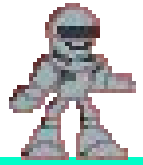
Drexel Game Devs



Meet A.D.A.M.!

Team Members and Roles 

Joseph Lipinski - Project Lead, Level Designer

Sarah Kushner - Game Artist

Zachary Lopez - Technical Lead

Matthew Freihofer - Player Programmer/Physics Programmer

Patrick Nwanah - A.I. Programmer

Cory Cellucci - Programmer

Problem to be addressed

Bored? Do you have nothing to do? Especially not reading assignments or homework assignments during finals week?

Instead of being bored, why not play our game!?

Broader Impact

As addressed in our Developer Documentation, this game is easily extensible! To anyone who wants to create a game in Python (specifically Pygame), this is a great way to start!

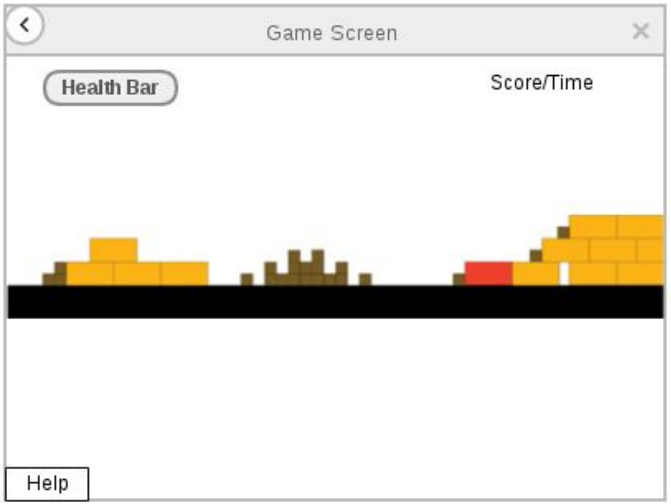
Each member of our group also had the chance to try completing a task they had never done before this class! Very exciting!

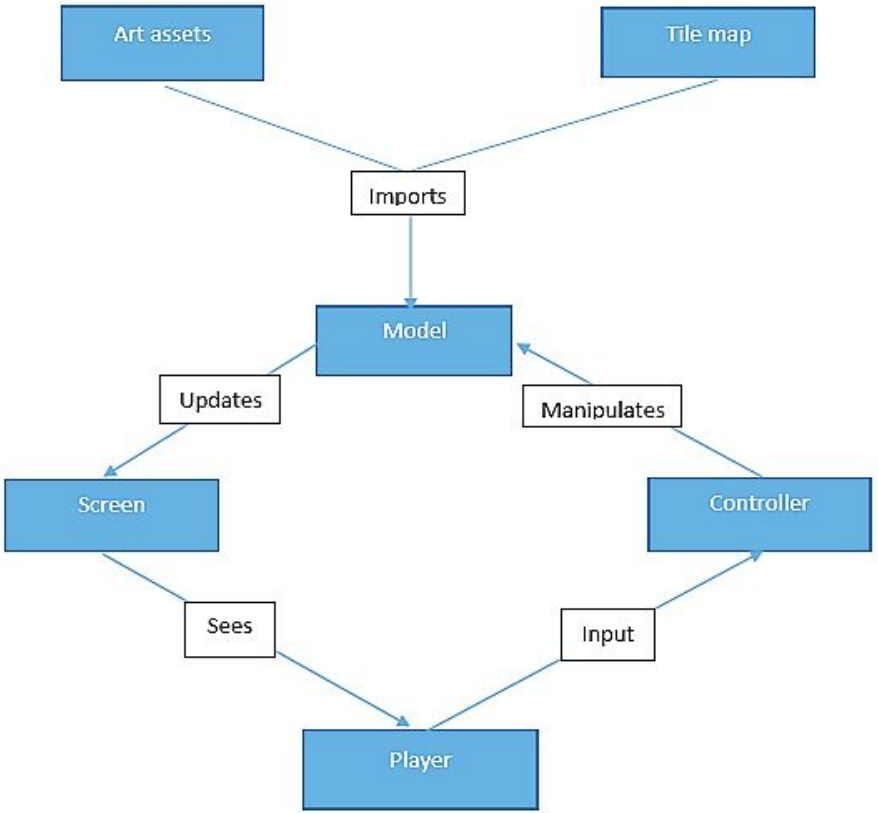
What We Did

- Procedural level generation
- A.I. behavior responsive to player’s position
- Physics affects enemies and player alike
- 2D platforming and side-scrolling shooting
- 8-bit art style
- Scoring system

How We Did It

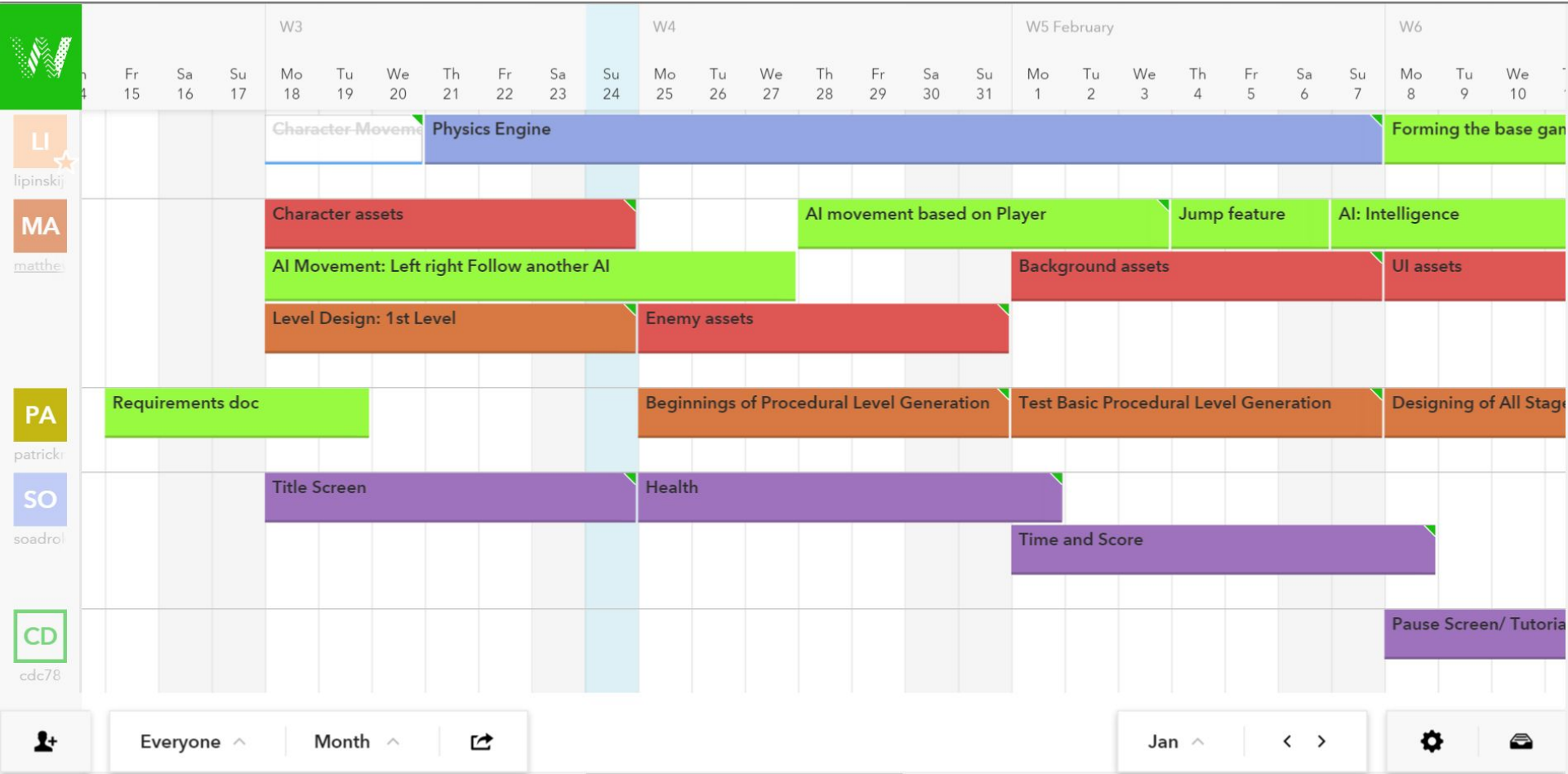
Design

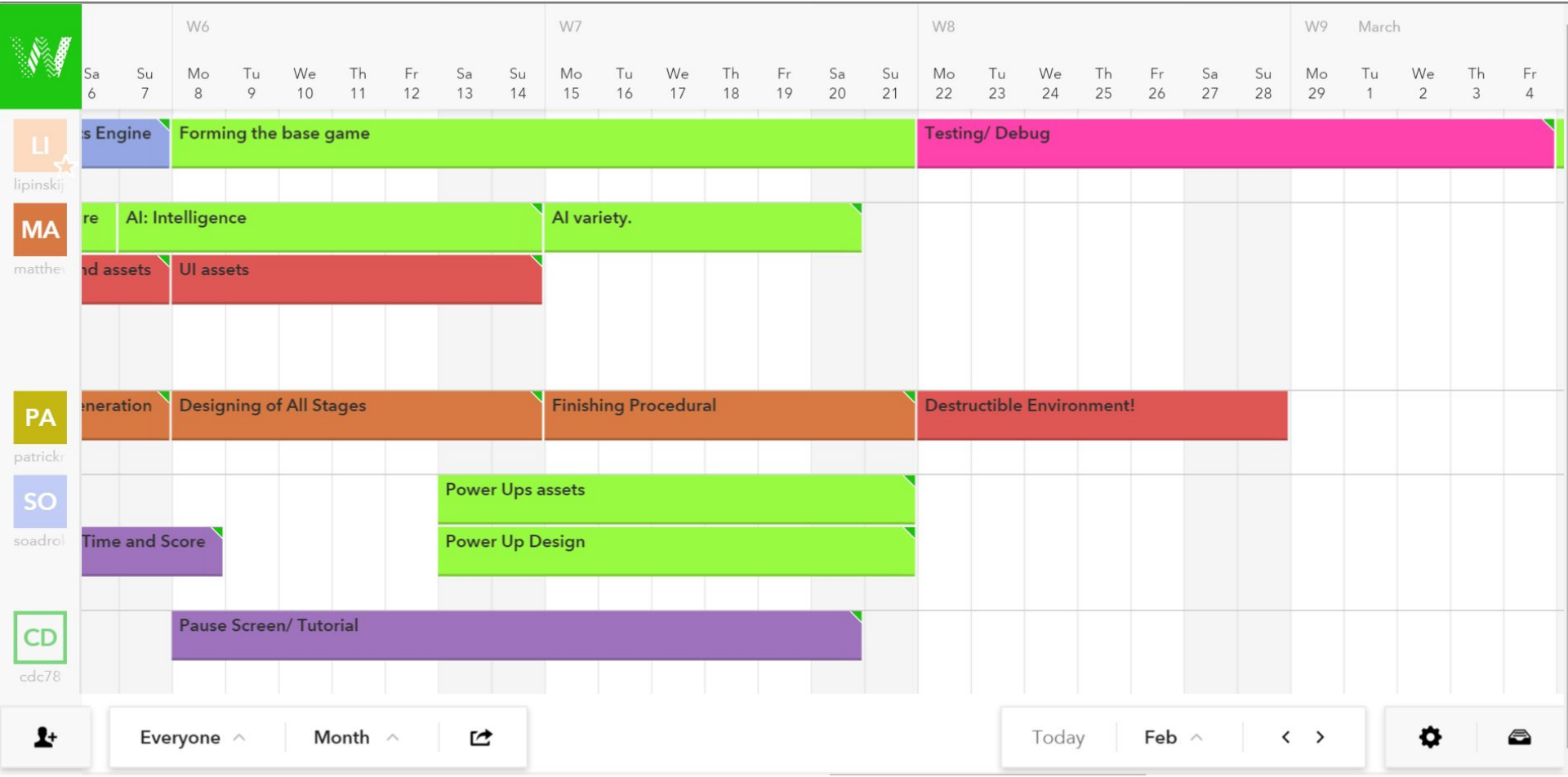


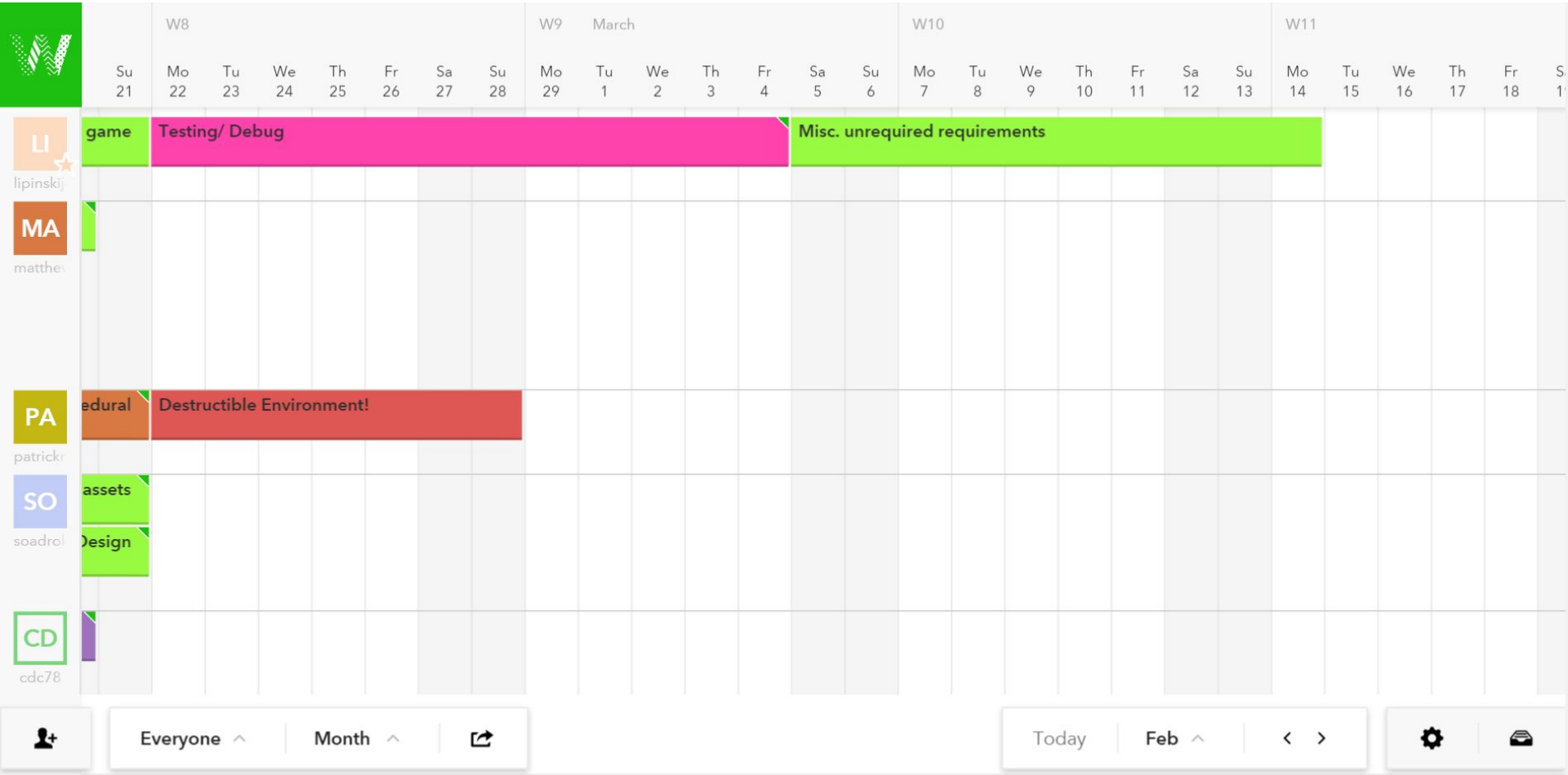


Plan

- Our design document and team roles made for convenient independent work
  - Easier to collaborate by combining parts in the end
- Gantt Chart to show responsibilities, timeline, and dependencies
- Incorporate feedback from playtesting


W3
W4
W5 February
W6
Fr 15
Sa 16
Su 17
Mo 18
Tu 19
We 20
Th 21
Fr 22
Sa 23
Su 24
Mo 25
Tu 26
We 27
Th 28
Fr 29
Sa 30
Su 31
Mo 1
Tu 2
We 3
Th 4
Fr 5
Sa 6
Su 7
Mo 8
Tu 9
We 10
LI
lipinskij
Physics Engine
Forming the base gan
MA
Character assets
AI movement based on Player
Jump feature
AI: Intelligence
AI Movement: Left right Follow another AI
Background assets
UI assets
Level Design: 1st Level
Enemy assets
PA
patrickr
Requirements doc
Beginnings of Procedural Level Generation
Test Basic Procedural Level Generation
Designing of All Stage
SO
soadrol
Title Screen
Health
Time and Score
CD
cdc78
Pause Screen/ Tutoria
Everyone
Month
Jan


W6
W7
W8
W9
March
Sa 6
Su 7
Mo 8
Tu 9
We 10
Th 11
Fr 12
Sa 13
Su 14
Mo 15
Tu 16
We 17
Th 18
Fr 19
Sa 20
Su 21
Mo 22
Tu 23
We 24
Th 25
Fr 26
Sa 27
Su 28
Mo 29
Tu 1
We 2
Th 3
Fr 4
LI
lipinskij
s Engine
Forming the base game
Testing/ Debug
MA
matthe
re
AI: Intelligence
AI variety.
nd assets
UI assets
PA
patrickr
eneration
Designing of All Stages
Finishing Procedural
Destructible Environment!
SO
soadrol
Power Ups assets
Time and Score
Power Up Design
CD
cdc78
Pause Screen/ Tutorial
Everyone
Month
Today
Feb


W8
W9
March
W10
W11
Su 21
Mo 22
Tu 23
We 24
Th 25
Fr 26
Sa 27
Su 28
Mo 29
Tu 1
We 2
Th 3
Fr 4
Sa 5
Su 6
Mo 7
Tu 8
We 9
Th 10
Fr 11
Sa 12
Su 13
Mo 14
Tu 15
We 16
Th 17
Fr 18
LI
lipinskij
game
Testing/ Debug
Misc. unrequired requirements
MA
matthe
PA
patrickr
edural
Destructible Environment!
SO
soadrol
assets
Design
CD
cdc78
Everyone
Month
Today
Feb

How It Actually Went

Some setbacks:

- General time management
- Pygame/Tiled/PyTMX version issues
  - Led to a plan change
  - Needed another feasible way to procedurally generate levels
- Slower progress than expected, but we still did it!

SetBacks Elaborated: Tile mapping

Original Plan was to use Tiled/PyTMX for streamlined Map Creation. Issues with this are listed below:

- PyTMX requires specific versions of Pygame (2.7/3.2)
- PyTMX requires specific Python libraries (libraries not inherently apart of 2.7 or 3.2).
- PyTMX requires specific Pygame libraries (libraries that were discontinued after Pygame 2.7).
- Pygame 2.7 binaries do not mesh with 3.2 binaries
- The list goes on... You get the picture.

Spent a week sifting through issues. Abandoned the idea. Why?

- Too much dev time wasted. We cut our losses. Looked for another solution.

SetBack solution: Tile mapping

- Much of our original code was still useful, thanks to modular design.
- We looked for solutions online that seemed to use our individual modules.
- Found a plethora of tutorials. We stitched our code together following their format, and was able to produce the final product.

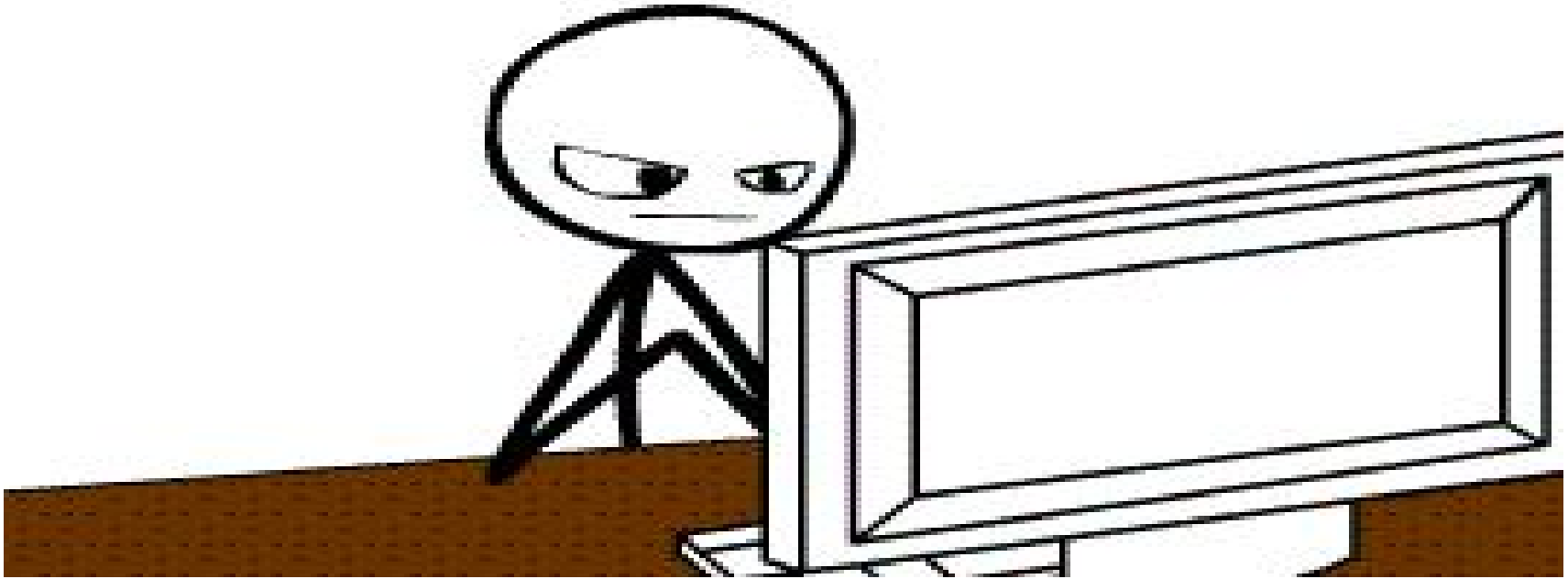
SetBacks Elaborated: A.I.

- Integrating A.I. into the system caused problems with the way the world shift (side scrolling) was implemented.
  - Dependent on the x position of the character
  - Character walk cycles would only work when the screen did not shift forward (i.e. getting to the edge of the screen, walking backwards)

SetBacks Elaborated: Time constraints

Not all the art was integrated into the final system due to time constraints. So, here it is: the city level.

SetBack Summary



Lessons Learned

1. We all developed a better understanding of Python, and Game Development in general.
2. We learned a hard lesson regarding reinventing the wheel.
    a. We set out to individually develop our own AI system; our own Physics engine; our own Procedural generation.
        i. While this, without a doubt, gave us a better understanding of each topic, we may have learned more if we had used other tools instead.
            1. HINT HINT: UNITY!!

3. All of us now know the value of documenting and planning (Thank you Professor Mongan :D !).

Demo!