

Coulianos

method for pacing analysis

Foreword

Foreword

Games are improving in quality year by year. It's mostly noticeable by the ever increasing visual quality, however the level design quality also has to improve to make a better game. Too often I see feedback solely based upon screenshots or graphics on both forums and reviews; they don't tell anything about the level design! Some people even seem to think that level design and art is the same thing. The word pacing has become more and more of a buzzword, but no one has really defined what it is. In this article I will present you my view of what pacing is, and how you can measure it and adjust it to perfect the end user experience.

This whole thing started out as I wanted to see if I could analyze the way Valve timed things to create the great single player experiences they have been so known for creating. I invented a dirty, quick and rough method to analyze pacing in linear computer games in an effort to find the magic recipe. In this article I will share how I used it to create Project 25, a 35 minute single player episode set in the Half-life 2 universe, as well as suggesting how you can use it to create your own stuff.

Background

I'm Filip Coulianos, an aspiring level designer who has studied computer game development at the University of Skovde/Sweden with an emphasis on graphics. My multiplayer experience comes mainly as a core member of the Decadence development team contributing with environment art, level design and game design. I also have released numerous maps for various mods powered by GoldSource and Source engine prior to Decadence. My main single player achievement comes from a small single player episode called Project 25 which I will use as my main source when I write this article.

Introduction

Introduction

This method is all about thinking about level design in the most abstract way; a long chain of events with different game play elements that must be balanced to one another to keep it varied, consistent and interesting. The relations between the links are as important as their content. None of the links may be too large to avoid player fatigue, nor should they be too short to keep the player focused on the challenges that are presented. This method adds a layer to the typical abstract when designing single player games because of the timestamps. First off, let's have a look at an abstract from Duke Nukem Forever;

DNF World Chart		EDF Base Continued	
Mission	RC Car Driving 2	Cloning Research	Platform Ride Part 1
Filename (Chunk#)	map14d (20)	map16 (21)	map16c (22)
Major Plot Points	Still looking for Graves Bombshell continues navigating the RC car in the EDF Base clone meat cooking, deboning and grinding facility while avoiding Giant Pigcops, Rats, Octababies, spinning bone crushers, flame jets, death pits, and waste compactors. She must get off the RC car a number of times solving puzzles to progress: Jumping on waste compactor gears and rotating fuses to enable and disable pistons. Finally reaching the clone meat Cryotube entry station Bombshell grows and solves a Cryotube switchboard puzzle riding a Cryotube to exit.	Riding a Cryotube system into the Clone Research Facility Bombshell navigates a Cryotube puzzle, combats Duke clones and saves the real Duke with a lap dance. Now free Duke helps Bombshell escape out a vent and is escorted by Pigcops through the facility where he learns of the Duke Clone army and must win an arm wrestling contest against a Duke clone to progress. Reaching a warehouse Duke acquires a Hover lift pushes it into place and rides it using shootable impulse beams. Reaching the Hover lift depot Duke opens the exit doors and rides the Hover lift out as the alarms sound and the Clones are now aware he is an imposter.	Duke rides the Hover lift searching for Graves and the Teleporation Facility while combating enemies and avoiding trip mine traps. The Platform stops several times forcing Duke to get off opening giant doors and using a crane and forklift to solve puzzles to progress
Non Standard Game play	Play as Bombshell Shrunk Game Play Shrunk Driving	Cryotube Ride Arm wrestle clone Lift impulse puzzle	Rail shooter sequence Forklift ride
New Skills			
New Weapons		Freeze Ray	
New Enemies		Duke Clones	
Time of Day	Mon, April 3rd - 12:00 AM	Mon, April 3rd - 2:00 AM	Mon, April 3rd - 4:00 AM
Level Exit	Ride a Cryotube out	Ride the Hover lift out	Open Blast Doors and ride the Hoverlift out
% Complete	60	40	80

Source: www.nofrag.com

This abstract is very clear about what will happen in each level. It efficiently communicates player progression by stating when new enemies show up as well as new abilities and skills that adds to the core game play. It's a great start and a good way to flesh out all your ideas and to communicate your ideas to the rest of the team. What this chart does not tell you is how everything relates time-wise. How much time is the player supposed to spend flying around with the jet pack compared to the combat the player experiences later? How long are the cut scenes? What should be the target play time for a section?

How do I use it?

Break down game play

The idea behind the method is simple. Break down the core game play of your game to a handful of different game play elements. Let's take Half-Life 2 as an example which I think has the following game play elements:

Puzzles; non combat sections where the player has to solve a logical puzzle to proceed. (very low stress)

Dialogue; non combat sections where either pieces of the story is being revealed, or player gets new weapons. (no stress)

Arena; enclosed and heavily scripted areas in which the player faces multiple enemies and/or a big boss and must kill them all in order to proceed. (high stress)

Soft resistance; areas in which the player simply travels through from with enemies scattered around to keep the player somewhat busy. (low stress)

Vehicle ride; areas in which the player drives through with either the airboat or the buggy. (low/medium stress)

You then let someone play through your game/mod and time the tester. For each new game play element, take a note and set a time stamp. You then use the time-stamps and create a chart to illustrate how the player progressed through the different areas and can easily see if the player spends too much time doing the same thing, which is something you want to avoid.

How do I use it?

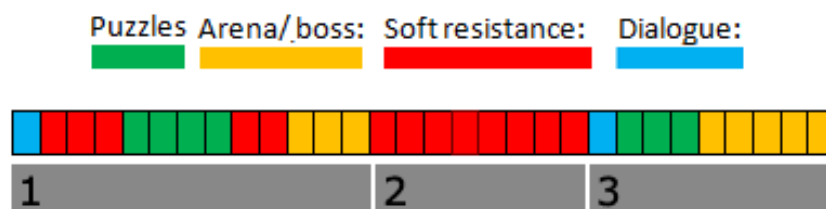
Continued...

Project 25 example

When designing Project 25 I had a couple of goals. Project 25 had to be at least 25 minutes of game play for the average player, it also had to be doable in 2,5 months. What I basically did was to come up with an abstract similar to the one in the Duke Nukem Forever example. I then designed my own chart, putting game play elements into it and an estimated completion time for each element. The basic abstract of project 25 was like this:

“The player gets introduced to game and her goal (reach station 25) in a rebel base located in a tunnel. The player then progresses through the tunnel, and gets up to street level. There the player battles through combine forces until she finds and saves a couple of rebels. The rebels lead the player to station 25 where the player gets ambushed, survives and finds Alyx.”

I then made a breakdown with timestamps of what should happen in my mod. This is a breakdown of the pacing for the episode, each cell in the chart represents one minute of playtime, the chart is supposed to be read from left to right:



1. First; one minute of dialogue (blue) in which the player is introduced to the game and the goal. This followed by a couple of minutes with just basic exploration and sporadic encounters of zombies (red). Then a puzzle (green) to get the player relaxed and not threatened. When the puzzle is solved the player quickly gets attacked by zombies and an arena follows with combine soldiers breaking into the tunnel (orange).

2. The player gets through the tough fight and now continues into a sewer system with little to none enemies to get the stress down (red). Finally the player gets up from the sewers and fights her way through a backyard (red).

3. The player saves rebels who were held hostage in one of the houses at the backyard. After a short dialogue the player is being shown where station 25 is. The player now has to solve a puzzle to get through (green). The pacing is very low as the player solves the puzzle and opens a gate, only to be surprised by an ambush by the combine. A final battle takes place in station 25's ruins.

How do I use it?

Continued...

Playtesting

After I was done with the abstract and had timing in place I now knew I had to come up with two puzzles, write and design dialogue for 2 occasions, a long arena event in the end (the boss fight) as well as one in the middle of the game. This was of course all subject to change, but now I had a very clear picture about what to implement and a rough idea about how big each area should be. Next stop was to get the first iteration up and running as soon as I possibly could.

When first iteration was done, it was time to do the most important thing in the development process; test the product. To get most out of my method one has to do this as early as possible. I had a fully working prototype about three weeks into the project. It was very rough, buggy and all that, but the project was fully playable from the beginning to the end. From this time and on I had regular play tests once or twice a week with friends who played it through. Below are a few handy tips and tricks that you could use when conducting your own playtests:

Always be present during the tests and take notes about what the testers had difficulty with and what they didn't understand. Also time the players to get a feeling of how well your target time compare to the average player.

Have an interview after the playtest. Ask the playtester what she liked and what she didnt like. Ask things based on your observations such as if the player behaved strangely during a specific challenge.

The playtester will always ask questions regarding the game and the challenges that are presented when she plays; dont answer these questions. Instead, take notes about the questions and answer them after the playtest. Pay close attention to what the tester asks about, this is vital information that will help you create a smoother experience!

Try to avoid "the scientist effect" at all costs. The playtester will most probably feel more or less uncomfortable about having you sitting behind her back taking notes about every move she does, and because of that will play differently than she would otherwise. Do everything you can to make the playtester feel comfortable. If you can, conduct the test at the playtesters home.

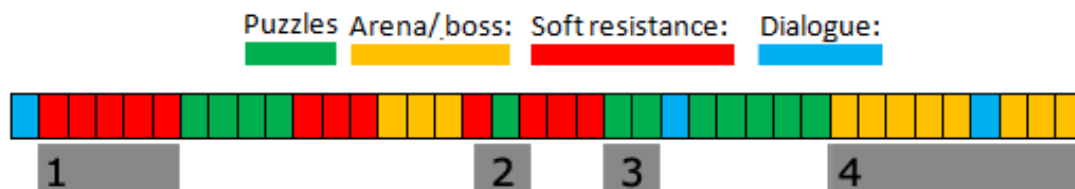
If you can, try to avoid showing the tester that you actually time her. This will only stress her and make her play faster or slower than she would otherwise. You also could record a demo of the playsession and time it later to avoid this issue.

How do I use it?

Continued...

Changes due to testing

In total I did 10 play tests of project 25 before release. Project 25 was fully playable from beginning to end when I had been developing about 1/3 into the project. I then continually tested throughout the development once or twice a week improving the game each time. I also took notes about completion time for each gameplay element and compared the data between experienced and inexperienced players. Below are the changes that I did as Project 25 was being playtested, this chart is based upon the completion time for the average player:



1. The first area got slightly bigger to make sure the player spent a bit longer in-game time before the first puzzle emerged.
2. A small puzzle was added in the sewers to make sure pacing was low before the fight at the backyard. I added this puzzle because players would otherwise do nothing but kill stuff for 13 minutes straight – which is bad variation.
3. I also added a small puzzle right before the dialogue to lower the stress on the player prior to the dialogue.
4. The final battle was interesting but some testers got a bit fatigued by the intense fighting. Because of this I changed the final battle in such a way that the combine forces attacks in two waves and new rebel characters shows up and helps out in the fight, hence the one minute dialogue in the middle of two arenas.

As you can see the average player actually gets to a new game play element in roughly every five minutes.

I found that the sweet spot is about 8 play tests (or iterations if you will) is about enough to collect information you need, after that information starts to get redundant. Interestingly enough I found that the game studio REBELLION (the creators of Alien Versus Predator series) uses a minimum of 7 iterations before giving a section of the game green light for shipping. However really big projects most probably have A LOT more than 8 play tests.

Summary

Enhancements

As I used this method for analysis of an existing game and used it to plan my own project I also found a couple of flaws with it. There simply isn't enough information being presented in the chart to make it easy to communicate with other teammates, leads etc. I suggest that one adds notes and comments around the chart and also adds a stress curve to illustrate the amount of stress the games puts on the player.

Summary

To think about what should happen in your project is of course essential, but don't forget to think about how much TIME everything should take as well. How long should each game play moment be? Work with target completion time for each event and think about how it will relate to what will happen after!

Get first prototype up and running as early as possible. Being able to play from start to finish 1/3 into the project proved to be extremely successful from a feedback and quality perspective. (This is perhaps not possible if new game play mechanics are to be implemented in the project, but I do think you should aim for this goal).

Play test often and with different people. Seven to ten play tests proved to be a good target to aim for, but you can never get too much. Don't forget to take notes and interview the tester afterwards.

Time the testers! How long did it take for them to get through the experience? Did the average players experience correspond to your estimates? Do you shift pace often enough to avoid fatigue?

A shift in pace every 5-10 minutes proved to be a good recipe to keep the players feel immersed and find the game play experience varied and interesting.

Pacing is just a piece of the puzzle. Don't forget to think about the quality at each event as well! Make every puzzle, scripted sequence or whatever the challenge really awesome! Think about what you want the player to feel at each event and how that feeling will be in relation to text event.

This method can also be used by quality assurance teams to show actual numbers on sections of the game in development that doesn't have good variation or pacing. Simply time the testers, if the pace of the average player doesn't shift very often you can easily prove this to whoever is in charge and ask for a improvement.