Klasy i struktury danych w projekcie Sudoku

Kamil Kowalski
Politechnika Śląska
wydział Automatyki, Elektroniki i Informatyki
Semestr 2 Sekcja 12

2021-06-13

0.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Rozdział 1

Opis programu

1.1 Ogólny opis

Celem programu jest umożliwienie rozgrywki sudoku w różnych trybach i z dowolnym interace użytkownika. W programie polmorfizm zostanie zastosowany jako warstwa abstrakcji pomiędzy klasami pochodnymi klasy Game i klasami pochodnymi UserInterface, co pozwoli wybierać tryb gry i jej interface niezależnie od siebie.
Algorytm generujący sudoku zawiera się klasie SudokuGenerator i jest oparty na algorytmi DLX, operującym na cyklicznej liście dwukierunkowej (LinkedNode), obecna implementacja pozwala na generacje wypełnionego sudoku 25x25 w zadawalającym czasie $< 1s$
Metoda generate() klasy SudokuGenerator zwraca w pełni wypełnione sudoku, w celu usunięcia części pól, należy wywołać funkcje deleteFromBoard() Funkcja ta będzie stopniowo usuwała komórki z sudoku i sprawdzała czy możliwe jest rozwiązanie bez zgadywania, do momentu aż sudoku osiągnie zadany poziom trudności
Interface użytkownika i tryb gry zostaną odczytane z argumentów wejściowych programu, w zależności od wybranej kombinacji zostaną zainstancjonowane odpowiednie klasy pochodne
Komunikacja między interface użytkownika a grą będzie odbywała sią za pomocą 2 kolejek fifo zawierających odpowiednio pochodne klasy Event lub obiekty std::string zawierające wiadomości zwrotne.

1.2 Przykłady

Program mógłby zostać uruchomiony z lini poleceń w następujący sposób:

```
Sudoku limitedTime cli 3
```

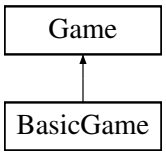
Oznaczało by to że sudoku zostanie uruchomione w konsoli, w trybie z ograniczonym czasem, a wygenerowane sudoku będzie rozmiaru 9x9 (3^2, długość boku poprawnego sudoku jest kwadratem)

Rozdział 2

Class Documentation

2.1 BasicGame Class Reference

Inheritance diagram for BasicGame:



Public Member Functions

- **BasicGame** (UserInterface &interface, LinkedList< std::unique_ptr< Event >> &queue)
- void **init** ()
- void applyMove (Move &move)
- void retractMove ()
- void askForHint (Coordinates coords)
- void **changeState** (Game::GameState &newState)
- void **gameLoop** ()
- void **updateLeaderboard** (Leaderboard &leaderboard)

Additional Inherited Members

2.1.1 Member Function Documentation

2.1.1.1 applyMove()

```
void BasicGame::applyMove (
            Move & move )  [virtual]
```

Applies move to sudoku

Implements Game.

2.1.1.2 askForHint()

```
void BasicGame::askForHint (
            Coordinates coords )  [virtual]
```

Reveals one hidden cell

Implements Game.

2.1.1.3 retractMove()

```
void BasicGame::retractMove ( )  [virtual]
```

Undoes a last move

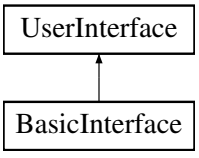
Implements Game.

The documentation for this class was generated from the following files:

- GameModes/BasicGame.h
- GameModes/BasicGame.cpp

2.2 BasicInterface Class Reference

Inheritance diagram for BasicInterface:



Public Member Functions

- **BasicInterface** (LinkedList< std::unique_ptr< Event >> &eventQueue, LinkedList< std::string > &messageQueue)
- void initiate ()
- void **render** (Sudoku &sudoku)
- void **render** (Timer &timer)
- void **render** (Hint &hint)
- void **render** (MistakeCounter &mistakeCounter)
- void **render** (CountdownTimer &countdownTimer)
- void **message** (const std::string &msg)
- void display ()
- void input ()

Additional Inherited Members

2.2.1 Member Function Documentation

2.2.1.1 display()

```
void BasicInterface::display ( )  [virtual]
```

Displays, or in case of CLI flushes, rendered elements. Meaningful mainly in GUI applications

Implements UserInterface.

2.2.1.2 initiate()

```
void BasicInterface::initiate ( )  [virtual]
```

invoked before rendering of UI

Implements UserInterface.

2.2.1.3 input()

```
void BasicInterface::input ( ) [virtual]
```

Gathers input from user and passes it to _eventQueue of appropriate type

Implements UserInterface.

The documentation for this class was generated from the following files:

- UserInterfaces/BasicInterface.h
- UserInterfaces/BasicInterface.cpp

2.3 Coordinates Class Reference

```
#include <Sudoku.h>
```

Public Member Functions

- **Coordinates** (uint8_t row=0, uint8_t column=0)

Public Attributes

- uint8_t **_row**
- uint8_t **_column**

2.3.1 Detailed Description

Class representing Sudoku coordinates

The documentation for this class was generated from the following file:

- Sudoku/Sudoku.h

2.4 CountdownTimer Class Reference

```
#include <Misc.h>
```

Public Member Functions

- **CountdownTimer** (unsigned int timeLimit)
- void **start** ()
- unsigned int **limitAsSeconds** ()
- unsigned int **asSeconds** ()

2.4.1 Detailed Description

Minor class representing countdown timer

The documentation for this class was generated from the following files:

- GameModes/Misc.h
- GameModes/Misc.cpp

2.5 Event Class Reference

`#include <Event.h>`

Public Member Functions

- virtual void run (Game &game)=0

2.5.1 Detailed Description

Base class , instances are meant to have varying constructors and act as lambda captures

2.5.2 Member Function Documentation

2.5.2.1 run()

```
virtual void Event::run (
            Game & game )  [pure virtual]
```

Only virtual method, executes class specyfic action on Game object

The documentation for this class was generated from the following file:

- Event/Event.h

2.6 EventQueue Class Reference

Public Member Functions

- void **push_back** (std::unique_ptr< Event > &&event)
- void **pop** (Game &game)
- void **run** (Game &game)
- void **clear** ()
- unsigned int **size** ()

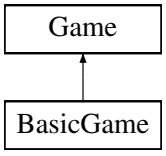
The documentation for this class was generated from the following files:

- Event/EventQueue.h
- Event/EventQueue.cpp

2.7 Game Class Reference

```
#include <Game.h>
```

Inheritance diagram for Game:



Public Types

- enum GameState {
  Loading , **Play** , **Pause** , **TimeOut** ,
  FinalMistake , **GameOver** }

Public Member Functions

- template<typename T >
  Game (UserInterface &interface, LinkedList< T > &eventQueue)
- virtual void applyMove (Move &move)=0
- virtual void retractMove ()=0
- virtual void askForHint (Coordinates coords)=0
- virtual void **init** ()=0
- virtual void **gameLoop** ()=0

Protected Attributes

- std::shared_ptr< UserInterface > _interface
- GameState **_state**
- Sudoku _baseSudoku
- Sudoku _sudoku
- Sudoku _filledSudoku
- LinkedList< Move > _moves

2.7.1 Detailed Description

Abstract class representing game mode It's responsible for modyfing sudoku board and controling game state

2.7.2 Member Enumeration Documentation

2.7.2.1 GameState

`enum Game::GameState`

Enum representing state of game

2.7.3 Member Function Documentation

2.7.3.1 applyMove()

```
virtual void Game::applyMove (
            Move & move )  [pure virtual]
```

Applies move to sudoku

Implemented in BasicGame.

2.7.3.2 askForHint()

```
virtual void Game::askForHint (
            Coordinates coords )  [pure virtual]
```

Reveals one hidden cell

Implemented in BasicGame.

2.7.3.3 retractMove()

```
virtual void Game::retractMove ( )  [pure virtual]
```

Undoes a last move

Implemented in BasicGame.

2.7.4 Member Data Documentation

2.7.4.1 _baseSudoku

Sudoku Game::_baseSudoku [protected]

Generated sudoku, with gaps

2.7.4.2 _filledSudoku

Sudoku Game::_filledSudoku [protected]

Generated filled sudoku

2.7.4.3 _interface

std::shared_ptr<UserInterface> Game::_interface [protected]

Pointer to interface used by game mode

2.7.4.4 _moves

LinkedList<Move> Game::_moves [protected]

List of moves

2.7.4.5 _sudoku

Sudoku Game::_sudoku [protected]

Sudoku in progress

The documentation for this class was generated from the following file:

- GameModes/Game.h

2.8 Hint Class Reference

#include <Misc.h>

Public Member Functions

- **Hint** (unsigned int hintCount)
- bool **uncover** (Coordinates pos, Sudoku &filledSudoku, Sudoku &sudoku)
- unsigned int **getHintCount** ()
- unsigned int **getMaxHintCount** ()

2.8.1 Detailed Description

Minor class representing hints mechanism

The documentation for this class was generated from the following files:

- GameModes/Misc.h
- GameModes/Misc.cpp

2.9 LinkedList< T > Class Template Reference

Public Member Functions

- template<typename... V>
  LinkedList (const V &...v)
- template<typename... V>
  void **push_back** (T value, const V &...v)
- template<typename... V>
  void **push_front** (T value, const V &...v)
- T & **getElement** (unsigned int i)
- void **pop_front** ()
- void **pop_back** ()
- LinkedNode< T > * **getRoot** ()
- template<typename... V, typename F >
  void **iterate** (const F &function, V &...v)
- template<typename... V, typename F >
  void **reverseIterate** (const F &function, V &...v)
- uint32_t **count** ()

The documentation for this class was generated from the following file:

- LinkedList/LinkedList.h

2.10 LinkedNode< T > Class Template Reference

Public Member Functions

- **LinkedNode** (T value)
- **operator T&** ()
- void **setValue** (T value)
- void **insertBefore** (LinkedNode< T > *node)
- void **insertBefore** (T value)
- void **insertAfter** (LinkedNode< T > *node)
- void **insertAfter** (T value)
- void **popIn** ()
- void **popOut** ()
- LinkedNode< T > * **erase** ()
- bool **insideChain** ()
- bool **selfReference** ()
- LinkedNode< T > * **next** (unsigned int distance)
- LinkedNode< T > * **prev** (unsigned int distance)
- LinkedNode< T > * **next** ()
- LinkedNode< T > * **prev** ()
- template<typename F >
  LinkedNode< T > * **select** (const F &f)
- template<typename F , typename... V>
  void **iterateForward** (const F &f, V &...v)
- template<typename F , typename... V>
  void **iterateBackward** (const F &f, V &...v)
- uint32_t **count** ()

Static Public Member Functions

- static void **advance** (LinkedNode< T > *&ptr, uint32_t dist)
- static void **advance** (LinkedNode< T > *&ptr)
- static void **recede** (LinkedNode< T > *&ptr, uint32_t dist)
- static void **recede** (LinkedNode< T > *&ptr)

Public Attributes

- T **_value**

The documentation for this class was generated from the following file:

- LinkedNode/LinkedNode.h

2.11 MistakeCounter Class Reference

`#include <Misc.h>`

Public Member Functions

- **MistakeCounter** (unsigned int tolerance)
- unsigned int **getMistakes** ()
- unsigned int **getTolerance** ()
- void **reset** ()
- void **increment** ()
- bool **gameOver** ()

2.11.1 Detailed Description

Minor class representing mistake counter

The documentation for this class was generated from the following files:

- GameModes/Misc.h
- GameModes/Misc.cpp

2.12 Move Class Reference

`#include <Move.h>`

Public Member Functions

- **Move** (Coordinates pos, uint8_t number)
- void **apply** (uint8_t **board)
- void **retract** (uint8_t **board) const

Public Attributes

- Coordinates **_pos**
- uint8_t **_number**

2.12.1 Detailed Description

Class representing single move by player, meant to be able to retrace itself

The documentation for this class was generated from the following files:

- Move/Move.h
- Move/Move.cpp

2.13 Sudoku Class Reference

`#include <Sudoku.h>`

Public Member Functions

- **Sudoku** (uint8_t rootSize, uint8_t **board)
- **Sudoku** (const Sudoku &sudoku)
- **Sudoku** (Sudoku &&sudoku)
- Sudoku & **operator=** (const Sudoku &sudoku)
- Sudoku & **operator=** (Sudoku &&sudoku)
- bool isComplete ()
- void applyMove (Move &move)
- bool applyMoveConditionally (Move &move)
- void retractMove (const Move &move)
- uint16_t **getSize** ()
- uint16_t **getRootSize** ()
- uint8_t * **operator[]** (uint16_t row)
- uint8_t & **operator[]** (const Coordinates &coords)

Friends

- std::ostream & **operator<<** (std::ostream &stream, const Sudoku &sudoku)

2.13.1 Detailed Description

Class responsible for holding, and manipulating sudoku board

2.13.2 Member Function Documentation

2.13.2.1 applyMove()

```
void Sudoku::applyMove (
            Move & move )
```

Applies move with no regard for correctness

2.13.2.2 applyMoveConditionally()

```
bool Sudoku::applyMoveConditionally (
            Move & move )
```

Applies move if valid

2.13.2.3 isComplete()

```
bool Sudoku::isComplete ( )
```

Check if sudoku is filled

2.13.2.4 retractMove()

```
void Sudoku::retractMove (
            const Move & move )
```

Undoes a move

The documentation for this class was generated from the following files:

- Sudoku/Sudoku.h
- Sudoku/Sudoku.cpp

2.14 SudokuGenerator Class Reference

```
#include <SudokuGenerator.h>
```

Public Member Functions

- **SudokuGenerator** (uint16_t rootSize)
- Sudoku **generate** ()

Friends

- uint32_t **count** (LinkedNode< SudokuGenerator::SudokuNode * > *node)

2.14.1 Detailed Description

Class responsible for sudoku generation

The documentation for this class was generated from the following files:

- SudokuGenerator/SudokuGenerator.h
- SudokuGenerator/SudokuGenerator.cpp

2.15 Test Class Reference

The documentation for this class was generated from the following file:

- test.h

2.16 Timer Class Reference

`#include <Misc.h>`

Public Member Functions

- void **start** ()
- unsigned int **asSeconds** ()

2.16.1 Detailed Description

Minor class representing timer

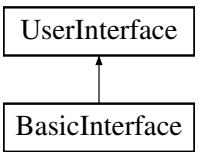
The documentation for this class was generated from the following files:

- GameModes/Misc.h
- GameModes/Misc.cpp

2.17 UserInterface Class Reference

`#include <UserInterface.h>`

Inheritance diagram for UserInterface:



Public Member Functions

- template<typename T >
  UserInterface (Game &game, LinkedList< T > &eventQueue, LinkedList< std::string > &messageQueue)
- virtual void initiate ()=0
- virtual void render ()=0
- virtual void display ()=0
- virtual void input ()=0

Protected Attributes

- LinkedList< std::string > _messageQueue
- Game & _game

2.17.1 Detailed Description

Base interface to be used by all derived classes implementing UserInterface, graphical or in console

2.17.2 Member Function Documentation

2.17.2.1 display()

```
virtual void UserInterface::display ( )  [pure virtual]
```

Displays, or in case of CLI flushes, rendered elements. Meaningful mainly in GUI applications

Implemented in BasicInterface.

2.17.2.2 initiate()

```
virtual void UserInterface::initiate ( )  [pure virtual]
```

invoked before rendering of UI

Implemented in BasicInterface.

2.17.2.3 input()

```
virtual void UserInterface::input ( )  [pure virtual]
```

Gathers input from user and passes it to _eventQueue of appropriate type

Implemented in BasicInterface.

2.17.2.4 render()

```
virtual void UserInterface::render ( )  [pure virtual]
```

Renders, or in case of CLI displays, UI elements. Game rendered is specified to have select few base components, such as: SudokuBoard, Timer or MistakeCounter and others that will be drawn by UI instance

2.17.3 Member Data Documentation

2.17.3.1 _game

```
Game& UserInterface::_game  [protected]
```

Reference to instance of class deriving from abstract class Game

2.17.3.2 _messageQueue

```
LinkedList<std::string> UserInterface::_messageQueue  [protected]
```

Queue for passing messages between Game and UserInterface derived instances

The documentation for this class was generated from the following file:

- UserInterfaces/UserInterface.h

Skorowidz