DANGER Docs

Danger Docs Justification Document
Alex Chojnacki, Benjamin Lapid, Eliana Abraham, Morgan Erlich

Creative Process

The project began with a handful of meetings in the first week of project 4 dedicated to committing to a clear concept of what we wanted our game to be. We all agreed it would be important for everyone to share a vision of what they wanted the final game to look like. We stressed this not only because it leads to fewer conceptual pivots in development, which carry a heavy time cost, but also because we believed that it would be far easier to stay motivated if we all had a passion for creating that game.

We started with a few simple questions -- effectively ad-libbing a logline. "{GameName} is a(n) {adjective} {2d, top down, 3d, first person} {multiplayer,single player} {action, adventure, party} game where [the] player(s) {verb}." We quickly settled on "cooperative party game" as an overall constraint and decided to iterate through general themes and their prospective gameplay mechanics. We thought about team games which teammates were forced to communicate as an explicit gameplay mechanic, like the instruction system in SpaceTeam, or the daytime deliberation in werewolf or mafia. We were also very entertained by the idea of a "wacky" or otherwise less-than-serious themes.

After much deliberation, Eliana suggested an idea for a game in which players commanded cells inside of a human body, directing its actions and facilitation normal body functions ala "Osmosis Jones" or "Inside Out". This theme got everyone pretty excited, so we stuck with it a bit and quickly iterated the idea into a "wacky" cooperative surgery game. Core mechanics would include the availability of different tools to complete surgeries and a bear which could end the game. The idea stuck with us and the theme changed very little from that point onward.

After the initial concept phase, we researched a few games after our initial pitch. One game in particular proved heavily inspirational: Overcooked. A fantastic little game, praised for it's character, approachability, and brevity, provided a great basis from which we based our game. From overcooked, we took the idea of environmental factors calling players away from their current tasks, forcing asynchronous gameplay decisions and task scheduling. We also sought to emulate the hectic atmosphere, though this proved to be the most difficult aspect to emulate -- a goal which we could only approach but never truly achieve without an incredible amount of playtesting and fine tuning.

Development and Playtesting

Playtesting was instrumental for our game, both from in class demos and the official playtesting much later. Our first big learning experience came from the an in-class demo. We noticed that with the patient staying in one place and all the tools being required to be used on the patient there was very little incentive to leave the patient's side. From this observation, we thought conceived of the tool-stealing raccoons, a mechanic which successfully drew doctor's away from the patient. The bear also came from this critique -- an event which forces player collaboration in order to deter it, though admittedly, this mechanic was not demoed until far later in the class due to technical bugs, some of which persisted until the gamma (sorry).

The second big win for playtesting came from the beta testing feedback. Because our game contained so many individual mechanics, it was very overwhelming to be dropped into game sessions where the fail state could come about within 15 seconds of starting and players would have virtually no idea why. This incentivized us to add an in-depth tutorial that was flashy, satisfying, and really effectively communicated the mechanics of the game to players without overstaying its welcome. This gives the players excellent instruction and eliminates most of the ambiguity from the main game, allowing players to focus on reacting to the various events in the game and less ad-hoc investigating the game mechanics. From this same feedback session, we learned that we needed a much cleaner user interface with higher quality graphics to communicate controls and instructions. Taking this to an extreme, our main game includes no text -- which we feel was an improvement over the necessity for a text interface to explain the mechanics during real gameplay.

Playtesting was also instrumental in identifying some bugs which we didn’t anticipate. We wondered why our demos were so much more buggy than our personal playtests, eventually learning of a inconsistency in the C# event system which caused some of our code to fail to execute on windows machines after a specific instance of an item was stolen by a raccoon. After fixing the bug by building destructors for the appropriate classes--thus no longer relying on C# to clean up the instance for us-- at least three game breaking bugs were fixed. We would not have been able to identify this if it weren’t for other playtesters, as three of our four groupmates were using macs.