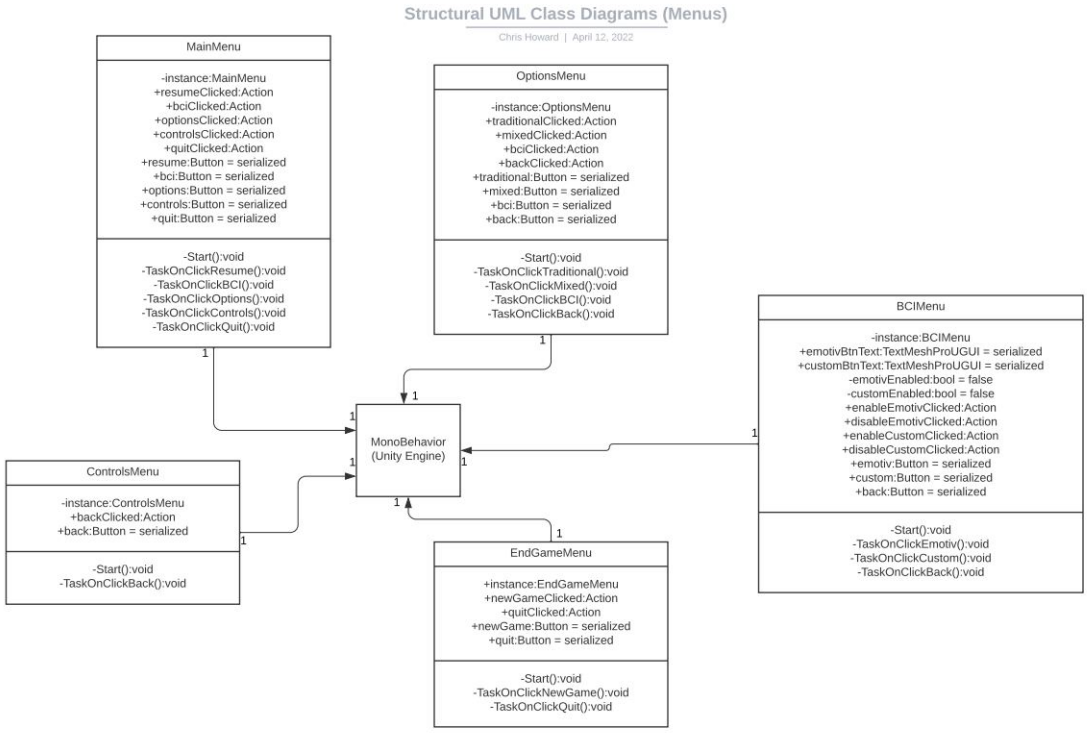
10.6 Structural UML Class Diagram (Animation)



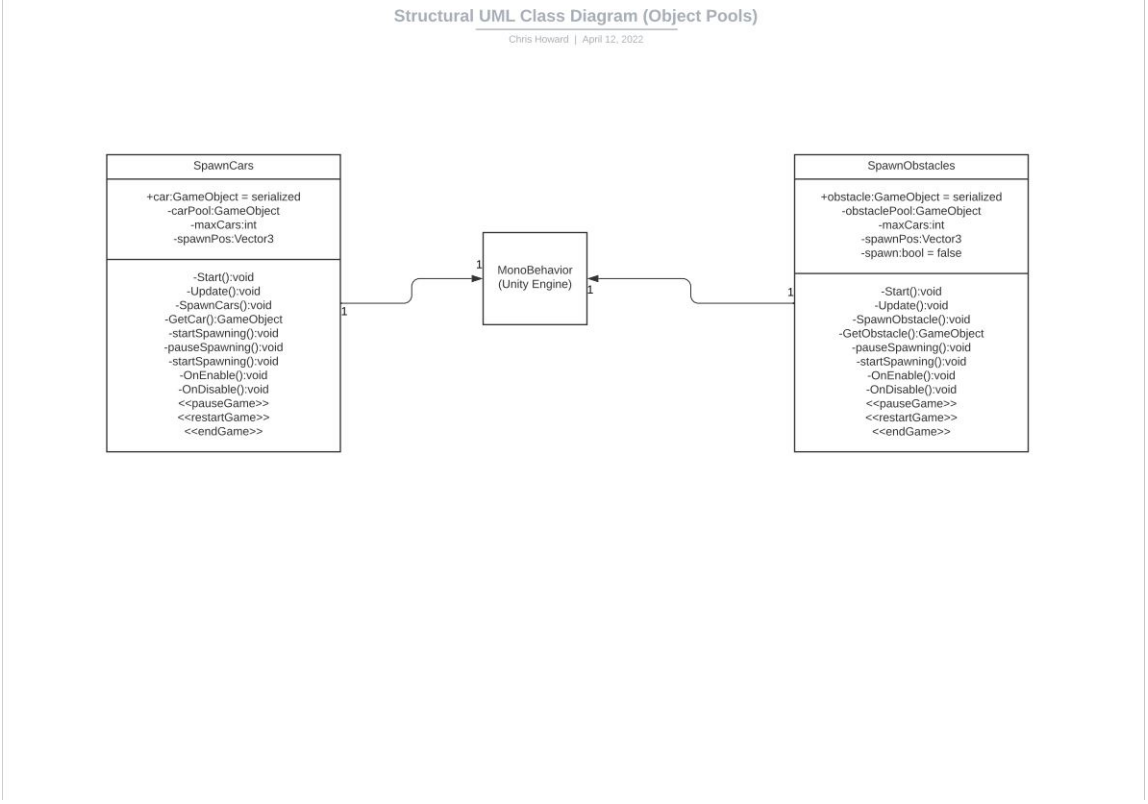
10.7 Structural UML Class Diagram (Managers)



10.8 Structural UML Class Diagram (Menus)



10.9 Structural UML Class Diagram (Object Pools)



11 References

Pimax Vision 8K X. (2021, December 10). Retrieved from https://pimax.com/product/vision-8k-x/

Varjo XR-3 - The industry's highest resolution XR headset: Varjo. (2021, November 01). Retrieved from https://varjo.com/products/xr-3/

VIVE Flow. (n.d.). Retrieved from https://www.vive.com/us/product/vive-flow/overview/

PlayStation VR: Live the game with the PS VR headset. (n.d.). Retrieved from https://www.playstation.com/en-us/ps-vr/?smcid=pdc:en-us:ps-vr:primary nav:msg-hardware:ps-vr

Posted November 19, 2. B. (2021, January 14). Introducing: Galea. Retrieved from https://openbci.com/community/introducing-galea-bci-hmd-biosensing/

(www.dw.com), D. W. (n.d.). Facebook rebrands as 'Meta' in new focus on metaverse: DW: 28.10.2021. Retrieved from https://www.dw.com/en/facebook-rebrands-as-meta-in-new-focus-on-metaverse/a-59650386