

CSCI-466

CMP

Dominion Game

October 3, 2017

Mathew Griffin

Carlos Perez

Brent Parker

Contributions:

Matthew Griffin:

Project management, and overlook of diagrams.

Class diagrams (corrupt).

Carlos Perez:

Project management, Export Log diagram.

Brent Parker:

Cover page, contents, select opponent and view high scores sequence diagrams.

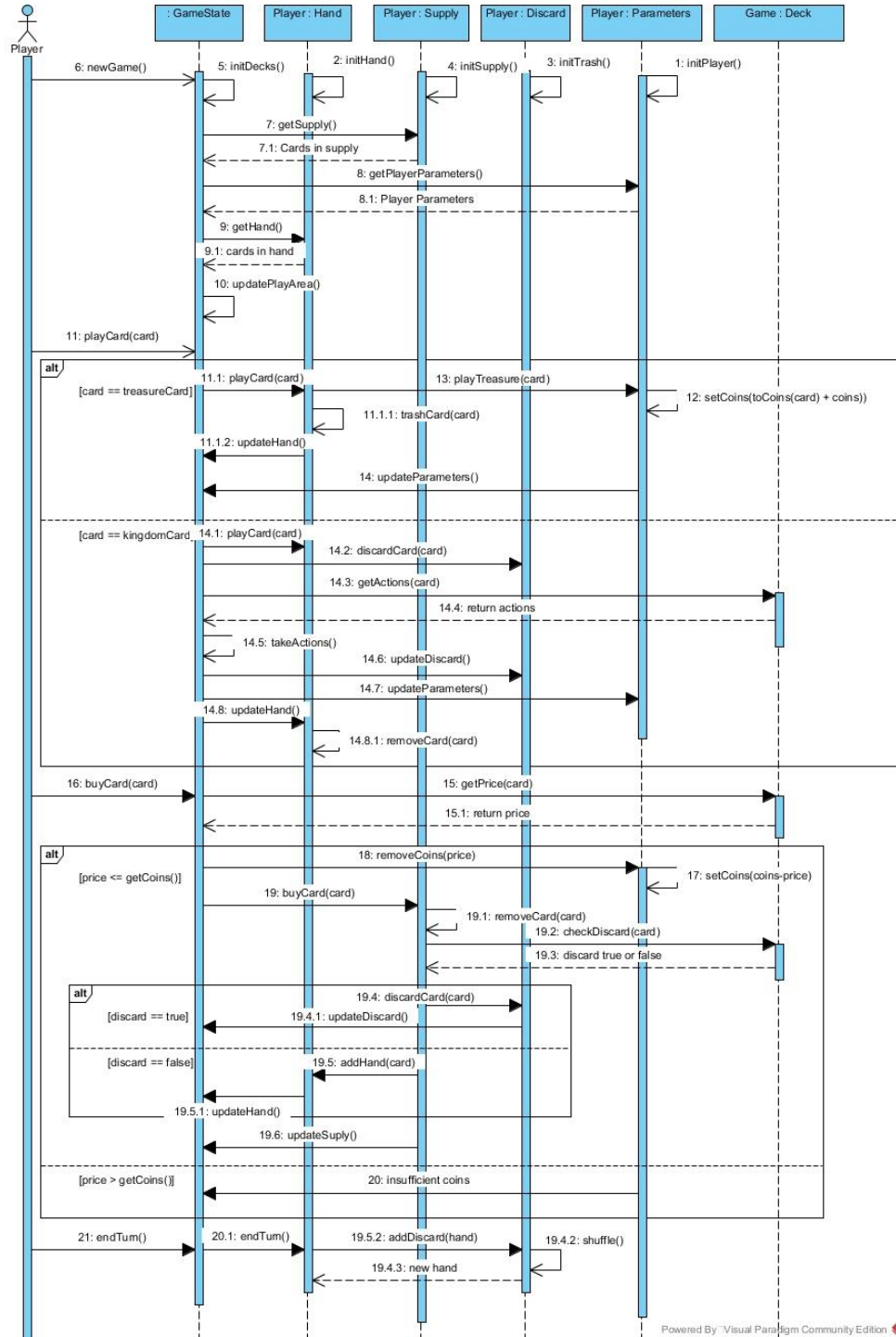
System Architecture.

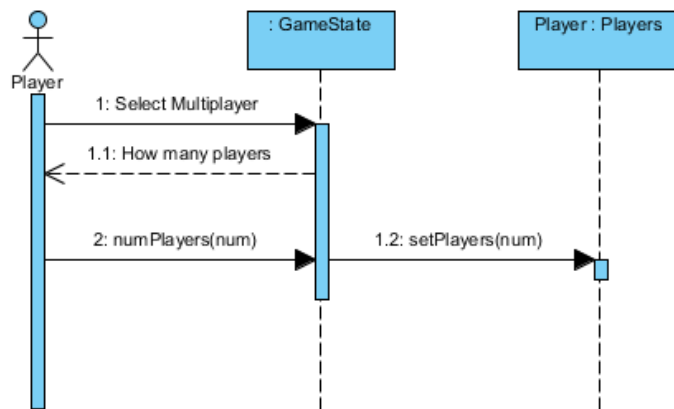
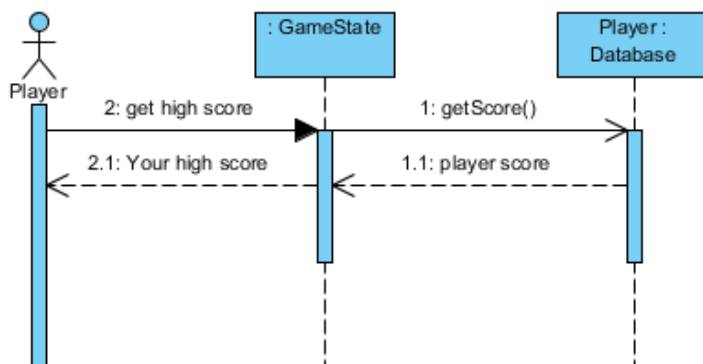
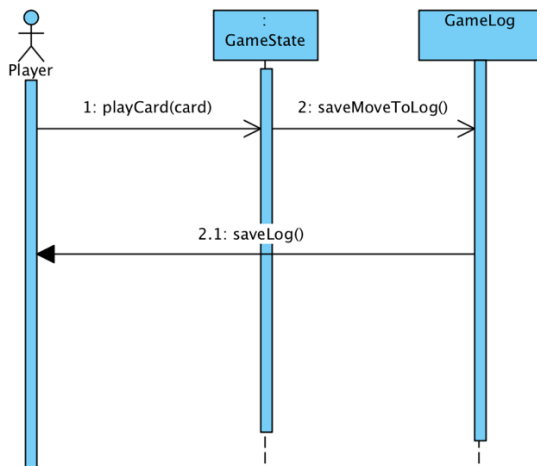
Contents

Contributions:	2
Interaction Diagrams	4
Sequence Diagrams.....	4
System Architecture.....	6
Architecture Styles	6
UML Package Diagram	6
Mapping Subsystems to hardware	6
Persistent Data Storage	6
Network Protocol.....	6
Global Control Flow	7
Execution Order:	7
Time Dependency:	7
Hardware Requirements.....	7
Progress Report.....	7
Project Management	7
References	10

Interaction Diagrams

Sequence Diagrams



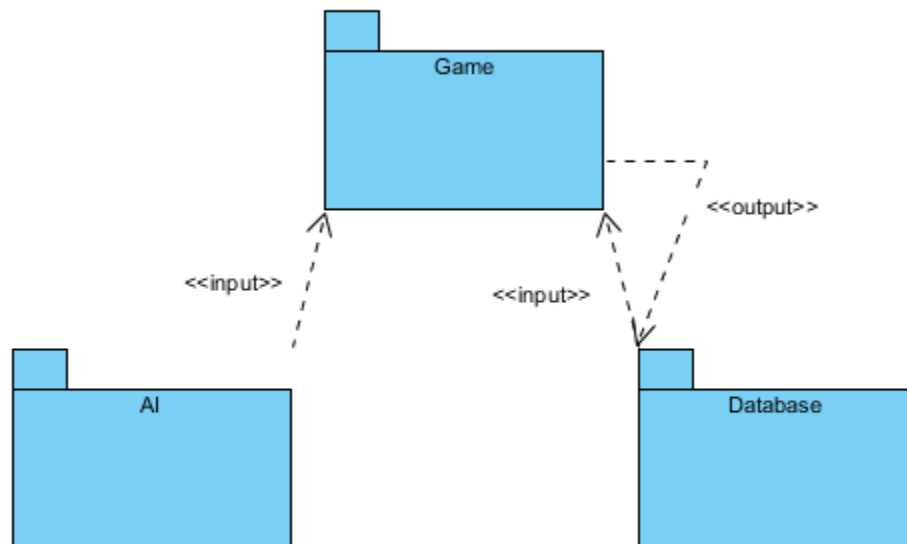
sd Select Opponent**sd** View High Scores**sd** DominionPlayGame

System Architecture

Architecture Styles

The bulk of this system is built upon a **component-based** architectural style. The system is built from functions and objects that are dedicated to their own purpose so that they can be replaced, used in a similar program, and easily extended. This architectural style is easier to use than other architectures. In the future we may implement a **client-queue-client system** for multiplayer games.

UML Package Diagram



Mapping Subsystems to hardware

The subsystems are all client based for the time being.

Persistent Data Storage

The players high score needs to be persistent at some point in development. This may not be included in the initial release of the game. The player high scores will be stored in a relational database in the format:

Player name	High Score	Date
-------------	------------	------

Network Protocol

We're using **HTTP** for this project because it's simple to use, and we want our game to be easily accessible so that users can simply go to our website and start playing the game.

Global Control Flow

Execution Order:

The system will be event-driven. As with most websites the system will wait for the user to act before it does anything.

Time Dependency:

The system won't have any timers.

Hardware Requirements

- Color display with minimum of 1024 X 768 resolution,
- Minimum 1.5ghz CPU with integrated graphics
- 2GB of RAM
- 500MBs of free hard drive space
- Mouse
- Keyboard
- Internet connection with minimum bandwidth of 56Kbps
- Chrome, Firefox, or Edge Internet browser

Progress Report

We have implanted all the front-end work for our game. Our board is loading with all the cards. All our assets are loaded. Functions have been started and the deck hand function has been programmed. End/New Game button also functional.

We are working on shuffling the deck on initial game load, and giving the player his hand while working on the supply pile and trash pile functionality.

This week we are aiming on finishing the basic initial game functionality to stay on track of our project schedule.

Project Management

Week	Tasks
9/11 – 9/17	<ul style="list-style-type: none"> • Carlos: Design basic board the board, and place where elements will be placed. Design in CSS Kingdom cards and place in board. • Matthew: Design in CSS Treasure and Curse cards. Prepare all strings different cards will need. • Brent: Design in CSS Victory and Trash pile cards. For victory cards, generate the strings we will need • Goal for the week is to start to have our assets ready for functionality, and communicate with the team
9/18 – 9/24	<ul style="list-style-type: none"> • Finish task from last week and share with the team and get feedback from everyone. • Carlos: Functionality to start board's deck with 10 Kingdom cards and give the user 5 random cards from the deck. • Matthew: Treasure cards functionality. Assign the value of the cards, and basic functionality that will decrease the buy actions if clicked on a card. • Brent: Give the player their 7 copper cards and 3 estates cards (actions don't need to work perfect at this point, just make sure he gets the cards when he starts the game). • Goal for the week is to start with basic functionality that will be the framework of the game, and start shaping gameplay.
9/25 – 10/1	<ul style="list-style-type: none"> • Carlos: Action cards, once a player click on an action card. Perfect the action (coins, moves, whatever the card value/action says) • Matthew: Give a player a card after using his coins from the coin system. • Brent: Trash pile functionality. Move cards from the trash to the top supply, only if supply is empty. • Goal for the week is to get the card functionality in check, and perform quick moves with the cards.
10/2 – 10/8	<ul style="list-style-type: none"> • Carlos: When no more moves are available, and player wishes to move all cards to pile, also move 5 cards from supply into the deck • Matthew: Buy cards from the supply into a player's deck functionality and move card from supply to deck

Week	Tasks
	<ul style="list-style-type: none"> • Brent: End game logic, if there are no more cards to grab, or any 3 supply piles are empty, end the game and give final score/coins left. • This week's goal: Wrap up functionality as far as gameplay.
10/9 – 10/15	<p><u>Make up week.</u> If we are behind, or we need time catch up. Also test our code, and fix bugs from the functionality that should be complete.</p>
10/16 – 10/22	<ul style="list-style-type: none"> • Carlos: Calculate final store, when a game ends. • Matthew: Treasure cards functionality. Assign the value of the cards, and basic functionality that will decrease the buy actions if clicked on a card. • Brent: Give the player their 7 copper cards and 3 estates cards (actions don't need to work perfect at this point, just make sure he gets the cards when he starts the game). • Goal for the week is to start with basic functionality that will be the framework of the game, and start shaping gameplay.
10/23 – 10/29	<p>Prepare demo for next week. Use this time to fix final bugs, and present also prepare the report for the demo. Iron our bugs and prepare to turn in final demo one next week. Depending on the stage of the game, different people can pick up different bugs and functionality that needs to be done.</p>
10/30 – 11/5	<p>DEMO 1 DUE 11/6</p> <p>Make sure game is in good standing to turn in.</p>
11/6 – 11/12	<ul style="list-style-type: none"> • Carlos: High score functionality, add up the score of the player and store. • Matthew: Rules of the game, add a tutorial to let the user know how to play the game. • Brent: Hint system. Give player tooltips for help, and add (?) icons for players to seek help if needed. • Work on UI, make the experience easier, and show the player how to work.
11/13 – 11/19	<ul style="list-style-type: none"> • Carlos: Work on animations, make the game more presentable. • Matthew: Iron out cards, graphics and work on CSS to make the game easier on the eyes.

Week	Tasks
	<ul style="list-style-type: none"> Brent: CSS fixes, bugs, and make sure the game works on different browsers (chrome,ie,firefox).
11/20 – 11/26	SHORT WEEK – HAPPY THANKSGIVING Easy week, fix bugs that we find and work on smaller items as we feel, but mostly taking it easy for this week.
11/27 – 12/3	<ul style="list-style-type: none"> Carlos: Fixes, bugs and documentation for next week. Matthew: Documentation, reporting and final stages of the game. Brent: Testing the game, provide feedback to others and work on a final bug list we can all work on during the week.
12/4 – 12/10	DEMO 2 DUE 12/11 – FINAL VERSION Go over the list of final bugs and fixes that Brent worked on last week.
12/11	Turn in final Project

References

Microsoft. (n.d.). *Architectural Patterns and Styles*. Retrieved from microsoft.com:

<https://msdn.microsoft.com/en-us/library/ee658117.aspx?f=255&MSPPErr=-2147217396>