**FRANCISCO JAVIER LOZANO GONZALEZ**

**A00401813**

Damage and Reaction Calculator for the Game Genshin Impact

A calculator is desired to be developed for characters in the game **Genshin Impact** that calculates damage based on the selection of reaction types: transformative damage, amplifying damage, and additive damage. Additionally, it calculates the damage inflicted by a single character or a team (4 characters), according to a specific reaction type and based on the enemy's resistance multiplier. The system will allow storing the values of the last 10 recorded damages and will display the highest damage achieved in the calculator so far.

**Entradas:**

* **@param** double character\_level= character level
* **@param** double character\_attack = character attack
* **@param** double critical\_damage\_probability = probability critical damage
* **@param** double critical\_damage\_percentage = percentage critical damage
* **@param** double elemental\_mastery = elemental mastery
* **@param** string choice = indicating the desired type of multiplier.
* **@param** double reaction\_multiplier = reaction multiplier

**Salidas:**

* **@return** doublebase\_damage = base damage
* **@return** doubletransformative\_damage = it has the transformative damage
* **@return** double amplificative\_damage = it has the amplificative damage
* **@return** double additive\_damage = it has the additive damage
* **@return** doubletotal\_damage = it has the total damage
* **@return** print in screenthe max damage in the moment
* **@return** print in screen last 10 damage saved in the calculator
* **@return** all other texts with instructions for the users and ask for information.

**Methods:**

/\*

\* aleatory\_multiresistance method

\* Description: Calcul a number between 0.5 and 2

\* Precondition: None

\* Postcondition: Does not affect global variables.

\* @param None

\* @return <multiresistance> value type double that represents the multiplier

resistance

\*/

/\*

\* base\_damage\_method {

\* Description: The method is responsible for calculating base damage through

character attack, critical damage probability, and entered percentage of critical damage.

\* Precondition: The method receives 3 variables and returns a result without affecting global variables.

\* Postcondition: Does not affect global variables.

\* @param <character\_attack> A float-type variable whose assignment has already occurred.

\* @param <critical\_damage\_probability> A float-type variable whose assignment has already occurred.

\* @param <critical\_damage\_percentage> A float-type variable whose assignment has already occurred.

\* @return base\_damage A float-type variable, the result of the operation using

the input data character\_attack, critical\_damage\_probability, and

\* critical\_damage\_percentage.

\*/

------------------------------------------------------------------------------------------

/\*

\* Transformative Damage Method {

\* Description: This method calculates the %bonusEM (Elemental Mastery bonus percentage)

based on the previously entered Elemental Mastery. Additionally, with the

character's level, chosen transformative multiplier,

and resistance multiplier, it calculates the transformative damage.

Furthermore, a switch statement is used to obtain the value of the previously.

entered transformative multiplier.

Precondition: The method receives 4 variables and returns a result that

affects the global variable total\_damage.

\* Postcondition: Affects the global variable total\_damage.

\* @param <transform\_multiplier> A float-type variable with an already assigned value.

\* @param <character\_level> An int-type variable with an already assigned value.

\* @param <resistance\_multiplier> A float-type variable with an already assigned value.

\* @param <elemental\_mastery> A float-type variable with an already assigned value.

\* @param <choice> A string-type variable with the response indicating the

desired type of transformative multiplier.

\* @return transformative\_damage A float-type variable, the result of the

operation using the input data transform\_multiplier,

character\_level, resistance\_multiplier, elemental\_mastery, and choice.

\*/

------------------------------------------------------------------------------------------

/\*

\* amplificative\_damage\_method {

\* Description: This method calculates the %bonusME based on the previously

entered elemental mastery.

Additionally, using the reaction multiplier and %bonusME, along with the

previously calculated amplification multiplier plus the base damage,

it computes the amplificative damage.

Furthermore, a switch statement is used to obtain the value of the previously.

entered amplificative multiplier.

\* Precondition: The method receives 3 variables and returns a result that

affects the global variable total\_damage.

\* Postcondition: Affects the global variable total\_damage.

\* @param <amplification\_multiplier> A float-type variable with an already

assigned value.

\* @param <reaction\_multiplier> A float-type variable with an already assigned value.

\* @param <elemental\_mastery> A float-type variable with an already assigned value.

\* @param <choice> A string-type variable with the response indicating the

desired type of amplificative multiplier.

\* @return amplificative\_damage A float-type variable, the result of the

operation using the input data elemental\_mastery, amplification\_multiplier,

and reaction\_multiplier.

\* }

\*/

------------------------------------------------------------------------------------------

/\*

\* additive\_damage\_method {

\* Description: This method calculates the %bonusME based on the previously

entered elemental mastery. Additionally,

using the additive multiplier, %bonusME, character level, and resistance

multiplier, it computes the additive damage.

Furthermore, a switch statement is used to obtain the value of the previously.

entered additive multiplier.

\* Precondition: The method receives 4 variables and returns a result that

affects the global variable total\_damage.

\* Postcondition: Affects the global variable total\_damage.

\* @param <additive\_multiplier> A float-type variable with an already assigned value.

\* @param <resistance\_multiplier> A float-type variable with an already assigned value.

\* @param <character\_level> An int-type variable with an already assigned value.

\* @param <elemental\_mastery> A float-type variable with an already assigned value.

\* @param <choice> A string-type variable with the response indicating the

desired type of additive multiplier

\* @return additive\_damage A float-type variable, the result of the operation

using the input data additive\_multiplier,

resistance\_multiplier, character\_level, and elemental\_mastery.

\*/

------------------------------------------------------------------------------------------

/\*

\* saving\_damage method

\* --------------------------------

\* Description: let the user can see the contents of the list of damage, print

the last 10 damages

\* Preconditions: need a list previously saved

\* Postcondition: Don’t affect global variables

\* @param <list. damages> the list that contains the damage.

\* @return <> print in screen the last 10 damages in the list

\*/

------------------------------------------------------------------------------------------

/\*

\* calc\_highestdamage method

\* Description: Calcul the highest damage value in the list "damages" and print

it in the output

\* Precondition: the list "damages"

\* Postcondition: Don’t affect global variables

\* @param <list.damages> the list that contains the damage

\* @return <> print in screen the highest damage value registered in the list

"damages"

\*/

**Example:**

**--Damage calculator—**

* "What process do you want to perform?

1- Calculate damage

2- View maximum recorded damage

3- View the last 10 recorded data

1. Exit calculator"

**-1**

* "Do you want continue evaluating characters?
* 1.si
* 2.no

**-1**

* "What type of reaction does the character have? (1. Transformative 2. Amplification 3. aditive)”

-**2**

* "Enter character attack value"

**-23**

* "Enter critical damage probability"

**-0.3**

* "Enter percentage amount of critical damage.”

**-0.2**

* "Enter elemental mastery value"

**-30**

* "Enter reaction multiplier value"

-2

* "What type of amplification multiplier do you use? 1. Vaporize 2. To melt"

**-1**

* The final damage is. 109.24
* The total damage is. 109.24
* "What process do you want to perform?

1- Calculate damage

2- View maximum recorded damage

3- View the last 10 recorded data

1. Exit calculator"

**-2**

* 109.24
* "What process do you want to perform?

1- Calculate damage

2- View maximum recorded damage

3- View the last 10 recorded data

1. Exit calculator"

**-3**

* 109.24
* …
* …
* …
* …
* …
* …
* …
* …
* …
* "What process do you want to perform?

1- Calculate damage

2- View maximum recorded damage

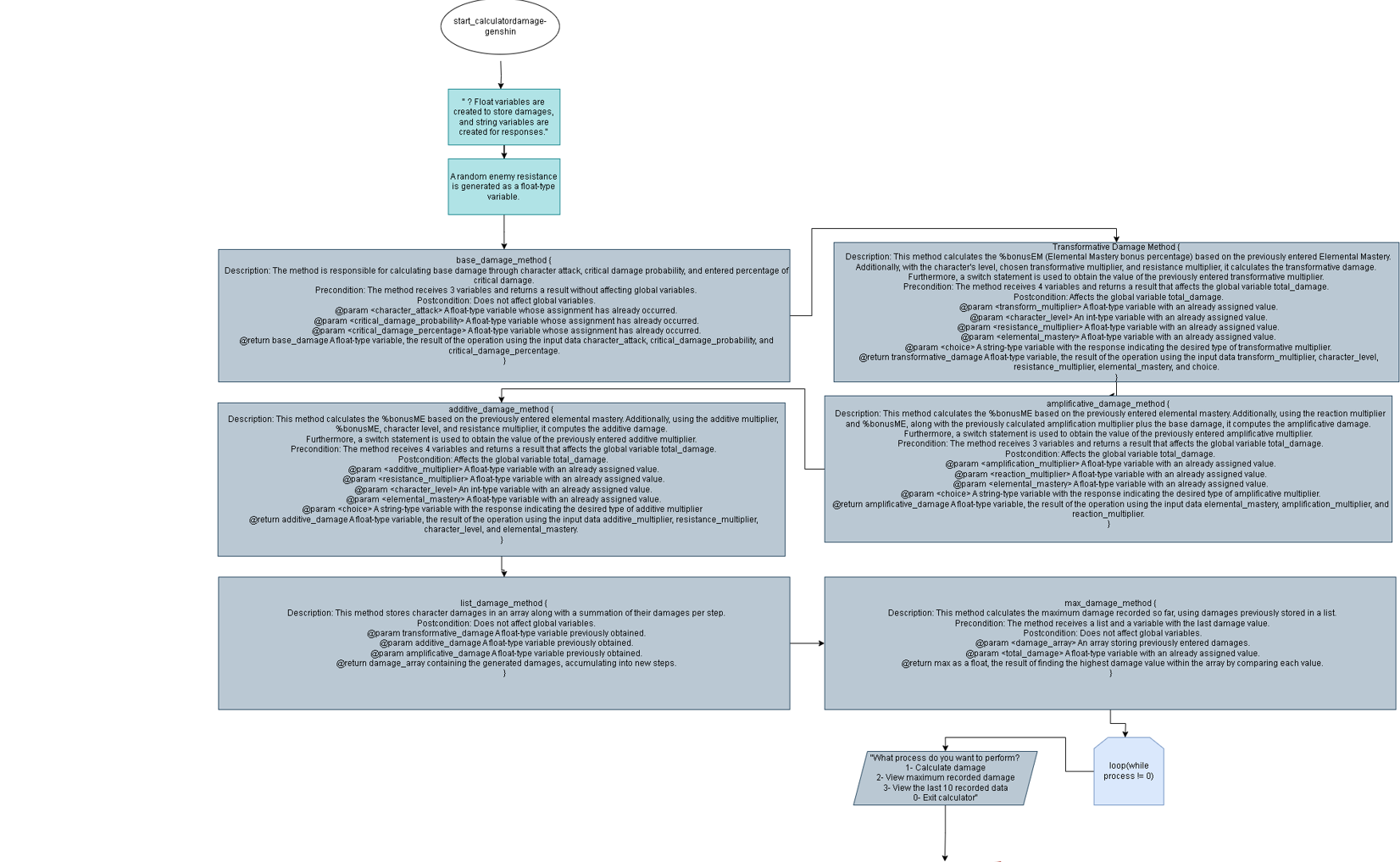
3- View the last 10 recorded data

1. Exit calculator"

**-0**

* "Thank you for using our calculator."

**Diagram:**



A diagram of a flowchart

Description automatically generated

A diagram of a flowchart

Description automatically generatedA diagram of a diagram

Description automatically generated with medium confidenceA diagram of a diagram

Description automatically generated