

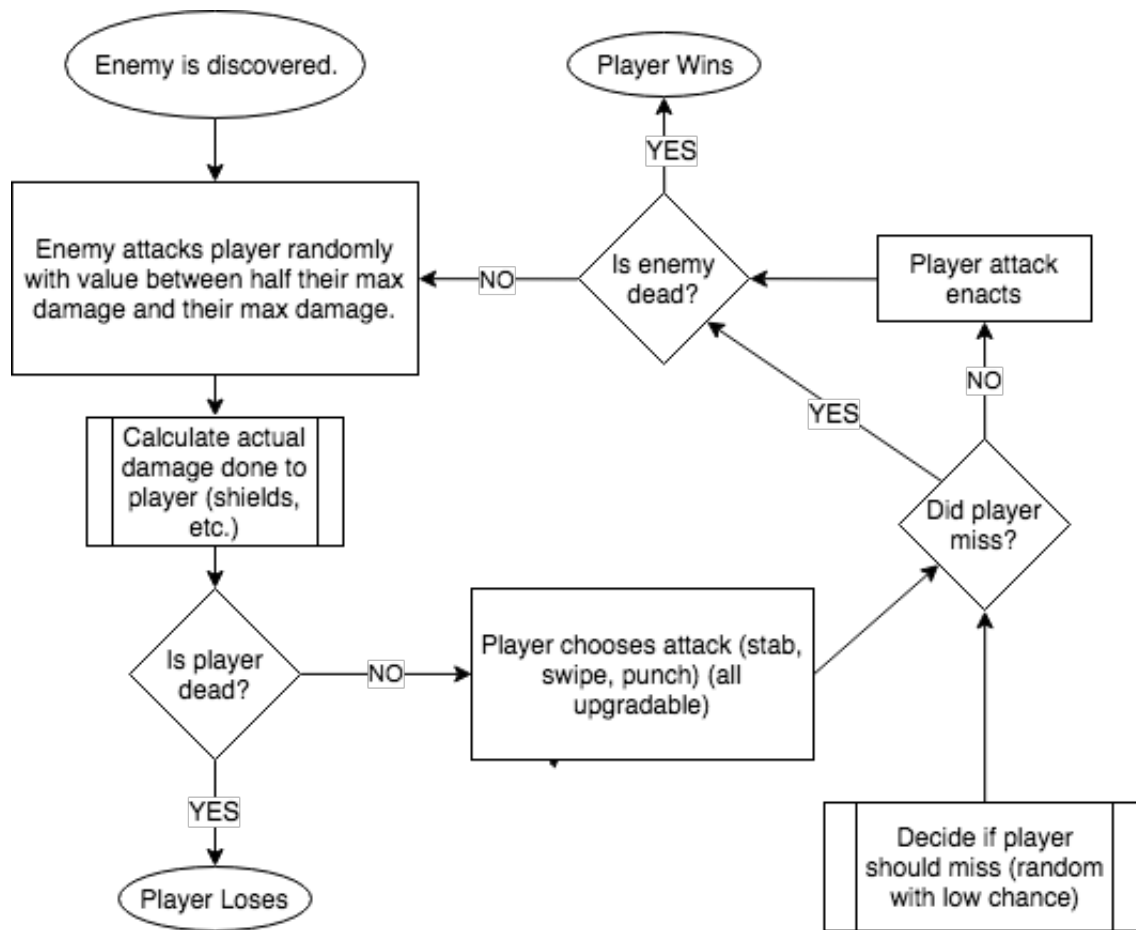
FuseBlade - Theory

Problem Definition

FuseBlade is a *Tile-Based RPG Adventure Game* That takes place on an island. The problem that this software solves is boredom, as is the nature of most games. To solve this problem we will make a coherent working game that will include the following features:

- A large 100x100 map loaded in from a file
- A spritesheet to handle all the textures
- An animated Character that can move
- Enemies
 - Enemy AI
 - Enemy stats and textures
 - random spawning of enemies
- Combat and Combat UI
- Leveling up
- Player and enemy death
- Player respawning and saving

System Flow Chart



Gantt Chart

Task	W1	W2	W3	W4	W5	W6
Problem Definition	<div><div></div><div>✓</div><div></div></div>					
System Flow Chart	<div><div></div><div>✓</div><div></div></div>					
Gantt Chart	<div><div></div><div></div><div></div></div>	✓				
Data Dictionary						<div><div></div><div>✓</div><div></div></div>
Setup and Framework Init		<div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div></div>	✓		
Map		<div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div></div>	✓		
Character Function			<div><div></div><div></div><div></div></div>		✓	
UI Popups				<div><div></div><div></div><div></div></div>	✓	
Combat					<div><div></div><div></div><div></div></div>	<div><div></div><div>✓</div><div></div></div>
Enemies					<div><div></div><div></div><div></div></div>	<div><div></div><div>✓</div><div></div></div>
Saving						<div><div></div><div>✓</div><div></div></div>
Development Journal	<div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div></div>	<div><div></div><div>✓</div><div></div></div>
Personal Reflection						<div><div></div><div>✓</div><div></div></div>

Data Dictionary

Field Name	Data Type	Data Format	Field Size	Description	Example
PLAYER_SPEED	Float	(primitive)	8 bytes	Player movement speed in pixels/second	20
CAM_ZOOM	Float	(primitive)	8 bytes	Camera Zoom Level	5
CAM_MODE	String	(primitive)	6 bytes	Camera follow type that determines how the camera will	"smooth"

				act	
CAM_LIMIT	Float	(primitive)	8 bytes	The camera pan distance limit	100
FIGHT_BOX_WIDTH	Float	(primitive)	8 bytes	Width of the fight box UI	100
FIGHT_BOX_HEIGHT	Float	(primitive)	8 bytes	Height of the fight box UI	70
ENEMY_COUNT	Float	(primitive)	8 bytes	Amount of enemies that will spawn on the map	100
ENEMY_AGRO_DIST	Float	(primitive)	8 bytes	Base number for how far away the player needs to be from an enemy for it to start following the player	5
ENEMY_SPEED	Float	(primitive)	8 bytes	default enemy movement speed in pixels/second	10
PLAYER_ENABLED	Bool	(primitive)	8 bytes	Whether the player can move or not	True
DEBUG_ENABLED	Bool	(primitive)	8 bytes	Whether the debug tools are enabled or not	False
PLACEMENT_TILES	Array(int)	[NNN, NNN, ...]	120	The locations that enemies can spawn	[123, 123, 321]

				(max 15 tiles)	
ENEMY_TYPES	Array(string)	[WORD, WORD, ...]	80	The list of different enemy types (max 8 enemies)	["funny", "words", "here"]
ENEMY_DATA	Object	{ enemyName: { <i>stats: { speed:</i> <i>enemySpeed,</i> <i>sight:enemySight,</i> <i>attack:enemyAttack},</i> <i>frame:</i> <i>enemyFrameName},</i> enemyName2: {...}}	10	The object that contains all the data and stats of all the different enemy types	{"Skeleton": stats: {speed:10, sight:5, attack:5}, frame: "skeleton"}

Personal Reflections

Marcus

Overall I thought the project went better then expected, we managed to make it look nice and something to be proud of, however we did have ambitious goals at the beginning that were not helped by a large portion of the time allocated to us working on the project was dedicated to trying to get the excalibur.js framework working, which in the end we ended up switching frameworks to phaser.js 3. Because of this setback, towards the end of the project we started to just build the project to the specifications instead of to our original plan. This focused our work efforts but because of the lack of features compared to what we had originally planned and that I have become quite fond of this project, I will most likely continue working on it after it is handed in as I would like to see it completed with the full list of features.

Ben

This has been one of my favourite projects ever, and I have really, really enjoyed making this. At first I was keen for a group project, then realised how my somewhat perfectionist/controlling nature could morph this into something where I would take on all the responsibility, crashing, and eventually burning, BUT in the end I was able to make sure this didn't happen.

I was actually quite surprised with how well we were able to work together across different branches and completely different structures, but I think with both of us fully having our head around the structure of the game was a huge help, and allowed us to completely understand each other, all the time. This coupled with

Marcus' amazing ability to write bulletproof, accessible, well-thought out and quite frankly beautiful classes made this project so much easier and more fun.

My only wish was that we had discovered Phaser.js earlier! Something I wish I hadn't done was persist on staying with Excalibur.js for so long, as we lost a precious week or three there :/. This caused us to have to grind hard, real hard, in the last week and was a push for both of us. We did manage to get it done through some hours-long discord calls and constantly shooting ideas back and forth.

Next time, I'd like to make sure I pick a robust, well-documented and example-heavy library, as this will make the whole thing a *lot* easier. As well as this, a better plan into how exactly our codebase is to be structured would be very beneficial, and assist us in not ending up with 30+ primitive global variables.

This is a project I have enjoyed so, so much, and I know that I will continue to develop it with Marcus from here on out. It's gonna become more than just another *siDe pROjEcT*.