**Amy Mejia**

**Description of Solution Concepts:**

A brief description of each solution concept should be given. These concepts do not each need to fulfill the entire problem requirements; parts of the problem can be addressed by each concept and combined into a full solution concept if necessary. Conceptual solutions should be distinct from one another and should not just be minor variations of one main concept. Specific details such as dimensions, masses, brands, etc. do not need to be given at this stage. The document should “paint a picture” of the overall idea of each concept.

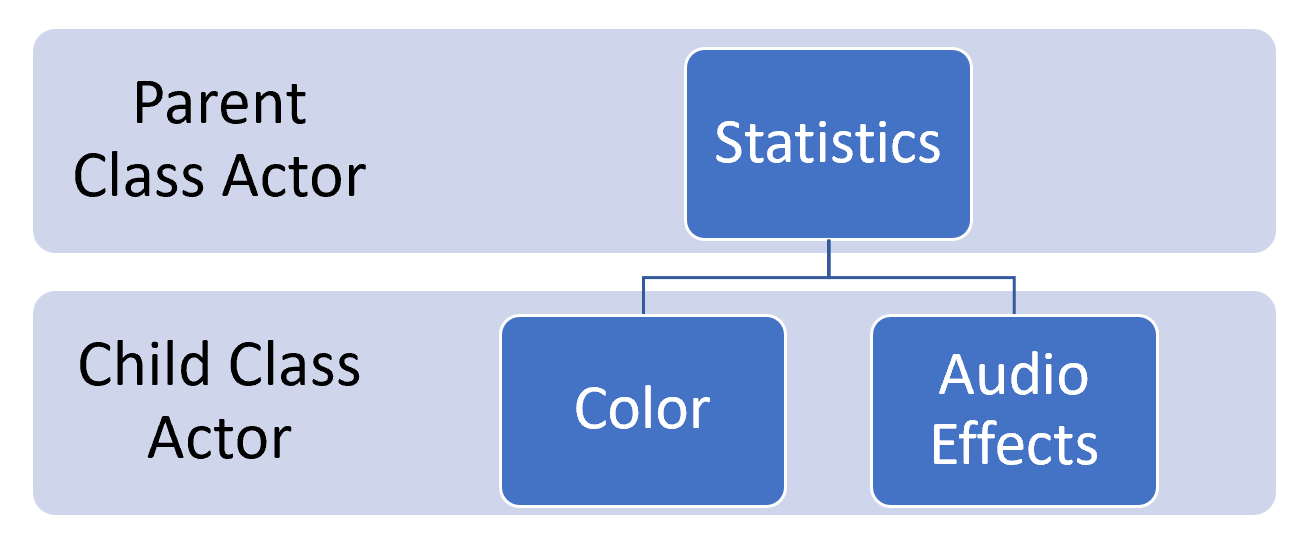
**—------------------------------------------------------------------------------------------------------------------**

The AR/VR device is at the core of the project composed of two primary constituents with the primary being modeled towards the device’s performance which includes the memory, battery life, management app, and runtime. While the secondary is customized towards achieving a higher quality player’s game for unskilled gamers. This will be realized with the addition of visual effects such as overlays and player performance statistics, object identification, and audio effects. Here the hardware and software improvements would amplify the player’s experience towards an informed and option driven run.

There have been a few algorithms that the team has considered for object identification as this would allow the unskilled player to utilize the AR/VR device for player winner outcomes. The algorithm being pursued right now in this phase of the project is YOLO. YOLO is marketed for its speed and accuracy. It appealed to the team because it utilized the C programming language. That is the one language every member knows and understands to some degree. Exploring the options suited for the AR/VR device and the management app software not using C programming language was pursued. As a team, there exists uncertainty around commitment to a singular language due to programming language barriers. After numerous meetings the team chose to stay open minded on the software languages and packages or potential plug-ins.

Over the Summer one of the team members had a dry run on the plan of execution for a particular chosen software and there were issues with language translation between different interfaces. Therefore, the teams have chosen to remain open and try numerous software models before moving forward to see which yields the best result and overall device interface.

The original sponsor of this project wanted to push the limits of the device and venture into hardcoding new technology specs for the AR/VR device. To balance the technical performance of the device, audio and visual effects were decided upon for implementation. For visual effects object detection, overlays, and player statistics throughout the game will be rendered. The player statistics will be using an algorithm focused on learning the player’s chance of winning by counting cards of the game. The object detection was discussed in the section above concerning the YOLO algorithm. Concerning the overlays, originally the discussion included overlays that were more player engaging and fun using color scheme for next move options that included the statistic for winning. Other overlays included a box encapsule for the object being identified. Unity game engine will be used for most of the overlays and audio effects as class actors, game actors, and child actors of classes (e.i.) sound can be a child actor of the colored statistics being displayed. Individual sounds and color will be components of the parent class actor statistics. Visual scripting will be the first approach to explore audio and visual effects. Hardcoding in the engine will require language research on macros and language specifiers which will take longer and may face more interface backlash. If the algorithms interfacing will require hardcoding them that route will be taken otherwise live scripting in Unity Game Engine will be the first approach.

****