CS 242 Final Project Proposal

Classical Game Collection

Timeline:

1.1. Week 1

- Implement back-end logic for the board
- Implement back-end logic for the snake
- Implement back-end logic for movement
- Implement static GUI

1.2. Week 2

- Complete back-end logic for the game
- Implement game-loop
- Implement dynamic GUI for movement
- Implement dynamic GUI for eating apple and end of game
- let user pause/resume the game
- Implement user-score interaction
- Implement forfeit/restart menu
- implement console status helper
- refractor code

1.3. Week 3

- polish and make a vivid UI(grass, head,body,stone,fruit)
- Implement gaming environment changing option
- Implement a dynamic realtime score board containing speed info , user nickname and scores
- Implement the second snake; different skin color between 2 snakes; identify two different snakes and operate them differently in the backend logic

Bonus extra features:

- concurrent game loop: two player are able to operate the games and be able to attack each other using keyboard
- new game end logic, have pop up dialogue to tell users which snake died by its special color(using HashMap)

1.4. Week 4

- implement game collection's main menu page and navigation schema
- add AI component for the snake game
- add sudoku game backend
- add sudoku game dynamic GUI
- integrate multiple games into one application

Rubric:

Week1:

Category	Weight	Scoring	Notes
Basic Preparation	2	0-1	
Cleverness	2	0-2	
Rubric	2	0-1	
Code Submission	4	0-2	
Decomposition	4	0-2	
Documentation	4	0-2	
Effort	2	0-2	
Naming	2	0-2	
Overall Design	5	0-2	
Participation	5	0-2	
Presentation	4	0-2	
Requirements 1 Requirements 2 Requirements 3 Requirements 4 Testing	5 5 5 5 5	0-2	Implement backend logic for the board Implement backend logic for the snake Implement backend logic for move and gameloop Implement static GUI backend unit test,Data Structures,Logic
Schedule	2	0-1	Revised final project schedule if necessary
Total	**58**		

Week2:

Category	Weight	Scoring	Notes
Basic Preparation	2	0-1	
Cleverness	2	0-2	
Rubric	2	0-1	
Code Submission	4	0-2	
Decomposition	4	0-2	
Documentation	4	0-2	
Effort	2	0-2	
Naming	2	0-2	
Overall Design	5	0-2	
Participation	5	0-2	
Presentation	4	0-2	
Requirements 1 Requirements 2 Requirements 3 Requirements 4 Testing	5 5 5 5 5	0-2	Implement dynamic UI let user pause/resume the game Implement user-score interaction Implement forfeit/restart menu implement console status helper refractor code backend unit test, manual test plan
Schedule	2	0-1	Revised final project schedule if necessary
Total	**58**		

Week3:

Category	Weight	Scoring	Notes
Basic Preparation	2	0-1	
Cleverness	2	0-2	
Rubric	2	0-1	
Code Submission	4	0-2	
Decomposition	4	0-2	
Documentation	4	0-2	
Effort	2	0-2	
Naming	2	0-2	
Overall Design	5	0-2	
Participation	5	0-2	
Presentation	4	0-2	
Requirements 1 Requirements 2 Requirements 3 Requirements 4 Testing	5 5 5 5 5	0-2	● polish and make a vivid UI(grass, head,body,stone,fruit) ● Implement gaming environment changing option ● Implement dynamic realtime core board containing speed info , user nickname and scores ● Implement the second snake; different skin color between 2 snakes; identify two different snakes and operate them differently in the backend logic Bonus extra features: ● concurrent game loop: two player are able to operate the games and be able to attack each other ● new game end logic, have pop up dialogue to tell users which snake died by its special color(using HashMap) backend unit test, manual test plan

Schedule	2	0-1	Revised final project schedule if necessary
Total	**58**		

Week1:

Category	**Weight**	**Scoring**	**Notes**
Basic Preparation	2	0-1	
Cleverness	2	0-2	
Rubric	2	0-1	
Code Submission	4	0-2	
Decomposition	4	0-2	
Documentation	4	0-2	
Effort	2	0-2	
Naming	2	0-2	
Overall Design	5	0-2	
Participation	5	0-2	
Presentation	4	0-2	
Requirements 1 Requirements 2 Requirements 3 Requirements 4 Testing	5 5 5 5 5	0-2	● add AI component for the snake game ● implement game collection's main menu page and navigation schema ● add sudoku game backend ● add sudoku game dynamic GUI ● integrate multiple games into one application backend unit test,Data Structures,Logic
Schedule	2	0-1	Revised final project schedule if necessary
Total	**58**		

Classical Game Collection

1. Abstract

1.1. Project Purpose

What is the purpose of this project?

Build a classical game collection app for---

Nostalgic people: who likes to play old past games.

Patients: use this app to train their cognitive consistency and psychological balance

Children: for gaming enlightenment

All age ranges: use this app to help them kill boring times.

Preserve the past history for future generations.

What problems do you want to solve?

1. Learn to implement different game logics
2. Concise and elegant GUI Design.
3. Generate different level of game hardness
4. More advanced GUI interaction

1.2. Background/Motivation

Why are you interested in doing this project? Have you worked on something similar before?

Brief: collaboration plays a significant role in industries. Thus, we want to grasp this last chance and practice our coordination skills before stepping into the company.

We all love playing games. As object oriented design and programming is extremely important in industry, after having some experience with Java GUI development, we want to implement more games and strengthen our OOD skills.

2. Technical Specifications

2.1. Platform: PC

2.2. Programming Languages: Java

2.3. Stylistic Conventions: concise comments, naming conventions, camelCase, maintainable

2.4. SDK: Java

2.5. IDE: Eclipse, IntelliJ

2.6. Tools/Interfaces: Google Chrome

2.7. Target Audience: Nostalgic people, Patients, Children, all ages

2.8. Members: xzhu42, hliu82 , txu25

3. Functional Specifications

3.1. Features

Several bullet points of what kind of functionality your project will feature. What should the user be able to do?

- Play different classical Games
- Have fun!

3.2. Scope of project

What are some of the limitations of this product?

Currently this is a free app. We hope to link it with some financial platform so that players can purchase VIP skins.

4. Future Enhancements

What are some cool tweaks you'd want to make to your product after the core functionality is done? Are you planning to work on it in the future?

1. Added more games
2. Likely