

LAPORAN TUGAS BESAR

IF2210/Pemrograman Berorientasi Objek

ArkavQuarium

Dipersiapkan oleh:
K02M - Deny In Spanish


13515020 - Daniel Christian Pradipta Baso

13515116 - Aries Tri Sutrisno K A

13516026 - William Juniarta Hadiman

13516041 - Felix Septianus Darmawan

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung
Jl. Ganesha 10, Bandung 40132

	Sekolah Teknik Elektro dan Informatika ITB	Nomor Dokumen		Halaman
		<i>IF2210-TB-K02M-2</i>		<i>30</i>
		<i>Revisi</i>	<i>0</i>	<i>2018-04-11</i>

STEI- ITB	<i>IF2210-TB-K02M-2</i>	Halaman 1 dari 30 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		

Daftar Isi

Ringkasan	3
Penjelasan Tambahan Spesifikasi Tugas	3
Background	3
Main Menu	3
Penentuan Posisi Makanan Ikan Dilakukan Menggunakan Mouse	3
Pemain Dapat Mengambil Coin Menggunakan Mouse	3
Ikan Memiliki Tampilan Berbeda Saat Lapar	3
Rancangan Kelas	3
Perubahan dari Tugas Kecil	15
Rincian Kelas	18
Aquarium	18
Vector2	18
AquariumObject	19
Creature	19
Fish	19
IDestructible	20
Guppy	20
Piranha	20
Snail	21
Coin	21
Food	21
Program Utama	22
Test Script	22
Pembagian Kerja dalam Kelompok	25
Lampiran	26
Form Asistensi	26
Log Activity Anggota Kelompok	28
Screenshot Program	29

1 Ringkasan

Dalam tugas besar ini kami diminta untuk mengimplementasikan permainan Insaniquarium dengan nama ArkavQuarium, dimana Insaniquarium adalah sebuah game di mana kita dapat memelihara ikan yang akan menghasilkan uang yang dapat kita kumpulkan. Terdapat beberapa ikan dan hewan lain yang dapat dipelihara.

Laporan yang kami buat berisi tentang spesifikasi tugas tambahan yang kami tambahkan dengan penjelasan tiap spesifikasi, lalu ada rancangan kelas yang kami buat berisi Diagram Kelasnya dan juga rincian isi tiap kelasnya. Selanjutnya kami memberikan penjelasan mengenai program utama yang kami buat lalu yang terakhir adalah pengetesan program yang kami lakukan.

Hasil tugas besar ini sangat memuaskan. Dengan desain kelas pada tugas kecil yang baik, implementasi dan interaksi antar kelas menjadi sangat mudah dan rapi. Tugas besar ini sangat baik untuk mengaplikasikan pemrograman berorientasi objek.

2 Penjelasan Tambahan Spesifikasi Tugas

2.1 Background

Penambahan fungsi `draw_background`. Fungsi ini akan menerima path ke gambar, lalu mencetak gambar sehingga memenuhi layar tanpa mengubah aspect ratio.

2.2 Main Menu

Memunculkan tombol “Play Game” di awal permainan. Pemain akan menekan tombol tersebut untuk masuk ke game.

2.3 Penentuan Posisi Makanan Ikan Dilakukan Menggunakan Mouse

Posisi X dari cursor akan menentukan posisi X makanan muncul ketika tombol mouse ditekan, namun posisi Y tetap sama dengan nol.

2.4 Pemain Dapat Mengambil Coin Menggunakan Mouse

Coin dapat diambil oleh pemain jika pemain menekan mouse dan posisi mouse berada dalam jarak 32 pixel dari tengah koin.

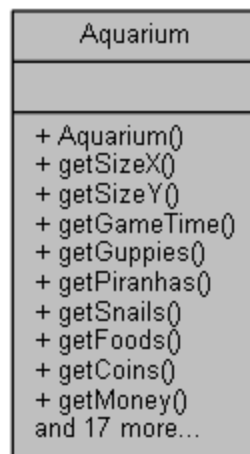
2.5 Ikan Memiliki Tampilan Berbeda Saat Lapar

Ikan yang sedang dalam keadaan lapar akan memiliki tampilan yang berbeda.

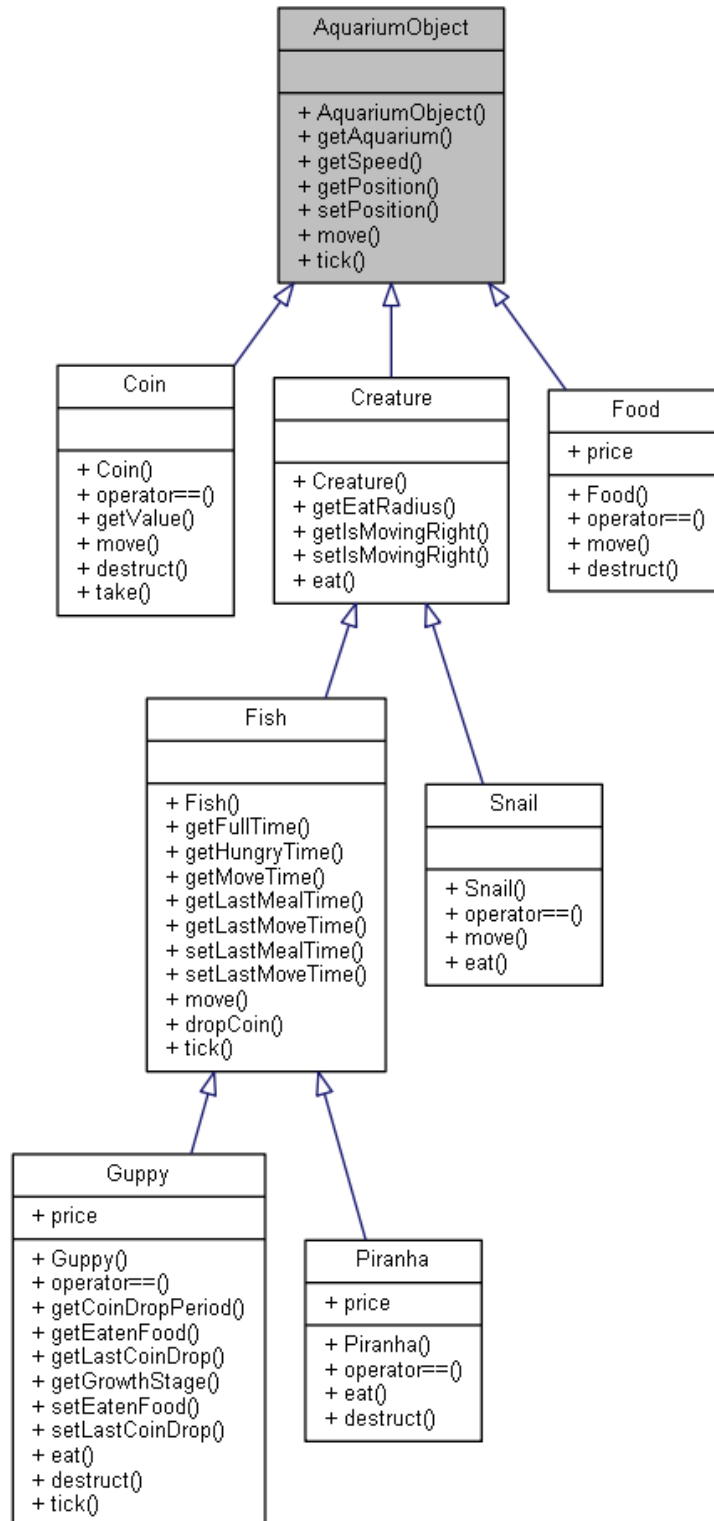
3 Rancangan Kelas

Rancangan kelas yang kami buat berisi 1 interface dan 12 kelas. Interface tersebut bernama kelas `IDestructible`. Sebelas kelas tersebut adalah `Aquarium`, `AquariumObject`, `Coin`, `Creature`, `Fish`, `Food`, `Guppy`, `LinkedList`, `Piranha`, `Snail`, `Vector2`. Dari sebelas kelas tersebut, kelas yang tidak

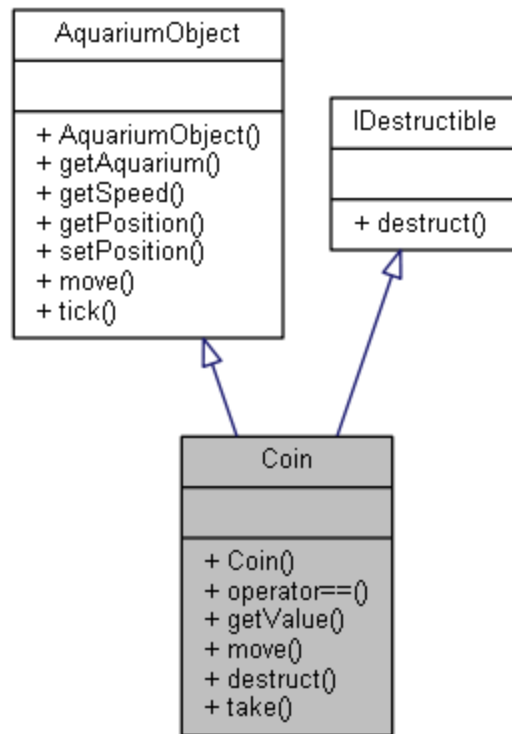
abstrak hanyalah Vector2, Aquarium, Coin, Food, Guppy, Piranha, dan Snail. LinkedList adalah kelas generic yang berfungsi sebagai penyimpanan data.



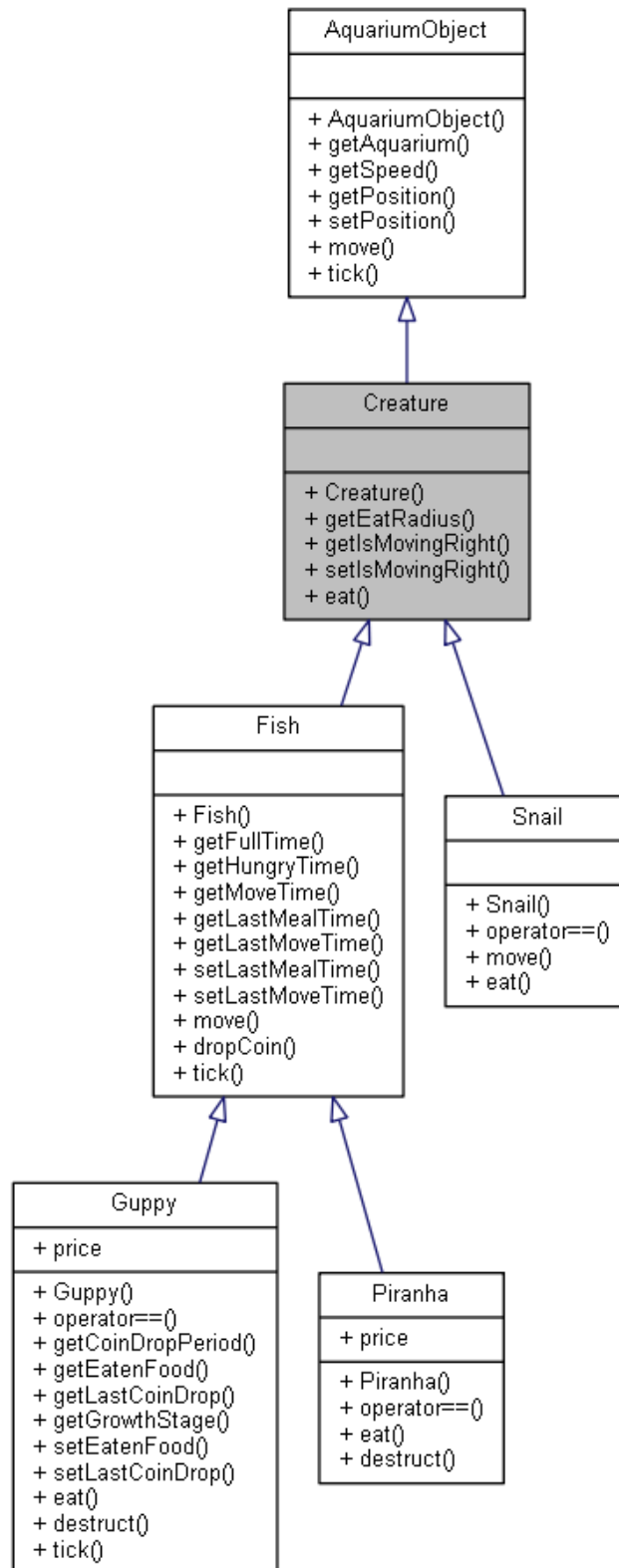
Gambar 3.1. Diagram Kelas Aquarium



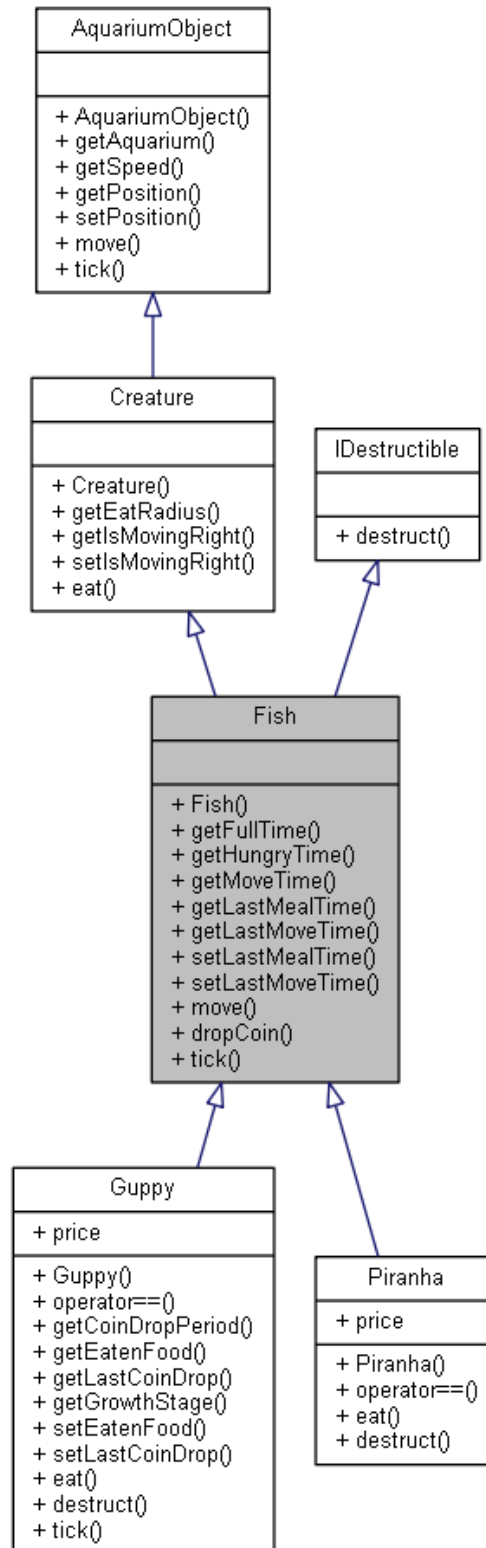
Gambar 3.2. Diagram Kelas AquariumObject



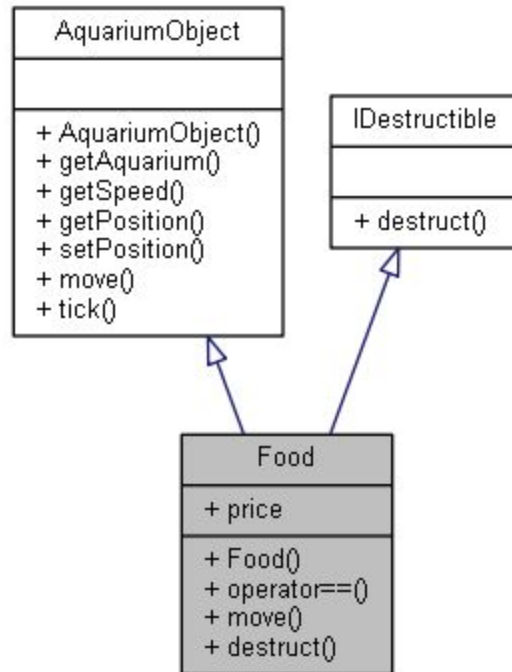
Gambar 3.3. Diagram Kelas Coin



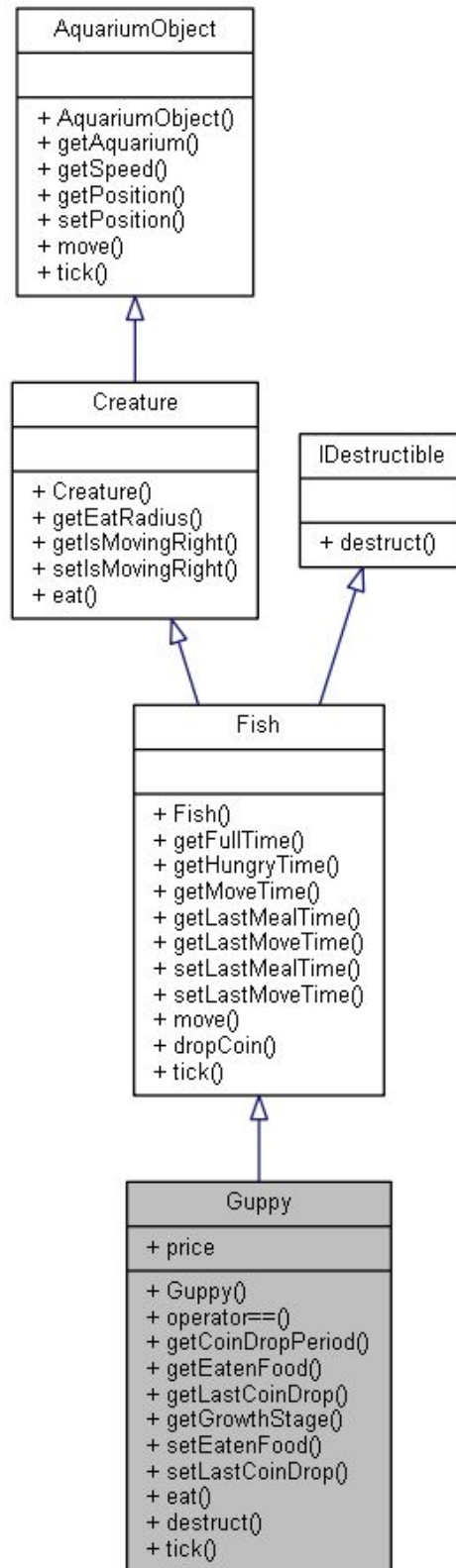
Gambar 3.4. Diagram Kelas Creature



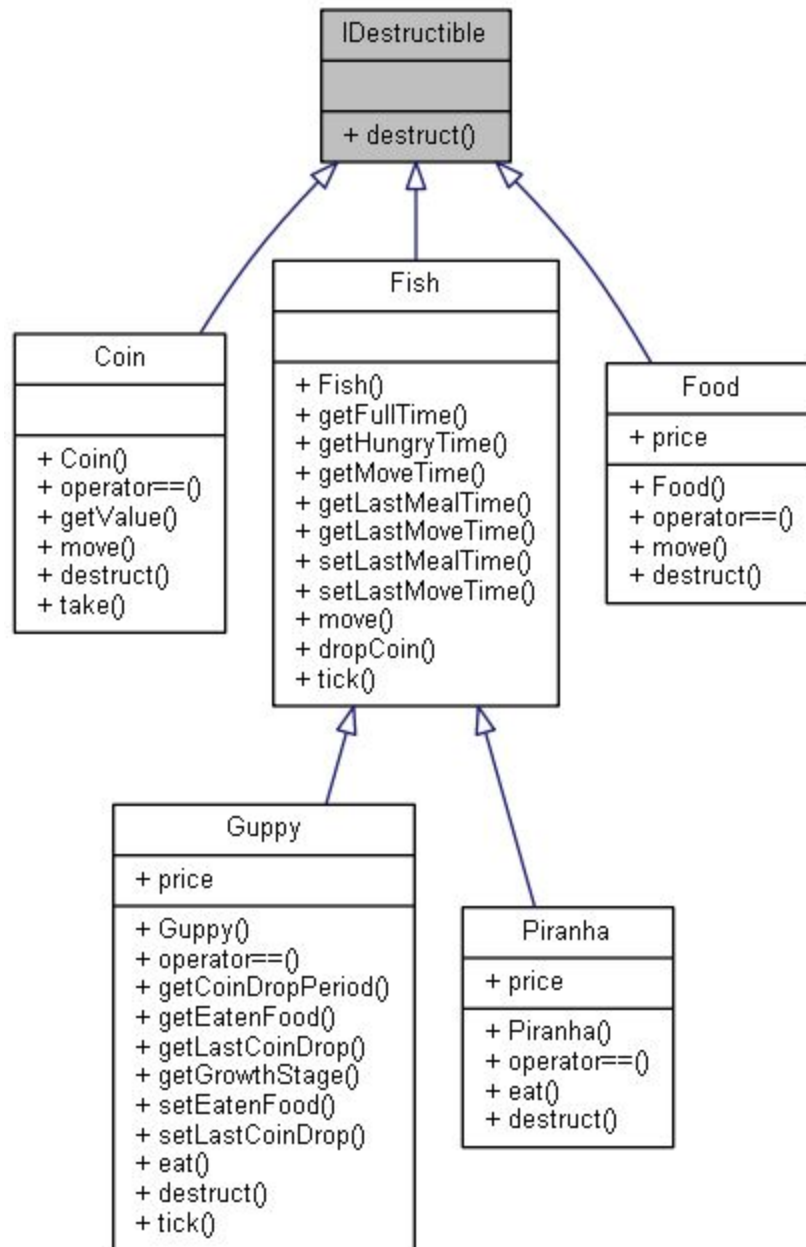
Gambar 3.5. Diagram Kelas Fish



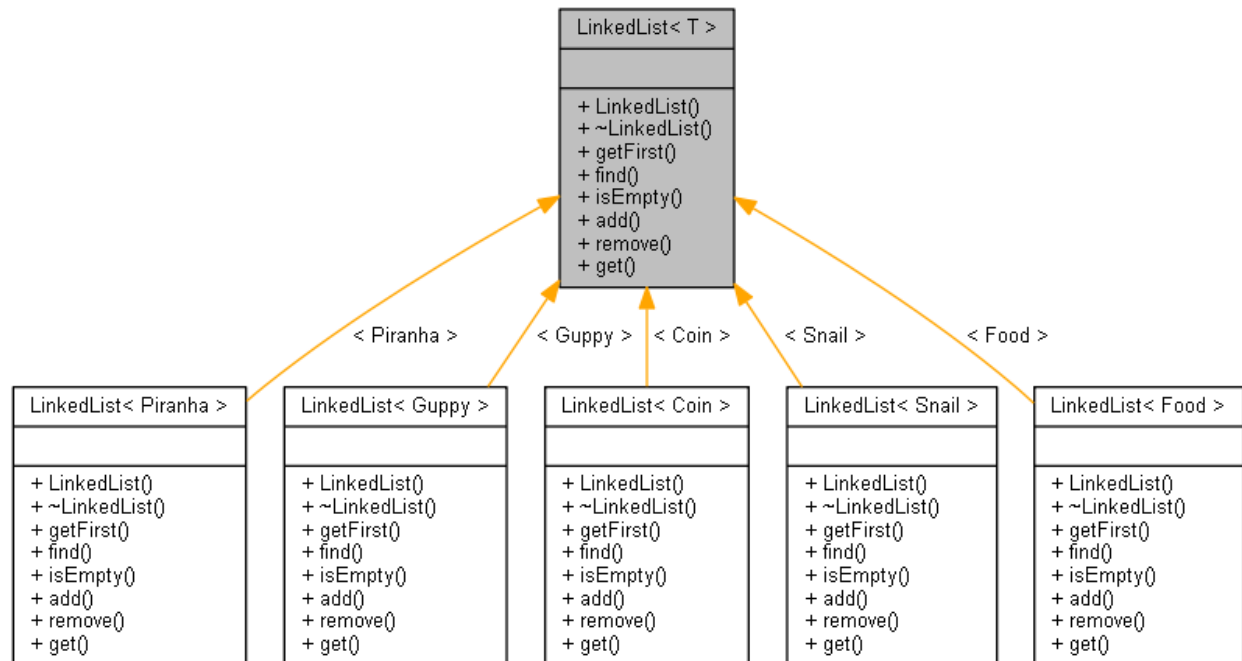
Gambar 3.6. Diagram Kelas Food



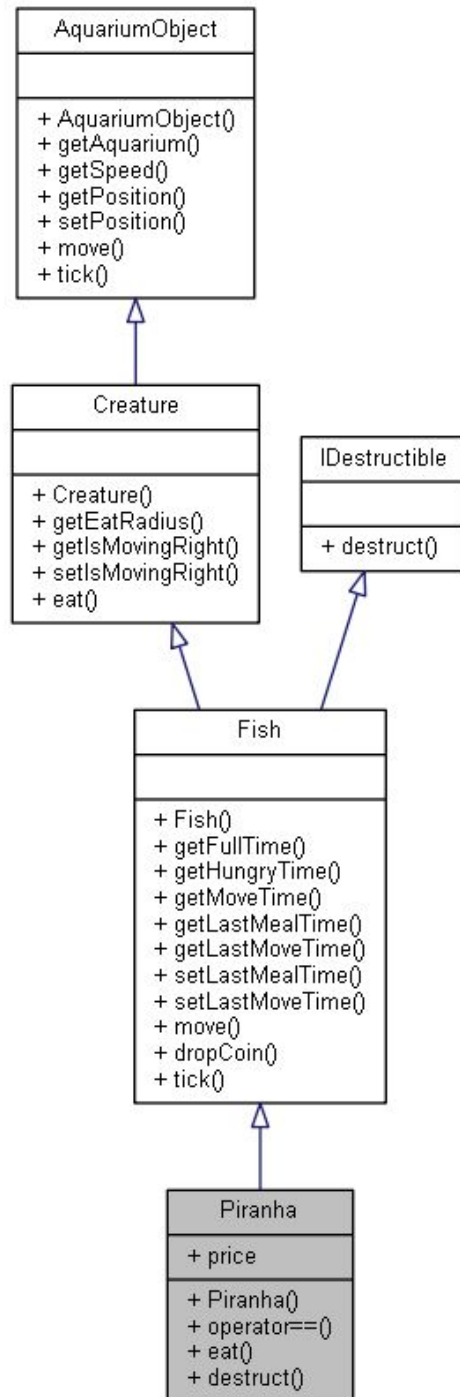
Gambar 3.7. Diagram Kelas Guppy



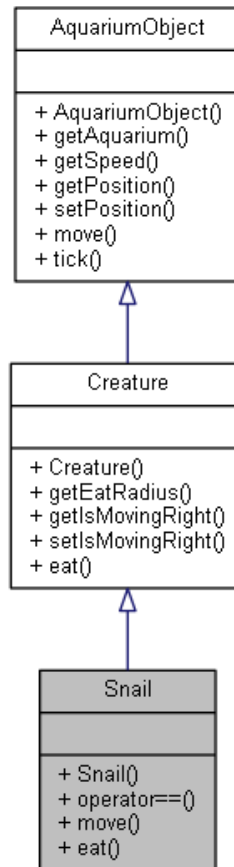
Gambar 3.8. Diagram Kelas IDestructible



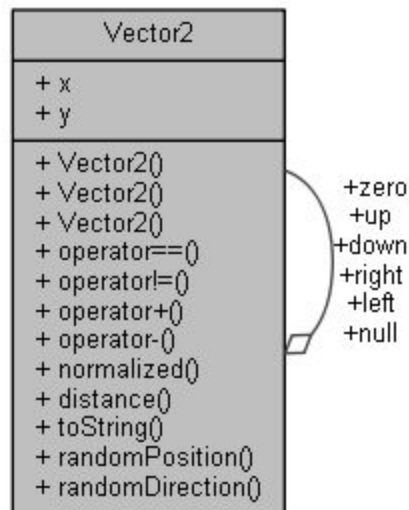
Gambar 3.9. Diagram Kelas LinkedList



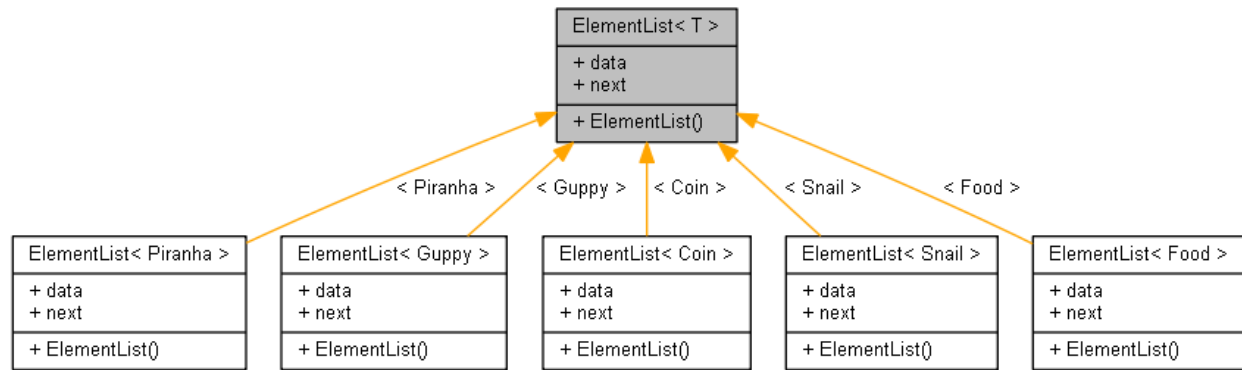
Gambar 3.10. Diagram Kelas Piranha



Gambar 3.11. Diagram Kelas Snail



Gambar 3.12. Diagram Kelas Vector2



Gambar 3.13. Diagram Kelas ElementList

3.1 Perubahan dari Tugas Kecil

Kelas	Perubahan
Aquarium	<ul style="list-style-type: none"> • Constructor dan Copy Constructor dihilangkan dan diganti Constructor ber-parameter. • Destructor dihilangkan. • Operator= dihilangkan. • Menambahkan method <code>getMoney()</code>, <code>getEgg()</code>, <code>getEggPrice()</code>, <code>setMoney(int)</code>, <code>setEgg(int)</code>, <code>setEggPrice()</code>. • Method <code>setGameTime()</code> diubah menjadi <code>setGameTime(int)</code>. • Method <code>spawnGuppy(int)</code>, <code>spawnPiranha(int)</code>, <code>spawnSnail(int)</code>, <code>spawnFood(int)</code> diganti menjadi <code>add(const Guppy&)</code>, <code>add(const Piranha&)</code>, <code>add(const Snail&)</code>, <code>add(const Food&)</code>. • Menambahkan method <code>add(const Coin&)</code> untuk menambahkan coin ke dalam akuarium. • Menambahkan method <code>remove</code> untuk setiap method <code>add</code> yang ada. • Menambahkan atribut <code>money</code>, <code>egg</code>, dan <code>eggPrice</code> bertipe <code>int</code>.
AquariumObject	<ul style="list-style-type: none"> • Menghapus ADT <code>Vector2</code>. • Atribut <code>const Aquarium* aquarium</code> diubah menjadi <code>Aquarium& aquarium</code>
Coin	<ul style="list-style-type: none"> • Constructor berparameter <code>int</code> – <code>Coin(int)</code> – diganti menjadi berparameter <code>Aquarium&</code>, <code>Vector2</code>, dan <code>int</code> – <code>Coin(Aquarium&, Vector2, int)</code>. • Menambahkan method <code>operator==(const Coin&) const</code>,

	<p><code>destruct()</code>, dan <code>take()</code>.</p> <ul style="list-style-type: none"> • Method <code>getValue()</code> menjadi <code>const – int getValue() const</code>.
Creature	<ul style="list-style-type: none"> • Mengubah parameter konstruktor yang awalnya <code>Aquarium*</code> menjadi <code>Aquarium&</code> • Menambahkan method <code>getIsMovingRight()</code> dengan tipe balikan boolean dan <code>setIsMovingRight(bool)</code> dengan tipe balikan void. • Menambahkan atribut <code>isMovingRight</code> dengan tipe <code>bool</code>. • Menghapus method <code>findFood()</code>. • Mengubah tipe method <code>eat()</code> dari boolean menjadi <code>Vector2</code> dan mengubah menjadi <code>Abstract</code>.
Fish	<ul style="list-style-type: none"> • Konstruktor Fish yang awalnya berparameter <code>Aquarium*</code>, <code>float</code>, <code>float</code>, <code>int</code> – <code>Fish(Aquarium*, float, float, int)</code> – diubah dengan menghilangkan parameter <code>int</code> dan mengubah parameter <code>Aquarium*</code> menjadi <code>Aquarium&</code> – <code>Fish(Aquarium&, float, float)</code> • Mengubah tipe pengembalian pada method <code>getFullTime()</code>, <code>getHungryTime()</code>, dan <code>getLastMealTime()</code> dari <code>int</code> menjadi <code>float</code>. • Menghilangkan method <code>getPrice()</code>, <code>isMovingRight()</code>, dan <code>setIsMovingRight()</code>. • Menambahkan method <code>getMoveTime()</code> dan <code>getLastMoveTime()</code> dengan tipe pengembalian <code>int</code> dan <code>const</code>, dan <code>setLastMoveTime(int)</code>. • Mengubah method <code>setLastMealTime()</code> menjadi berparameter <code>int</code> – <code>setLastMealTime(int)</code>. • Menambah method <code>tick()</code>. • Menambah atribut <code>moveRandomly()</code> dengan tipe <code>void</code>, <code>moveTime</code> dengan tipe <code>int</code>, <code>direction</code> dengan tipe <code>Vector2</code>. • Menghilangkan atribut <code>price</code> dan <code>isMovingRight</code>. • Mengubah tipe atribut <code>fullTime</code>, <code>hungryTime</code>, dan <code>lastMealTime</code> dari <code>float</code> menjadi <code>int</code>.
Food	<ul style="list-style-type: none"> • Mengubah konstruktor menjadi berparameter <code>Aquarium&</code> dan <code>float</code> – <code>Food(Aquarium&, float)</code> • Menambahkan method <code>operator==(const Food& other)</code> <code>const</code> dan <code>destruct()</code>. • Menambahkan atribut <code>price</code> dengan tipe <code>int</code>.
Guppy	<ul style="list-style-type: none"> • Mengubah konstruktor menjadi berparameter <code>Aquarium&</code>. • Menghapus method <code>getSpawnPeriod()</code> dan <code>findFood()</code>.

	<ul style="list-style-type: none"> ● Menghapus atribut <code>spawnPeriod</code>. ● Menambahkan method <code>operator==(const Guppy&) const, destruct(), tick()</code>. ● Menambahkan atribut <code>price</code> bertipe <code>int</code>, <code>spawnPeriod</code> bertipe <code>int</code>, dan <code>findFood()</code> bertipe <code>Food*</code>. ● Mengubah method <code>eat()</code> yang awalnya bertipe <code>boolean</code> menjadi bertipe <code>Vector2</code>.
LinkedList	<ul style="list-style-type: none"> ● Tidak menginclude <code><cstddef></code> ● <code>ElementList</code> tidak menggunakan <code>class</code> tetapi menggunakan <code>struct</code>. ● Atribut <code>data</code> diinisiasi menggunakan constructor initialization list dan <code>next</code> diubah dari <code>NULL</code> menjadi <code>0</code> ● Atribut dengan tipe <code>ElementList*</code> diubah menjadi generik dengan mengubahnya menjadi <code>ElementList*<T></code> ● Konstruktor <code>LinkedList</code> menginisiasi <code>first</code> menjadi <code>0</code>, yang awalnya adalah <code>NULL</code>. ● Menambahkan Copy Constructor. ● Mengubah implementasi destructor. ● Menambahkan method <code>getFirst()</code> dengan tipe balikan <code>ElementList<T>*</code>. ● Menghapus <code>operator=</code> ● Mengubah implementasi untuk method <code>find(const T&), isEmpty(), add(const T&), remove(const T&), dan get(int)</code>.
Piranha	<ul style="list-style-type: none"> ● Konstruktor <code>Piranha</code> menjadi berparameter <code>Aquarium&</code>. ● Menghapus method <code>findFood()</code>. ● Menambahkan method <code>destruct()</code> dan <code>operator==(const Piranha&) const</code>. ● Mengubah tipe balikan <code>eat()</code> dari <code>bool</code> menjadi <code>Vector2</code>. ● Menambah atribut <code>price</code> dengan tipe <code>int</code> dan <code>findGuppy()</code> dengan tipe <code>Guppy*</code>.
Snail	<ul style="list-style-type: none"> ● Konstruktor <code>Snail</code> menjadi berparameter <code>Aquarium&</code>. ● Menghapus method <code>findFood()</code>. ● Menambahkan <code>operator==(const Snail&) const</code>. ● Mengubah tipe balikan <code>eat()</code> dari <code>bool</code> menjadi <code>Vector2</code>. ● Menambah atribut <code>findCoin()</code> dengan tipe balikan <code>Coin*</code>.
Vector2	<ul style="list-style-type: none"> ● Menambah kelas <code>Vector2</code>, beserta method dan attributnya.

4 Rincian Kelas

4.1 Aquarium

Kelas Aquarium berfungsi sebagai pengontrol waktu game dan penyimpanan objek yang ada di dalam game.

Method:

- Constructor: Membuat objek
- Getter: Mengembalikan nilai variabel
- Setter: Mengubah nilai variabel
- Add: Menambahkan objek ke aquarium
- Remove: Menghapus objek dari aquarium
- Tick: Menambah gameTime sebanyak satu dan memanggil tick dari semua objek yang ada di aquarium

Atribut:

- SizeX: Lebar aquarium
- SizeY: Tinggi aquarium
- GameTime: Unit waktu dari game
- Guppies: Penyimpanan guppy
- Piranhas: Penyimpanan piranha
- Snails: Penyimpanan snail
- Foods: Penyimpanan food
- Coins: Penyimpanan coin
- Money: Jumlah uang yang dipakai untuk membeli
- Egg: Jumlah telur yang telah dibeli
- EggPrice: Harga telur

4.2 Vector2

Kelas Vector2 berfungsi sebagai tipe bentukan untuk posisi dan arah.

Method:

- Constructor: Membuat objek
- Getter: Mengembalikan nilai variabel
- Setter: Mengubah nilai variabel
- Operator==: Membandingkan kesamaan dengan Vector2 lain
- Operator!=: Membandingkan perbedaan dengan Vector2 lain
- Operator+: Menambahkan nilai x dan y dengan Vector2 lain
- Operator-: Mengurangi nilai x dan y dengan Vector2 lain
- Normalized: Membuat ukuran vektor sebesar 1 tanpa mengubah arahnya
- Distance: Menghitung jarak ke Vector2 lain
- Operator*: Mengalikan nilai x dan y dengan k
- RandomPosition: Memberikan Vector2 yang memiliki x dalam range 0-x, dan y dalam range 0-y

- RandomDirection: Memberikan Vector2 yang memiliki arah acak dengan ukuran 1
- ToString: Mengembalikan Vector2 dalam bentuk string

Atribut:

- Null: Mengembalikan Vector2 dengan $x = -1$ dan $y = -1$
- Zero: Mengembalikan Vector2 dengan $x = 0$ dan $y = 0$
- Right: Mengembalikan Vector2 dengan $x = 1$ dan $y = 0$
- Left: Mengembalikan Vector2 dengan $x = -1$ dan $y = 0$
- Up: Mengembalikan Vector2 dengan $x = 0$ dan $y = -1$
- Down: Mengembalikan Vector2 dengan $x = 0$ dan $y = 1$

4.3 AquariumObject

Kelas AquariumObject berfungsi sebagai kelas paling mendasar dari objek yang ada di dalam aquarium

Method:

- Constructor: Membuat objek
- Getter: Mengembalikan nilai variabel
- Setter: Mengubah nilai variabel
- Move: Fungsi abstrak untuk bergerak
- Tick: Fungsi virtual untuk melakukan aksi setiap satu satuan waktu

Atribut:

- Aquarium: Referensi ke aquarium ikan ini berada
- Speed: Kecepatan objek bergerak
- Position: Posisi objek berada

4.4 Creature

Kelas Creature berfungsi sebagai kelas dari objek yang hidup. Kelas ini adalah turunan dari AquariumObject

Method:

- Constructor: Membuat creature
- Getter: Mengembalikan nilai variabel
- Setter: Mengubah nilai variabel
- Eat: Fungsi abstrak untuk makan

Atribut:

- EatRadius: Jarak maksimum creature dapat meraih makanan
- IsMovingRight: Menentukan arah horizontal creature bergerak

4.5 Fish

Kelas Fish berfungsi sebagai kelas dari objek yang dapat menghasilkan coin. Kelas ini adalah turunan dari Creature.

Method:

- Constructor: Membuat fish
- Getter: Mengembalikan nilai variabel

- Setter: Mengubah nilai variabel
- Move: Menggerakkan fish jika lapar dan terdapat makanan, jika kondisi tersebut tidak terpenuhi, maka ikan akan bergerak acak dan berganti arah setiap moveTime
- MoveRandomly: Mengarahkan fish untuk bergerak secara acak
- DropCoin: Membuat coin di posisi fish
- Tick: Menghilangkan fish jika fish lapar melebihi hungryTime

Atribut:

- FullTime: Durasi fish kenyang sebelum lapar
- HungryTime: Durasi fish lapar sebelum mati
- MoveTime: Durasi fish akan bergerak ke arah yang sama sebelum berganti arah
- LastMealTime: Waktu terakhir fish makan
- LastMoveTime: Waktu terakhir fish berganti arah
- Direction: Arah ikan ini bergerak

4.6 IDestructible

Kelas IDestructible berfungsi sebagai kelas dari objek yang dapat dihancurkan dari akuarium. Kelas ini adalah interface.

Method:

- Destruct: Menghilangkan objek dari akuarium

4.7 Guppy

Kelas Guppy berfungsi sebagai kelas dari Guppy. Kelas ini adalah turunan dari Fish.

Method:

- Constructor: Membuat guppy
- Operator==: Operasi komparasi yang bersifat object comparison
- Getter: Mengembalikan nilai variabel
- Setter: Mengubah nilai variabel
- GetGrowthStage: Mengembalikan tahap pertumbuhan guppy dari jumlah makanan yang telah dimakan
- Eat: Fungsi untuk memakan food di sekitar guppy
- FindFood: Fungsi untuk mendapatkan lokasi food di akuarium
- Destruct: Menghilangkan guppy dari list of guppies
- Tick: Menjatuhkan koin setiap coinDropPeriod

Atribut:

- CoinDropPeriod: Periode guppy akan menjatuhkan coin
- EatenFood: Jumlah food yang telah dimakan guppy
- LastCoinDrop: Waktu terakhir guppy menjatuhkan coin
- Price: Harga dari guppy

4.8 Piranha

Kelas Piranha berfungsi sebagai kelas dari Piranha. Kelas ini adalah turunan dari Fish.

Method:

STEI- ITB	IF2210-TB-K02M-2	Halaman 20 dari 30 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		

- Constructor: Membuat piranha
- Operator==: Operasi komparasi yang bersifat object comparison
- Eat: Fungsi untuk memakan guppy di sekitar piranha
- FindFood: Fungsi untuk mendapatkan lokasi guppy di akuarium
- Destruct: Menghilangkan piranha dari list of piranhas

Atribut:

- Price: Harga dari piranha

4.9 Snail

Kelas Snail berfungsi sebagai kelas dari Snail. Kelas ini adalah turunan dari Creature.

Method:

- Constructor: Membuat snail
- Operator==: Operasi komparasi yang bersifat object comparison
- Move: Fungsi untuk bergerak horizontal di dasar akuarium
- Eat: Fungsi untuk memakan coin di sekitar snail
- FindFood: Fungsi untuk mendapatkan lokasi koin di akuarium

4.10 Coin

Kelas Coin berfungsi sebagai kelas dari coin. Kelas ini adalah turunan dari AquariumObject.

Method:

- Constructor: Membuat coin
- Operator==: Operasi komparasi yang bersifat object comparison
- Getter: Mengembalikan nilai variabel
- Move: Menggerakan coin ke dasar akuarium
- Take: Menambahkan money sebanyak value
- Destruct: Menghilangkan coin dari list of coins

Atribut:

- Value: Nilai dari coin

4.11 Food

Kelas Coin berfungsi sebagai kelas dari coin. Kelas ini adalah turunan dari AquariumObject.

Method:

- Constructor: Membuat food
- Operator==: Operasi komparasi yang bersifat object comparison
- Move: Menggerakan food ke dasar akuarium dan menghancurkan food ketika menyentuh dasar akuarium
- Destruct: Menghilangkan food dari list of foods

Atribut:

- Price: Harga dari food

5 Program Utama

Program utama akan menginisialisasi window untuk game. Selanjutnya, main menu akan ditampilkan dan loop untuk menangkap klik dijalankan. Jika pemain menekan tombol “Play Game” maka loop akan berhenti dan game akan diinisialisasi dengan membuat akuarium dan mengeset waktu. Kemudian, loop game akan dijalankan. Isi dari loop game adalah menjalankan tick jika waktu dari tick sebelumnya telah melebihi speed / tps. Kemudian loop akan menangkap input user dan memberikan reaksi. Kemudian tampilan akan diatur kembali. Background akan digambar terlebih dahulu, lalu UI, lalu text, lalu objek-objek di dalam game. Terakhir, loop akan mengecek kondisi menang kalah. Jika kondisi terpenuhi, loop akan berhenti dengan nilai variabel win yang telah terisi. Jika win berisi true, maka akan dituliskan text win, jika tidak maka akan dituliskan text lose. Setelah itu pemain dapat menekan tombol apapun untuk keluar dari game.

6 Test Script

No.	Kelas	Nama File Driver	Fitur/Method yang diuji	Kasus Pengujian	Hasil yang Diharapkan	Hasil yang Keluar
1	Vector2	d_Vector2.cpp	1. default ctor 2. ctor berparam 3. cctor 4. operator!= 5. operator+ 6. operator- 7. normalized() 8. distance() 9. operator*(param1) 10. operator*(param2) 11.	7. normalized() -1- x: 10 y: -10 -2- x: 10 y: 20	7. -1- x : 0.707107 y : -0.707107 -2- x : 0.447214 y : 0.894427	7. -1- x : 0.707107 y : -0.707107 -2- x : 0.447214 y : 0.894427
2	Coin	d_Coin.cpp	1. ctor berparam 2. operator== 3. getValue() 4. move() 5. take()	4. move() x : 20 y : 30 x : 49 y : 49 5. take() test1.take() //Coin value 5 test2.take() //Coin value 10	4. x : 20 y : 31 x : 49 y : 50 5. Current aqtest money: 105 Current aqtest money: 115	4. x : 20 y : 31 x : 49 y : 50 5. Current aqtest money: 105 Current aqtest money: 115
3	Food	d_Food.cpp	1. ctor berparam	3. move()	3.	3.

			2. operator== 3. move()	-1 move- x : 40 y : 0 -26 move- x : 8 y : 0	x : 40 y : 2 x : 8 y : 52	x : 40 y : 2 x : 8 y : 52
4	Guppy	d_Guppy.cpp	1. ctor berparam 2. operator== 3. getCoinDropPeriod() 4. getEatenFood() 5. getLastCoinDrop() 6. getGrowthStage() 7. setEatenFood() 8. setLastCoinDrop() 9. eat() 10. tick()	10. tick() setGameTime(200) setGameTime(500)	10. Coin value dropped after tick() : 2 Coin value dropped after tick() : 2	10. Coin value dropped after tick() : 2 Coin value dropped after tick() : 2
5	Piranha	d_Piranha.cpp	1. ctor berparam 2. operator== 3. eat()			
6	Snail	d_Snail.cpp	1. ctor berparam 2. operator== 3. move() 4. eat()	3.move() //including eat() Snail position before move : 25 Coin is placed in absis of 40	3. Snail position after move : 28 Snail did move.	3. Snail position after move : 28 Snail did move.
7	Aquarium	Aquarium_driver.cpp	1. getSizeX() 2. getSizeY() 3. getGameTime() 4. getGuppies() 5. getPiranhas() 6. getSnails() 7. getCoins() 8. add() 9. remove() 10. getMoney() 11. getEgg() 12. getEggPrice() 13. setGameTime(_gameTime) 14. setMoney(_money) 15. setEgg(_egg)	1,2,10,12 : Value must be equal with value in constructor parameters. 3. initialized when constructed (check the value) 4,5,6,7,8 : after adding list 4,5,6,7 using add, then check the list is null or not	1. 10 2. 20 3. 0 4,8 : not NULL 5,8 : not NULL 6,8 : not NULL 7,8 : not NULL 4,9 : NULL 5,9 : NULL 6,9 : NULL 7,9 : NULL 10. 1000 11. 0 12. 100 13. 20 14. 1500 15. 1	1. 10 2. 20 3. 0 4,8 : not NULL 5,8 : not NULL 6,8 : not NULL 7,8 : not NULL 4,9 : NULL 5,9 : NULL 6,9 : NULL 7,9 : NULL 10. 1000 11. 0 12. 100 13. 20 14. 1500 15. 1

			16. setEggPrice (_eggPrice) 17. tick()	4,5,6,7,9 : after removing list 4,5,6,7 using remove, then check the list is null or not 11. initialized when constructed (check the value) 13,14,15,16 : set the value then using getter to check if the value is changed. 17. check setGameTime method and the rest will be checked in other drivers	16. 150 17. 21	16. 150 17. 21
8	Linked List	LinkedList_Driver.cpp	1. add() 2. find() 3. getFirst() 4. get(_value) 5. remove() 6. isEmpty()	1. add() method will be checked (not direct) by method 2,3,4, and 5 2. find() will be checking 2 condition. If found equal value, then will return index of the first value. if not found equal value, then will return -1 3. must be the first value of the list 4. return index of element list with value same with parameter	2. 0 (when found) 2. -1 (when not found) 3. 5 4. 1 6. True	2. 0 (when found) 2. -1 (when not found) 3. 5 4. 1 6. True

				5. will be checked (not direct) by isEmpty() 6. after removed all element, then check the list which the list must be empty		
--	--	--	--	------------------------------------------------------------------------------------------------------------------------------------	--	--

7 Pembagian Kerja dalam Kelompok

NIM - Nama	Pekerjaan
13515020 - Daniel C. P. B.	Dokumentasi dan laporan
13515116 - Aries T. S. K. A.	Pembuatan driver dan laporan test script
13516026 - Willian J. H.	Pembuatan driver dan laporan test script
13516041 - Felix S. D.	Implementasi kelas dan program utama dan laporan kelas dan program utama

8 Lampiran

8.1 Form Asistensi

**Form Asistensi Tugas Besar
IF2210/Pemrograman Berorientasi Objek
Sem. 2 2017/2018**

No. Kelompok/Kelas : 13/402
Nama Kelompok : K02M
Anggota Kelompok (Nama/NIM) :
1. Daniel Christian / 13515000
2. Annes Tri S KA / 13515116
3. Felix Septamus / 13516041
4. William Kurniata

Asisten Pembimbing :

Asistensi I

Tanggal : 5 April 2018

Tempat : Lab Perograman

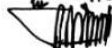
Kehadiran Anggota Kelompok:

No
NIM
Tanda tangan

1
13518020



2
13515116



3
1351604



4
13516026



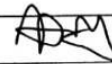
5

6

Catatan Asistensi:

1. Tidak pakai library self, jadi ga ke compile di Windows
2. Perubahan yang dilakukan, lebih ke kenapa kelasnya diganti atau dihapus, (contoh: lupa di kasih parameter, returnnya kurang ke referensi).
3. Kalo yg custom harus di definisikan
4. Driver nya per kelas tapi harus bisa ngerti semua method

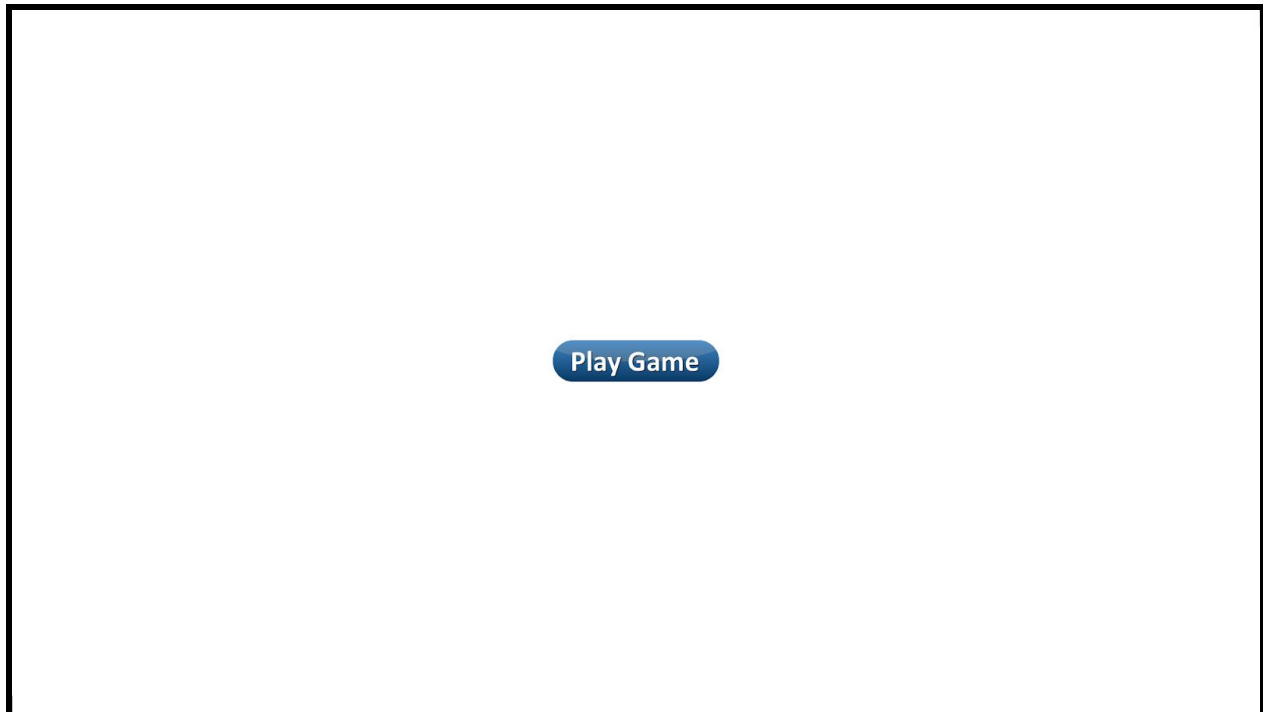
Tanda Tangan Asisten:



8.2 Log Activity Anggota Kelompok

Tanggal	13515020 Daniel	13515116 Aries	13516026 William	13516041 Felix
Kamis 5 April	Asistensi	Asistensi	-	Asistensi
Jumat 6 April	-	-	-	Implementasi kelas
Sabtu 7 April	Dokumentasi doxygen	-	-	Implementasi program utama
Minggu 8 April	-	-	-	Perbaikan bug dan tampilan
Senin 9 April	Update dokumentasi	Pembuatan driver	-	Implementasi bonus
Selasa 10 April	Dokumentasi komentar	Pengujian driver	Pembuatan driver	Penambahan komentar
Rabu 11 April	Pembuatan laporan	Pembuatan laporan	Pembuatan laporan	Pembuatan laporan

8.3 Screenshot Program



Win