

POLARIS: THE 4 CARDINALS

CHI IEONG NG, , TOMMY DINH, KENNETH WOO, IVAN HERNANDEZ, JOSUE FLORES, JOSUE REYES

DEPARTMENT OF COMPUTER SCIENCE

CALIFORNIA STATE UNIVERSITY, NORTHRIDGE

COMP 491 Final Presentation



Outline

- Introduction
- Background knowledge
- Related Work
- Design
- Implementation
- Demo
- Results
- Conclusion and Future Work
- References

Introduction

- Polaris: The 4 Cardinals is a Role-Playing Game with top down RPG and platformer elements using Unity Engine
- Video games can build positive contribution and have positive effects
- Video games can be a great tool and provide an environment for players to learn a specific field of knowledge. [6]
- Game development is actually very beneficial and some of the major keys in whether students can learn well with this method is based on teamwork, a student background, teacher requirements and time constraints and workload. [10]

Presenter: Chi leong Ng

Background Knowledge



Unity Engine



Platformer game



Top Down RPG

Related Work - 2D Game Design Software

- Level Design in 2D Games
 - Guidance - Nonverbal guidance to tell the player where to go.
 - Safe Zone - Places of respite
 - Foreshadowing - A concept where a mechanic gets introduced to a player, then makes that mechanic more complex.
 - Layering - Combination of mechanics to create a new experience.
 - Branching - Branching paths,



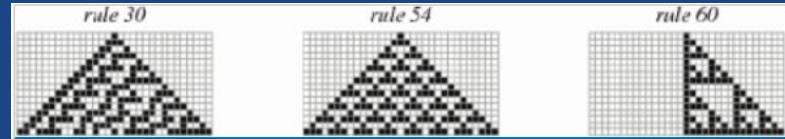
[4]



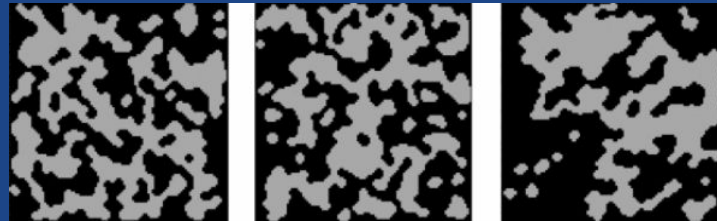
[4]

Related Work - 2D Game Design Software

- Procedural Generation
 - Creating procedural levels is efficient, but lacks practical use.



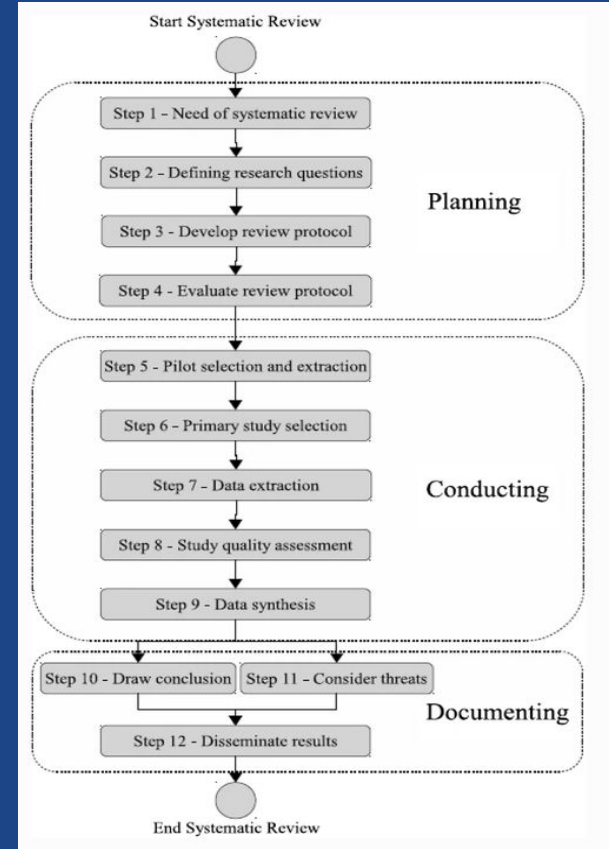
[5]



[5]

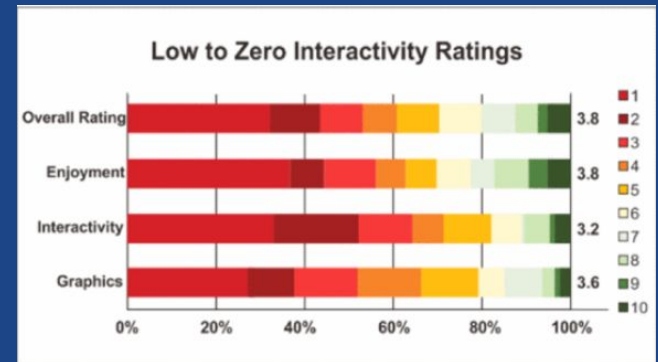
Video Game Based Learning

- Process for game development is different than that of programming.
- Goal is to examine what needs more attention while creating video games.
- The hope is that better games can be made.



Game Refinement Settings on Idle Games

- Studies games such as Cookie Clicker to see how much interaction is indeed for a player to enjoy a game.
- Games with little interaction from the player aren't enjoyable.
- Math was used to try and find the optimal interaction between the player and the game for maximum enjoyment.



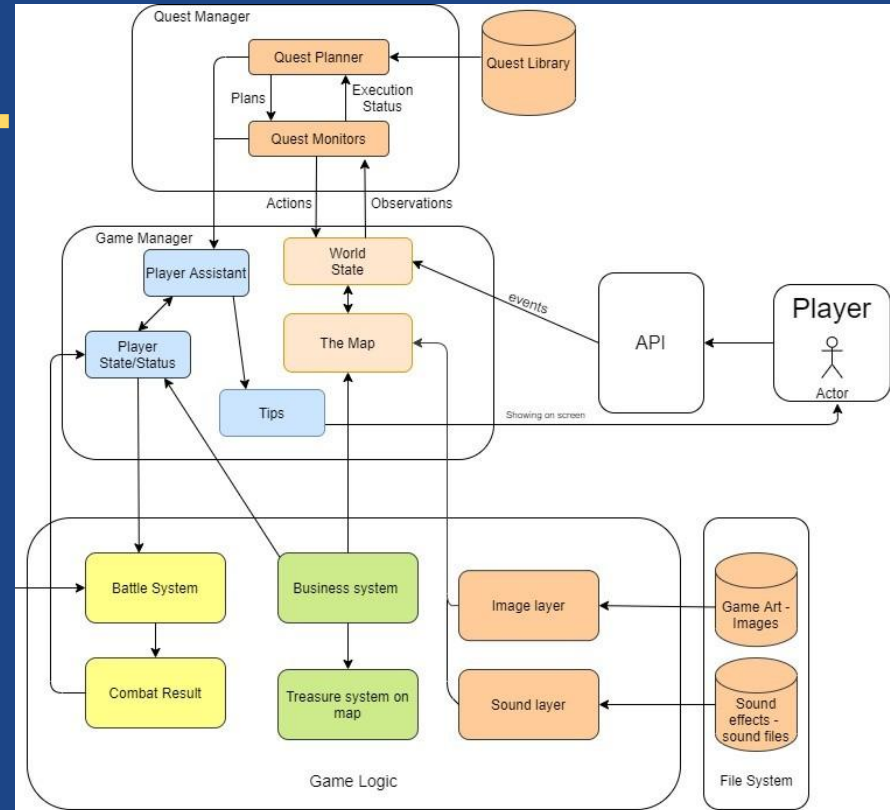
Goals

- Created an entertaining RPG game with story and challenging battles
- Designed interesting characters
- Top-down map and scroll vision(platformer) battle system
- Quest system
- Inventory, Item and equipment system
- Combat and skill system
- Multiple behaviour enemy (scripted AI)
- 4 dungeons
- 4 Bosses

Design Game Manager

- **World State and Map**
 - Loads locations and activities.
- **Player Status**
 - Shows the objectives and status of player characters.
- **Player Assistant**
 - Holds player tips and shows relevant info for the area
 - Takes input from quest planner and player state.

Presenter: Ivan Hernandez

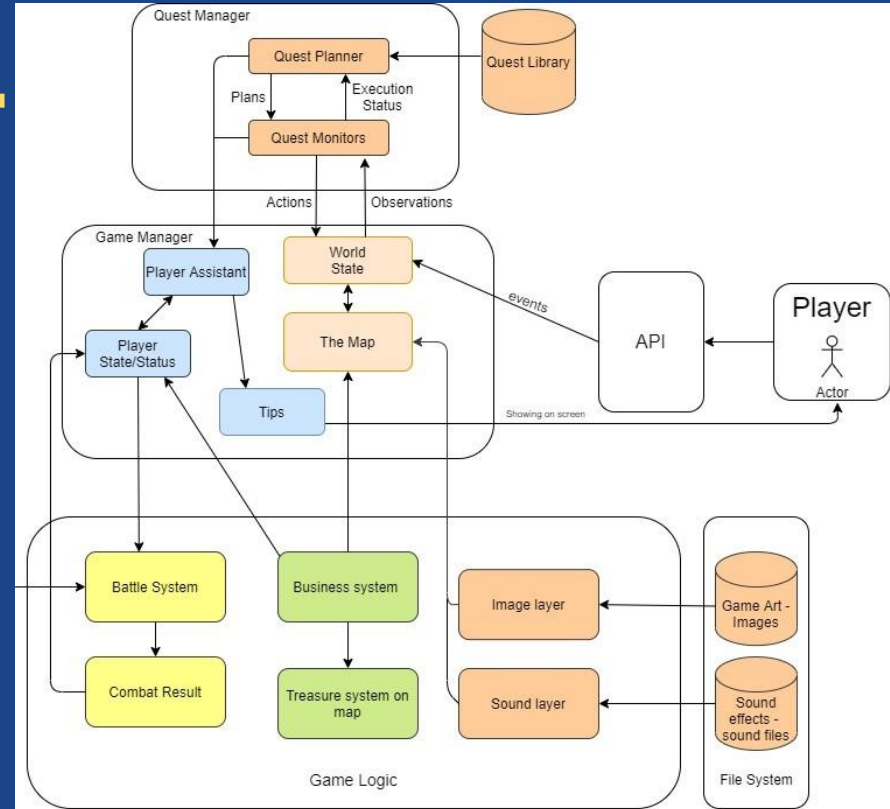


The middle section of the framework. Which the player directly interacts with the most.

Design Quest Manager

- **Quest Library**
 - Database for quests.
- **Quest Planner**
 - Reads and distributes quest information to player and quest monitor.
- **Quest monitor**
 - Communicates with Game Manager on quest activation, progress, and completion

Presenter: Tommy Dinh



The top section of the framework. Primarily controls events that occur on the world map such as quests and dialogue.

Design Game Logic and Enemy Manager

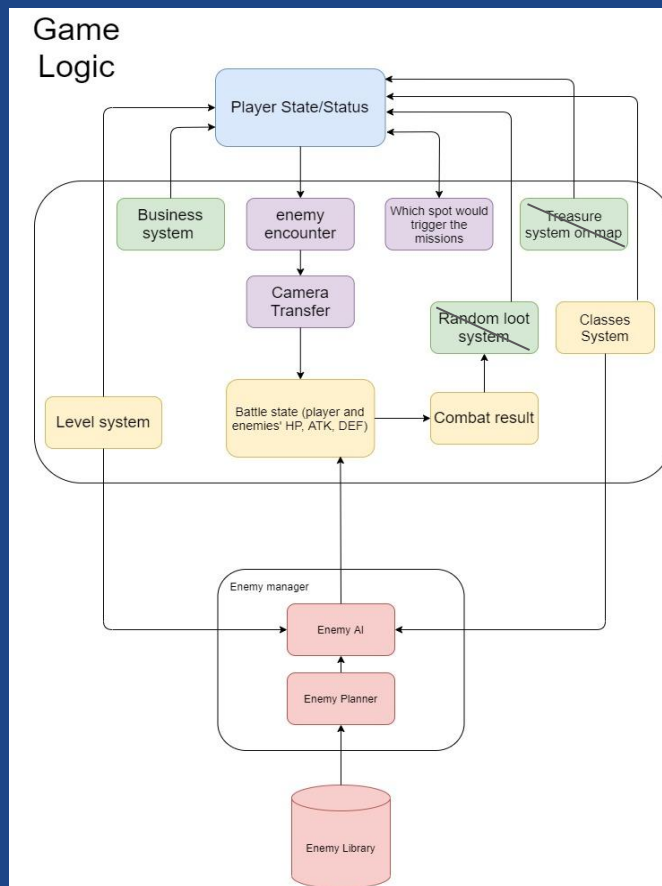
Business system - Item/Equipment stores

Quest system

Level system

Enemy State Machine

Presenter: Chi leong Ng



Middle part is the logic between Player and other interactive objects,, and bottom part is the enemy logic which is the state machine

Design Player

- 2d sprite
- Movement
 - Walk
 - Jump
 - Ledge Climb
- Skills
 - Dash
 - Double Jump
 - Increased Jump Height
 - Damage Multiplier

Presenter: Josue Reyes



Design Combat

Presenter: Josue Reyes

- Sword
 - Close Quarters Combat
 - High Risk High Reward
- Bow and Arrow
 - Ranged Combat
 - Deals less damage than sword



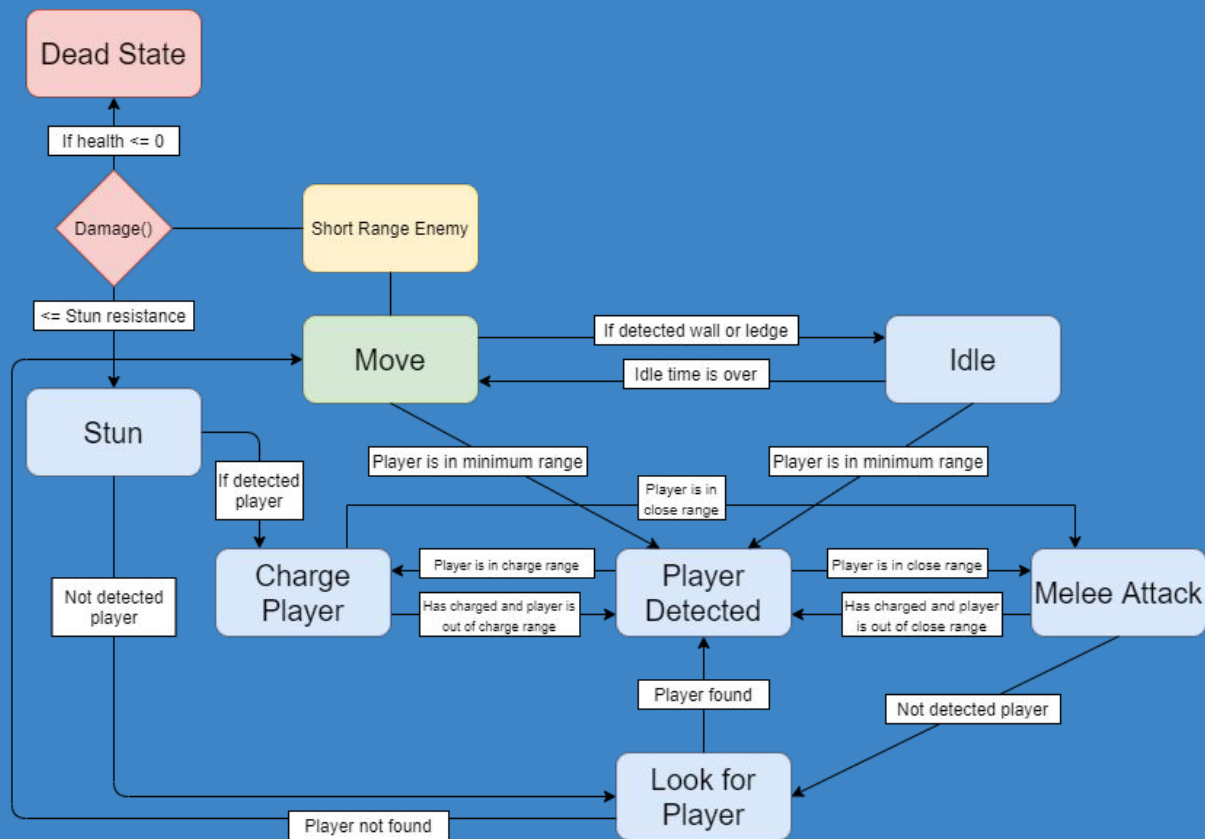
Implementation

- Our implementation
 - We all contributed to the design of the map
 - We then split our work evenly, with everyone taking a major component of the game for themselves
 - Kept constant communication with each other to make sure we were on track to finish
 - Met every friday to discuss progress and more ideas

Implementation

Presenter: Chi leong Ng

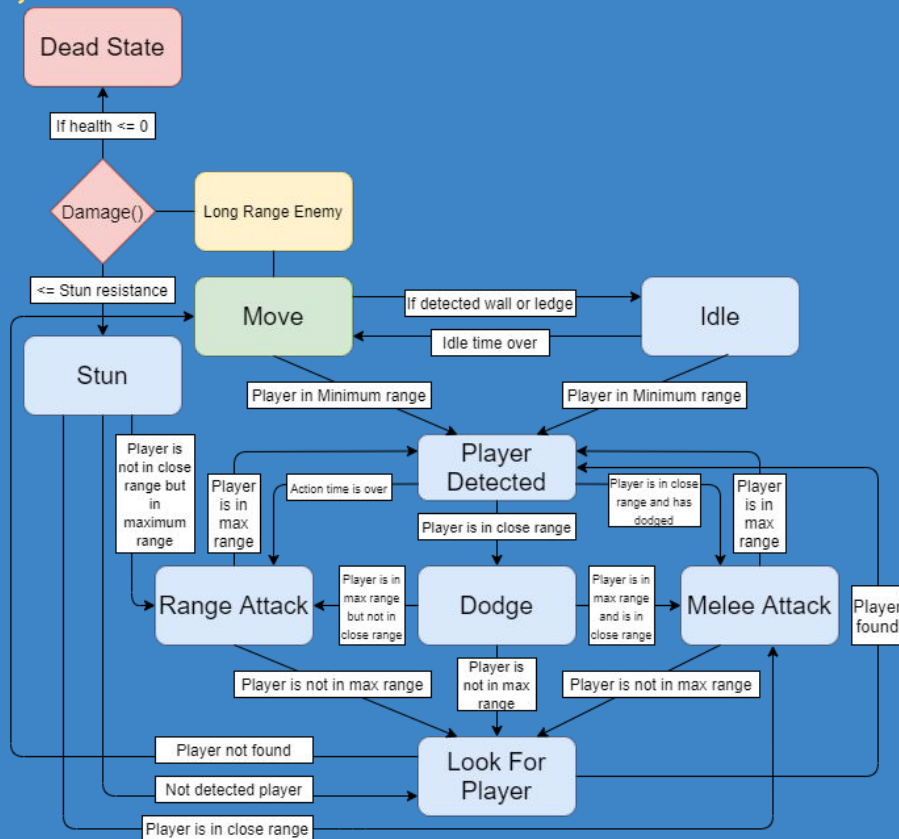
- Short Range Enemy State Machine



Implementation

Presenter: Chi leong Ng

- Long Range Enemy State Machine



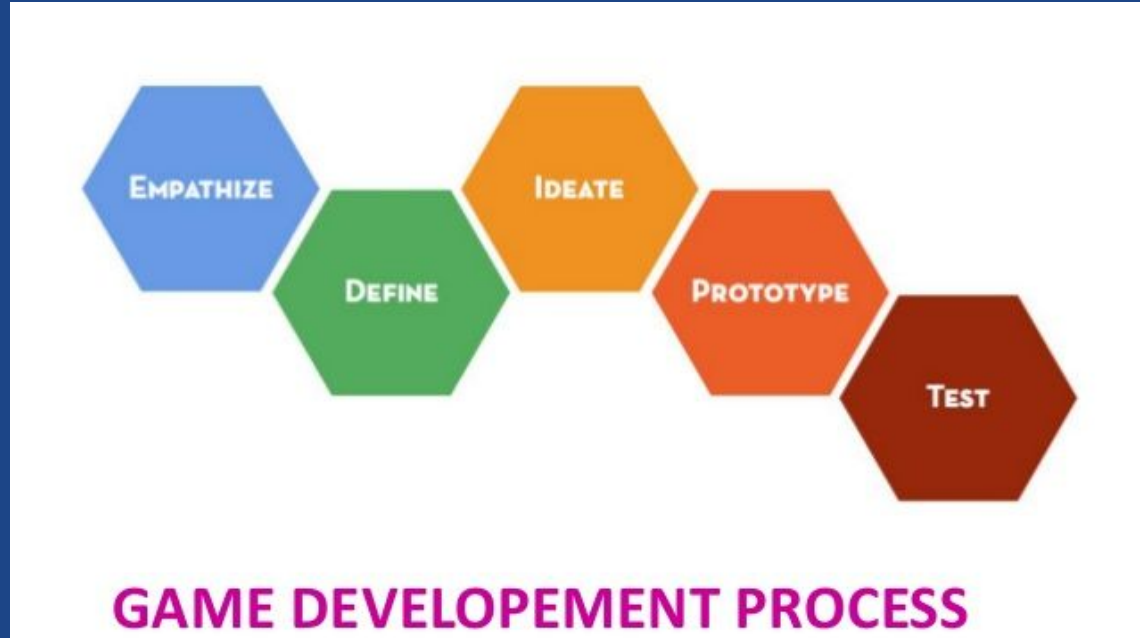
Demo of Your Project



Results

Process

1. Idea creation
2. Discussion
3. Preliminary implementation
4. Testing
5. Decision
6. Re-implementation
7. Final Discussion
8. Removal or Finalization



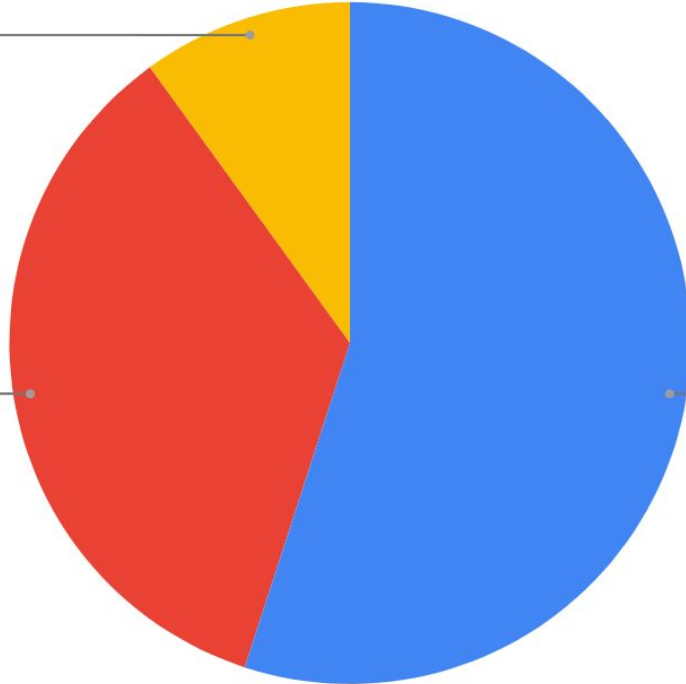
Data of Users Opinion: Platforming

Count of Platforming

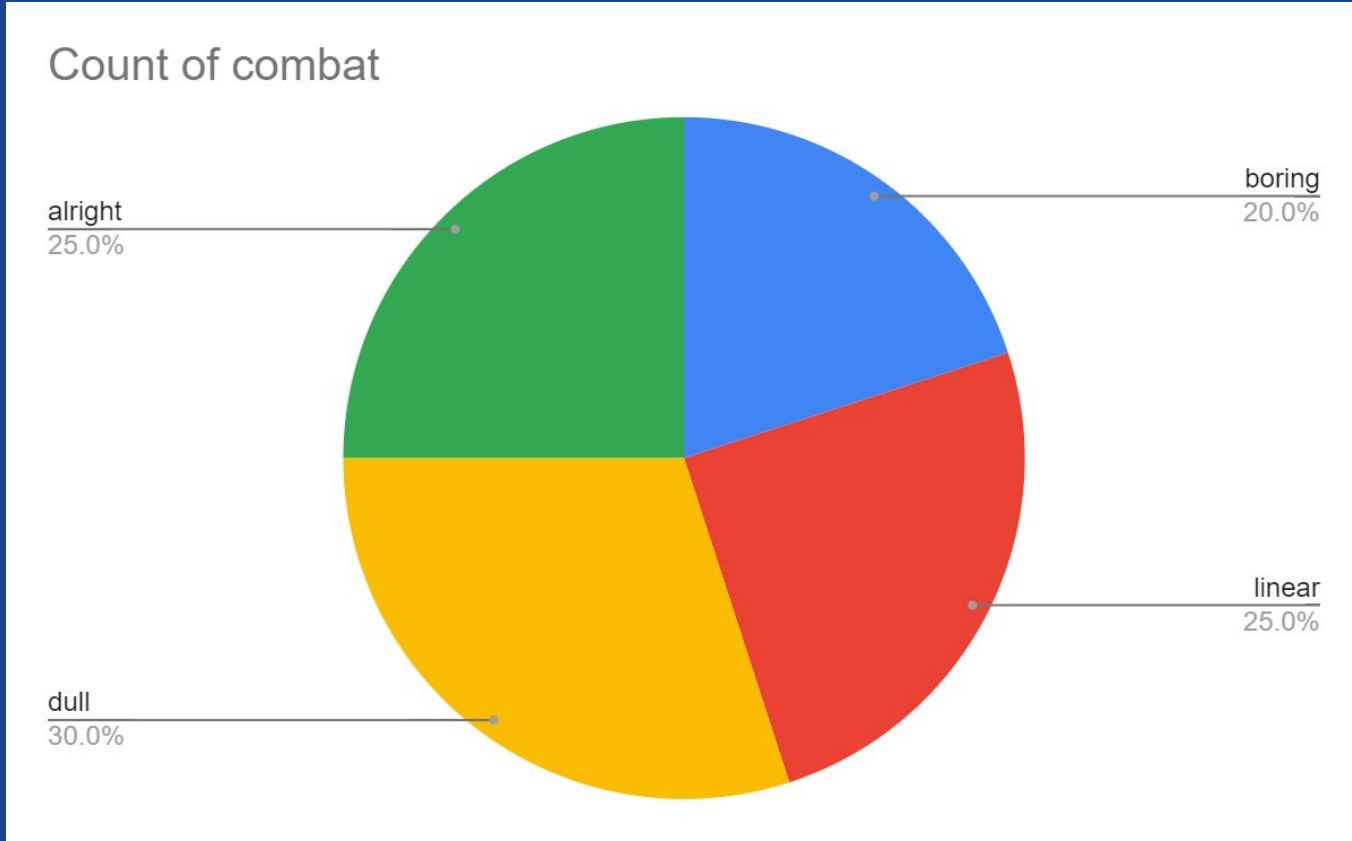
alright
10.0%

too precise
35.0%

difficult
55.0%

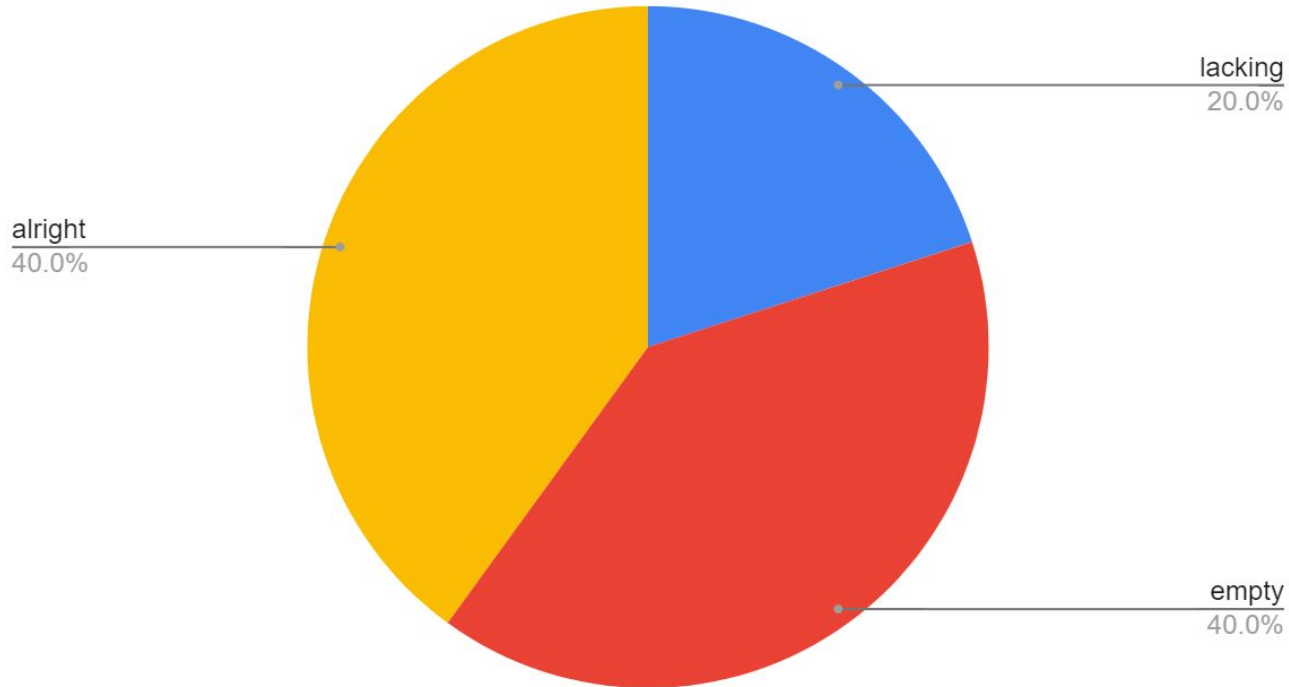


Data of Users Opinion: Combat



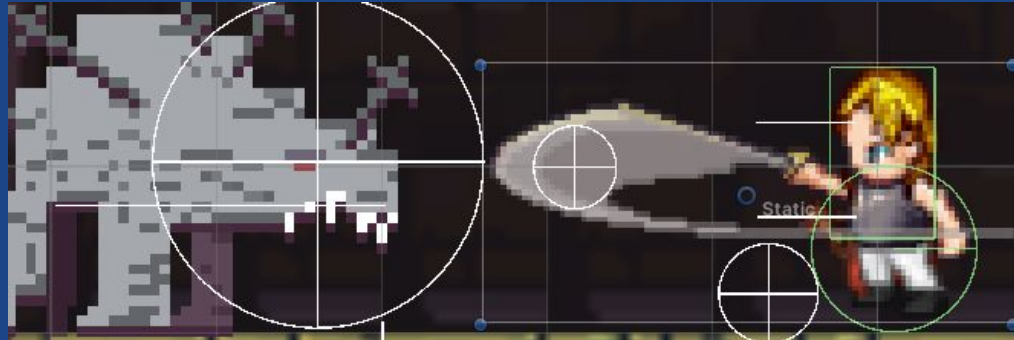
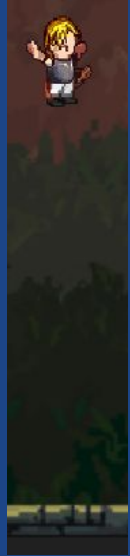
Data of Users Opinion: World

Count of world



Gameplay Evaluation

- Trial and Error
 - In-Game testing
 - Required no errors between game objects
- Features
 - Improve
 - Create
 - Modify
- Adjustments
 - Team Experience
 - Enjoyment
 - Fairness



Presenter: Josue Flores

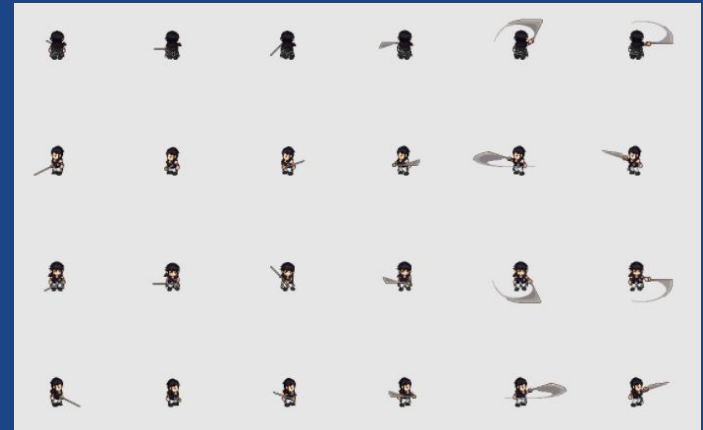
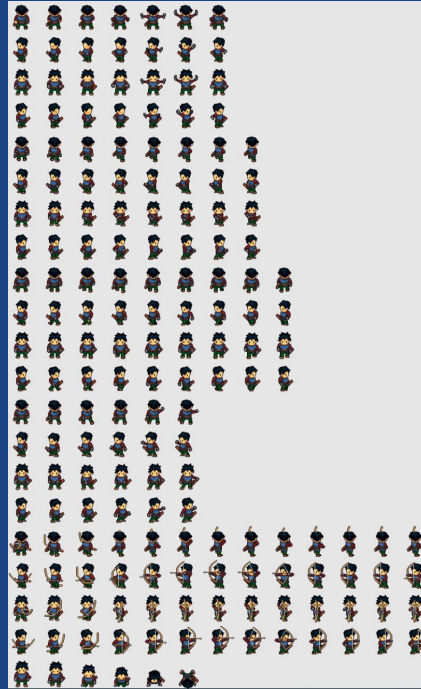
Conclusion

- Game development
 - It evolves
 - Creative freedom
 - Unity engine
 - Provides challenges
 - Learning experience
- Game Design
 - World exploration
 - Story telling
 - Interactive
 - Fun



Future Work

- Future Additions
 - Party System
 - Fleshed out Story
 - Additional Characters
 - Additional Skills
 - Ways to acquire skills



References

[1]

Aleem, S., Capretz, L. F., and Ahmed, F. Game development software engineering process life cycle: a systematic review. *Journal of software engineering research and development* 4, 1 (2016), 1–30.

[2]

Chover, M., Marín, C., Rebollo, C., and Remolar, I. A game engine designed to simplify 2d video game development. *Multimedia tools and applications* 79 , 17-18 (2020), 12307–12328.

[3]

Jacob, L., Clua, E., and de Oliveira, D. Oh gosh!! why is this game so hard? identifying cycle patterns in 2d platform games using provenance data. *Entertainment computing* 19 (2017), 65–81.

[4]

Khalifa, A., de Mesentier Silva, F., and Togelius, J. Level design patterns in 2d games. In *2019 IEEE Conference on Games (CoG)* (2019), IEEE, pp. 1–8.

[5]

Macedo, Y. P. A., and Chaimowicz, L. Improving procedural 2d map generation based on multi-layered cellular automata and hilbert curves. In *2017 16th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)* (2017), IEEE, pp. 116–125.

[6]

Rankin, Y. A., Gold, R., and Gooch, B. 3d role-playing games as language learning tools. *Eurographics (Education Papers)* 25 , 3 (2006), 33–38.

References

[7]

Sazaki, Y., Primanita, A., Hashim, S. Z. M., and Ayu, I. D. Game refinement settings on idle games. In 2019 IEEE 5th International Conference on Wireless and Telematics (ICWT) (2019), IEEE, pp. 1–6.

[8]

Thominet, L. Games we envision: Rhetorics of potentiality in open video game development projects. In 2019 IEEE International Professional Communication Conference (ProComm) (2019), IEEE, pp. 141–144.

[9] Vogel, M. W. Japanese independent game development. PhD thesis, Georgia Institute of Technology, 2017.

[10]

Wu, B., and Wang, A. I. A guideline for game development-based learning: A literature review. International journal of computer games technology 2012 (2012), 1–20.

Q & A

