

Input Output

Input	Java Data
Player name	String
Up Button	-1
Down Button	1
Left Button	-1
Right Button	1
Teleport x	short
Teleport y	short
Y btn	bool
N btn	bool
Find type	bool
Find ID	int

Data Types

Data	Type
Item grafiks	2D char []
Character grafiks	2D char []
Terrain grafik	2D char []
Item Color	Color
Character Color	Color
Terrain Color	Color
ENRG	int
SATT	int
SDEF	int
AGI	int
RSTMN	int
RENRG	int
MNBST	short
MMBST	short
Wheight	int
PWR	int
STMN	int
ENRG	int
TYPE	short

Impact	short
Source	short

Data Types 2

HP	int
STMN	int
STR	int
NDEF	int
ENRG	int
SATT	int
SDEF	int
AGI	int
RSTMN	int
RENRG	int
MNBST	short
MMBST	short
Wheight	int
PWR	int
STMN	int
ENRG	int
TYPE	short
Impact	short
Source	short

Program Class Interaction Procedure

Combat System

Combat flow 1.0

1 Fight starts

- 2 Items mod dynamic creature classes
- 3 Players choose what to do
- 4 Dynamic actions classes get created
- 5 Items mod dynamic action classes
- 6 Actions mod dynamic creature classes
- 7 Creatures get checked for HP<0
- 8 HP<0 dynamic creature classes get destroyed

9 the number of dynamic creature classes on both sides gets checked

10 if a side has 0 classes the fight is over and the other sides dynamic inventory gets modded

11 else the goto step 3

Combat flow 2.0

1. Player enters combat via trigger in map. Loads EIDs classes with data
2. Player decides moves added to arraylist including ID TTI target speed and caster
3. Ai decides moves added to arraylist including ID TTI target speed and caster
4. arraylist gets sorted into an array
5. array is run through and attack .dothing() method gets run
6. check health levels of both sides
7. if team or enemy HP>0 goto 2. else end combat with win lose depending on side

Creature system

DMG = (NATT or SATT *PWR+MSTR ÷(enemy (NDEF or SDEF* BSTs))
*BSTs*TYPE*STAB))

BST= BST + .5 - (BSTC^2÷MNBST or MSBST)

Creature Management systems:

Wilds

Required variables:

ID: 1 int

types: 3 ints

base stats: 10 ints

stat modifiers>> accessible externally

attacks: 4 overall point quantity bound int and 1 ID bound int

attacks to point quantity 4 randomly picked from attack database

obtained points: 1 int defined by secty and sectx

//ability: 1 thingy

Player

Required variables:

when fighting:

ID: 1 int

types: 3 ints

base stats: 10 ints

attacks: 4 overall point quantity bound int and 1 ID bound int

attacks to point quantity:

obtained points: Stored

//ability: 1 thingy

when teaching:

ID: 1 int

points obtained: 1 int

attacks at overall points: an array of ints

Attacks:

Stored in a class contains:

ID: int

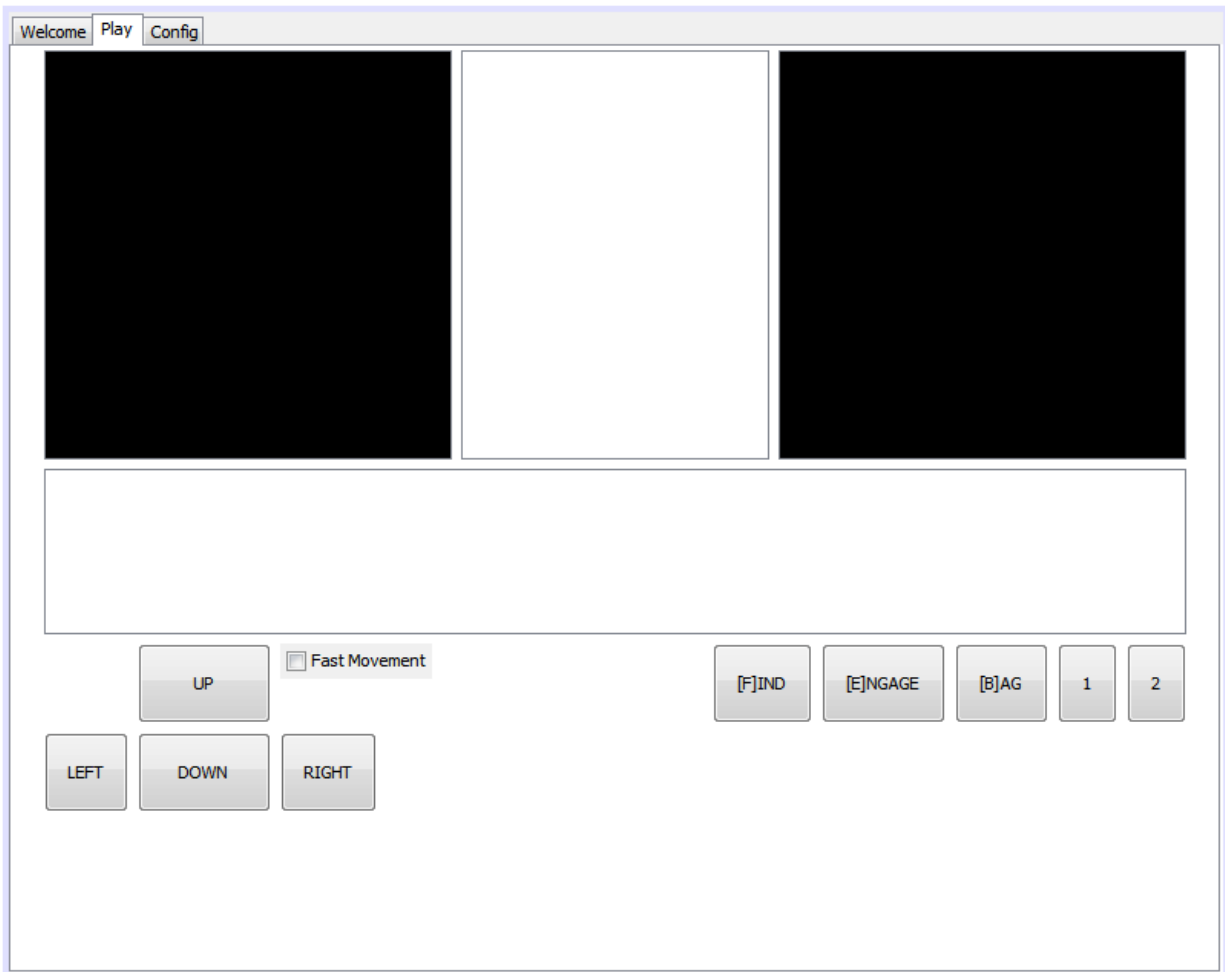
Name: String

DMG: int

Type: int int

Effects: Method that modifies the Dynamic creature states in a battle

Final Prototype



Chronological Development Plan

Test table functionality

Make a timer class

Make movement mnemonics

Build the map layout

- Make all the classes

- give them the correct file names

Complete mapmanager

- make gets for rooms

- make gets for colors

- make gets for difficulty

Populate the map

- translate the map terrain into colors and char layouts

gameGUI

- lock movement on interaction with things
- add movement mnemonics

CreatureManager

- Creature files
 - design creature template
 - Setup 20 creatures
- Switch Structure with all gets

Item manager

- item files
 - armor
 - weapon
 - potion
 - misc
- Switch structure with all gets

Store GUI

- Design the GUI
- Pass item IDs only

Fight GUI

- Design the GUI
- Get loading of creatures set up

Saving & loading

- write/read creature stats as displacement
- write/read inventory as quantities
- write/read room states as numbers

Make Find work

- Design find GUI
- limit to items and creatures in the room

Make Teleport work

- Design teleport GUI
- limit range

Make Autopilot work

Make Inventory work

- Design Bag GUI

Design Armor GUI

Make Engage activate un automated tasks

Create displacement animation engine that runs at 10FPS

Testing Plan

Normal Behaviour

Opening Tab

Start button loads a new file or interprets the save file depending on what is chosen

Load button selects and loads the save file

Store

select button(chooses right item)

quantity buttons (quantity is correct and limited to 99)

confirm button confirms an action

money is correctly calculated

Combat

target selection attacks land on selected target

attack selection attack chosen is used

combat screen closes once combat is over

damage is correctly calculated

boosts are correctly calculated

Map

Autopilot does everything in the room

Can't TP of the side

only certain objects don't collide

Find finds what you're looking for

NPCs can be interacted with using engage button

dialog paths follow player choices

1 and 2 buttons control dialog flow

trigger stores or combat

Colors load

Mnemonics for the up down left right buttons work

Art screens

Art isn't larger than the display

Colors load

Note: I do not have much to test because user input is heavily restricted. Limiting the number of sources of error to the absolute minimum.

