**CP Election Documentation**

**Created by**

Niti Assavaplakorn 6031031221

Natchapol Srisang 6031308121

**2110215 Programming Methodology**

**Semester 1 Academic Year 2018**

**Chulalongkorn University**

**CP Election Documentation**

**Introduction**

CP Election is a single player survival game. The goal is to fight against enemies and finally beat the boss of the game. In addition to normal attack, the player can use various types of helpful(?) items dropped from enemies to make the game easier.

**Implementation Details**

**\* Noted that Access Modifier can be listed below**

+ (public)

# (protected)

- (private)

Underlined (static)

*Italic (abstract)*

**1. Package application**

1.1. Class Main

1.1.1. Fields

|  |  |
| --- | --- |
| - Stage stage | Store the start stage. |
| - StartScene startScene | Store the start scene. |
| - Scene instructions | Store the instruction scene. |
| - Scene gameScene | Store the game scene. |

1.1.2. Methods

|  |  |
| --- | --- |
| + void main(String[] args) | Launch the game. |
| + void start(Stage primaryStage) | The main entry point of JavaFX application. |
| + void setCenter() | Set the stage at the center of screen. |
| Getters of every fields & setter of startScene |  |

**2. Package constants**

2.1. Enum CCType (NONE, STUN, SLOW, SILENCE)

This enum contains all crowd control (CC) status types.

2.2. Class Images

This class contains all images.

2.2.1. Fields

|  |  |
| --- | --- |
|  |  |

2.3. Class Sounds

This class contains all sounds.

2.3.1. Fields

|  |  |
| --- | --- |
|  |  |

**3. Package controller**

3.1. Class ?

3.1.1. Fields

|  |  |
| --- | --- |
|  |  |

**4. Package exception**

This package contains all exceptions.

4.1. CannotAttackException

This exception is thrown when Character cannot attack.

4.2. CannotMoveException

This exception is thrown when Character cannot move.

4.3. CannotUseItemException

This exception is thrown when Character cannot use item.

4.4. InventoryFullException

This exception is thrown when Player’s inventory is full.

4.5. InventoryEmptyIndexException

This exception is thrown when Player tries to use item that player does not have.

4.6. ItemTypeNoExistException

This exception is thrown when Player tries to use item that type does not exist in the game.

**5. Package input**

5.1. Class KeyInput

This class contains all input keys available in the game.

5.1.1. Fields

|  |  |
| --- | --- |
| - Set<KeyCode> activeKeys | Store the start stage. |
| - Queue<KeyCode> triggerkeys | Store the start scene. |
| - final Set<KeyCode> UNPOLLABLE\_KEYS | Store the instruction scene. |

5.1.2. Methods

|  |  |
| --- | --- |
| - void addKey(KeyCode code) |  |
| - void removeKey(Keycode code) |  |
| + boolean pressingKey(KeyCode code) |  |
| + void clear() |  |
| + boolean isPollAvailable() |  |
| + KeyCode pollKey() |  |
| + void bindScene(Scene scene) |  |

**6. Package model**

6.1. Interface IUpdatable

This interface is for entities that need to be updated.

6.1.1. Methods

|  |  |
| --- | --- |
| + void update() |  |

6.2. Interface Iuseable

This interface is for entities that can be used.

6.2.1. Methods

|  |  |
| --- | --- |
| + void use() |  |

6.3. Class Frame

This class is frames of every entities in the game.

6.3.1. Fields

|  |  |
| --- | --- |
| # double posX | x-coordinate of the frame, start from left to right |
| # double posY | y-coordinate of the frame, start from top to bottom |
| # double width | Width of the frame |
| # double height | Height of the frame |

6.3.2. Constructor

|  |  |
| --- | --- |
| + Frame(double posX, double poxY, double width, double height) | Initialize every fields. |

6.3.3. Methods

|  |  |
| --- | --- |
| + boolean isCollideWith(Frame f) | Return true if this frame collides with frame f, otherwise false. |
| Getters & Setters of every fields |  |

6.4. *Class Entity (abstract)*

This class represents every entities in the game.

6.4.1. Fields

|  |  |
| --- | --- |
| # String name | Entity’s name |
| - Image image | Entity’s image |

6.4.2. Constructor

|  |  |
| --- | --- |
| + Entity(double posX, double posY, double width, double height, String name, Image image) | Initialize every fields. |

6.4.3. Methods

|  |  |
| --- | --- |
| + void render(GraphicContext gc) | Render the entity at coordinate (posX, posY) relative to the screen. |
| Getters of every fields & setter of image |  |

6.5. *Class MoveableEntity (abstract)*

This class represents every moveable entities of the game.

6.5.1. Fields

|  |  |
| --- | --- |
| + final int LEFT = -1 | Static values that represent the entity’s facing direction |
| + final int RIGHT = 1 |
| # double speedX | Speed in x-axis |
| # double speedY | Speed in y-axis |
| # int facing | Entity’s facing direction (LEFT or RIGHT) |
| - Image imageL | Image when entity is facing left |
| - Image imageR | Image when entity is facing right |

6.5.2. Constructors

|  |  |
| --- | --- |
| + MovableEntity(double posX, double posY, String name, Image imageL, Image imageR) | Initialize every fields, with width and height of the object equal to the width and height of imageL respectively, image is null, and is facing right. |

6.5.3. Methods

|  |  |
| --- | --- |
| + void render(GraphicContext gc) | Render the entity at coordinate (posX, posY) relative to the screen. |
| + void move() | Move the entity. |
| + void setFacing(int facing) | Set facing direction of the entity, and change the image that corresponds to the direction |
| Getters of every fields except facing, and images & Setters of speedX, speedY |  |

6.6. *Class Character (abstract)*

This class represents every characters of the game, including player, enemies and boss.

6.6.1. Fields

|  |  |
| --- | --- |
| # int hp | Character’s health point |
| # int maxHp | Character’s maximum health point |
| # int minAtk | Character’s minimum attack point |
| # int maxAtk | Character’s maximum attack point |
| # int def | Character’s defense point |
| # CCType status | CC effect applied to the character (default is NONE) |
| - int attackTick | Attack frame counter count since last attack frame (start from 0). |
| - int attackCooldown | The number of frame for character to be able to attack again. |
| - int ccedTick | CC frame counter since last CC applied frame (start from 0). |
| - int ccedDuration | The number of frames which CC is applied |
| - boolean isAttacking | True if the attack is cooling down, otherwise false. |

6.6.2. Constructors

|  |  |
| --- | --- |
| + Character(double posX, double posY, String name, Image imageL, Image imageR, int maxHp, int atk, int def, int attackCooldown) | Initialize every fields. |

6.6.3. Methods

|  |  |
| --- | --- |
| + boolean isDead() | Return true if the character is dead, otherwise false. |
| + boolean canAttack() | Return true if the character can attack, otherwise false. |
| + boolean canMove() | Return true if the character can move, otherwise false. |
| + boolean isCCed() | Return true if the character was applied CC effect, otherwise false. |
| + boolean isStunned() | Return true if the character is stunned, otherwise false. |
| + boolean isSlowed() | Return true if the character is slowed, otherwise false. |
| + boolean isSilenced() | Return true if the character is silenced, otherwise false. |
| + int getDamage() | Return random damage between minAtk and maxAtk |
| + boolean takeDamage(int damage) | Called when the character is damaged. The damage is reduced by defense point. Return true if the character took damage (damage is more than 0), otherwise false. |
| + void resetAttackTick() | Reset attackTick back to 0. |
| + void addAttackTick() | Increase attackTick by 1. If attackTick equals to attackCooldown, reset attackTick and allow the character to attack. |
| + void resetCCedTick() | Reset ccedTick back to 0. |
| + void addCCedTick() | Increase ccedTick by 1. If ccedTick equals to ccedDuration, reset ccedTick and reset CC effect of the character. |
| + void renderNormalAttack(GraphicContext gc) | Render normal attack animation. |
| + *void attack()* | *Throws CannotAttackException* |
| + *void dead()* |  |
| Getters & Setters of every fields, except attackTick and ccedTick |  |