Team: Camilla Lambrocco, Edwin Coy III, William Dell'Anno

Title: Strategy Game Arcade

Project Summary: Our program will have several strategy style arcade games including chess, checkers, tic-tac-toe, and connect four. One and two player gameplay will be possible and an AI built to defend against a loss (vice work for a win) will be implemented. A GUI will be built which the user can interact with using a mouse.

Project Requirements:

Business Requirements		
ID	**Description**	**Priority**
This project has no business requirements.		

User Requirements		
ID	**Description**	**Priority**
UR-01	User can select a game to play.	Critical
UR-02	User can use a mouse to make selections.	Critical
UR-03	User can view the score.	High
UR-04	User can select a game mode.	High
UR-05	User can access the arcade menu from any game.	High
UR-06	User can quit a game.	High
UR-07	User can select a one or two player mode.	High
UR-08	User can reset the score for a set of games.	Medium
UR-09	User can restart a game.	Medium
UR-10	User can view high scores.	Low
UR-11	User can enter their name.	Low
UR-12	User can select the difficult level of the AI.	Low

Functional Requirements		
ID	**Description**	**Priority**
FR-01	Application AI will compete against single player.	High
FR-02	AI will select game moves defensively.	High
FR-03	Game pieces are distinguishable between players.	High
FR-04	Game will alternate between player turns.	High
FR-05	AI will select game moves offensively.	Medium
FR-06	Game moves and actions will trigger audio effects.	Low
FR-07	High difficulty AI will make offensive and defensive moves.	Low
FR-08	Low difficulty AI will make defensive moves only.	Low

Non-Functional Requirements		
ID	**Description**	**Priority**
NFR-01	Invalid player selections will not crash the application.	Critical
NFR-02	Games have real time response.	High

Users and Tasks:
Our application will have two types of users: a single player user and a two player user. Each user will need to accomplish similar tasks during game play (i.e., executing a move and monitoring the other player's move). However, a single player user must make a mode and game selection at the menu instance of the application. A two player user may or may not have to make this same selection.

Use Case ID:	UC-01
Use Case Name:	Select player mode
Description:	The user can select either single player mode or two player mode.

Actors:	User
Pre-condition:	The user launched the application.
Post-condition:	The user is either in single player mode or in two player mode.
Frequency of use:	Each instance of the application.
Flow of events:	<table><tr><th>#</th><th>Actor action</th><th>System response</th></tr><tr><td>1</td><td>Select mode</td><td>Set up platform response depending on selection made.</td></tr></table>
Variations:	None.
Notes and issues:	
Developer notes:	

Use Case ID:	UC-02
Use Case Name:	Single Player Mode
Description:	The user can play any game in the arcade in single player mode (against the AI).

Actors:	User
Pre-condition:	The user selected single player mode.
Post-condition:	The user can play any game against the AI.
Frequency of use:	Each instance the user selects single player mode and selects a game to play.
Flow of events:	<table><tr><th>#</th><th>Actor action</th><th>System response</th></tr><tr><td>1</td><td>Select game</td><td>Selected game is launched and AI is set up for the game selected</td></tr></table>
Variations:	None.
Notes and issues:	
Developer notes:	

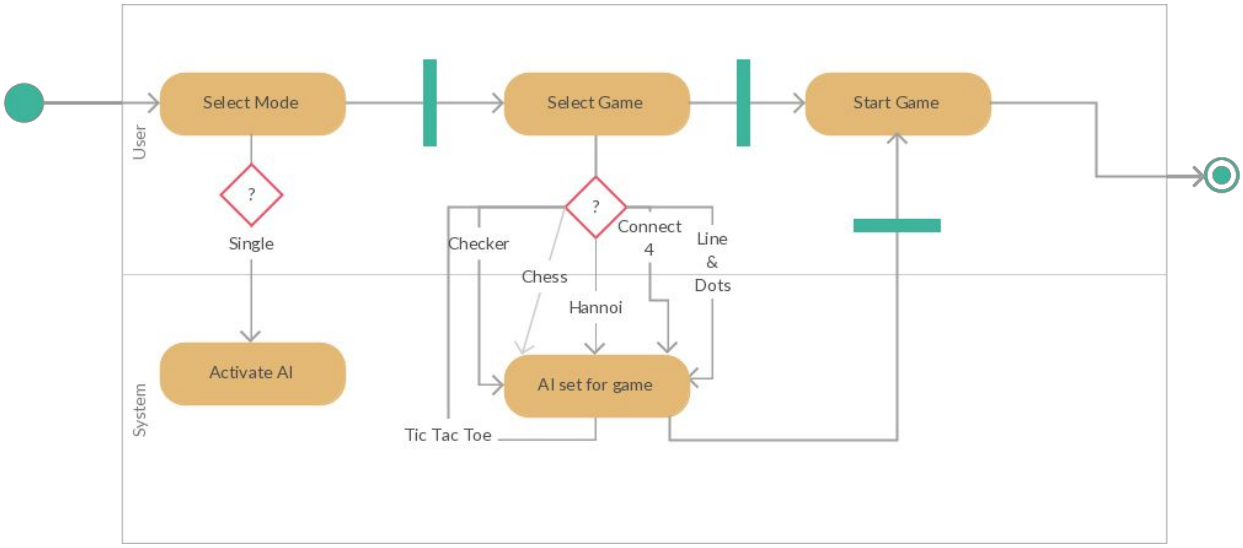
Use Case ID:	UC-03
Use Case Name:	Two Player Mode
Description:	The users can play any game in the arcade in two player mode (against each other).

Actors:	User
Pre-condition:	User selected two player mode.
Post-condition:	The users can play any game in the arcade in two player mode (against each other).
Frequency of use:	Each instance the user selects two player mode and selects a game to play.
Flow of events:	<table><tr><th>#</th><th>Actor action</th><th>System response</th></tr><tr><td>1</td><td>Select game</td><td>Selected game is launched and awaits a player to execute a move</td></tr></table>
Variations:	None.
Notes and issues:	
Developer notes:	

Use Case ID:	UC-04
Use Case Name:	Execute Move
Description:	The user executes a move in the game of choice.

Actors:	User
Pre-condition:	A game has been launched in either single or two player mode.
Post-condition:	A user's desired move is executed.
Frequency of use:	When a game is launched and has not yet been won or tied.
Flow of events:	<table><tr><th>#</th><th>Actor action</th><th>System response</th></tr><tr><td>1</td><td>User selects desired move</td><td>Desired move is validated and executed graphically.</td></tr></table>
Variations:	If an invalid move is selected, the move is not executed.
Notes and issues:	
Developer notes:	

Activity Diagram:



Data Storage:

Our system will not require long term data storage. Variables that are initiated with the application will store short-term data such as number of games, player scores, etc.

UI Mockups:

The following User Interface (UI) mockups show the initial menu window and the six game windows. Common elements such as the window (frame), the four buttons, and the score tallies will be the same objects for each game. The center area of each game frame will be for the game board, which will be common among certain games (e.g., chess and checkers) and unique for others (e.g., Towers of Hanoi). Reusable objects reduce the number of classes, methods, and objects that need to be coded.

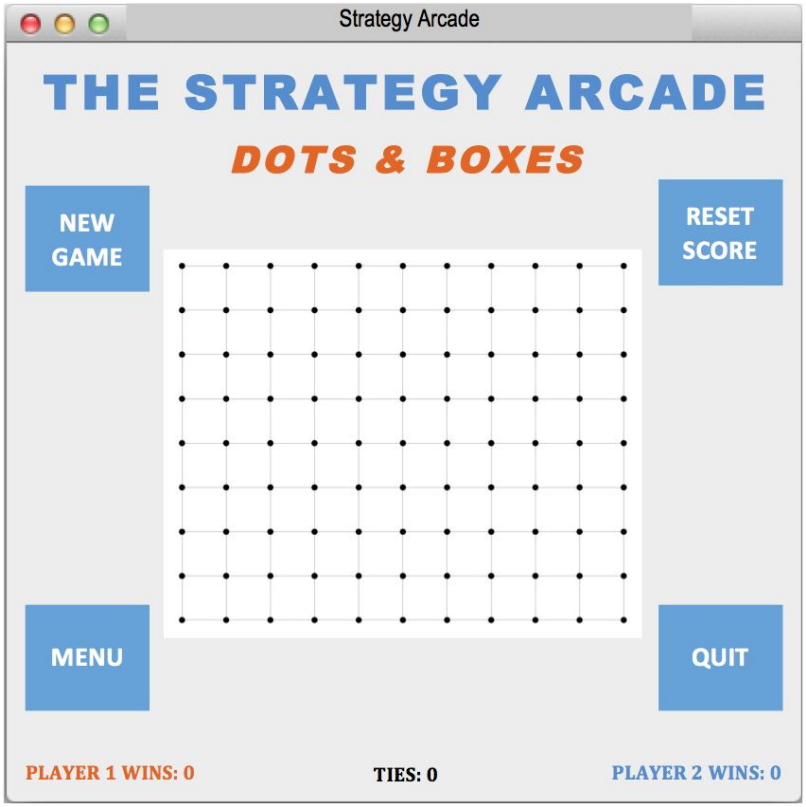
Strategy Arcade
THE STRATEGY ARCADE
TIC TAC TOE
NEW GAME
RESET SCORE
MENU
QUIT
PLAYER 1 WINS: 0
TIES: 0
PLAYER 2 WINS: 0

Strategy Arcade
THE STRATEGY ARCADE
CHESS
NEW GAME
RESET SCORE
MENU
QUIT
PLAYER 1 WINS: 0
PLAYER 2 WINS: 0

Strategy Arcade
THE STRATEGY ARCADE
CHECKERS
NEW GAME
RESET SCORE
MENU
QUIT
PLAYER 1 WINS: 0
PLAYER 2 WINS: 0

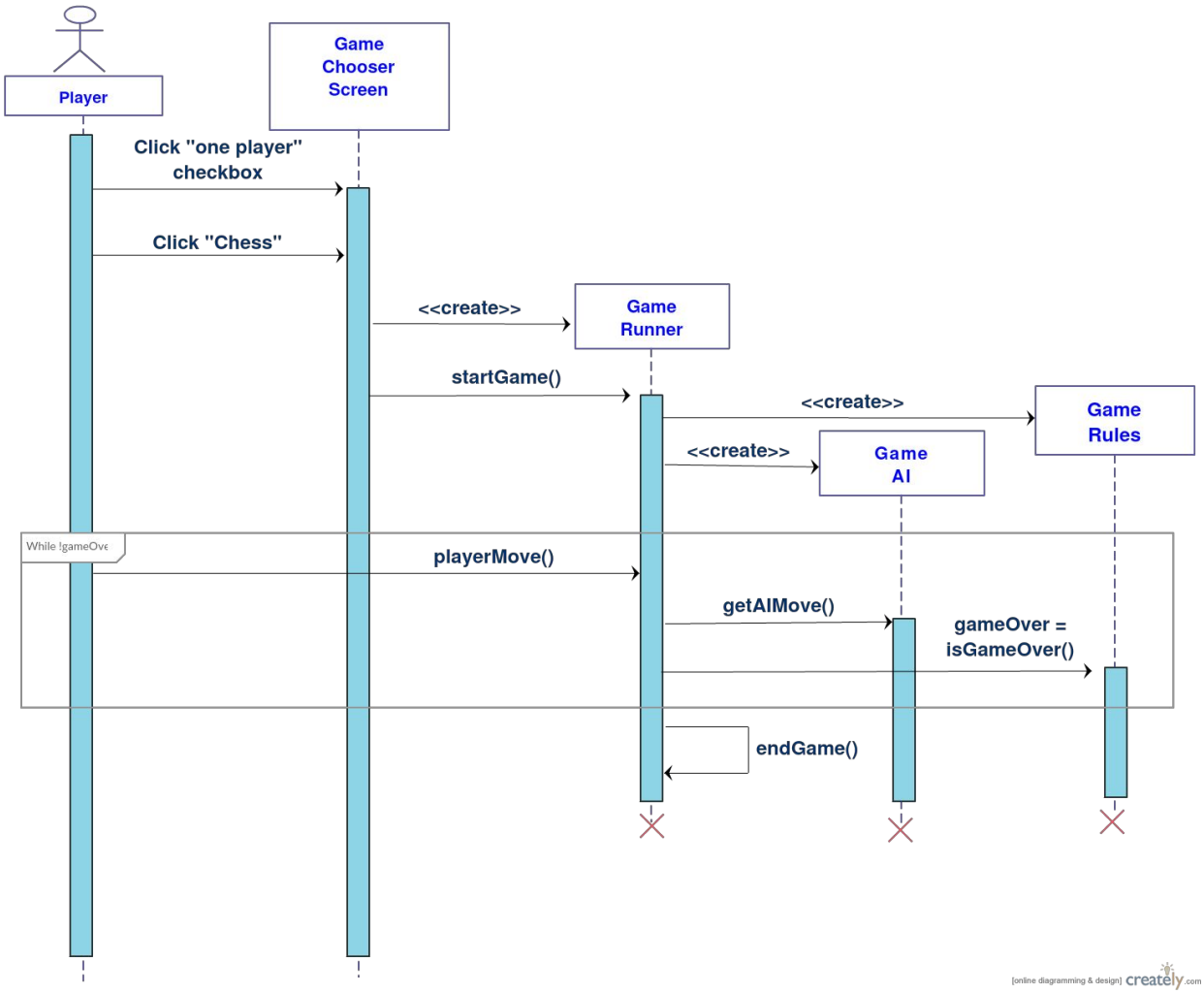
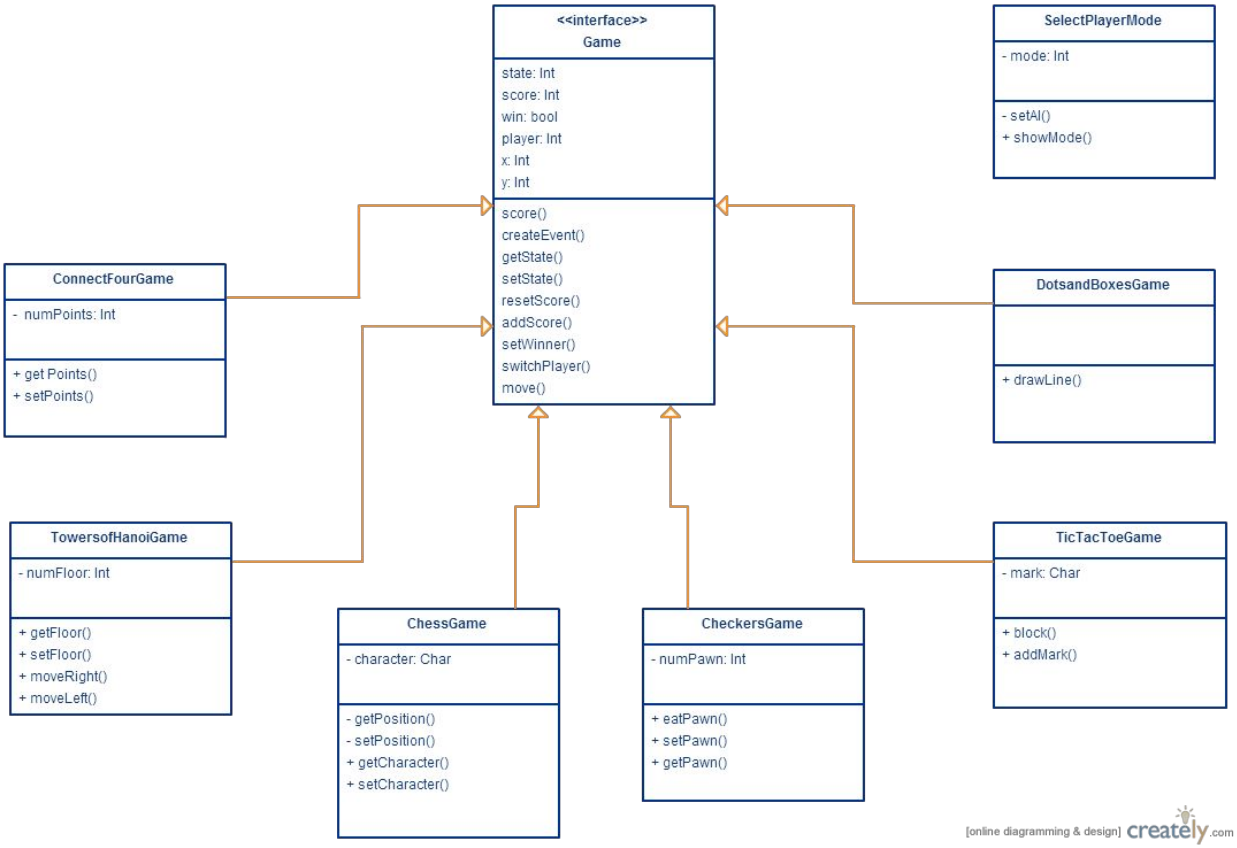
Strategy Arcade
THE STRATEGY ARCADE
CONNECT FOUR
NEW GAME
RESET SCORE
MENU
QUIT
PLAYER 1 WINS: 0
TIES: 0
PLAYER 2 WINS: 0

Strategy Arcade
THE STRATEGY ARCADE
TOWERS OF HANOI
MENU
NEW GAME
RESET SCORE
QUIT
PLAYER 1 WINS: 0
TIME: 00:00
PLAYER 2 WINS: 0


Strategy Arcade
THE STRATEGY ARCADE
DOTS & BOXES
NEW GAME
RESET SCORE
MENU
QUIT
PLAYER 1 WINS: 0
TIES: 0
PLAYER 2 WINS: 0

User Interactions:

In the sequence diagram below, the single player will choose chess. However, the remainder of the interaction (and therefore the diagram) is left nonspecific. We will be programming to an interface; the game runner does not need to know which game it is currently running.



Class Diagram: