

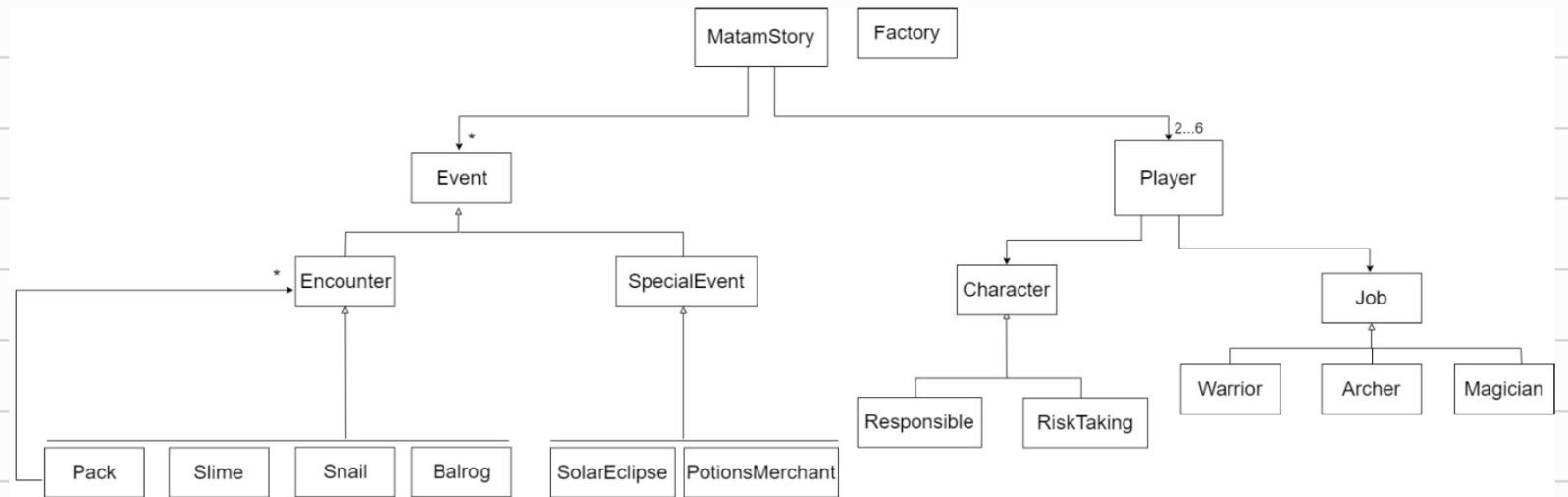
עקבים - נס - גלגלן
318403193 'יכנס' כוונת
208638171 ג'נובה' ג'נובה

שרב להקוחה במדמיה

Pack 2 Snail Snail
Pack 3 Pack 2 Balrog Slime Snail Balrog
Pack 4 Snail Snail Slime Pack 2 Slime Slime
Pack 2 Snail Slime
Pack 2 Pack 2 Pack 2 Balrog Balrog Slime Snail
Pack 3 Slime Snail Snail
Pack 3 Pack 3 Balrog Slime Snail Balrog Balrog
Pack 4 Snail Snail Slime Pack 2 Slime Slime
Pack 2 Snail Slime
2 Pack 3 Pack 2 Balrog Balrog Slime Slime Snail
Pack 2 Snail Snail
Pack 3 Pack 2 Balrog Slime Snail Balrog
Pack 5 Snail Snail Snail Slime Pack 2 Slime Slime
Pack 2 Snail Slime
Pack 2 Pack 2 Pack 2 Balrog Balrog Slime Snail
Pack 2 Snail Snail
Pack 3 Pack 2 Balrog Slime Snail Balrog
מה שכאבי מדמיינים
Pack 2 Snail Slime
imgflip.com



1 Notice



2 Notice

הנתקן פולחן מילויים עליונים

Encounter-וְיַדְעָה רְאֵת Pack גְּמֻלָּה - Composite (1)

הנתקן יתגלה כ-Encounters, ו-תפקידו ניכר יותר

Encounter דוחן ריג'וסט רְקִיבָּה וְעַמְּקָם עֲמָלָק

הנתקן מילויים יתגלה כ-Strategy (2)

Player קידם מילויים עליון לירובם בפונקציית Job -> (k)

: מילויים Job -> מילויים מילויים יתגלו כ-

מילויים יתגלו כ- getCombat מילויים יתגלו כ-

מילויים יתגלו כ- Job מילויים יתגלו כ- Job מילויים יתגלו כ-

הנתקן מילויים יתגלו כ- Job, מילויים יתגלו כ- Combat ->

player.getJob().getCombat(level, force) :

PotionsMerchant -> מילויים יתגלו כ- Character מילויים יתגלו כ-

Character יתגלו מילויים יתגלו כ- "מילויים" מילויים יתגלו כ-

Player.getCharacter.buy(Player)

מילויים יתגלו כ- Character יתגלו כ- מילויים יתגלו כ- Job

מילויים יתגלו כ- Job יתגלו כ- Character יתגלו כ- מילויים יתגלו כ-

ו-
ל-
.player ל-
.player ל-
.player ל-
.player ל-

strategy -
strategy -

player ל-
player ל-
player ל-
player ל-
player ל-
player ל-

command (3)
player ל-
player ל-
player ל-

Factory (4)

player ל-
player ל-
player ל-
player ל-

player ל-
player ל-
player ל-
player ל-

player ל-
player ל-
player ל-
player ל-

13) INTRODUCE POLYMORPHISM BY ADDING IS_ABLE WHICH
IS_ABLE HAS A BASE CLASS String WHICH IS_ABLE
FROM WHICH THE DERIVED CLASS COLOR CAN USE
METHODS ALREADY FOUND, CREATE 'IF' INSTEAD OF WHICH
WE CAN USE AS IF FOR THIS PURPOSE WE CAN USE POLYMORPHISM

3. Interface

Class Rouge: Job {
public:

Rouge();

Rouge(const Rouge&) = delete;

Rouge& operator=(const Rouge&) = delete;

std::unique_ptr<Job> clone() override; }

THIS IS HOW WE CAN DO IT WITH JOB CLASS WHICH IS
NOTHING BUT A CLONE OF THE PREVIOUS ONE WHICH IS ALREADY
DEFINED. THE ONLY DIFFERENCE IS THAT IT IS NOT
A POLYMORPHIC CLASS AS IT IS NOT DERIVED FROM
A BASE CLASS.

ALSO, WE CAN USE THE FOLLOWING (2)
Factory ->

13) (C) PRACTICE AGAIN AND AGAIN TO GET USED TO IT (3
FIRSE IN), Job FOR ALREADY EXISTING CLASS OF WHICH
getCombat() OR Warrior

Rouge - סטראטגי

: פונקציית getCombat מתקבלת כערך של int, ופונקציית setCombat מקבלת כערך של int.
הפונקציה getCombat מקבלת כערך של int ופונקציית setCombat מקבלת כערך של int.

רפלקס גז, פונקציית getCombat מקבלת כערך של int ופונקציית setCombat מקבלת כערך של int.

Function getCombat (unsigned int force, unsigned int level, unsigned int enemyCombat = 0);

Function setCombat (int combat);

Function getCombat (int combat);

, פונקציית setCombat מקבלת כערך של int ופונקציית getCombat מקבלת כערך של int.

Function setCombat (int combat);

Function setCombat (int combat);

Function fight

Function fight מקבלת כערך של int ופונקציית setCombat מקבלת כערך של int.

Function setCombat מקבלת כערך של int ופונקציית getCombat מקבלת כערך של int.

: Function getCombat מקבלת כערך של int ופונקציית setCombat מקבלת כערך של int.

unsigned int Rouge::getCombat (...) level ... force ... enemyCombat) {

unsigned int combat = level + force;

if (enemyCombat >= 2 * combat) {

return enemyCombat + 1;

}

return combat;

הצגה סטראטגי

Function play מקבלת כערך של int ופונקציית setCombat מקבלת כערך של int.

void Encounter::play(Player& player) {

:

void Encounter::Play (Player &) {

 Player.getJob().Fight(*this, Player &);
}

void Job::Fight (Encounter & encounter, Player & player) {
 Job job;

 job.setJobType (JobType::FIGHT);
 job.setJobName ("Fight");
 job.setJobDescription ("Fights the player");
 job.setJobIcon (ICON_FIGHT);
 job.setJobColor (COLOR_FIGHT);
 job.setJobLevel (LEVEL_FIGHT);
 job.setJobExperience (EXPERIENCE_FIGHT);
 job.setJobSkill (SKILL_FIGHT);
 job.setJobSkillLevel (LEVEL_FIGHT_SKILL);
 job.setJobSkillExperience (EXPERIENCE_FIGHT_SKILL);
 job.setJobSkillIcon (ICON_FIGHT_SKILL);
 job.setJobSkillColor (COLOR_FIGHT_SKILL);
 job.setJobSkillDescription ("Fights the player");
 job.setJobSkillType (JobType::FIGHT_SKILL);
}

void Job::Fight (Encounter & encounter, Player & player) {
 Job job;

 job.setJobType (JobType::FIGHT);
 job.setJobName ("Fight");
 job.setJobDescription ("Fights the player");
 job.setJobIcon (ICON_FIGHT);
 job.setJobColor (COLOR_FIGHT);
 job.setJobLevel (LEVEL_FIGHT);
 job.setJobExperience (EXPERIENCE_FIGHT);
 job.setJobSkill (SKILL_FIGHT);
 job.setJobSkillLevel (LEVEL_FIGHT_SKILL);
 job.setJobSkillExperience (EXPERIENCE_FIGHT_SKILL);
 job.setJobSkillIcon (ICON_FIGHT_SKILL);
 job.setJobSkillColor (COLOR_FIGHT_SKILL);
 job.setJobSkillDescription ("Fights the player");
 job.setJobSkillType (JobType::FIGHT_SKILL);
}

Rouge ->N P11 Jobs -> Sf p'k JJA k1,1
.12e jnnna nrajjja2 l2lne l2en nojia j12 1d,1,2

void Rouge :: fight (Encounter & encounter, Player & player) {
 if (encounter.getCombat >= 2 * this->getCombat()) {

 print RougeEscapeMessage();
 return;

}

 if (encounter.getCombat > this->getCombat()) {
 ... f2cc l1N'N, pen,

}
 else { . . .

3

!Strategy -> l1N'N, ADOIJ AN213 IS

4. nfcle

!>> Mofn A'N"NR nrcra m'lc f'gol,ld p'lj

A'N"NR A'N"NR nrcra f'gol,ld p'lj (1
: specialEvent A'N"NR

Class DivineInspiration : SpecialEvent {

public:

DivineInspiration();

DivineInspiration(const DivineInspiration&) = delete;

DivineInspiration& operator=(const DivineInspiration&) = delete;

Void play(Player &) override;

}

רְגָבָן יְמִינָה כְּלַיְלָה אֶתְנָה תְּמִימָה כְּלַיְלָה
מְלֵאָה כְּלַיְלָה תְּמִימָה כְּלַיְלָה אֶתְנָה תְּמִימָה כְּלַיְלָה

(Factory-) מְנוּעָה אֶתְנָה תְּמִימָה כְּלַיְלָה תְּמִימָה כְּלַיְלָה
: אֶתְנָה תְּמִימָה כְּלַיְלָה תְּמִימָה כְּלַיְלָה (3)

Void Player::setJob(uniqueptr<Job> newJob) {

this->Job = std::move(newJob);

}

רְגָבָן לְנָדָב getRandomJob() כְּלַיְלָה תְּמִימָה כְּלַיְלָה (4)

לְנָדָב סְקָלָה אֶתְנָה תְּמִימָה כְּלַיְלָה תְּמִימָה כְּלַיְלָה
size -> if (size == 0) return Job(); else return Job();

לְנָדָב קְרָבָה אֶתְנָה תְּמִימָה כְּלַיְלָה תְּמִימָה כְּלַיְלָה
size > 0) return Job(); else return Job();

לְנָדָב קְרָבָה אֶתְנָה תְּמִימָה כְּלַיְלָה תְּמִימָה כְּלַיְלָה
uniqueptr<Job> newJob = getRandJob();

DivineInspiration & newJob; return newJob;

: void DivineInspiration::Play() כְּלַיְלָה תְּמִימָה כְּלַיְלָה (5)

Void DivineInspiration::Play(Player& player) {

std::unique_ptr<Job> newJob = getRandomJob();

player.setJob(std::move(newJob));

printDivineInspirationMessage(player, player.getJob().getName());

}

!ASI