

</CODE BLOCKER>



Not Fast, Just Furious

Powered by



Ludicrous Engine

Revision 1.0.0

Overview:

Who are we?

What is the Ludicrous Engine?

Working Title

Genre

Core Gameplay

Targeted Platform

Project Scope

Influences

Elevator Pitch

What sets this Project apart?

Core Gameplay Mechanics

Story

Assets Needed:

- 2D
- Sound
- /CODE

Schedule:

- Milestone 1
- Milestone 2
- Milestone 3
- Milestone 4

Overview:**Who are we?**

Naveen Prasad, Colin Pugh, Christian Lee

What is the 'Ludicrous Engine'?

It's designed to power awesome 2D platformers with DirectX.

Working Title:

</CODE BLOCKER>

Genre:

Survival Platformer

Core Gameplay:

You will play as "char acter;", a variable tasked with making it to the end of the "PrgRAM" a virtual simulation created by "The Creator". Along the way the player comes across bugs of various forms that it must avoid or defeat with his "Debugger" tool. At the final stage of the "PrgRAM" the player must fight through an "Exception E" boss to successfully compile. The player will only have 1 health and must dodge all attacks to survive. Upon death "The Creator" must click recompile to respawn "char acter;" so that it can restart its journey. The game will have 8-bit graphics and music and include enormous amounts of programming puns.

Targeted Platform:

This game is designed to be played on the PC using DirectX.

Project Scope:

- Built in Visual Studio 2017 with D3D12
- 2D Graphics
- Single Player

- Played with Keyboard
- Enemy AI
- Simple HUD to display health
- Audio
- Will feature Collisions & Gravity
- All assets will be created by Team: Not Fast, Just Furious

Influences:

- I Wanna Be The Boshy
- MegaMan
- NES Super Mario Bros

Elevator Pitch:

</CODE BLOCKER> is a 2D platformer where the player is a variable in a program that goal is to be compiled. Through his journey he comes across various bugs that try to attack him and he must either dodge their attacks or debug them (shoot them). The game will have a classic retro feel with 8-bit graphics.

What sets this Project Apart?

Core Gameplay Mechanics:

Mechanic 1:

- The player will only have 1 health, and this is because he is a variable created by “The Creator” and if destroyed by the bugs in the “PrgRAM” the simulation ends and “The Creator” must recompile the program again which sets the player at the initial position. The player has to dodge abilities thrown at them from all directions in order to make it through successfully. This is because it’s a test of individual skill (Player reaction times and speed).

Mechanic 2:

- The game will feature enemies and boss type enemies. The enemies and bosses will drop a variety of different power-ups that include double jump, double firing etc.

Mechanic 3:

- During play, the player will come across Ctrl+S savepoints which they can choose to activate in order to save their progress. The player will be able to respawn here if they die.

Story Characters (Subject to Change / Concept Art):

Player is called “char acter;” who holds a debugger weapon that shoots debug commands to destroy enemies.

“The Creator”

The Creator is the player who is playing the game and will be controlling “char acter;”. His goal is to see that his creation is successfully compiled with no compilation or runtime errors.

The “PrgRAM”

A program written by the creator where all the story characters exist.

Enemies:

Small Bug: Tries to run into player. Can be destroyed in 5 Debug commands.

Medium Bug: Can take up to 10 Debug commands from the player. Stationary but fires NULL_POINTERS towards the player

Big Bug: Can take up to 15 Debug commands from the player. Also fires NULL_POINTERS in all directions.

Argument: Can take up to 60 Debug commands. Hitting weak spots deals double damage but is only visible for a few seconds.

Destroying this boss rewards player with [Ctrl-S] power up.

Exception E: The final boss that has a shield that the player must TRY to throw first by hitting it while avoiding errors. When shields are down the “Exception E” will start throwing errors in all directions that the player must dodge. “Exception E” will use sin and cos waves to throw errors. Will also spawn bugs periodically.

Power-Ups:

Double: Player can double-tap.

Float: Player can double jump.

Ctrl-S: The player will respawn at Argument defeat location and not at the start of the game.

Assets Needed:

2D

- Character Sprite
- Enemy Sprites
- Power-Ups
- Environment
- Textures
- Projectile Sprites

Sound

- 8-bit themes
- Boss music

- Menu music
- Shooting clip
- Enemy sounds

/CODE

- Character scripts
- Enemy / Boss scripts
- Level Managers
- Sound Managers
- ETC

Schedule:

- Milestone 1
- Milestone 2
- Milestone 3
- Milestone 4