Team # 1

Game Title:

Ro(b)ot (B)eat

Theme:

A unicorn t-shirt and a party that got way out of hand.

Game Description:

Botty Holly, the world’s first robot musician, has gone to Unicorn Unicorn, a start-up music production company, to market his mixtape to them. The gameplay consists of Botty sneaking through the floors of the company without getting caught and playing his mixtape for the people on each floor before succeeding.

How to Play:

W and S move the player forwards and backwards, A and D turn the player clockwise and counterclockwise, E picks up t-shirts from each floor’s t-shirt bin, E also plays Botty’s mixtape when he is close to each floor’s boombox, and space places t-shirts on the ground.

Postmortem thoughts:

On the art side, the pipeline was a little messy. The models weren’t scaled well early enough and needed to be UV mapped before rigging. The rigged models don’t have any textures for that reason. There were also issues with the Pudgy Man asset which led to multiple reskinning. Just small issues which should have been resolved earlier which ended up snowballing.

In terms of the story, it took a fair amount of time in the beginning to come up with a cohesive story that fit with the concept of the game that we’d decided on, but after that, there weren’t any major issues vis a vis storyboarding or in-game writing. We just kept iterating until the story was as good as we could get it. As for the audio, it would have been better to get started on sound effects earlier in the pipeline. Most of those were implemented in the latter half of the second week.

As for the AI, we should have started with a navmesh instead of implementing our own pathfinding. There were a couple hurdles that had to be overcome to get it to work that wouldn’t be necessary with navmeshes.

For the gameplay programming, we definitely could’ve benefitted from making the SceneChanger script functions static. We ran into a few speed bumps in implementation because it wasn’t static and by the time we realized it could be a problem, changing the implementation became a headache.

It works well when the level is predesigned on grid-based paper prototype. It shows again the importance of getting things figured out before we make it into digital product. Meanwhile to have two writers on the team is very pleasing because both of us can give each other comments on different design decisions. However, some ideas, like putting instructions into the game, and most importantly, dynamic dialog was not implemented because of time restriction, but if they do exist, the narrative could be more cohesive.

Team Members & Responsibilities & Signatures:

Grant Doney:

Modeled all 3D assets, rigged and animated the Pudgy Man and the Robot, textured shirts, box of shirts, the shirt stack, the walls, boombox, Pudgy Man, Robot, door, and table, created Win, Loss, and all three exposition screens, and designed unicorn logo

Jordan Peloquin:

Created all music tracks, created/retrieved all sound effects save one, collaborated with Grant on the storyboard for cutscenes, narrative design and planning, all in-game text, level design, implementing most audio and all sound effects, and implementing cutscene graphics.

Jimmy Jin:

Mostly assist Jordan in the story and sound effect (i.e. commenting and sourcing), collaborated with Jordan in designing puzzle elements to be solved in the level design, designed the instruction board to assist the general narrative environment.

Eric Franco:

Player movement, placing shirts down, boombox (toggling music on/off), set up all animations through code and controller, scene management, grabbing shirts from boxes, UI text, title screen (image and implementation), game over sceen. Script files edited: Player, SceneChanger, BoomBox, GameOver, ShirtCount, ShirtText, PathfindingScript (for animations).

Jared Okun:

The enemy AI including pathfinding. Level generation. The scripts for the intro screen and ending scene. Made the camera follow the player. Script files edited: CameraFollowScript, CreditScript, EndScript, FlockingScript (unused), GenerateLevel, IntroScript, LevelManagerScript, PathfindingScript, and PathsListScript.