## GameTime

### CRC

|  |  |
| --- | --- |
| GameTime | |
| Responsibilities: The purpose of the GameTime class is to keep track of the time while a game is currently in progress. | Collaborators: GameTime collaborates with Game. |

### Description / Overview

The GameTime class is a part of the Model in our architecture and is a class that simply keeps track of all aspects of time for a game. GameTime controls how long a robot takes to complete a turn, as well as how long a human player has to complete a turn. GameTime is able to change the think time for a robot, pause the timer, reset the timer, pause the game, and keep track of the overall time for the game. It can also reset the game timer at the start of a new game.

### Instance Variables

#### robotThinkTime

Data Type:Integer

robotThinkTime is an integer that determines how long a robot takes to act.

#### playTimer

Data Type:Timer

playTimer is a timer that keeps track of time as long as a game is being played and is used for any case within the game that uses time.

#### startGameInMillis

Data Type:Long

startGameInMillis gets the current time when the game starts in milliseconds.

### Method Overview

[*public GameTime GameTime(int think)*](#_Void_method1(String_arg0,)

[*public void setThinkTime(int i)*](#_Void_method2(String_arg0))

[*public void resetPlayTimer()*](#_Void_method3(String_arg0,)

[*public void toggleTimer()*](#_Void_method2(String_arg0))

[*public bool isPaused()*](#_Void_method3(String_arg0,)

[*public int getThinkTime()*](#_Void_method4())

[*private void pauseTime()*](#_Void_method4())

[*private void resumeTime()*](#_Void_method1(String_arg0,)

### Method Writeups

*public GameTime GameTime(int think)*

Constructor that allocates a robotThinkTime, creates a playTimer of type Timer, and allocates startGameInMillis to the current time in type long.

#### public void setThinkTime(int i)

This method sets the think time for the robot to some integer I.

*public void resetPlayTimer()*

This method resets the play timer back to 0.

*[public void toggleTimer()](#_Void_method2(String_arg0))*

This method switches toggles the timer being paused or running. If the timer is paused it will resume and if it is running it will pause.

*public bool isPaused()*

This method checks whether the timer is paused or running.

*public int getThinkTime()*

This method returns the current think time for the robot.

*private void pauseTime()*

This method pauses the timer, allowing it to be resumed at the same number at any time.

[*private void resumeTime()*](#_Void_method2(String_arg0))

This method resumes the timer from the same time it was paused.