REVEREX: DX Vitalist

COSC 3P99 - Final Technical Report

Supervisor: Prof. Renata Dividino

Joshua Varghese
Department of Computer Science
Brock University
St. Catharines, Ontario
ln20fb@brocku.ca | 7123870

I. Overview

Reverex was first conceptualized in September 2023 and has since evolved into an award-winning game developed by a talented team. During the development of Reverex: DX, the focus was on overseeing project progression and coordinating efforts among programmers to implement, enhance, and optimize the game for Steam publication. Initially, there were plans to create a networking serialization system for Steam network synchronization. However, priorities shifted to address more immediate tasks, including rewriting the Vitalist player code for improved efficiency across various minigames. This adjustment facilitated a smoother implementation of networking capabilities later in production, handled concurrently by another team member.

II. Introduction

Roles and Gameplay

Reverex: DX operates as a two player experience where two players control two sides of a TV screen. The left most side is controlled by the “Navigator” player and the right most side is controlled by the “Vitalist” player. The Navigator is responsible for completing a series of parkour coupled with changing gravity mechanics through a dream world. The Vitalist player is responsible for playing minigames to increase the Navigator’s potential to complete the parkour as quickly as they can.

Player Interactions

The Navigator and Vitalist are meant to “push” and “pull” on each other during gameplay, where each can affect the other positively and negatively. The Navigator’s primary focus is to complete the game with as little interruption as possible by maintaining a steady heart rate, displayed at the bottom of their screen. If their BPM surpasses 200 they will suffer a heart attack and need a defibrillation. BPM of the Navigator increases the more they move and “exert” themselves, with no method to circumvent this without help from the Vitalist. The Navigator must also be cautious not to fall into the void of the game while platforming to save time and not require extra help from the Vitalist.

III. System Design

The Vitalist System

The Vitalist system [1] entails nine total minigames that can be played from a

small defined area to the right-hand side of the game screen. These minigames affect how the Navigator on the left and side of the screen plays the game. Each minigame is categorized into three variants, selectables, pop-ups, and overridables.

Selectables

Selectables are minigames that influence the Navigator positively, that can be played at the Vitalist player's discretion by selecting from a list of UI buttons that persist on the right-hand screen by default [4]. All selectables have a set time limit of ten seconds to complete, which otherwise will end the minigame without providing its beneficial effect. Each effect also lasts fifteen seconds after completing a minigame successfully, which locks its corresponding minigame counterpart from being played until the effect's time elapses. These minigames are labelled as adrenaline, electrolytes, and antibiotics which increase the amount of speed, jump height, and heart stability the Navigator has respectively.

- **Adrenaline Minigame** [7]: Inspired by *Flappy Bird* [14], players control a flying pill by tapping a button to jump, avoiding pill bottles, and guiding it into a mouth at the end to signify administering adrenaline.
- **Electrolytes Minigam**e [9]: Players manage a rotating can that tilts downward to pour a drink into a person's mouth. The objective is to tap a button to tilt the can upward and pour steadily.
- **Antibiotics Minigame** [8]: Based on *Pop The Lock* [15], players time button presses as a spinner overlaps germs. Each successful hit destroys a germ and reverses the spinner's direction.

Each one of these minigames will improve the Navigator's abilities during play and act as optional tasks the Vitalist can perform while idle.

Pop-ups

Pop-ups are minigames that influence the Navigator negatively. They have no time limit to complete and persist until they are completed unless interrupted by an overridable. It is important to note that pop-ups immediately grant the negative effect to the Navigator, which is only removed when the minigame is complete or interrupted. These minigames are labelled as exhaustion, fracture, iv, and eye drops which decrease the amount of speed, jump height, heart stability, and visual clarity the Navigator has respectively.

- **Exhaustion Minigame** [10]: Similar to *Subway Surfers* [16], players control a runner navigating three lanes to avoid scrolling obstacles, such as beds, until reaching the finish line.
- **Fracture Minigame** [12]: Inspired by classic 2D platformers like *Doodle Jump* [18], players guide a character upward by jumping between platforms to reach a flag at the top.
- **IV Minigame** [13]: Resembling the precision-based gameplay of *Operation* [19], players steer a

free-falling needle to hit a marked target on an arm.

- **Eye Drops Minigame** [11]: Modelled after *Cookie Clicker* [17], players rapidly click a button to fill a tube with water until a droplet forms and falls onto an eyeball.

Timers and Queue System

Pop-ups operate on a continuous timer that ticks for fifteen seconds when a popup or overridable isn't currently playing out. When this timer hits zero, if the Vitalist player is currently in a selectable minigame, it will play immediately after their current game is over regardless if it was completed successfully or not. If the Vitalist player is idling, it will immediately take over their screen and force them to play the popup until it is complete or interrupted by an overridable. Pop-ups are also directly tied to the minigame queue system, which controls what popup to prompt the player when the popup timer's time elapses.

The minigame queue [2] systems holds references to each minigame and will push minigames to the end of the queue if they have been played already or if the selectable variant of a popup has been played and completed successfully. This ensures granting a positive effect to the Navigator is not immediately followed up by its corresponding negative effect, and acts as a further reward for completing selectable games successfully.

Overrideables

Overridables are minigames that come as a direct result of the Navigator misplaying or not being fully attentive to their game. They can interrupt both selectable and pop-up minigames immediately without having to account for timers or effects. Overridables also pause the pop-up timer mentioned prior. The main purpose of overridable minigames is to "reset" the Navigator if they've reached one of two states. These states occur when they have either fallen out of the world and are in need of a reset to a past position, or they have exerted themselves past 200 BPM and need to have their vitals reset. Overridables will persist until they have been completed and have no time limit. These minigames are labelled as rewind, and defibrillation.

- **Rewind Minigame** [5]: The player must rotate their left joystick (or mouse) counter-clockwise for two seconds before the minigame completes.
- **Defibrillation Minigame** [6]: The player must shuffle their joysticks up and down (or rotate their mouse) to fill a progress meter, after which they must press down on two buttons to complete the minigame. Defibrillation will also cause every negative effect to appear on the Navigator at once until it is completed.

Overridable Interactions

It is important to note that the rewind minigame stands apart from the defibrillation minigame in the sense that rewind can override the defibrillation minigame and not vice versa. They also act differently under different circumstances during gameplay. If a selectable minigame is

currently at play, both rewind and defibrillation can override the current minigame and upon either's completion will return the player to the default selectable menu. However, if a pop-up minigame is currently at play, rewind and defibrillation will act differently from each other. If defibrillation triggers during a pop-up, it will automatically remove any and all effects currently on the Navigator, regardless if it is positive or negative, before giving the Navigator every negative effect. After it's completion, the Navigator returns to normal and the Vitalist screen returns to the default selectable menu. If rewind triggers during a pop-up it will remove the effect of the pop-up and save a reference to it and after rewind completes, it will automatically play the prior pop-up and not return to the default selectable menu. If a rewind triggers during a defibrillation minigame is playing, the same rules apply, except it will not remove the negative effects from the Navigator and just play the defibrillation minigame after rewind is completed.

IV. Overarching System

The Vitalist system operates as a state-machine [3] to cleanly define when minigames should be playing and to easily switch in and out of states at any given time. It also allows for specific data to be centralized in one particular script so that any state can reference relevant persisting information. One such example is the difficulty levels of each minigame. Each minigame except for overridables were designed to be scalable in difficulty and increase every in difficulty after every completion. After reaching the maximum level of difficulty and upon completion of a minigame, it will randomize the difficulty level between a set range to circumvent minigames feeling repetitive.

Tutorial Mode

The Vitalist also has a variable called "Tutorial Mode" which puts limitations on the Vitalist during the tutorial of the game. When loading into the game and choosing not to skip the tutorial, the Vitalist is by default set to "Tutorial Mode" where certain abilities are restricted. The electrolytes and antibiotics minigames are both locked to the player and are unlocked as the Navigator progresses through their part of the tutorial. The Vitalist also has their pop-up timer turned off so that no pop-ups interrupt game flow like they normally do, and the Navigator's BPM does not increment. This was done to slow down the gameplay for first time players so they could get a better grasp of how the Vitalist works and gradually grow to understand how to play.

V. Mistakes

When first approaching Reverex: DX and the new systems I had planned for it, I had greatly underestimated how much optimization would be needed. Certain systems like minigame timing and the current queue system were not written in a neat way, and it was only later into development that I realized they needed to be redone to run more efficiently. So, I took it upon myself to make the pop-up timer and selectable minigame timer run as coroutines, separate from the main thread. This way, timing minigames would not take away CPU time dedicated to the game itself. I had also

realized I could separate my queue system into another script entirely to work outside the main Vitalist code. However, I had failed to account for the static references I was once using and in doing, so I had created persisting references that existed through loads of the game from prior games. This also affected how the tutorial triggers in the level to unlock and play selectables were referencing the minigames statically, and in the end simply switching the state machine reference to be static instead of the state machine itself fixed the issue, allowing everything to reset properly.

VI. Conclusion

The Vitalist system a complex web of smaller subsystems and entire games all running together in harmony. The core state machine that drives the Vitalist made its recreation a lot more complex when porting from the original Reverex, but it made a huge difference for optimizing the code as well. This became especially apparent when it came time to sync all the necessary components of the Vitalist over a peer-to-peer network, where changing states could easily be done through a networked function call. The Vitalist system was also meant to communicate other big systems in the game as well, such as the Navigator system and Stats Manager system, which is responsible for keeping track of the positive and negative effects currently at play in the game as well as their appropriate visual effects.

I wanted to write a system that could be reliable and robust and able to account for any scenario that could play out in game. Overridable minigames in combination with ongoing timers made this very difficult in this regard, but were necessary to provide instant feedback to the Vitalist to save the Navigator and keep gameplay progressing as quickly as possible. Ultimately, it has always been clear that making a secondary player feel just as engaging and meaningful as the main player is far more important than focusing solely on the main player.

Citations

[1] LegendaryChibi, “Vitalist/Vitalist.cs,” GitHub, 2025. Available: https://github.com/LegendaryChibi/Reverex-DX-Vitalist/blob/main/Vitalist/Vitalist.cs. [Accessed: Jan. 4, 2025].

[2] LegendaryChibi, “Vitalist/MinigameQueue.cs,” GitHub, 2025. Available: https://github.com/LegendaryChibi/Reverex-DX-Vitalist/blob/main/Vitalist/MinigameQueue.cs. [Accessed: Jan. 4, 2025].

[3] LegendaryChibi, “Vitalist/Minigames/MinigameStateMachine.cs,” GitHub, 2025. Available: https://github.com/LegendaryChibi/Reverex-DX-Vitalist/blob/main/Vitalist/Minigames/MinigameStateMachine.cs. [Accessed: Jan. 4, 2025].

[4] LegendaryChibi, “Vitalist/Minigames/States/Selectables/MinigameMenuState.cs,” GitHub, 2025. Available: https://github.com/LegendaryChibi/Reverex-DX-Vitalist/blob/main/Vitalist/Minigames/States/Selectables/MinigameMenuState.cs. [Accessed: Jan. 4, 2025].

[5] LegendaryChibi, “Vitalist/Minigames/States/Overrideables/MinigameRewindState.cs,” GitHub, 2025. Available: https://github.com/LegendaryChibi/Reverex-DX-Vitalist/blob/main/Vitalist/Minigames/States/Overrideables/MinigameRewindState.cs. [Accessed: Jan. 4, 2025].

[6] LegendaryChibi, “Vitalist/Minigames/States/Overrideables/MinigameDefibState.cs,” GitHub, 2025. Available: https://github.com/LegendaryChibi/Reverex-DX-Vitalist/blob/main/Vitalist/Minigames/States/Overrideables/MinigameDefibState.cs. [Accessed: Jan. 4, 2025].

[7] LegendaryChibi, “Vitalist/Minigames/States/Selectables/MinigameAdrenalineState.cs,” GitHub, 2025. Available: https://github.com/LegendaryChibi/Reverex-DX-Vitalist/blob/main/Vitalist/Minigames/States/Selectables/MinigameAdrenalineState.cs. [Accessed: Jan. 4, 2025].

[8] LegendaryChibi, “Vitalist/Minigames/States/Selectables/MinigameAntibioticsState.cs,” GitHub, 2025. Available: https://github.com/LegendaryChibi/Reverex-DX-Vitalist/blob/main/Vitalist/Minigames/States/Selectables/MinigameAntibioticsState.cs. [Accessed: Jan. 4, 2025].

[9] LegendaryChibi, “Vitalist/Minigames/States/Selectables/MinigameElectrolytesState.cs,” GitHub, 2025. Available: https://github.com/LegendaryChibi/Reverex-DX-Vitalist/blob/main/Vitalist/Minigames/

States/Selectables/MinigameElectrolytesState.cs. [Accessed: Jan. 4, 2025].

[10] LegendaryChibi, "Vitalist/Minigames/States/Popups/MinigameExhaustionState.cs," GitHub, 2025. Available: https://github.com/LegendaryChibi/Reverex-DX-Vitalist/blob/main/Vitalist/Minigames/States/Popups/MinigameExhaustionState.cs. [Accessed: Jan. 4, 2025].

[11] LegendaryChibi, "Vitalist/Minigames/States/Popups/MinigameEyedropsState.cs," GitHub, 2025. Available: https://github.com/LegendaryChibi/Reverex-DX-Vitalist/blob/main/Vitalist/Minigames/States/Popups/MinigameEyedropsState.cs. [Accessed: Jan. 4, 2025].

[12] LegendaryChibi, "Vitalist/Minigames/States/Popups/MinigameFractureState.cs," GitHub, 2025. Available: https://github.com/LegendaryChibi/Reverex-DX-Vitalist/blob/main/Vitalist/Minigames/States/Popups/MinigameFractureState.cs. [Accessed: Jan. 4, 2025].

[13] LegendaryChibi, "Vitalist/Minigames/States/Popups/MinigameIVState.cs," GitHub, 2025. Available: https://github.com/LegendaryChibi/Reverex-DX-Vitalist/blob/main/Vitalist/Minigames/States/Popups/MinigameIVState.cs. [Accessed: Jan. 4, 2025].

[14] "Flappy Bird," Wikipedia, 2025. Available: https://en.wikipedia.org/wiki/Flappy_Bird. [Accessed: Jan. 4, 2025].

[15] "Pop the Lock," Bay Tek Entertainment, 2025. Available: https://www.baytekent.com/pop-the-lock/. [Accessed: Jan. 4, 2025].

[16] "Subway Surfers," Wikipedia, 2025. Available: https://en.wikipedia.org/wiki/Subway_Surfers. [Accessed: Jan. 4, 2025].

[17] "Cookie Clicker," Wikipedia, 2025. Available: https://en.wikipedia.org/wiki/Cookie_Clicker. [Accessed: Jan. 4, 2025].

[18] "Doodle Jump," Wikipedia, 2025. Available: https://en.wikipedia.org/wiki/Doodle_Jump. [Accessed: Jan. 4, 2025].

[19] "Operation (game)," Wikipedia, 2025. Available: https://en.wikipedia.org/wiki/Operation_(game). [Accessed: Jan. 4, 2025].