

Adan Gomez

Game Technology and Coding II

Shocker Studios | Prof. Brian Foster

Due: Tuesday, Nov 14th, 2022

Abstract

During the *Week 11* lecture, in the *MART 361 Game Tech. and Coding II* class, the module introduced how to implement saving and loading systems within *Unreal Engine* games. To help build the students as Game Designers as well as test their application of the concept, students are challenged to create a game that displays the feature.

Through the methods that students learned, players would be able to not only save what items they have collected to that point, they could also save the current position and rotation of the player as well. This versatility allows designers to not only implement progression systems in their games, but also be creative in the way those systems are implemented.

Projected Practicum II

Saving as a Mechanic

Normally in video games, saving as a whole is not that unusual as a concept; On the contrary it is almost always expected in some way shape or form for most current games. However, outside of some *Indie* titles, saving is almost always used as a system or a feature rather than as a mechanic. While there are examples of Save Systems not being strictly bound as an action that solely takes place within UI, these edge cases are few and far between.

One notable example of saving being more interactive is in *Shovel Knight*; In the game players are able to choose whether or not they decide to take advantage of the save points scattered throughout the levels. In *Shovel Knight*, this type of Save System gives players the option of whether or not they want to interact with the system, allowing for a greater freedom of choice. Players who are more experienced may choose to destroy a save point and are rewarded for their progression with gold instead.

Other games that have gone without rigid Save Systems, fall into the category of *Rougelike* games. The inherent nature of these games is that the dungeons players explore are completely procedural, allowing for greater replayability, and a “fresh” experience through every playthrough. Having procedural dungeons, in other words, a dungeon that is generated by the computer every time an area is entered, would experience more drawbacks from including the “standard” Save System, then any benefits it could receive.

With these types of save systems in mind, my intention with this save system is that it saves the player “separate” from the world. To further elaborate on this concept, the world itself would exist as something that is dynamic and constantly moving, player actions would have limited intervention on the world at large. However, when the player activates a save, their position would be relative to all the progress they made up to that point; the twist with the mechanic being that the world itself “restarts” as if it were the player's first time loading the game. As an example, as the game is played a bridge suddenly collapses leaving the

player unable to cross to the other side; however if the player were to load a save directly in front of the bridge, they would be able to cross the bridge before it collapses.

Puzzles

Aside from the environmental puzzles one of the requirements of this assignment is the use of the AI that had been worked on throughout the semester. This presents a great opportunity to include stealth elements to the game. Being cognizant of the changes in the world, while also focusing on the enemies actively targeting the player seems like a great way to engage players, at least in a memorable way.

Paragraph

Paragraph

Victory

Paragraph

Paragraph

Paragraph