LAPORAN TUGAS BESAR

IF2111 Algoritma dan Struktur Data

Candy Crush Saga

Groar

Dipersiapkan oleh:

Groar

Rifda Annelies Az Zahra / 18216001

Alessandro Ario Wibowo / 18216002

Athur Naufan Muharram / 18216003

Al Varrel Putra Kusuma / 18216020

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Sekolah Teknik Elektro dan Informatika ITB** | Nomor Dokumen | Halaman |
| *IF2111-TB-2-01* | *11* |
| *10/11/17* |

Daftar Isi

[1 Ringkasan 4](#_Toc499667339)

[2 Penjelasan Tambahan Spesifikasi Tugas 4](#_Toc499667340)

[2.1 Load Score 4](#_Toc499667341)

[2.2 High Score 4](#_Toc499667342)

[2.3 Save Score 4](#_Toc499667343)

[2.4 Update Score 4](#_Toc499667344)

[2.5 Tampilan POP! saat candy pecah 4](#_Toc499667345)

[2.6 Shuffle 4](#_Toc499667346)

[3 Program Utama 5](#_Toc499667347)

[3.1 Welcome Screen 5](#_Toc499667348)

[3.2 Menu 5](#_Toc499667349)

[3.3 Play 5](#_Toc499667350)

[3.3.1 smartRandom 5](#_Toc499667351)

[3.3.2 AcakIsi 5](#_Toc499667352)

[3.3.3 PrintKotak 5](#_Toc499667353)

[3.3.4 Move 5](#_Toc499667354)

[3.3.5 GetChain 6](#_Toc499667355)

[3.3.6 Update 6](#_Toc499667356)

[3.3.7 FallingCandy 6](#_Toc499667357)

[3.3.8 RemoveZero 6](#_Toc499667358)

[3.3.9 IsChain 6](#_Toc499667359)

[3.3.10 detectPossibleSwap 6](#_Toc499667360)

[3.3.11 IsValidMove 6](#_Toc499667361)

[3.3.12 MovePoint 6](#_Toc499667362)

[3.3.13 Swap 7](#_Toc499667363)

[3.3.14 PrintInfo 7](#_Toc499667364)

[3.3.15 updateSkor 7](#_Toc499667365)

[3.3.16 IsEmptyFile 7](#_Toc499667366)

[3.3.17 IsNameEq 7](#_Toc499667367)

[3.3.18 Inisialisasi 7](#_Toc499667368)

[3.3.19 Save 7](#_Toc499667369)

[3.3.20 Load 7](#_Toc499667370)

[3.4 How To Play 8](#_Toc499667371)

[3.5 High Score 8](#_Toc499667372)

[3.6 About 8](#_Toc499667373)

[3.7 Exit 8](#_Toc499667374)

[4 Algoritma Menarik 8](#_Toc499667375)

[4.1 detectPosibble 8](#_Toc499667376)

[4.2 smartRandom 8](#_Toc499667377)

[5 Pembagian Kerja dalam Kelompok 9](#_Toc499667378)

[6. Lampiran 9](#_Toc499667379)

[5.1 Deskripsi Tugas Besar 9](#_Toc499667380)

[5.2 Notulen Rapat 10](#_Toc499667381)

[6.3 Log Activity Anggota Kelompok 11](#_Toc499667382)

# Ringkasan

Pada tugas besar kali ini kami membuat game <nama game> yang terinspirasi dari game Candy Crush Saga. Fungsi-fungsi dalam game ini dibuat dengan melakukan pengamatan game Candy Crush, dan dibuat dengan dengan tujuan akhir meniru game tersebut.

Program utama dari game ini adalah menunjukan matrix berukuran 8x8 yang berisi candy. Nantinya, candy tersebut harus disusun menjadi suatu baris atau kolom sebanyak tiga, empat, atau lima untuk mendapatkan skor. Kesempatan yang diberikan untuk setiap pemain adalah 18 kali untuk melakukan pertukaran candy dengan candy disebelahnya secara vertikal maupun horizontal. Apabila sudah tidak terdapat langkah lagi, game tidak lantas kalah, namun melakukan shuffling menjadi papan baru, namun kesempatan tetap melanjutkan dari sebelumnya. Permainan akan dinyatakan menang jika score yang didapatkan dapat melebihi batas yaitu 5000.

Dalam game ini, terdapat fitur yang menampilkan papan high score untuk setiap pemain yang telah melewati skor 5000. Selain dengan limitasi tersebut, papan high score juga akan menampilkan nama dari sepuluh pemain dengan skor tertinggi secara terurut mengecil. Papan high score akan secara otomatis muncul ketika permainan telah selesai. Selain itu, papan highscore juga dapat diakses langsung dari menu.

# Penjelasan Tambahan Spesifikasi Tugas

## Load Score

Load score dilakukan saat page welcome screen, player dapat memasukkan nama. Nama yang dimasukkan list player dan akan di *pass* sepanjang program.

## High Score

Selain ditampilkan pada saat selesai bermain, high score juga dapat diaskses melalui menu.

Ditampilkan maksimal 10 pemain terurut mengecil.

## Save Score

Score akan disave bila pemain memenuhi target permainan yaitu 5000 poin.

## Update Score

Score akan diupdate setiap mendapatkan chain dalam permainan.

## Tampilan POP! saat candy pecah

Saat move berhasil dan candy pecah akan terdapat tulisan pop! yg berkedip. Algoritma berkedip/ blink menggunakan fungsi delay.

## Shuffle

Apabila saat permainan sedang dijalankan dan turn masih ada, sedangkan sudah tidak terdapat candy yang bisa bertukar (bila ditukar mendapat point), maka program akan menampilkan kata Shuffle dan board akan diacak ulang oleh sistem.

# Program Utama

## Welcome Screen

Pada saat pertama kali program dipanggil akan menampilkan welcome screen yaitu halaman untuk menampilkan judul game dan juga input nama pemain. Pada saat input nama pemain, terjadi load dari file eksternal highscore yang digunakan.

## Menu

Halaman ini akan ditampilkan setelah user memasukkan nama pemain, halaman langsung diarahkan menuju halaman menu ini. Pada menu, pemain dapat memilih beberapa page lain yaitu: Play, How To Play, High Score, About, dan juga Exit. Setiap fungsi yang dipilih selesai dijalankan, page akan dikembalikan menuju menu kembali.

## Play

Halaman ini adalah halaman utama permainan. Pada halaman ini terdapat beberapa fitur yaitu:

### smartRandom

Fungsi ini berfungsi untuk mengacak matrix sedemekian rupa sehingga saat permainan dimulai, skor pasti dimulai dari 0. Algoritma ini dijalankan dengan cara mengecek sebelah kiri dan atas dari suatu kotak apakah warna sudah muncul sebanyak 2 kali atau belum. Sehingga, tidak mungkin terjadi chain saat permainan dimulai.

### AcakIsi

Prosedur ini berfungsi untuk mengisi kotak dengan mendapatkan nilai dari smartrandom. Nilai yang sudah didapat ini nantinya akan ditampilkan ke matrix dengan fungsi PrintKotak. Algoritma ini berjalan dengan melakukan iterasi matrix dan mengisinya satu persatu.

### PrintKotak

Prosedur ini berfungsi untuk melakukan print dari matrix. Algoritma ini menggunakan library online rlutil.h untuk mendapatkan fungsi textcolor dan dapat melakukan pewarnaan pada isi dari kotak. Fungsi ini juga menganalogikan candy dengan beberapa simbol-simbol: @@, $$, \*\*, ##, &&. Simbol-simbol ini mewakilkan candy dan suatu warna tertentu. Algoritma ini berjalan dengan melakukan iterasi matrix dalam melakukan printing.

### Move

Prosedur ini berguna untuk melakukan pengecekan apakah suatu gerakan dapat dilakukan atau tidak dengan melihat koordinat yang dimasukan oleh pengguna. Apabila valid, maka akan dilakukan pertukaran antara kedua candy yang bersebelahan.

### GetChain

Prosedur ini berguna untuk melakukan pengecekan terhadap matrix apakah terdapat suatu chain atau tidak, dan berapa banyak chain yang terjadi. Setelah mengetahui chainnya, koordinat dari chain dimasukan ke dalam stack.

### Update

Prosedur ini digunakan untuk mengubah warna dari candy yang koordinatnya terdapat di stack menjadi 0 atau tidak berwarna (dalam print, disimbolkan menjadi ‘ ‘).

### FallingCandy

Prosedur ini melakukan swapping antara candy yang memiliki nilai 0 dengan candy yang terdapat di atasnya. Hal ini dilakukan hingga nilai 0 tersebut terdapat di bagian atas matrix.

### RemoveZero

Prosedur ini berfungsi untuk mengubah nilai 0 yang sudah mengumpul pada bagian atas matrix setelah prosedur FallingCandy. Prosedur ini memanggil smartRandom untuk mendapatkan nilai yang menggantikan nilai 0 pada matrix.

### IsChain

Fungsi ini mengembalikan nilai Boolean setelah melakukan pengecekan terhadap bagian atas, kiri, kanan, dan bawah sebanyak 2 kali apakah terdapat chain atau tidak.

### detectPossibleSwap

Prosedur ini berfungsi untuk mengecek apakah masih terdapat move/gerakan yang dapat dilakukan oleh user. Algoritma dari prosedut ini memanfaatkan 3 stack yaitu ori, right dan down. Stack ori menyimpan koordinat dari titik yang sedang dicek, sedangkan stack right menyimpan koordinat dari sebelah kanannya, dan stack down menyimpan koordinat dari titik bagian bawahnya. Jika terdapat chain, maka akan dilakukan push ke stack, karena masih terdapat move yang bisa dilakukan.

### IsValidMove

Fungsi ini menghasilkan nilai Boolean untuk melakukan pengecekan apakah move yang dilakukan user valid atau tidak. Pengecekan yang dilakukan adalah pada bagian pinggir matrix, agar move tidak dilakukan dengan sesuatu yang tidak ada.

### MovePoint

Fungsi ini mengembalikan nilai point, yang berupa terjemahan dari masukan user (A, W, S, D) dan bagaimana pergerakan yang dilakukan.

### Swap

Prosedur ini berfungsi untuk menukar isi (candy) dari koordinat sebuah point.

### PrintInfo

Prosedur ini berfungsi untuk menampilkan isi dari List Player. Algoritma ini menampilkan sepuluh nama pemain yang memiliki nilai tertinggi.

### updateSkor

Prosedur ini berfungsi untuk melakukan updating list berdasarkan skor terakhir pemain. Pada algoritma ini, terlebih dahulu dicek apakah nama player terdapat didalam list. Apabila ada, maka dicek apakah skor sekarang lebih tinggi dari skor sebelumnya. Jika iya, elemen list tersebut akan dihapus dan dicari posisi yang sesuai berdasarkan urutan skor. Jika tidak, maka akan dibiarkan dan tidak diubah. Apabila nama player merupakan nama baru maka akan dicari posisi yang sesuai berdasarkan skor player tersebut. Urutan skor dipertahankan mengurut mengecil.

### IsEmptyFile

Fungsi ini menggunakan pointer-to-FILE sebagai parameter dan menghasilkan true jika FILE eksternal kosong. File eksternal terlihat kosong apabila pada posisi file ke 0 terlihat end-of-file. Jika tidak maka file eksternal tidak kosong.

### IsNameEq

Fungsi ini menggunakan (N1, N2) array of character sebagai parameter. Apabila parameter terkait memiliki panjang yang sama akan dicek kesamaan karakter. Apabila berbeda akan dihasilkan nilai true. Selain itu dihasilkan nilai false.

### Inisialisasi

Prosedur ini digunakan untuk menginisiasi nilai variable player (yang bertipe infolist). Yaitu dengan membaca nama player dari input user dan mengeset skor player menjadi 0.

### Save

Prosedur ini digunakan untuk menyimpan list of player kedalam file eksternal. Pada prosedur ini digunakan dua parameter yakni (Input char (nama file) dan Input/Output List). Pada prosedur ini pertama dilakukan pengecekan apakah list kosong. Apabila tidak kosong dilakukan pengambilan nilai sekaligus penghapusan nilai dari list. Nilai tersebut kemudian disimpan kedalam file eksternal dengan menggunakan fungsi bawaan fprintf.

### Load

Prosedur ini digunakan untuk membaca nilai-nilai yang terdapat pada file eksternal kedalam list of player. Pada prosedur ini digunakan dua parameter yakni (Input/Output List dan Input char (nama file)). Terlebih dahulu diassign nama lojik untuk file eksternal tersebut. Kemudian dicek apakah file eksternal kosong atau tidak, jika tidak maka akan dibaca setiap nilai pada file eksternal setiap line. Pembacaan dilakukan sampai end-of-file.

## How To Play

Halaman ini akan menunjukkan bagaimana cara memainkan permainan ini. Dijelaskan bagaimana kriteria permainan tersebut dikategorikan menang atau kalah. Bagaimana cara akses board game. Jumlah skor yang didapat apabila dapat menjadikan 3, 4 atau 5 candy.

## High Score

Halaman ini memunculkan suatu tabel yang berisi list of infolist (Player, Score). Pemain yang ditampilkan adalah maksimal 10 pemain dengan skor tertinggi ditampilkan secara terurut mengecil.

## About

Halaman ini memunculkan identitas dari programmer dan tujuan game ini dibuat.

## Exit

Halaman ini digunakan untuk portal keluar dari permainan. Pemain akan dikonfirmasi kembali sebelum keluar dari permainan ini. Apabila pemain benar-benar mau keluar maka terdapat perintah untuk save game yang tidak ditampilkan ke layar. Save yang dimaksud disini adalah menyimpan data dari list of infolist (Player, Score) ke dalam file external agar bisa diakses di kemudian kesempatan.

# Algoritma Menarik

Dalam tugas besar ini, terdapat dua algoritna menarik yang digunakan di dalam game <nama game> ini. Dua fungsi tersebut adalah detectPossible dan smartRandom. Untuk detectPossible, algoritma ini menarik karena tingkat kesulitannya dalam pembutuan. Sedangkan, untuk smartRandom, algoritma ini menarik karena dapat mengendalikan sesuatu yang bersifat random.

## detectPosibble

Algoritma ini menarik dikarenakan tingkat kesulitannya untuk diselesaikan dan cukup lama dipikirkan dalam banyak rapat. Algoritma ini digunakan didalam loop permainan untuk menentukan apakah masih ada langkah yang dapat dilakukan atau tidak. Algoritma ini bekerja dengan cara melakakukan swap terhadap kotak di atasnya dan di kanannya dan melakukan pengecekan apakah terdapat chain atau tidak. Dengan cara ini, dapat dideteksi apakah masih ada langkah yang dapat dilakukan atau tidak.

## smartRandom

Algoritma ini menarik dikarenakan dengan menggunakan ini, kita dapat mengontrol sesuatu yang bersifat random. Algoritma ini digunakan pada saat permainan baru saja dimulai, sehingga pada saat dimulai permainan, tidak terdapat chain yang panjang dan mengumpul sekaligus. Algoritma ini bekerja dengan cara melakukan pengecekan pada kotak sebelumnya, apakah sudah ada yang sama dikiri atau dibawahnya, apabila sama, akan dilakukan random sampai mendapat hasil yang berbeda.

# Pembagian Kerja dalam Kelompok

|  |  |
| --- | --- |
| **Nama/Nim** | **Fungsi** |
| Rifda Annelies A/ 18216001 | * menu * about * highscore * welcomeScrn * gameOverScrn * move * isValidMove |
| Alessandro A/ 18216002 | * Smartrandom * Update * FallingCandy * RemoveZero * GetChain * IsChain * detectPossibleSwap |
| Athur Naufan M/ 18216003 | * Save * Load * IsEmptyFile * IsNameEq * ConvertSkor * inisialisasi |
| Al Varrel P /18216020 | * Acakisi * PrintKotak * Penggabungan fungsi * Desain Tampilan |

# Lampiran

## Deskripsi Tugas Besar

*Candy Crush Saga* adalah sebuah permainan teka - teki.. Langkah dasar permainan ini adalah menukar atau membalikkan posisi permen yang berdekatan secara vertikal atau *horizontal* untuk menciptakan barisan permen dengan tiga atau lebih warna yang sama.. Pemain kalah dari permainan jika *move* telah habis dan tidak mencapai target *score* yang telah ditentukan.

## Notulen Rapat

* Rapat 1

Pada rapat pertama dibahas fungsi-fungsi yang akan dibuat secara kasar, yaitu Inisialisasi board (termasuk ADT board, candy), sistem skor, pemain, move, pengecekan candy yang pecah, update board, update skor, cek game over, dan save serta load.

* Rapat 2

Pada rapat kedua dibahas algoritma-algoritma yang digunakan untuk menentukan move, pecahnya candy, shuffle candy, dan menggeser turun candy ketika pecah. Terutama untuk case candy yang berbentuk T atau L.

* Rapat 3

Pada rapat ketiga dibahas pembagian kerja yang dilakukan masing-masing anggota kelompok, sebagai berikut:

Rifda: Main menu, about, how to play, dan move.

Sandro: Pop Candy, shuffle candy, gerakan candy setelah pecah.

Athur: Pemain, skor.

Varrel: Pengacakan matrix, tampilan board, penyatuan.

## 6.3 Log Activity Anggota Kelompok

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nama | 11/11 | 12/11 | 13/11 | 14/11 | 15/11 | 16/11 | 17/11 | 18/11 | 19/11 |
| Rifda | Rapat #1 |  | Rapat #2 | asistensi |  | Rapat #3 |  | Coding | Coding |
| Sandro | Rapat #1 |  | Rapat #2 | Asistensi |  | Rapat #3 | Coding | Coding |  |
| Athur | Rapat #1 |  | Rapat #2 | Asistensi |  | Rapat #3 | Coding |  | Coding |
| Varrel | Rapat #1 |  | Rapat #2 | Asistensi |  | Rapat #3 |  | Coding |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Nama | 20/11 | 21/11 | 22/11 | 23/11 | 24/11 | 25/11 | 26/11 |
| Rifda |  | Coding | Coding | Penyusunan laporan | Menggabungkan program  Penyusunan laporan | Debugging  Penyusunan laporan | Testing  Penyusunan laporan |
| Sandro | Coding |  | Coding | Coding | Menggabungkan program | Debugging | Testing  Penyusunan laporan |
| Athur |  | Coding | Coding | Coding | Menggabungkan program | Debugging | Testing  Penyusunan laporan |
| Varrel | Coding | Coding | Coding | Coding | Menggabungkan program | Debugging  Penyusunan laporan | Testing  Penyusunan laporan |

|  |  |  |
| --- | --- | --- |
| Nama | 27/11 | 28/11 |
| Rifda | Penyusunan laporan  Debugging | Penyusunan laporan |
| Sandro | Penyusunan laporan  Debugging | Penyusunan laporan |
| Athur | Penyusunan laporan  Debugging | Penyusunan laporan |
| Varrel | Penyusunan laporan  Debugging | Penyusunan laporan |