Scraps Postmortem

Introduction

In the last month, Team Asparagus set out to create a fully-featured level for the *Scraps* video game – complete with a custom scripted sound system for adjusting the volume of music and voice over on the fly. We aimed to create a level which would showcase individual talent and highlight the skills that we have learned while attending Full Sail University. Our team consisted of six team members who all together dedicated an average of 650 man-hours each week to the project. Overall, our team is proud of what we were able to accomplish in a short amount of time, and we all plan on highlighting our level in our respective design portfolios.

Hours

~16 hours per day. (Weekends included.)

What Went Right?

1) Creativity

Our team strived to express creativity every step of the way this month, and we gave this project our absolute all with a unified goal to produce the best *Scraps* level that Full Sail University has ever been presented. Above all, we realize that breaking into the gaming industry as game designers is going to require a lot of hard-work and dedication from each of us, so we wanted to demonstrate our potential by producing an excellent addition to our individual portfolios.

Throughout this month, our team aimed to create a highly immersive player experience by applying the knowledge and tools that we have gained while attending Full Sail University. We strived to remain resourceful and adept in solving our design and technical problems by seeking out additional resources. Along the way, we were able to overcome many design and development hurdles, and we have immensely improved our game design and problem solving skills.

Our team routinely decided on how to ensure player progression stayed intact while pushing the limits of what was possible - in terms of design. Initially, the first area of our level used a transition point to teleport the player into the next area, but we decided that setup detracted from the overall immersion of our level. So, we removed the transition point at the end of area one, and we actually built a network of twisting drainage pipes between the two areas for the player to traverse through, and we added a few interactions along the way to keep the player engaged.

2) Problem Solving.

Our team routinely encountered problems which we solved, but we encountered the most problems when we started adding new features. The scope of our project grew the most during our second week because we added many new features to our individual areas. The setups in our alpha build worked flawlessly, but during the second week we quickly became aware of conflicts (scripts).

There were two recurring problems that continued to plague us:

1) Duplicate Scripts – We should have used the same script for instead of duplicating it. New problems tended to crop up after a team mate modified the original script, but not the duplicated script. I would suggest that future teams strive to reuse scripts, and I would suggest that they make their scripts more modular.

2) Lack of Prefabs – At first, team members were merely copying setups out of each other's scene, then pasting the setup in their individual scene. From the beginning, I warned that is very bad practice when you want all of your setups to remain consistent across the level. We established better organization of our assets and consistently created prefabs after submitting the beta build.

3) Combining Individual Areas

One of the first tasks that our team accomplished was to combine our individual areas, and it was by far the easiest part of this project – for us. We owe much thanks to the fact that most team members correctly followed the project export directions in the previous Level Design course. Although, a few team members had issues with importing their project assets – most notably the terrain data.

The few team members who had issues with importing their terrain was due to the fact that they did not export the terrain data which was associated with their terrain. For each team member, who was missing terrain data, we were able to recover terrain data from their Level Design work space on the Perforce server. Keep in mind that Unity usually places a terrain's data asset in your project's Assets folder. Some people may not think to check for assets (terrain data) in the Assets folder of their projects – when exporting a unity package.

4) Sounds and Voice Over

Our team would have liked to have had all of our sounds and voice over completed for the beta build, but we had to push their addition off till the very end. We were able to script a custom sound system which allowed us to cycle through background music tracks, and we were able to reduce the volume of the background music during the time that voice over was played. We did not want to

completely cut the background music off entirely, but at the same time we wanted to ensure that the player could clearly hear the voice over.

We recorded our voice overs with Audacity, and we actually chose to export our voice overs as WAV (mono) for our final build. We decided to go with mono due to the time crunch that we were under because some of the voice over clips had been recorded as mono instead of stereo. We are pleased with the overall sound quality of our voice overs in the end. The Art Lead, **Kelly Bruce**, did an exceptional job in providing us with the high quality voice overs, and I was utterly blown away by her talent for voice acting.

5) Consolidating Inventory

Early on, our team began to realize that we would need to consolidate our inventory items in order to prevent the inventory icons from overflowing the UI. We did not consolidate the inventory until after the beta build. We began seeing problems in our scripted setups where the player would have the item in their inventory, but some of our scripts did not seem to detect the item in the inventory. Immediately, we consolidated all of our item into a handful of items, and we were able to alleviate the problem. Before going that route, we tried inspecting the ASCore inventory script system to try to figure out what the issue was, but we couldn't find any apparent problem.

What Went Wrong?

1) Terrain cannot Rotate

When we began moving our scene nodes into place. We encountered a problem when trying to rotate our individual scene nodes – specifically we experienced a problem with rotating the terrain within each scene node. We exhausted time searching for a way to rotate our terrains within Unity, but we learned that Unity does not allow the terrain to be rotated – without using a third-party solution.

Solution:

We exported the height map of the terrains which needed to be rotated, then we imported them into Adobe Photoshop where we were able to rotate the height map images – 90 degrees clockwise or counter-clockwise. We exported the height maps out of Photoshop in RAW format, and we imported our new terrain height maps back into Unity. Unfortunately, we ran into more problems with the terrain, so we decided to redo the entire terrain for our level.

2) ASCore Scripts

Objective List (Script): Our team has noticed a considerable amount of bottlenecks on the CPU which are caused by scripts provided in the ASCore package. For instance, the Objective List was one of the biggest culprits, and we did actually modify that script without changing the core functionality the game. The main problem with the Objective List is that the Update function is causing the UI to redraw the Objective List too often. It would not have really been a problem, but the Text in the Scraps UI is using both a Shadow and Outline filter – and that's where it becomes a problem.

Solution:

The System Lead, **Alexander Shepherd**, modified the script to update the Objective List only if it needs to be updated. After we cleared out that bottle neck, then that gained us around 20+ frames per second throughout all of our areas.

Extra Solutions:

Minimap (Script): The Minimap script is trying to find LBM type game objects way too often in the Update function. So, the System Lead modified the Minimap script to find objects of type LBM in the start function of the Minimap script. Although, in the end, we decided to explicitly cull the objects that the minimap displays in order to prevent all of our LBM game objects from being placed on the LBM layer.

Transition System (Script): The Transition System is another ASCore provided script which bottlenecks the CPU because it is searching for Transition Points in the Update function. The System Lead did try to modify this script so that it only searches for transition points in its Start function, but it ended up breaking core functionality in our builds. We reverted the changes made to the Transition System, and we tried to reduce transition points used throughout our level. The best solution for us would have been to not use the Transition System which would have improved player immersion, but we would have needed to animate the player's avatar when it climbed up ladders. (We were hesitant to bring in new animations to the character).

3) Managing Multiple Terrains

Our level occupies a large amount of space, so we wanted to have a terrain which could house all of our individual areas. We wanted to have the flexibility to easily expand the the terrain out too. We planned to add mountain ranges in the distance. Unity does not provide a solution to stitch together terrain by default.

Solution:

We purchased Multi-Terrain Brush on the Unity Store.
https://www.assetstore.unity3d.com/en/#!/content/44037

Multi-Terrain Brush greatly improved our workflow when working with the terrain of our level. We were able to sculpt and paint multiple terrain data assets simultaneously, and we were able to seamlessly stitch together our terrains with ease.

4) Last Minute Builds

We thoroughly tested our level in the editor of Unity, but we always built our game towards the very end. Often times, we had very little time remaining before we were required to submit to the perforce server. We put ourselves through an immense amount of stress by waiting a couple of hours for our level to finish building. Our team was extremely lucky that our builds did not fail, and they played as was expected too.

Solution:

If we would have had a team member who continually built the game on their system throughout the week, then that would have helped save us a tremendous amount of stress and uncertainty. The initial build of our game always took the longest amount of time, but the subsequent builds were much quicker. I will never allow a future team that I am on to operate with such a haphazard work flow ever again. We will strive to build early and often.

5) Not Enough Polish; Too Many Features.

Our team planned to put a halt on adding new features before we began working on Gold. We had some team members who continued to focus on adding new features to their areas which gave us some problems to troubleshoot.

Solution:

While we didn't add any major new features to our game, I do not believe that it is a wise decision to continue to make additions to a game right before it is set to ship. Instead, we should have focused on polishing all of our already existing setups and features.

Advice to future students

Spend more time planning and organizing your assets upfront. Our team organized our project assets after the Beta build, and we did not wait a moment too late. We would have been better off if we would have organized early on before we began ramping up our effort to bring in new assets. Aside from organization, it is important that the Art Lead designates an area for prefabs to be stored, and the team must use only the prefabs from that designated location. It is important to maintain consistency through a level, and it is possible to maintain consistency when the whole team uses the same prefab game objects. Individuals should always request new art assets to be brought in and configured by the Art Lead.

Conclusion

In summary, all members on our team put their lives on hold in order to devote our time to the completion of this project. As I stated at the beginning, we wanted to make the best Scraps level that has ever been created. I am certain that we have created an outstanding portfolio piece which will help propel us into the gaming industry. Above all, we learn A LOT, and I was happy to learn that I have somewhat of a teaching skill as I spent much time instructing my team mates.