

Rogue Legacy Design Post Mortem: Budget Development

Teddy Lee

Game Designer, Cellar Door Games

Kenny Lee

Programmer, Cellar Door Games

Teddy: Hey everyone, thanks for coming to our talk today. My name is Teddy Lee, I am the Game Designer for Cellar Door Games.

Kenny: And I'm Kenny Lee, the programmer. If you haven't guessed it yet, we're brothers.

Teddy: We're here today to give a post mortem on Rogue Legacy, with an emphasis on budget design.

For those who don't know, Rogue Legacy is a casual man's roguelike. Even though many people still find it very difficult, it is far more accessible because we added permanency. So in the game, every time you die, you get to spend the money you found to improve your family. And I say family because with each death you never get to play the same hero. Instead, your children succeed you.

[NEXT SLIDE]

Budget Oriented

- Game mechanics first.
- Preference for cheap and efficient choices.
- We'll never try to make the perfect game.

Kenny:

To start, we consider ourselves to be a budget oriented team. That means we focus more on gameplay rather than trying to make expensive, highly detailed worlds.

That also means we're always on the look out for solutions that are cheap, fast, and reusable.

So we often implement solutions that aren't as good if the cost savings are justified.

So we'll never make our perfect game, and we're okay with that.

[NEXT SLIDE]

Previous Games



Kenny:

We've been doing this for a long time now. A lot of people think Rogue Legacy was our first foray into game development, but we actually made 8 games before it.

On average, each of these games cost us less than a thousand dollars to make, with development time ranging from 2 days to 3 and a half months.

And it's through this experience that we were really able to solidify how we go about making games.

[NEXT SLIDE]

Development Timeline

Kenny recklessly starts the project as programmer



Jan. 2012

Glauber enters the battle as artist



Jun. 2012

Gordon joins as composer and sound designer



Oct. 2012

Judson swoops in as music composer



Jan. 2013

Teddy hobbles forth as full time designer



Feb. 2013



June 27, 2013

Kenny:

So this is the development timeline for Rogue Legacy. As you can see it starts out with me working on the project solo, and as time went by more and more people joined on.

For Rogue Legacy, it took us a year and a half to make the game, and I was the only person who worked on it full-time from start to finish.

Teddy actually came on full-time 5 months before release and everyone else was contracted.

[NEXT SLIDE]

Rogue Legacy By the Numbers

- Development time: 1.5 years
- Development cost: \$14,878
- Copies sold a week after launch: 100,000

Kenny:

This is Rogue Legacy by the numbers.

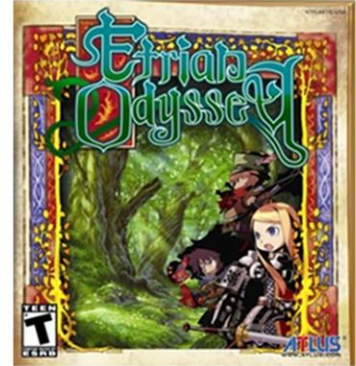
In total, our out of pocket costs for the project was less than \$15,000.

When the game went up for sale, we were able to cover all of our development costs in less than an hour.

And within the first week, we were able to sell over 100,000 copies of the game.

[NEXT SLIDE]

Rogue Legacy's Origins



Kenny:

To show how seriously we take budget, we need to talk about Rogue Legacy's beginnings.

Rogue Legacy was never meant to be a roguelike. Instead, it was a mash up of 3 games.

It had the death system of Dark Souls, the platforming of Castlevania, and the cartography of Etrian Odyssey. And it had no roguelike elements in it.

It was a game of crazy scope that emphasized exploration and puzzle solving.

[NEXT SLIDE]

The Rise of DS2D



Kenny:

This is Dark Souls 2D, the first build of our original non-nroguelike game, and it took about 3 months to make.

[PLAY VIDEO]

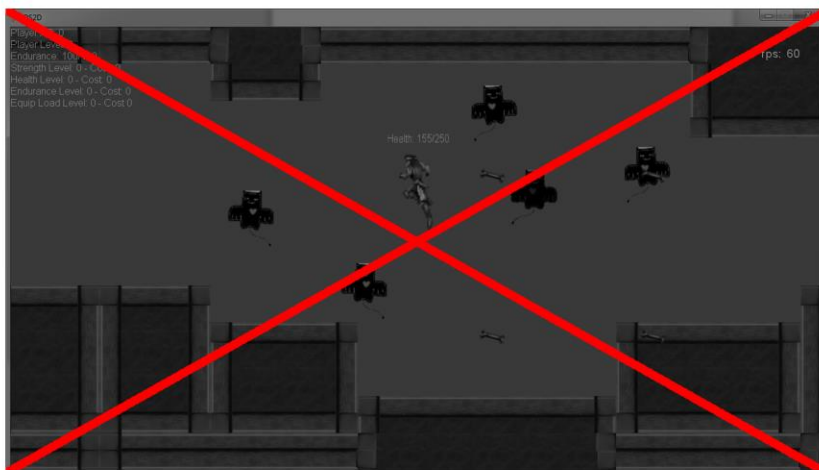
[THIS VIDEO MOST LIKELY NEEDS TO BE EDITED DOWN. I CANT SEE IT ON MY COMPUTER. ALSO, NO NEED TO NARRATE. IF IT'S LONG ENOUGH TO NARRATE, IT'S TOO LONG]

Kenny:

At this point we were feeling pretty good because it was already starting to feel like a game. It was even kind of fun! So we did what any developer would do in this position...

[NEXT SLIDE]

The Fall of DS2D



Kenny:

...we killed the project. At this point we were able to better evaluate the costs for the game and came to the fatal conclusion that we couldn't afford to make it.

It was a tough pill to swallow, but the experience of working on our previous games made us certain about these projections.

From here we went into salvage mode and tried to build a game from the scraps.

We had a basic 2D platforming engine, an AI logic system, and a level editor that could create rooms.

And that's when we decided to make Rogue Legacy.

Procedurally generated rooms resolved our biggest bottleneck, which was art, while opening up the strengths of our tools, which was rapid level development.

It was also a natural extension from Dark Souls 2D since a lot of the appeal to Dark Souls is its consequence upon death, which is similar to roguelikes but not as punishing.

[NEXT SLIDE]

The Rise of Rogue Legacy



Kenny: This is Rogue Legacy. After cancelling the first project, this version was created two and a half weeks later.

[PLAY VIDEO]

Kenny: Because budget drives us, we tried to salvage as much as we could. In the end we managed to salvage everything.

This was about a year before the game launched. We were able to launch so quickly despite the setbacks because we embraced the new direction immediately and pulled the scope down, with no regrets.

[NEXT SLIDE]

The Casual Roguelike



Teddy:

But the challenges didn't end there. We like to put twists on the genres we work on, so we decided to make a more casual roguelike experience. This meant changing some of the most common tenets found in roguelike games.

[NEXT SLIDE]

Make Death Fun



Teddy:

The first thing we wanted to fix was the harsh punishment of death. But for us it wasn't about making death less painful, but actually making it fun.

So we decided to add permanency to the game through the manor skill tree and equipment system. That way you never had to start from scratch. It was a contentious decision, because we knew we were diluting the spirit of roguelikes.

More Skill, Less Chance



Teddy:

We also wanted the game to have more skill and less chance.

A lot of roguelikes put emphasis on the roll of the dice.

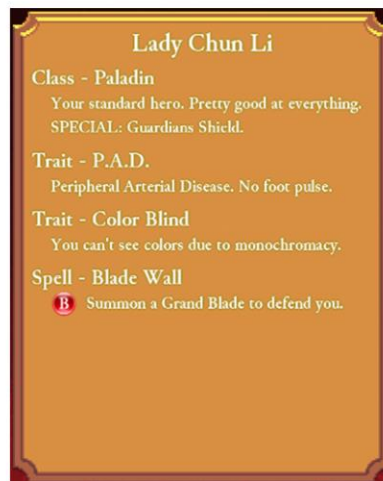
It often plays a bigger role in the player's success than their skill so we removed as much of it as we could.

Rogue Legacy has no critical misses, no instant death events, and we have a lot of tweaks inside the castle generation system to keep it fun and fair.

[TALK ABOUT CRITICAL STRIKES]

[NEXT SLIDE]

Reduce Obscurity



Teddy:

The next goal was to reduce obscurity.

Roguelikes often go out of their way to obscure information so that players can “discover” it. While some people enjoy that, we aimed to make our game more transparent and easier to memorize. So no pointless obfuscation like 50 different potions labeled purely by colour and stuff like that.

This is why we have descriptions for virtually everything in the game. Some are flat-out explained, while others provide hints so that the second time around, you more or less know what it does.

[NEXT SLIDE]

Faster and Shorter



Teddy:

Fourthly, we aimed to make it faster and shorter.

In most roguelikes, the first few minutes of play are usually the same until you start to power up and begin to take on unique characteristics.

We wanted to fix that, so we designed the lineage system to upfront all the change to the very start of the game.

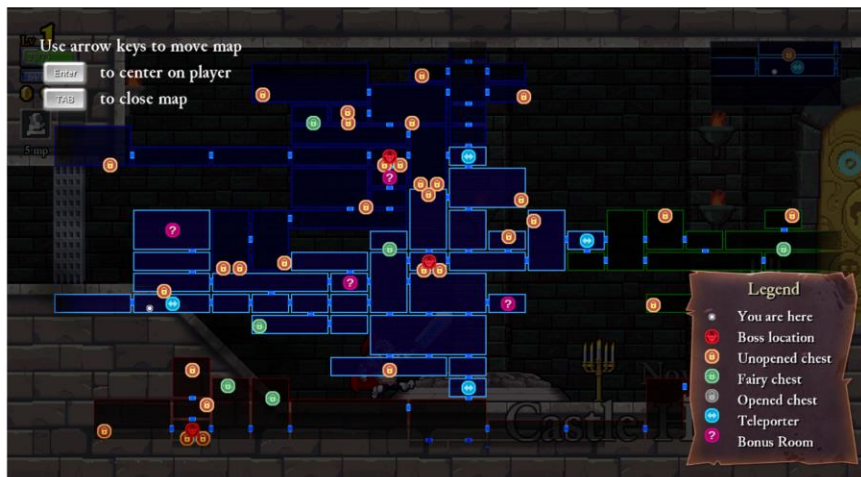
This was important because we wanted our game to be very easy to pick up and play. That meant making the average life of a player last around 1 minute. So if you died a lot in our game, don't worry, that was by design.

But these short lives also meant a lot of streamlining, because we didn't want people to play for 40 seconds, then spend the next 2 minutes navigating menus to play again. An example of this can be seen with the trait names. They were all purposefully very distinct, so that you could recognize them at

a glance and not have to read them every time. It's why we flip-flop between scientific terms and slang. [GIVE EXAMPLE OF HYPER/HYPOGONADISM AND MUSCLE WEAKNESS]

[NEXT SLIDE]

No Floor Clearing



Teddy:

The next point is No Floor Clearing.

A lot of roguelikes segment their game into floors, where the further you go, the harder it gets.

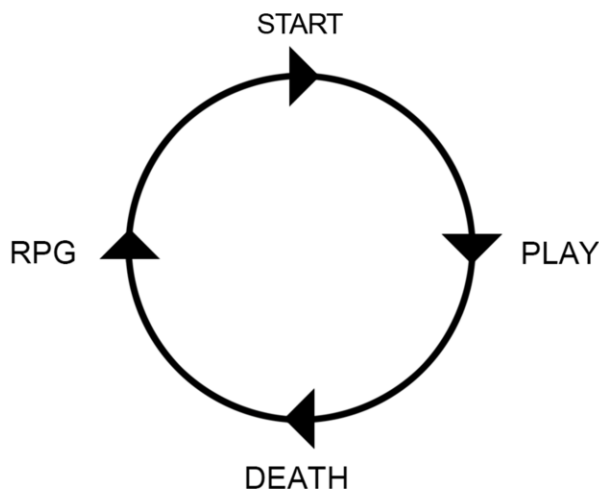
But if a player bypasses a floor without clearing it out, the enemies will outscale the player, putting you at a disadvantage.

This floor clearing slows the game down and limits player choice on how to progress.

We resolved this by removing all forms of character development during a run inside the castle.

[NEXT SLIDE]

The Game Loop



Teddy:

Our last point is actually the first decision we ever made for Rogue Legacy and the primary reason we stayed on budget.

We called it the Game Loop and it explains how the player goes through 1 life. This term might actually mean something else in Game Design, but I don't know. I didn't read any books.

It looks like a fairly standard loop, except the RPG mechanics happen after death and before you explore the castle; instead of during play like most games.

It seems like a minor change, however, almost all of the previous design choices we spoke of were built to support it.

By putting the RPG mechanics after dying, you had something to look forward to, making death fun. And because no character development happened during gameplay, we could lessen downtime, and resolve the issues related to floor clearing.

And best of all, it gave that "one more time" feel.

[NEXT SLIDE]

Rogue Legacy Alpha



Kenny:

This is what we had 8 months later.

[START VIDEO]

This is pretty much the retail version you see now with a bit of older art, but a lot of features missing.

Even though things like Charon, traits, and even the lineage system weren't in, we were certain it'd be done on time.

Honestly, a video of the final version of the game would be more appropriate here, but we thought this would be more interesting.

[NEXT SLIDE]

Budget Legacy



Kenny:

We've been pretty strict with the budget to every game we've made. Rogue Legacy was no different, but we did come into it with a fair bit of experience. Here are a few key tips that we try to use on every game that we make.

[NEXT SLIDE]

Autonomy

Work with people who can be independent

Kenny:

Autonomy. Work with people who can be independent. This is important for a small indie team, because everyone wears multiple hats and it's tough to micromanage people and fix their mistakes. They gotta be able to do that on their own.

This has implications though, that not every developer would agree with.

This means our development process is not very democratic. Opinions matter, but when it comes down to it, the most trust is placed on the expert and nothing is decided by a vote.

[NEXT SLIDE]

Autonomy



Kenny:

A good example of this happened with our artist, Glauber Kotaki. We originally had a separate character and background artist, and I was stuck in the middle, trying to get their styles to work together. You can see from this picture how much the background and foreground clashed with the character art.

Things were clearly not working, and Glauber stepped up to fix it.

[NEXT SLIDE]

Autonomy



Kenny:

And this is the final product. What's amazing is that Glauber had very limited experience on backgrounds before. This was his first attempt at something of this scale.

Budget-wise, it's crucial to work with these kinds of autonomous people. For Rogue Legacy, we didn't catch on fast enough and it cost us. Before Glauber fixed things, 20% of the entire project's costs were spent on this problem. We went through many different artists.

[NEXT SLIDE]

Custom Editors

Make the editor for the game, not the company

Kenny:

The second rule we follow is to make a custom editor for each game. This isn't for everyone, but for us and the size of our projects, it always turns out cheaper and faster.

[NEXT SLIDE]

Custom Editors



Kenny:

The map editor Rogue Legacy can only do one thing: make rooms.

Because it's so rigid, it only took about 2 weeks to make. There were tweaks that came afterward, but the bulk of the work was within that time frame.

What was really great though was that we could create and bug test a room in as little as five minutes.

There won't be much from this we can salvage for our next project, but that's fine because the time saved creating content more than makes up for the code lost.

[NEXT SLIDE]

Design Alternatives

Always have design alternatives

Kenny:

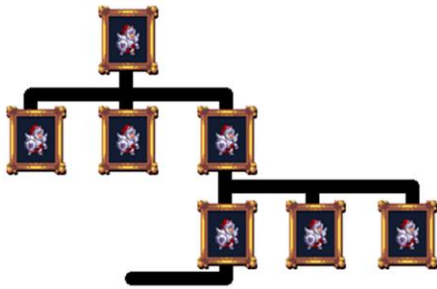
Always have design alternatives. This ties in with what we mentioned at the beginning of the talk, where we always look out for cheap, fast, and reusable solutions. It's usually necessary to come up with multiple ideas to find the one that best suits the resources you have.

[NO ADLIB]

[NEXT SLIDE]

Design Alternatives

LINEAGE TREE



LINEAGE LINE



Kenny:

When everything is categorized based off cost, it's a lot easier to gauge how far you can go with an idea.

For example, the original lineage screen was going to show your entire ancestral family tree. You could track spouses, children, branch offshoots, etc.

It was expensive, and we ditched it early for a lineage line that only shows your "preferred children". We lost some ornamental features with this change, but the idea of an expanding legacy was retained at a fraction of the cost.

[NEXT SLIDE]

Design Alternatives



Teddy: [NO ADLIB]

Another quick example of design alternatives is the skill tree economy. The choice we made for this was very contentious, and illustrates how budget can play a big role in the decisions we make.

In this case we actually chose a bad solution because it was so much cheaper.

The original skill tree economy was fairly straightforward. Every time you spent money to upgrade a skill, that particular skill's price would raise. It seems simple, but there's a hidden problem.

[NEXT SLIDE]

Player Level Over Time

Optimal Build
Without Price Modifiers



Teddy:

Since there was no way to save money in the game, because charon takes it at the beginning of every run, optimal skill builds always meant putting an equal number of points into each skill. So 10 points in health, then 10 points in damage, and so on.

In other words, no skill diversity.

We wanted to encourage players to personalize their builds, which meant there had to be consequences to player choice.

But harsh consequences are counter-intuitive to the casual theme, so the repercussions had to be very light.

[NEXT SLIDE]

Solution

- Buying a skill raised the price of all other skills by a small amount.

Teddy:

In the end we decided to use a universal modifier.

Every time you upgraded a skill, the cost of all other skills would go up by a very small amount.

[NEXT SLIDE]

Player Level Over Time

Optimal Build
With Price Modifiers



Teddy:

What this did was it deterred players from evenly leveling up their character throughout the game, resulting in more unique skill tree builds.

[NEXT SLIDE]

Solution

- Buying a skill raised the price of all other skills by a small amount.

Teddy:

Even though this solution worked, it was terrible for a lot of reasons.

[NEXT SLIDE]

Solution

- Buying a skill raised the price of all other skills by a small amount.

Cons

- Bad feel.
- More punishing towards casual players.
- Fix is apparent, but the problem is not.

Teddy:

It feels bad having everything go up in price.

It punishes casual players who blindly upgrade skills.

And the negative effects of the fix are noticeable, but the reasons for them are not.

[NEXT SLIDE]

Solution

- Buying a skill raised the price of all other skills by a small amount.

Pros

- Cost \$0.
- < 10 minutes to add.
- Minimal balance required.

Teddy:

On the other hand, the pros were too much to pass up.

This was a zero dollar solution. No art or audio was needed, and implementing it literally took less than 10 minutes.

And because it was so self-contained, only a minimal amount of time was needed to balance it.

Our game is full of these workarounds, some better than others. But they're a big reason why Rogue Legacy had such a small budget.

This example sort of encompasses our design mantra of "good, but not perfect."

[NEXT SLIDE]

Where do we go from here?

Teddy:

So where do we go from here?

We're now financially stable, and a lot of these budget workarounds might not be necessary anymore.

So the question is would we still do things this way?

And the answer is that we most likely will. There's more to it than just saving money. Finding simple solutions to big problems is useful in any scenario, and we often surprise ourselves with the things we come up with. For us it's also the most entertaining part of development because it's a challenge.

The End

QA - West Hall
Room 3000



Teddy:

Anyway, that's it. Thanks to everyone for attending this talk. We hope it was insightful and entertaining. Or at least one of the two.

If anyone has any questions, we invite you to join us in the wrap up room in the west hall, room 3000, in a couple minutes.

Kenny: Thanks everyone.

The clown will now see you out.

[END TALK]