

Adam Guo  
Daniel Lee  
Janice Lee

### Unit 3 Writeup Final

For our group's final unit project, we completed a single 3D game hitting 5 feature components. Given the difficulty of designing a 3D game as well as the time frame given to complete the entire unit, we tried our best to implement all 5 features cohesively, focusing more on game engine rather than game design. We created a game called "Hole-in-the-Wall", with the intent of mimicking the popular television game show, where the player tries to fit themselves into a literal hole in a wall coming towards them. Our game is nowhere near as complex as the actual game show, and instead is the "foundational" version of the game. In "Hole-in-the-Wall", the player controls a box using WASD/space to move a box and jump it into a hole in a wall coming at them. Successfully jumping through the hole adds to the score count while jumping into the wall causes the wall to break into pieces.

For our first component, we decided to implement menus/modal UI. Because we had already written code that constructed a menu/modal UI for our 2D games, we decided to reuse the concept of game states to implement this feature for our 3D game. In our start state, we essentially have 2 boxes, where the user is allowed to move right or left to collide with one of the boxes, thus either starting the game by proceeding to the game state (left box), or (right box) showing our scores. In our end state, the user is presented with the same set of options, thus restarting the game or thus fulfilling our menu/modal UI feature (1). The second feature we implemented was save/load progress. We completed this by again reusing code from our unit 2 projects, which in our case was simply keeping track of the highest score recorded (2). The third feature we tackled was spatial audio, footfalls, and 3D audio sources, tied into gameplay. Because our game by nature does not deal with character movement, but instead a box moving left/right and jumping, we handled spatial audio and 3-D audio sources. As the wall gets closer, the sound of an oncoming train physically feels as if it is approaching the player with a volume increase. Upon player collision with the wall, the wall makes one of two sounds, a wood wall breaking (brown wall) or glass shattering (glass wall). Also depending on where the player position is, the audio of the "breaking effect" is also 3D, meaning different breaks on different parts of the wall have directional audio (3). The fourth feature we hit was destructible/modifiable terrain, which was achieved by having the solid wall break into multiple cubes and flying all over the place when the player box makes contact with the wall and not the hole, completed by having restitution change velocity instead of position (4). The fifth feature we hit was collision beyond AABBs. Besides the obvious collision that occurs when the player box jumps into the wall we showed that our collision handles non-axis aligned boxes, so when the boxes rotate as the wall gets destroyed, collision is still maintained, as the user box cannot force itself into one of the broken boxes (5). Just as a backup plan, we would also like to suggest that we reused substantial code from our 2D engine in our menu/modal UI/save/load features in the case that our other features do not suffice as the 5 for maximum credit (6).

Our group had outstanding teamwork and communication, the only problem we faced being the lack of time to really create a game that we are happy with. We made sure not to overwork ourselves, but also to try our best to complete the final project, and having the three of

us on the same page/mentality at all times definitely was a plus (albeit there were two seniors in our group). Being allowed to use the game engine code provided was immensely helpful, and definitely made the project seem slightly more manageable given the time frame. However, it was still a daunting task nonetheless, as all three of us felt that there was simply too much to learn but not enough time to apply. 3D games really proved to be different from 2D games, which was our largest hurdle to overcome. We all played our part in contributing code to the project and the work was extremely well balanced between us. We worked synchronously through live share on VS code as well as asynchronously by pushing code onto Github. Overall, our group's decision to create a simple game which hit all 5 required feature points alongside each individual's tenacity to complete the game and not give up is what led to our ultimate success. We would not hesitate to work with one another again and really experienced how (however cliché this may sound) the power of teamwork can push each of us to work harder. There were no "poor" aspects of our teamwork other than the lack of time we had to really meet up with each other and work on the project.

From our demo day and playtesting our other classmate's games, we were definitely proud of how far our project had come. The most interesting aspect of playtesting was seeing how other groups interpreted what a sufficient 3D game would be and how they integrated their features together. Honestly, it was amazing to really experience the variety of game logic that can exist. A major issue we found with both our game and other group's games was (similar to unit 2 games) the lack of clear instructions. While the controls were not difficult to intuitively figure out, someone with no experience whatsoever playing games would definitely have a difficult time understanding how the game works. Another change that we would like to implement in our own group's game is having more shapes and a more elaborate wall. While others' games were as simple as ours, seeing certain games such as monkey ball really pushed our own desire to go above what we had already done. We also noticed some minor aspects of the game that we left unchanged such as camera movement detracted from the overall game experience. Our camera rotates upon mouse movement, but didn't really feel that it needed to be changed while testing. A pro that we realized about our game was that having 3D audio actually increases gameplay immersiveness a lot. While it seemed useful in our 2D games, 3D audio really makes the game feel interactive somewhat, as if the game is truly reacting to each of your inputs instead of just being a simple feature of the game.

We faced a lot of difficulty when we were coding our game up. There were many aspects we did not essentially plan out beforehand, such as the user being able to cheat boundaries, or collision pushing our player box back, and especially collision given rotational axes. We also dealt with a lot of sizing issues in implementing our box to fit in a hole as well as coordinate issues in implementing our audio. Overall, our team worked remarkably to put together a simple, yet fun game. While we did not create a second game for our unit 3 games, the largest hurdle we faced was translating what we had learned in our 2D games to 3D games. While the ideas were there and the code seemed similar, it was most definitely a different objective. Besides these coding issues and the usual bugs here and there, there were no other major glaring problems. Having experienced coding a 3D game now, if given more time, would definitely have more confidence in building our own, much more developed 3D game, beginning with the engine.