Додаток А Технічне завдання

КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. І. Сікорського

Кафедра

автоматизованих систем обробки інформації та управління

Затвердив

Керівник Головченко М.М.

«02» березня 2021 р.

Виконавець:

Студент Василенко П.О.

«02» березня 2021 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання курсової роботи

на тему: «Скарбнички»

з дисципліни:

«Основи програмування-2. Модульне програмування»

Київ 2021

* 1. *Мета*: Метою курсової роботи є розробка програмного забезпечення, що передбачає можливість гри в «Скарбнички» проти комп’ютера.
  2. *Дата початку роботи*: «02»березня 2021 р.
  3. *Дата закінчення роботи*: «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 202\_ р.
  4. *Вимоги до програмного забезпечення*.

1. Функціональні вимоги:

* Можливість грати з комп’ютерним опонентом
* Можливість випадкової генерації колоди карт
* Можливість дотримування правил гри:
* 1. Можливість вгадати карти суперника, й забрати їх собі
* 2.Можливість вгадати лише частину набору карт суперника
* 3.Можливість не вгадати жодної з карт суперника
* 4.Можливість збирати 4 однакові за значенням карти в «Скарбницю»
* 5.Можливість перемоги/поразки відповідно до кількості скарбничок гравців

1. Нефункціональні вимоги:

* Використання стандарту ES6 для Javascript
* Використання системи контролю версій Git
* Наявність зрозумілого користувачеві інтерфейсу
* Кросбраузерність (Google Chrome, Opera, Mozilla Firefox)
* Все програмне забезпечення та супроводжуюча технічна документація повинні задовольняти наступним ДЕСТам:

ГОСТ 29.401 - 78 - Текст програми. Вимоги до змісту та оформлення.

ГОСТ 19.106 - 78 - Вимоги до програмної документації.

ГОСТ 7.1 - 84 та ДСТУ 3008 - 95 - Розробка технічної документації.

* 1. *Стадії та етапи розробки*:

1. Об'єктно-орієнтований аналіз предметної області задачі (до\_\_.\_\_.202\_ р.)
2. Об'єктно-орієнтоване проектування архітектури програмної системи (до \_\_.\_\_.202\_р.)
3. Розробка програмного забезпечення (до \_\_.\_\_.202\_р.)
4. Тестування розробленої програми (до \_\_.\_\_.202\_р.)
5. Розробка пояснювальної записки (до \_\_.\_\_.202\_ р.).
6. Захист курсової роботи (до \_\_.\_\_.202\_ р.).
   1. *Порядок контролю та приймання*. Поточні результати роботи над КР регулярно демонструються викладачу. Своєчасність виконання основних етапів графіку підготовки роботи впливає на оцінку за КР відповідно до критеріїв оцінювання.
7. **Постановка задачі**

Розробити програмне забезпечення, що дає змогу користувачу грати в карткову гру «Скарбнички» проти комп’ютерного суперника.

Відкривши чи оновивши сторінку веб-застосунку перед користувачем з’явиться модальне вікно з кнопкою початку гри. Натиснувши на цю кнопку розпочнеться гра за відповідними правилами описаними в розділі «Теоретичні відомості». У випадку спроби порушення правил програма не дозволить цього зробити. Інтерфейс користувача включає в себе можливість слідкувати за кількістю карт в колоді, в руці суперника, кількістю скарбничок обох гравців, та картами в своїй руці. По закінченню гри має виводитися вікно з оголошенням перемоги/поразки гравця з кнопкою «Спробувати ще».

1. **Теоретичні відомості**

***Правила гри «Скарбнички»:***

Кількість колод: 1

Кількість карт в колоді: 36

Кількість гравців: 2

Старшинство карт: 6,7,8,9,10,J,Q,K,A

Ціль гри: зібрати якнайбільше скарбничок, до того моменту, як карти в колоді й у обох гравців закінчаться

Хід гри: Колода ретельно тасується. Кожен гравець отримує по 4 карти, а колода що залишилася кладеться на центр столу сорочкою догори. Перших ходить користувач – надалі по черзі. Спочатку гравець питає у суперника про наявність карти/карт таким чином: значення карти, кількість карт та масті. Гравець може спитати лише карту такого значення, яке є в нього в руці. Якщо у суперника наявні всі запитані карти – він віддає їх гравцю. Якщо у суперника не виявилося жодної з запитаних карт – гравець бере верхню карту з колоди. Якщо у суперника є лише частина з загаданих карт, то він віддає їх гравцю, а гравець бере карту з колоди. Якщо у гравця чи комп’ютера в руках виявиться 4 однакові за значенням карти – вони збираються в скарбничку й відповідний гравець прибирає їх зі своєї руки. Якщо після збору скарбнички у гравця виявилося менше 4-х карт – він бере з колоди рівно стільки карт, скільки йому не вистачає. Гру вважають закінченою, коли колода карт – пуста а у гравців нема карт в руках. Після цього рахуються скарбнички кожного та визначається переможець.

1. **Опис алгоритмів**

Перелік всіх основних змінних та їхнє призначення наведено в таблиці 3.1.

Таблиця 3.1 – Основні змінні та їхні призначення

|  |  |
| --- | --- |
| Змінна | Призначення |
| enemyHand | Рука комп’ютера – масив карт, які комп’ютер може використовувати |
| deckSize | Кількість карт, що залишилися в колоді (від 0 до 28) |
| imagineHand | Уявна рука гравця – комп’ютер запам’ятовує всі карти, що гравець в нього забирав, угадуючи, а також всі карти, що гравець питав, але не вгадував |
| lastUsed | Масив карт, що комп’ютер використовував нещодавно, тому немає сенсу використовувати ще раз. Максимальна к-сть карт – 3. |
| value | Значення карти – 6 7 8 9 10 J Q K A |
| suit | Масть карти ♣ ♠ ♦ ♥ |

* 1. Загальний алгоритм під час ходу гравця

1.ПОЧАТОК(отримання карт)

2.ПОКИ deckSize>0 && enemyHand.size>0;

Гравець запитує value карти,

**Якщо** гравець вгадав

Гравець запитує кількість таких карт,

**Якщо** гравець вгадав

Гравець вгадує масті карт

**Якщо** гравець угадав усі карти

Передати вгадані карти гравцеві

**Якщо** гравець не зібрав скарбничку

Записати віддані карти до imagineHand і додати value+”?” до imagineHand

**Інакше**

Нічого

**Все якщо**

Перейти до алгоритму 1.2

**Інакше якщо** гравець угадав частину карт

Передати вгадані карти гравцеві,

Записати віддані карти до imagineHand і додати value+”?”+масті, що гравець назвав та не вгадав до imagineHand  
Перейти до алгоритму 1.2

**Інакше якщо** гравець не вгадав жодної карти

Записати value+?+suit (всі масті, названі гравцем)

Перейти до алгоритму 1.2

**Все якщо**

**Інакше**

Записати value+”?” до ImagineHand

Перейти до алгоритму 1.2

**Все якщо**

**Інакше**

Записати value+”?” до ImagineHand

Перейти до алгоритму 1.2

**Все якщо**

КІНЕЦЬ ПОКИ.

1.2. Загальний алгоритм ходу комп’ютерного опонента

1.3. Загальний алгоритм вибору ходу

priority– 0,

bestCard– не обрана.

Для кожної card в руці,

**Якщо** копій цієї карти в руці 1:

**Якщо** ця карта є в lastUsed:

**Якщо** кількість точних мастей в ImagineHand 0:

Пріоритет 0

**Інакше якщо** кількість точних мастей в ImagineHand 1:

Пріоритет 0

**Інакше якщо** кількість точних мастей в ImagineHand 2:

**Пріоритет 2500. Якщо** він перевищує значення в priority:

Priority = 1000, bestCard = card

**Інакше якщо**кількість точних мастей в ImagineHand 3:

**Пріоритет 10000**

**Все якщо**

**Інакше:**

**Якщо** в ImagineHand 0 копій цієї карти:

Пріоритет 1000. **Якщо** він перевищує значення в priority:

Priority = 1000, bestCard = card

**Інакше якщо** в ImagineHand 1 копія цієї карти:

**Якщо** це точно відома масть:

Пріоритет 1800. **Якщо** він перевищує значення в priority:

Priority = 1800, bestCard = card

**Інакше**

Пріоритет 1100. **Якщо** він перевищує значення в priority:

Priority = 1100, bestCard = card

**Все якщо**

**Інакше якщо** в ImagineHand 2 копії цієї карти:

Якщо кількість точних мастей в ImagineHand 0:

Пріоритет 2100. **Якщо** він перевищує значення в priority:

Priority = 2100, bestCard = card

Якщо кількість точних мастей в ImagineHand 1:

Пріоритет 2500. **Якщо** він перевищує значення в priority:

Priority = 2500, bestCard = card

Якщо кількість точних мастей в ImagineHand 2:

Пріоритет 2800. **Якщо** він перевищує значення в priority:

Priority = 2800, bestCard = card

**Інакше якщо** в ImagineHand 3 копії цієї карти:

Пріоритет 10000.**Якщо** він перевищує значення в priority:

Priority = 2800, bestCard = card

**Все якщо**

**Все якщо**

**Інакше якщо** копій цієї карти в руці 2:

**Інакше якщо** копій цієї карти в руці 3:

**Інакше якщо** копій цієї карти в руці 4:

Зібрати в сундучок

**Все якщо**

Повернути найбільший пріоритет та обрану для ходу карту