

INFINITY TRAIN

DEV BOOK

Concept:

Roguelike game set in infinite train. Concept is based on the Adventure Time cartoon episode titled Dungeon Train.



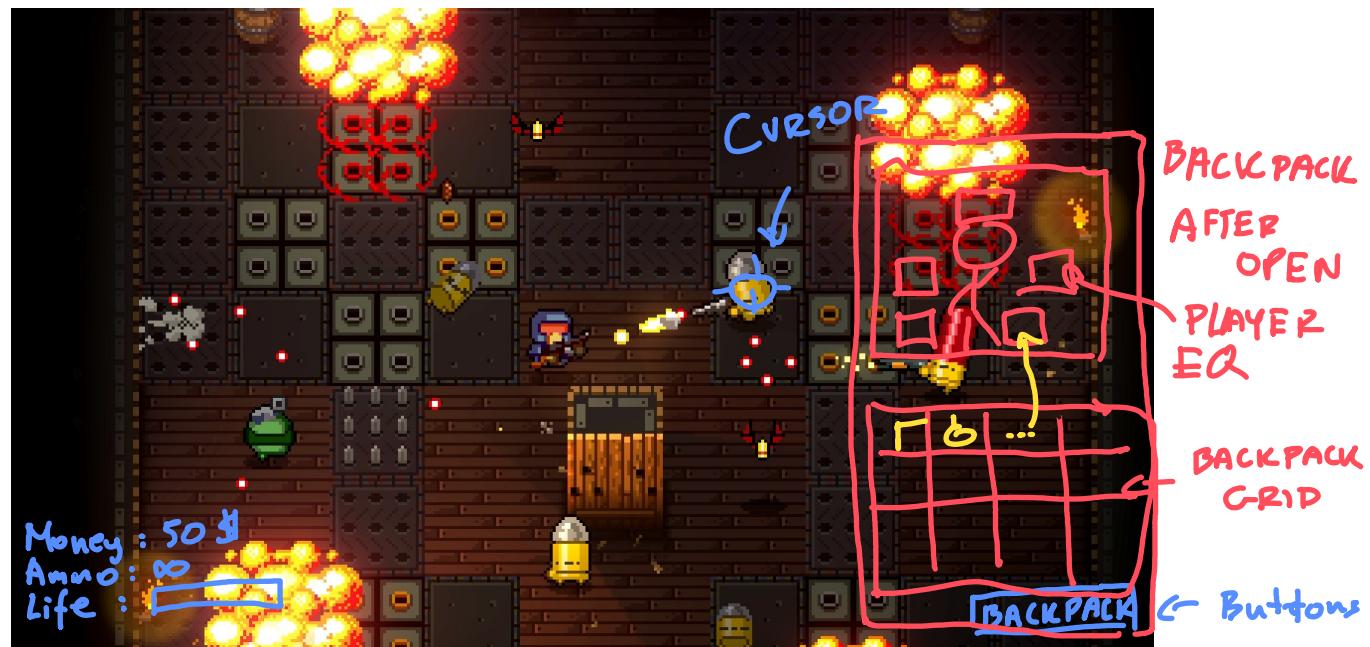
Mechanic:

Game will be the most similar to another rogue like game Enter the gungeon. It will be have similar movement style and shooting mechanic, but i want another mechanic for weapon. In my mind weapon will have some 'lifetime' like ammunition or wear, and I want cold armors and potions. Every item will be stored in backpack and you will be able to change what you currently using. In that case items will be very recently droped from opponents.

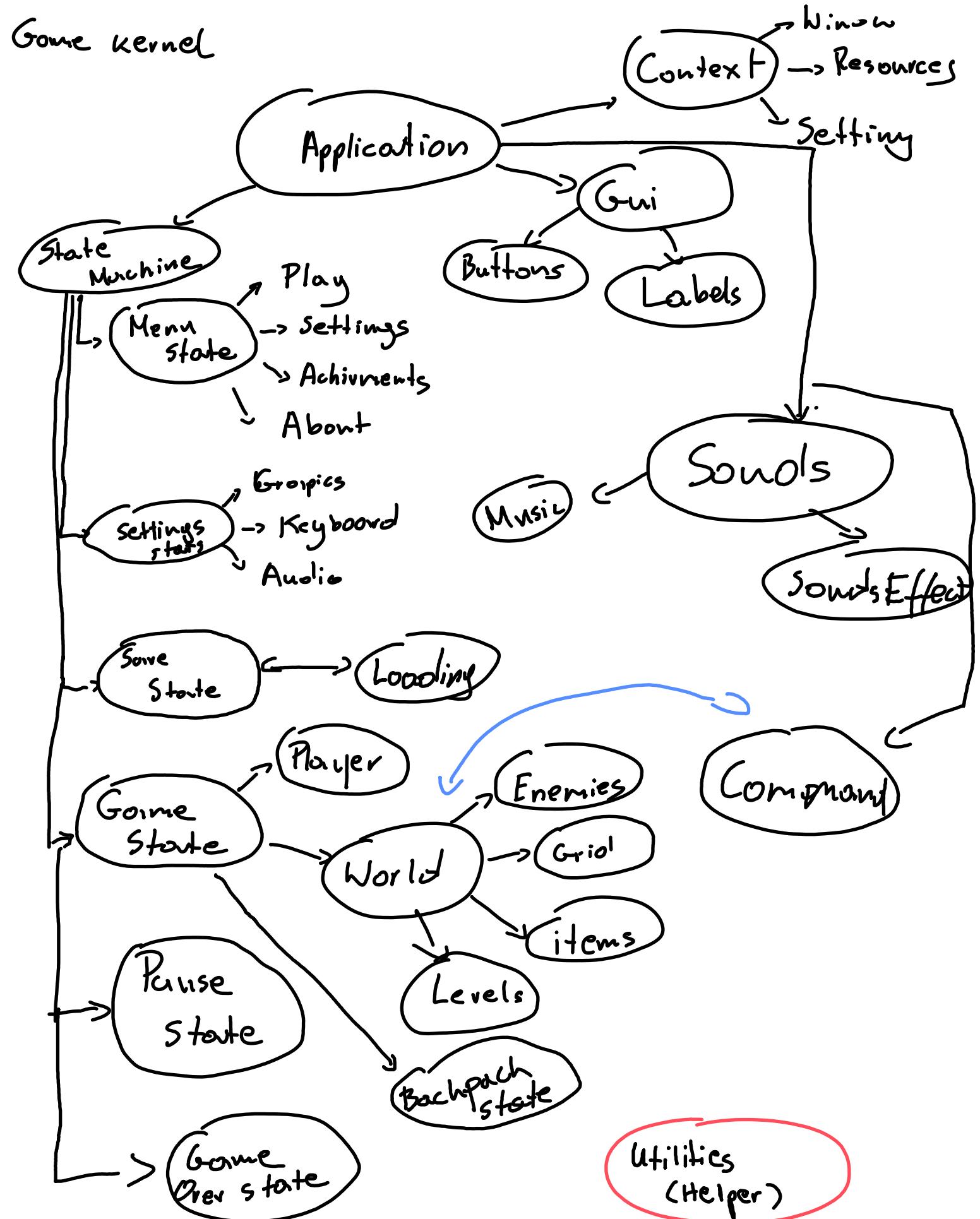
Levels (for start) will be similar to each other (same size of room) maybe with style style change. You will be come from bottom, destroy every monster in room and leave on top. There will be special rooms like shop, treasure room and boss room.

There will be few monsters. Each monster will have specific items they can drop. Level ends with boss room with special drop item

Example look:



Screen from Enter the Gungeon



MENU, SETTINGS, ACHIEVEMENT, ABOUT

SCREEN

INFINITY TRAIN



SOME GRAPHICS

MADE
IN SFML

Every thing here is simple. We have 4 buttons which everyone go to other state: Play - Game State, Settings - Settings State etc.

Settings is very similar only with options to change things. Achievements and About there is only text so it's not a problem.

Settings

INFINITY TRAIN

Settings

Resolution
Graphics → Fullscreen

Control → More

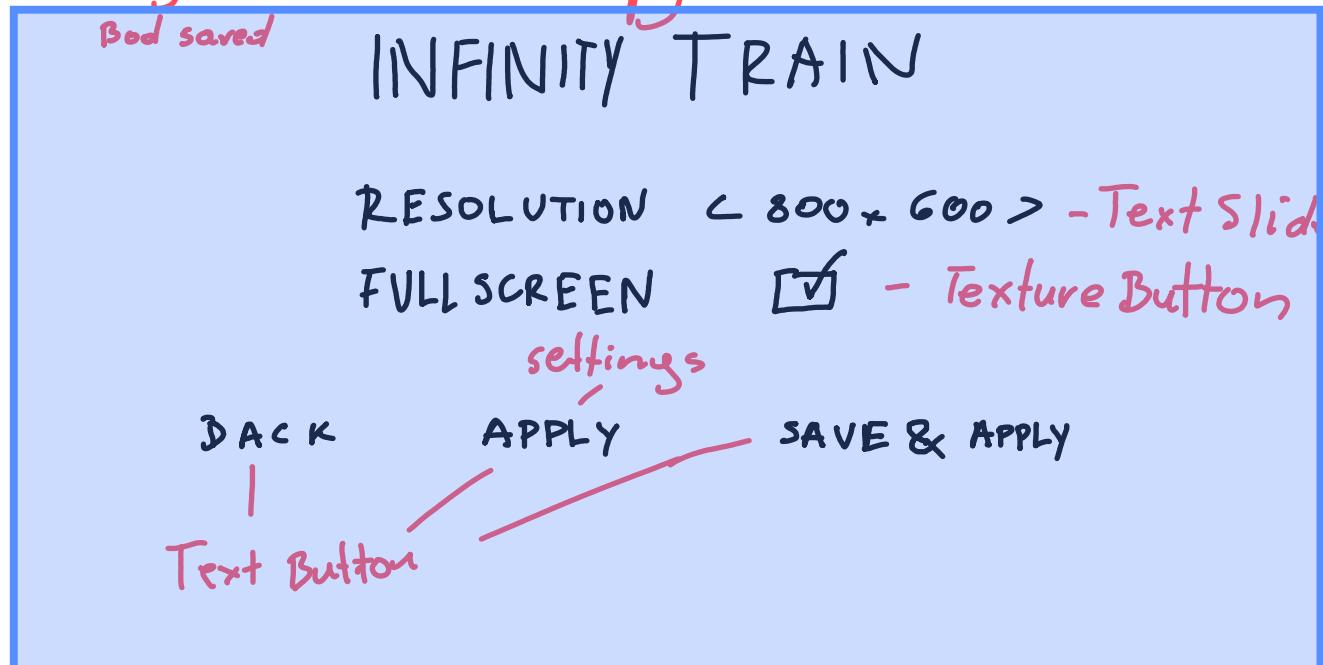
Audio → Sounds Volume
Music Volume

Save State

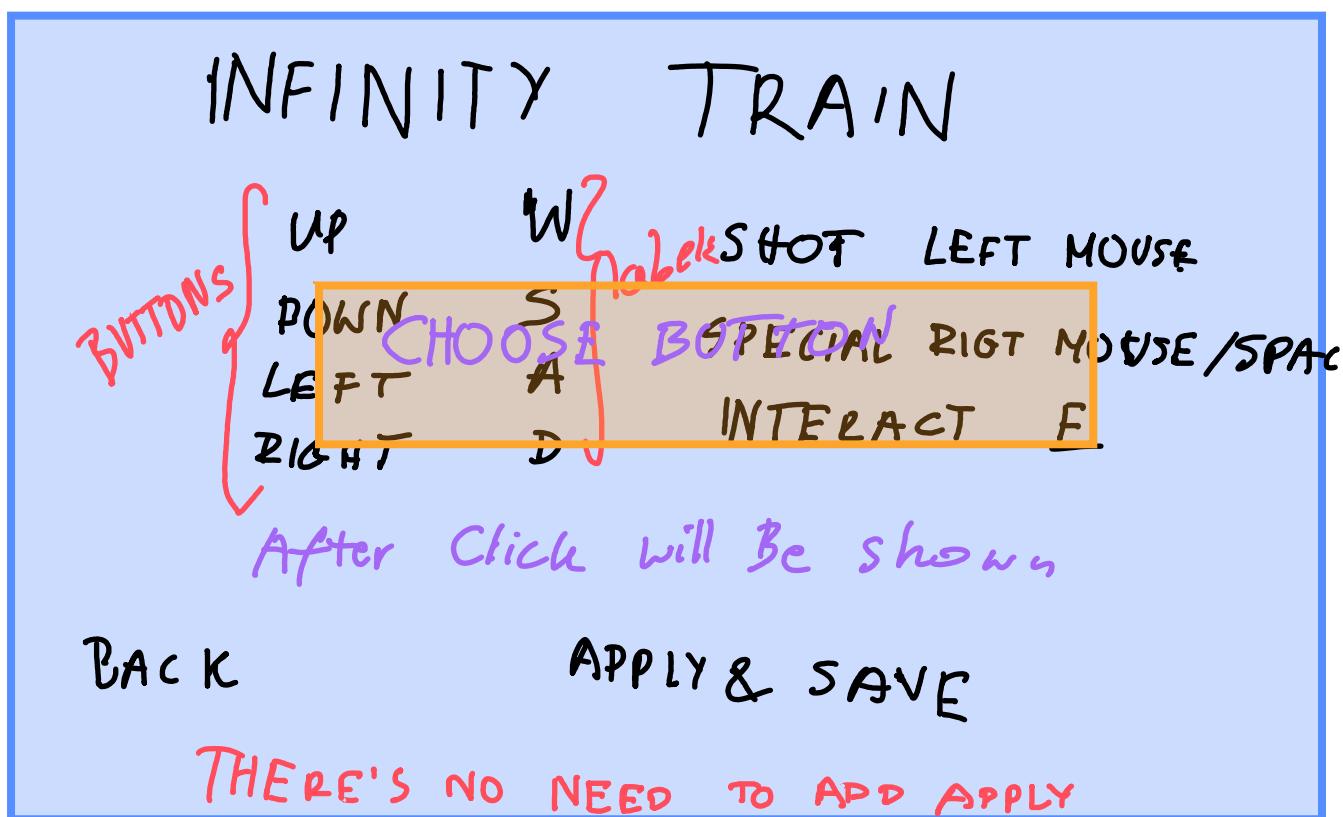
← BACK

Save state
save options
to settings
struct which
is in context

Look of Setting e.g. Graphics / Audio Settings is similar
After coming to this state copy settings

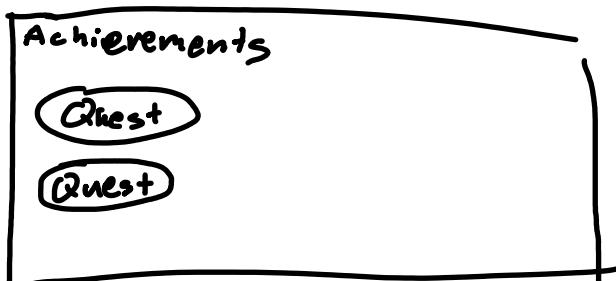


Control Settings



Context split to apply Graphics settings, Audio settings etc!

Achievements



How it work

In context we have achievement class

If we do something in game we have to refer to specific achievement

context.achievements.get(id).countUp()
or setAchieve

Examples of achievements

- Start the game
- Kill 1000 enemies
- Kill 10000 enemies
- Finish game once
- Finish game 10 times

| It's sub achievement
Can we combine?

Can be combined with statistic instead achievements because every data can be pull up from statistic

So if we will have statistic class with:

- killed enemies
- number of started game
- number of dead
- number of finished game

We can also store some special achievement but not showing it in statistic

Next idea: Replace achievement with statistic in menu, I think it will be better because have

Similar mechanics and will be a little easier

Statistic class



START

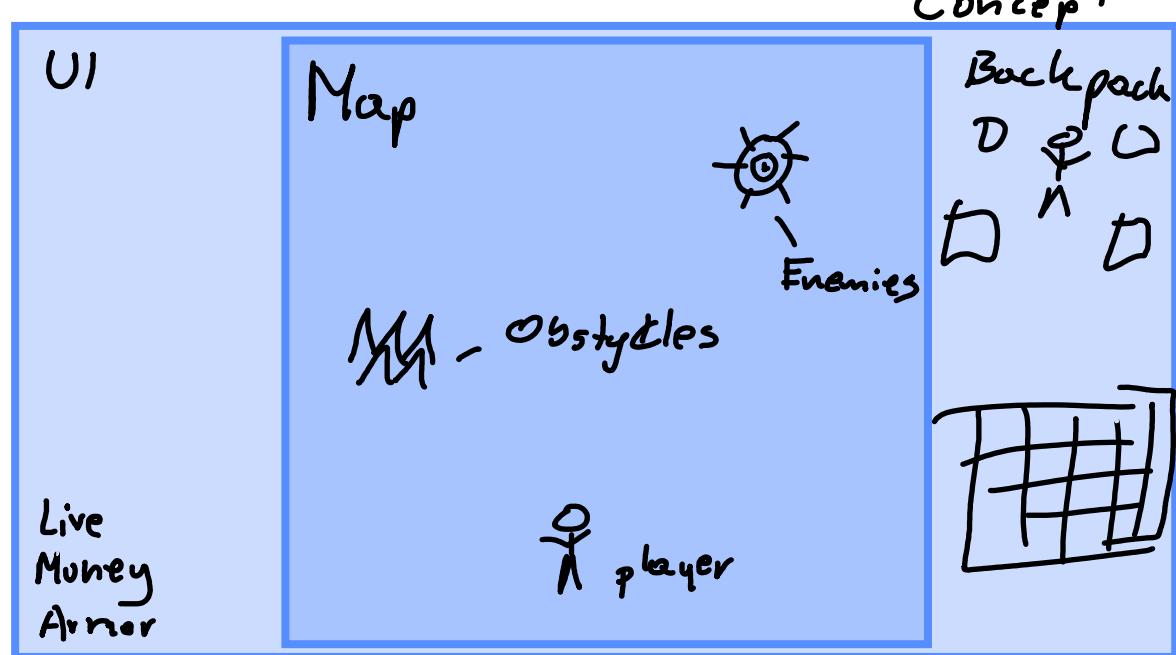
GAME

There are 3 things that need to be covered:

- | Level | Players | Enemies |
|-----------------|---------------|------------|
| - Map | - Mechanik | - Mechanik |
| - Drop items | - Interaction | |
| - Switching Map | | |

We start with simple level then creating Player at the end we create Enemies

Creating Level



At first we create map without obstacles and enemies, only with player-map collision

Player



Texture

Will be split



Body

- Hand
+ Weapon

Hand

Legs

- Legs
will have
animation

when
player
walk

Body will be static only
will be invert
depend if mouse is at left side of player or right side

Level - virtual class

level will be split to type levels:

- prehistory
 - Ancient
 - Medieval
 - Early modern
 - Present
 - Future
-

How level works

- Create enemies
- Open doors when kill every enemies
- Place player sprite
- Collision with wall and enemies

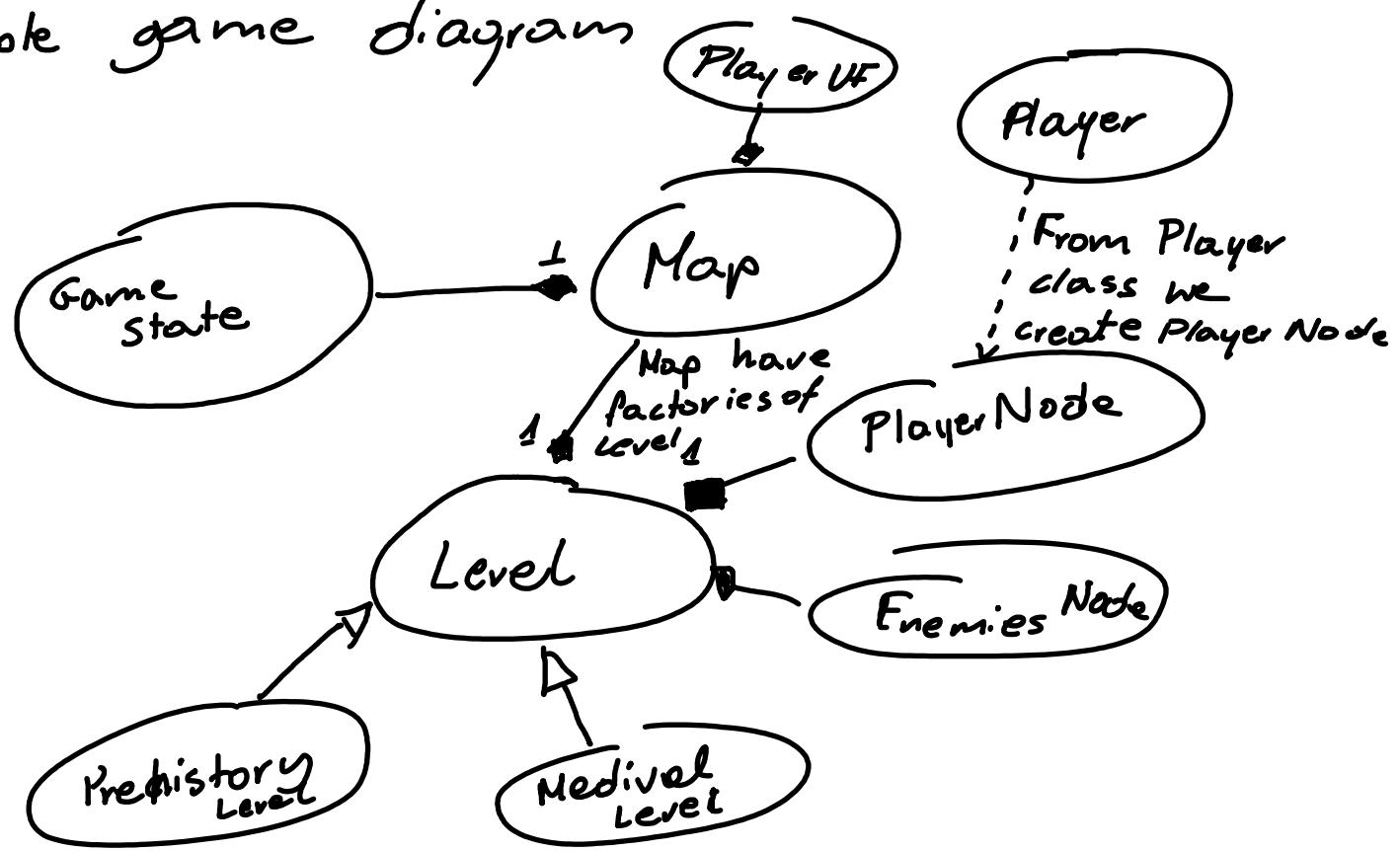
Contain:

Player, Enemies, WorldBounds, CommandQueue

Level number

Creating: Factories by Type.

Simple game diagrams



Player and Player Node

Class Player

- Have information between Levels (live, equipment, money)

Class Player Node

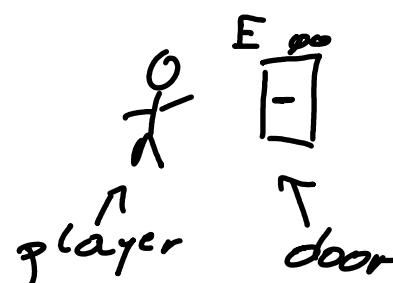
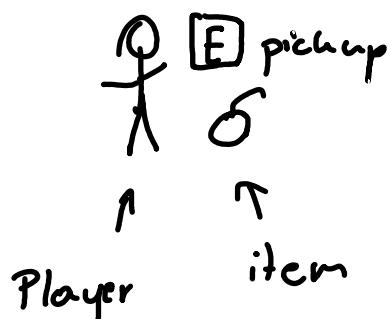
- Create graphic for player

Interactable class

How it works:

If player is near it show interact text. If player click interact button make some operation. e.g. Open door, pickup item.

How it looks



Code

Interactable class should be base class

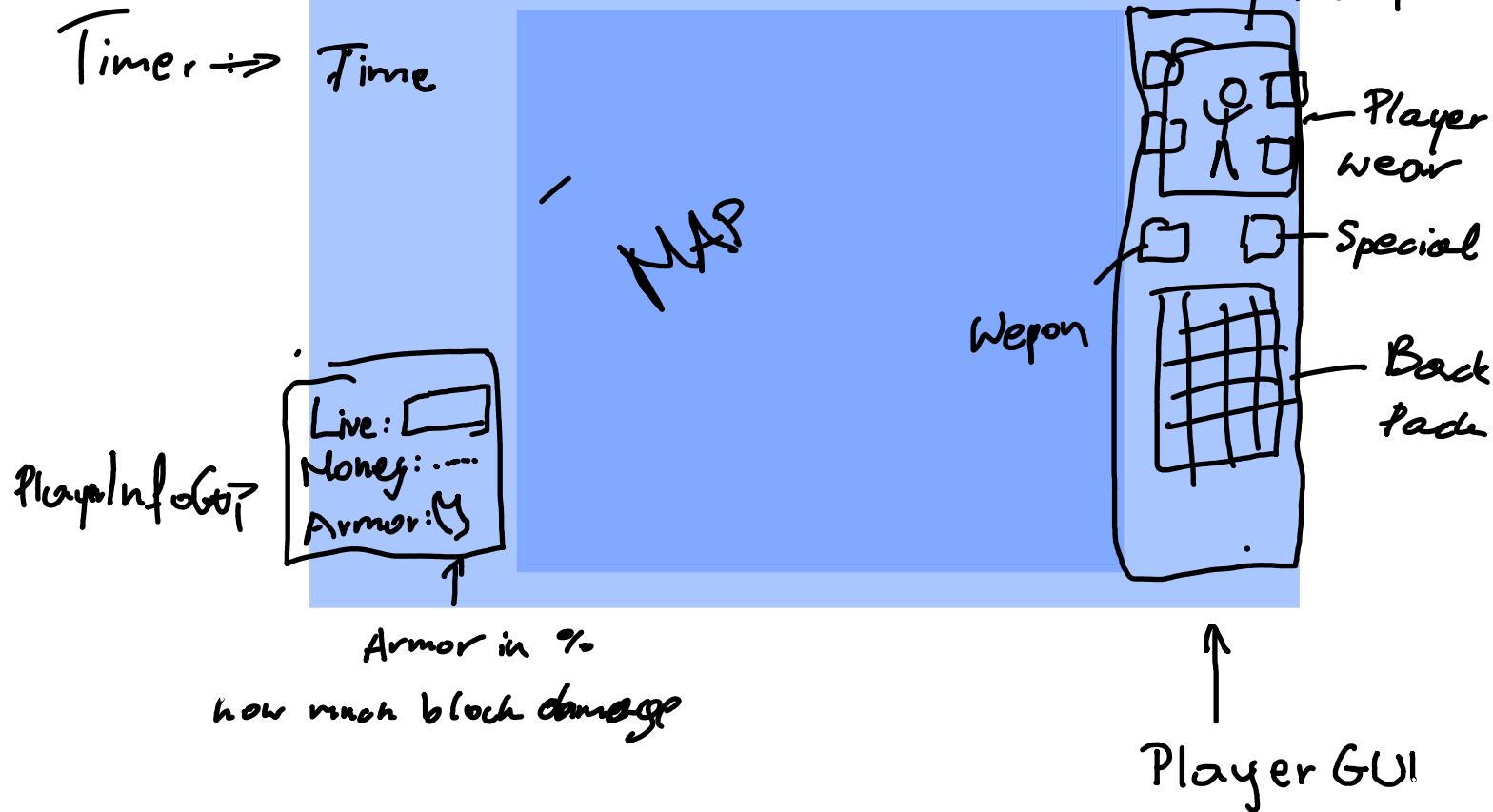
Params:

- Distance
- Text - text node
- Function - Command
- Command Queue

Methods:

- Draw
- Update
- Constructor

Player GUI



PLAYER INFO

{ Lives: uint
Money: uint
Backpack ← Contains 20 pickups
Wear ← Contain armor (helmet, armor, boots,
right hand, left hand)

{

BACKPACK

- Contain Pickable items
- If overflow or drop first one
- Can be controlled by mouse

```
{ int size = 20  
int fill = 0  
array: unique_ptr<Pickup>
```