

Adrina Khan

ENGL 3500

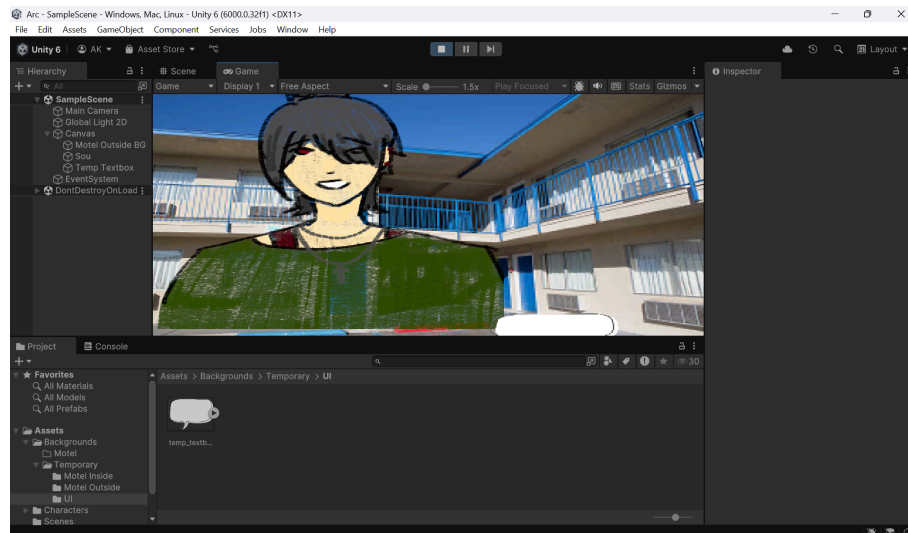
HW 11: Failure, Errors, and Debugging

April 4, 2025

In *Getting Over It with Bennett Foddy*, the player is tasked with maneuvering through a mountain range and various miscellaneous objects to reach outer space, except the player must do this through the use of a hammer and a man immobilized inside a cauldron. The game has some narration at certain checkpoints, in which Bennet Foddy will note something interesting about life, games, or random topics. The narration and calming music encourage the player to advance to the next “narration checkpoint”, but the gameplay does otherwise. The game is frustrating in that there are no save points to reset progress from; falling causes the player to start from the height that they fell to, be it a few feet or hundreds of feet. Those who pick up the game do so because it is known to be frustrating, challenging, and, at the end, rewarding. This plays into Juul’s idea of a central paradox in which players seek out games, knowing that they would fail, but play them anyway. Additionally, Juul calls out that this failure is taken to a more personal level; instead of noting that failure happens to the player character, Juul argues that the failure is incurred by the player themselves. This is especially apparent in *Getting Over It* in which the man in the cauldron does not fail when he falls down the mountain and instead the player feels like a failure. However, as Juul mentions, this sense of failure and acknowledgment of it is essential to games as it allows the success of the game to be more rewarding and meaningful.

Juul's perspective of the paradox of games can also be said about the making of a game as well. Game creators design and code games knowing that they will run into bugs and errors, but challenge themselves to do so anyway. The mistakes made throughout the game design process are not blamed upon the software (a mirror to the game's player character), they are pushed onto the creator (mirroring the failure felt by players). Though, once again, this failure is inevitable and essential to the game designing and creation process as it makes the little success much more rewarding and encouraging. While making my game, I ran into a few minor errors, two of which were related to images. The first error I encountered was importing assets without transparent backgrounds; the player sprites I drew had white backgrounds which would interfere with the background image they would lie on top of. To fix this, I redrew and reimported the assets with a transparent background (see sou_sprite in homework folder assets vs the ones imported into Unity). The other error I encountered was during the game creation process. When placing game objects onto the scene, in this case Sou's sprite, it was rendered poorly and too large. To fix this, I had to change the image settings through Unity and properly resize the image. Interestingly, and similar to Juul's ideologies, the minor bugs along the way make game creation fun as the little success it brings after motivates me to keep making the game.

Before resizing game objects and fixing resolution:



After resizing game objects and fixing resolution:

