*Code changes:*

* enum.h

Pada berkas ini, dilakukan penambahan *if-condition* untuk menangani kompatibilitas *entity* baru dengan *save profile* yang menggunakan versi lama yang tidak terdapat *entity* baru tersebut.

*Code* yang baru mengecek *tag* versi *game*. Jika *game* memiliki *tag* versi 34, maka monster Sheep dimunculkan. Jika *game* memiliki *tag* versi yang lebih tinggi dari versi 34, maka monster Dream Sheep yang dimunculkan. Penambahan juga dilakukan untuk menempatkan Dream Sheep ke dalam kelompok monster baru.

* godwrath.cc

Berkas ini berisi *function-function* terkait *god* atau dewa dalam *game*. Dungeon Crawl Stone Soup memiliki sistem penyembahan dewa di mana pemain dapat menjadi penyembah seorang dewa. *Function-function* tersebut sebagian besar menangani efek dari pemujaan dewa tertentu, serta konfigurasi monster-monster pengikut-pengikut setiap dewa. Pada berkas ini, tidak terdapat pengubahan yang mengubah baris *code*, pengubahan hanya terjadi pada sebuah baris *comment* terkait *entity* Sheep di mana kata ‘Sheep’ menjadi ‘Dream Sheep’.

* mgen-enum.h

Pada berkas ini, dilakukan pengubahan pada pendefinisian *band* di mana *band* Sheep diubah menjadi *band* Dream Sheep. *Band* merupakan sejumlah *entity* yang sama yang beraktifitas secara bergerombol.

* mon-abil.cc

Pada berkas ini, dilakukan implementasi *ability* Dream Dust yang dimiliki oleh Dream Sheep. Ada beberapa *function* dan *method* yang ditambahkan.

Pertama adalah *method* \_sheep\_message() yang menghasilkan output berupa teks yang mendeskripsikan aksi-aksi yang dilakukan oleh Dream Sheep, atau efek-efek yang dialami oleh pemain ketika berinteraksi dengan Dream Sheep. Terdapat beberapa tipe teks yang ditambahkan. Setiap tipe teks memiliki variasi masing-masing sesuai dengan ketepatan *event* yang terjadi.

* + Teks untuk mendeskripsikan tingkat kekuatan *ability* yang dilancarkan kumpulan Dream Sheep.
  + Teks untuk mendeskripsikan efek yang dialami karakter non-pemain ketika terkena Dream Dust.
  + Teks untuk mendeskripsikan efek yang dialami pemain ketika terkena Dream Dust.

Kedua adalah *method* \_dream\_sheep\_sleep() yang memiliki beberapa fungsi:

* + Mengecek sasaran yang akan diserang oleh seekor Dream Sheep. Jika tidak ada sasaran atau sasaran yang diincar oleh Dream Sheep telah mati, maka *function* akan melakukan *return*.
  + Mengatur posisi dan gerakan gerombolan Dream Sheep, serta menghitung jumlah Dream Sheep dalam sebuah gerombolan
  + Men-*generate* kemungkinan berhasilnya sebuah gerombolan Dream Sheep untuk melakukan *ability*.
  + Memanggil *method* \_sheep\_message() untuk mengeluarkan output sesuai dengan hasil perhitungan dalam *method* ini.
  + Menentukan durasi Sleep yang dialami karakter sasaran ketika Dream Dust berhasil.

Ketiga adalah *case* MONS\_DREAM\_SHEEP yang memiliki fungsi:

* + Mengecek apakah Dream Sheep tidak dalam kondisi bingung (*confused*) dan tidak sedang melarikan diri (*fleeing*), lalu mengecek apakah karakter sasaran tidak berada dalam sebuah *sanctuary*, sedang tidak tidur, dan tidak memiliki kekebalan terhadap Sleep.
  + Jika semua kondisi di atas terpenuhi, maka Dream Sheep memiliki peluang 20% untuk melakukan *ability*.
* mon-data.h

Pada berkas ini, Sheep ‘didaftarkan’ sebagai monster lama yang sudah tidak dipakai untuk versi baru, hanya disimpan untuk kompatibilitas versi lama. Selain itu, pengubahan juga banyak dilakukan untuk mengganti setiap *instance* dan konfigurasi yang terkait dengan Sheep agar menjadi Dream Sheep.

* mon-ench.cc

Secara umum, berkas ini berisi *function-function* yang mengatur karakteristik setiap *enchantment* yang terdapat dalam *game*, serta efeknya terhadap monster-monster yang ada. *Enchantment* merupakan jurus, atau *status effect* dari jurus yang dilancarkan oleh sebuah karakter atau *entity* dalam *game*, yang dapat memberikan efek tertentu pada sasaran yang menerima serangan jurus tersebut. Pada berkas ini, dilakukan pengubahan untuk *case* ENCH\_STICKY\_FLAME yang mengatur implementasi *spell* Sticky Flame. *Spell* tersebut memiliki keunikan berupa efektifitas yang sangat tinggi terhadap monster dengan *genus* MONS\_SHEEP. *Genus* merupakan kelompok monster yang masih memiliki relasi biologis atau anatomis yang dekat, fiktif atau non-fiktif. Perubahan yang dilakukan adalah mengganti efek *spell* tersebut yang tadinya ditujukan secara spesifik untuk monster Sheep, menjadi untuk kelompok *genus* dari Sheep, yang mengandung Sheep dan Dream Sheep.

* mon-pick-data.h

Berkas ini berisi data mengenai kemunculan monster seperti lokasi dan peluang muncul. Data tersebut disimpan dalam beberapa *const* yang masing-masing mewakilkan kelompok monster berdasarkan lokasi *spawn* atau muncul. Berkas *header* ini dimanfaatkan oleh berkas mon-pick.cc. Pada berkas ini, dilakukan penambahan data untuk MONS\_DREAM\_SHEEP, dan penghapusan data untuk MONS\_SHEEP. Dream Sheep dan Sheep diprogram untuk *spawn* di kategori level *dungeon* yang bernama Lair.

* mon-place.cc

Berkas ini berisi *function-function* yang digunakan ketika menempatkan monster di dalam *dungeon*. Bagian yang diubah pada berkas ini merupakan:

* + *const* map<monster\_type, band\_set> bands\_by\_leader yang berisi daftar monster khusus yang memiliki peluang untuk *spawn* diikuti dengan sebuah *band* monster tersebut atau *band* monster lain. Pengubahan yang dilakukan adalah:
    - Mengganti konfigurasi kemunculan *band* Sheep menjadi untuk *band* Dream Sheep.
    - Mengganti perhitungan peluang 20% untuk kemunculan *band* Sheep yang mengikuti monster Cyclops menjadi peluang 100% untuk kemunculan *band* Dream Sheep.
  + *const* map<branch\_type, band\_weights> bands\_pick yang berisi penanganan *band-band* yang muncul mengikuti monster-monster dengan karakteristik kepemimpinan *band* khusus. Pengubahan yang dilakukan adalah:
    - Mengganti konfigurasi untuk kemunculan *band* Sheep yang mengikuti monster Torpor Snail menjadi untuk *band* Dream Sheep. Konfigurasi ini terdapat pada sebuah *case* tersendiri untuk MONS\_TORPOR\_SNAIL di instansiasi band\_type \_choose\_band().
  + *const* map<band\_type, vector<member\_possibilites>> band\_membership yang berisi daftar *band* monster beserta konfigurasi kemungkinan monster-monster yang terdapat dalam *band* tersebut. Pengubahan yang dilakukan adalah:
    - Mengganti konfigurasi komposisi *band* Sheep menjadi untuk *band* Dream Sheep.
* mon-spell.h

Berkas ini berisi daftar *spell* yang dimiliki oleh semua monster yang terdapat dalam *game*. *Spell* disusun berdasarkan monster yang memiliki *spell* tersebut. Pada berkas ini, terdapat penambahan berupa sebuah kategori *spell* yang dimiliki oleh Dream Sheep. Kategori tersebut berisi sebuah *spell* yang memiliki nama obyek SPELL\_SLEEP.

* player-reacts.cc

Berkas ini berisi *function-function* yang berhubungan dengan karakter pemain. Sebagian besar *function* yang terdapat dalam berkas ini digunakan untuk mengatur *enchantment* pada karakter pemain. Pada berkas ini, terdapat pengubahan *typo* untuk *command* you.awake() menjadi you.awaken(). *Command* ini berfungsi untuk membangunkan pemain setelah Sleep yang dialami pemain kadaluarsa.

* player.cc

Berkas ini berisi *function-function* yang berhubungan dengan pemain. Perbedaan dengan player-reacts.cc adalah player.cc berisi *function* untuk penggunaan yang lebih umum. Terdapat beberapa penambahan dan pengubahan pada berkas ini:

* + Penambahan *bool* player::can\_sleep yang memeriksa apakah pemain dapat dibuat tidur oleh *enchantment* Sleep. Fungsi ini me-return true jika pemain tidak memiliki kekebalan terhadap Sleep.
  + Pengubahan *typo* untuk *method* player::awake() menjadi player::awaken()
  + Pengubahan instansiasi nilai kekebalan pemain terhadap Sleep di player::awaken(). Sebelum diubah, pemain di-set untuk memiliki kekebalan selama 1 turn. Setelah pengubahan, pemain hanya di-set untuk memiliki kekebalan selama 1 turn jika pada saat fungsi dipanggil pemain tidak memiliki kekebalan sama sekali.
  + Pengubahan *typo* untuk pemanggilan *method* awake() menjadi awaken() di *method* player::check\_awaken() yang memeriksa apakah pemain sudah terbangun.

Note: fungsi ini memiliki ~8000 baris code. Apa bisa dilakukan improvement agar code sebanyak itu tidak menumpuk ? Atau sudah best approach ?

* player.h

Berkas ini merupakan *header* untuk player.cc. Isi dari berkas ini merupakan inisialisasi variabel-variabel yang digunakan di player.cc. Pengubahan yang dilakukan pada berkas ini adalah pengubahan *typo* untuk inisialisasi awake() menjadi awaken().

Compatibility issue

Salah satu isu yang dapat diangkat dari pengubahan-pengubahan yang dilakukan oleh penambahan Dream Sheep ke *source code* adalah kompatibilitas. Kompatibilitas yang dimaksud adalah kompatibilitas pengadaan sebuah *entity* baru ke dalam *game* atau *save profile* dengan versi yang lebih lama.

Kegagalan kompatibilitas dapat menyebabkan monster baru tidak muncul di *game* dengan versi baru namun *save profile*-nya versi lama, atau justru membuat *save profile* tersebut *corrupt* dan tidak dapat dimainkan.

Dalam Dungeon Crawl Stone Soup, penanganan kompatibilitas dilakukan dengan cara menyimpan sebuah *tag* yang menunjukkan versi dari *game*, lalu mengelompokkan *entity-entity* yang ada menjadi *entity* lama dan *entity* baru.

...

Backward compatibility: it is hard to evolve the language when models (specs) created with it are already in place. **(p. 38)**

...

Backward and forward compatibility is an important concern when evolving a game SPL. Not rarely, refined versions of the DSLs can break existing model instances. For such scenarios, we suggest that migration tools are developed in order to assist game developers and designers to move their models to the new versions of the DSLs. **(p. 104)**

...

Another challenge that we faced with the approach relates to backward compatibility. More than once, new versions of the DSLs (still under the same case study) broke existing model instances. Since we did not develop migration tools to update models to the new versions of the DSLs, we had to update them manually. **(p. 144)**

**Domain Specific Game Development**

Buku untuk panduan/acuan *refactoring*

Object-oriented Programming in ANSI-C, Axel Schreiner

<https://www.cs.rit.edu/~ats/books/ooc.pdf>

Note:

Terdapat beberapa commit baru yang dilakukan oleh pembuat Dream Sheep untuk menyesuaikan beberapa konfigurasi game dengan Dream Sheep.

Interestingly, ketika mencari ‘dungeon crawl stone soup dream sheep’ di google, search result yang pertama muncul adalah sebuah post Reddit yang menyatakan kekesalan seseorang terhadap penambahan Dream Sheep dan meminta Dream Sheep dimodifikasi ulang atau dihilangkan lagi dari game. Post tersebut menyatakan bahwa Dream Sheep terlalu berbahaya jika *ability*-nya digunakan ketika pemain sedang menghadapi monster lain yang sangat kuat. Opini-opini yang ada pada thread tersebut 50:50. This seems out of topic tapi sepertinya ada pelajaran atau isu yang dapat diangkat dari komplain tersebut.